International Journal of Computational Intelligence Systems
Vol. **13(1)**, 2020, *pp.* 1447–1463
DOI: https://doi.org/10.2991/ijcis.d.200828.002; ISSN: 1875-6891; eISSN: 1875-6883
https://www.atlantis-press.com/journals/ijcis/

Research Article

A Novel Particle Swarm Optimization Approach to Support Decision-Making in the Multi-Round of an Auction by Game Theory

Trinh Ngoc Bao[1], Quyet-Thang Huynh[2], Xuan-Thang Nguyen[1], Gia Nhu Nguyen[3,4], Dac-Nhuong Le[4,5,6,*]

[1] *Hanoi University, Hanoi 100000, Vietnam*
[2] *Hanoi University of Science and Technology, Hanoi 100000, Vietnam*
[3] *Graduate School, Duy Tan University, Da Nang 550000, Vietnam*
[4] *Faculty of Information Technology, Duy Tan University, Da Nang 550000, Vietnam*
[5] *Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam*
[6] *Faculty of Information Technology, Haiphong University, Haiphong 180000, Vietnam*

ARTICLE INFO

Article History
Received 11 July 2020
Accepted 22 Aug 2020

Keywords
Multi-round procurement
Project conflicts
Game theory
Particle swarm optimization
Nash equilibrium
Decision support system
Multi-objective

ABSTRACT

In this paper, game-theoretic optimization by particle swarm optimization (PSO) is used to determine the Nash equilibrium value, in order to resolve the confusion in choosing appropriate bidders in multi-round procurement. To this end, we introduce an approach that proposes (i) a game-theoretic model of the multi-round procurement problem; (ii) a Nash equilibrium strategy corresponding to the multi-round strategy bid; and (iii) an application of PSO for the determination of the global Nash equilibrium point. The balance point in Nash equilibrium can help to maintain a sustainable structure, not only in terms of project management but also in terms of future cooperation. As an alternative to procuring entities subjectively, a methodology using Nash equilibrium to support decision-making is developed to create a balance point that benefits procurement in which buyers and suppliers need multiple rounds of bidding. To solve complex optimization problems like this, PSO has been found to be one of the most effective meta-heuristic algorithms. These results propose a sustainable optimization procedure for the question of how to choose bidders and ensure a win-win relationship for all participants involved in the multi-round procurement process.

1. INTRODUCTION

Project management is a key activity in software development organizations. One of the key problems in software project planning is the scheduling, which involves resource allocation and scheduling activities on time, optimizing the cost and/or duration of the project. The scheduling is a complex constrained optimization problem, because for the optimization of resources, time, and cost, it is necessary to consider a combination of variables, rules, and restrictions that cause the problem to be NP-hard search-based software engineering (SBSE) is an emerging area focused on solving software engineering problems by using search-based optimization algorithms. SBSE has been applied to several problems which arise during the software life cycle. The software project scheduling problem (SPSP) consists in finding a suitable assignment of employees to tasks, such that the project can be delivered in the shortest possible time and with the minimum cost [1,2].

In recent years, a considerable amount of research has been conducted around this topic. The studies that have been performed have focused on different aspects, for instance, on proposing models to solve the problem of allocating employees to tasks so that the project's cost and duration are minimized, on introducing or applying more efficient optimization algorithms, and on studying the performance of different metaheuristics to determine which may be more appropriate to solve the problem.

We summarize generalizations of the activity concept, the precedence relations and of the resource constraints. Alternative objectives and approaches for scheduling multiple projects are discussed as well. In addition to popular variants and extensions such as multiple modes, minimal and maximal time lags, and net present value-based objectives in Table 1 [1–6].

As we can see in Table 1, the RCPSP can be classified by preemptive scheduling, resource requests varying with time, setup times, multiple modes, tradeoff problems, minimal time lags, maximal time lags, release dates and deadlines, time-switch constraints, etc. The RCPSP is a general problem in scheduling that has a wide variety of applications in manufacturing, production planning, project management, and various other areas. In this study, we aim to identify the key aspects of procurement in the software project management context and their relation to project success [2].

* *Corresponding author. Email: ledacnhuong@duytan.edu.vn*

Table 1 The software project management problems.

Problem	Types	Ref
Project Scheduling Problem (PSP)	Single-objective	[1,2]
Project Scheduling Problem Work-Content (PSPWC)	Single-objective	[1]
Resource-Un-Constrained Project Scheduling Problem (RUCPSP)	Multi-objectives	[1]
Software Project Scheduling Problem (SPSP)	Multi-objectives	[3]
Resource Constrained Projects (RCP)	Multi-objectives	[2]
Resource Constrained Project Scheduling Problem (RCPSP)	Multi-objectives	[2]
Preemptive-Resource Constrained Project Scheduling Problem (P-RCPSP)	Multi-objectives	[1]
Resource-Constrained Project Scheduling Problem with Minimal and Maximal Time Lags (RCPSP/max)	Multi-objectives	[2]
Resource-Constrained Project Scheduling Problem with Flexible Work Profile (RCPSP-FW)	Multi-objectives	[1]
Flexible-Resource Constrained Project Scheduling Problem (F-RCPSP)	Multi-objectives	[1]
Preemptive-Flexible-Resource-Constrained Project Scheduling Problem (P-F-RCPSP)	Multi-objectives	[1]
Resource Constrained Multi-Project Scheduling Problem (RCMPSP)	Multi-objectives	[4]
Preemptive-Resource Constrained Multi-Project Scheduling Problem (P-RCMPSP)	Multi-objectives	[1]
P-RCMPSP with permitted Mode Change (P-MRCPSP-MC)	Multi-objectives	[1]
Flexible-Resource Constrained Multi-Project Scheduling Problem (F-RCMPSP)	Multi-objectives	[1]
Resource Availability Cost Problem (RACP)	Multi-objectives	[2]
Resource-Constrained Project Scheduling Problem with Generalized Precedence (RCPSP-GPR)	Multi-objectives	[4]
Resource constrained project scheduling problem with cash flows (RCPSPCF)	Multi-objectives	[3]
Multi-Mode Resource Constrained Project Scheduling Problem (MRCPSP)	Multi-objectives	[2]
Preemptive-Flexible-Resource-Constrained Multi-Project Scheduling Problem (P-F-RCMPSP)	Multi-objectives	[1]
Multi-Mode Resource Constrained Project Scheduling Problem with Renewable Resource (MRCPSP-RR)	Multi-objectives	[1]
Preemptive-Multi-mode Resource-constrained Project Scheduling Problem (P-MRCPSP)	Multi-objectives	[1]
MRCPSP Problem with Minimal and Maximal Time Lags (MRCPSP/max)	Multi-objectives	[2]
Resource-constrained project scheduling problem with time-dependent resource capacities	Multi-objectives	[2]
Resource Investment Problem with Minimal and Maximal Time Lags (RIP/max)	Multi-objectives	[2]
Search-based Software Engineering (SBSE)	Single-objective	[1]
Work Breakdown Schedule (WBS)	Multi-objectives	[4]
Team Software Project Management Problem (TSPMP)	Multi-objectives	[5]

Project management is the core process of most current business activities, which include projects that have the goal of creating products, services, or original results. Project management tasks include project scope management, quality, project schedule, budget, resources, and risks. Still, there are more internal problems affecting the project which is out of the management of risk management parts such as the conflicts between the project partners or the conflicts in risk management itself. Therefore, detecting and analyzing these problems brings a necessary supplement for the project management tasks, thus ensuring all matters arising to be controlled and also enhancing the quality and chances of success of the project [1,2]. Based on various factors such as the source, cause, role, or function of conflicts in the project, conflicts can be classified into several categories as follows:

- By source of conflict, conflicts are classified as (i) conflict from plan scheduling, (ii) conflict from determining the priority in performing project task, (iii) conflict from the power sources, (iv) conflict from technical problems, (v) conflict from administrative procedure, (vi) conflict from the private issue, and (vii) conflict from the expenditure.
- By cause of conflict, conflicts are classified as (i) conflict from different goals, (ii) conflict from resource disparity, (iii) conflict by other people's obstruction, (iv) conflict due to stress and psychological pressure from many people, (v) conflict due to ambiguity of jurisdiction, and (vi) conflict due to misleading communication.
- By role, conflicts are classified as (i) positive conflict and (ii) negative conflict.
- By function, conflicts are classified as (i) functional conflict and (ii) dysfunctional conflict.

Procurement is viewed as a strategic function to improve an organization's profitability. In recent years, it has been shown that there are many challenges and difficulties in procurement, especially in multi-round procurement, where the result of selecting bidders in one round affects bidding behavior in subsequent rounds [7].

- The first challenge is the difficulty in choosing the right bidders to maximize profits while also ensuring project completion, which becomes more and more challenging when it is divided into many stages with several auction-goers. The material depends on each company and their pricing policies, which must be considered in order to choose the most reasonable ones (*Quality and Cost-based Selection*). However, quality cannot be assured afterward [8]. This implies many risks, as there are no measures or evidence to prove that the chosen bidder is the best one. Moreover, a win-win relationship requires a fair auctioning environment, which is ideal when all bidders are granted the highest profits.
- Secondly, dividing the project into many stages affects the project's total cost, which is related to time. Good sustainability strategies require lowering the cost and, hence, increasing the profits and reducing the predicted price of each package; otherwise, it can lead to package over-budget, which causes incompletion and risks for the project.

In summary, a scientific and applicable method is necessary to make the final decision of the auction [8–13]. Apart from that, the main questions which must be considered for the problem are as follows:

1. When is the best time to open a bid?
2. Who are the most suitable auction-goers?

3. What are the most suitable quality and price at each stage?
4. How can we solve this?

The term sustainability strategy can be explained as a prioritized set of actions which provides an agreed framework in a cooperation situation. In other words, there are countless benefits from a good sustainability strategy, in terms of cooperation in a project, making the jobs of participants easier because long-term cooperation can be maintained by a rational plan of action. Rationality makes sustainability more effective and easier when it comes to project integration. Rationality can be obtained by forming a win-win situation for all sides.

Despite the fact that, in some special types of problems, Nash equilibrium may not perform as well as other models, such as competitive equilibrium (*i.e., when dividing the channels into two categories: high-quality and low-quality*), it is still the most suitable model to find the appropriate benefit for all players by determining a win-win relationship [14–17].

The multi-round procurement problem was being an NP-hard problem, solution methods are primarily meta-heuristics. The above questions have already been answered by some researchers using genetic algorithms (GAs) and the Nash equilibrium [14,18]. However, when the threshold is changed, the performance of each algorithm is different.

Particle swarm optimization (PSO) and evolutionary algorithms such as GA have been used to solve multi-objective problems; they have many similar behaviors but also have differences in performance across certain problem areas. They both have the same mechanism, in that the system is initialized by generating a population randomly and searching for optima by refining this population. Nonetheless, PSO does not use techniques such as cross-over and mutation for updating and, so, may prematurely converge to a local optimum. The authors prove that PSO and GA have the same effectiveness in solution quality, but that PSO has superior computational efficiency over GA. The advantages of PSO are that it uses fewer parameters and evaluation functions than GA [19].

The main idea of GA is combining father and mother singles to generate many descendants, in order to create a better generation. In contrast, the PSO Algorithm focuses on developing its own members to better themselves. Therefore, instead of using GA and generating 100 particles, we can use PSO with fewer individuals and let them improve themselves [20,21]. For these reasons, we propose a new approach using PSO and Nash equilibrium to identify a sustainability strategy in a multi-round procurement problem which has higher accuracy and better speed. The following sections explain further details about the problem under consideration.

Our main contributions is presented a novel approach for solving a multi-round procurement. In this approach, by combining knowledge from the fields of game theory, Nash equilibrium, and PSO algorithm, we presented a game theory model as an optimal cooperative solution for the multi-round procurement problem. The parameter setting strategy is selected in an attempt to fine-tune the PSO algorithm for this problem. The Nash equilibrium point can be considered as the best solution for the conflict. The Nash equilibrium point can be solved by multi-objectives evolutionary algorithm. Compared with another multi-objectives evolutionary algorithms, the result suggested that PSO algorithm was significantly better than others with shorter runtime and higher quality for finding Nash equilibrium candidate in large-scale dataset.

2. RELATED WORKS

The software project management was being an NP-hard problem, solution methods are primarily meta-heuristics. We classify and present the state-of-the-art hybrid meta-heuristics for software project management and multi-round procurement problems in Table 2 [20–29]. These methods can be categorized based on local search meta-heuristics, evolutionary and population-based, learning meta-heuristics, and our proposed algorithms

In [12], the authors designed a new approach to find a winner in a multiple-attribute auction. The new method could help to address a mechanism for decision-making in dealing with multiple-attribute and multiple-sourcing procurement. Multiple-sourcing is usually proposed to prevent a variety of procurement risk and uncertainty. In [22], a model was developed to assist participants in multiple-attribute procurement in inferring their preference, based on the difficulty of elicitation.

In [23], the ability to use Bayesian equilibrium to predict human behavior in sealed-bid split-award procurement auctions was analyzed. The auction type was a multi-object extension, in which bidders may require more than one unit. The authors applied the Bayes–Nash equilibrium in the expanded game to forecast the strategies of participants. Then, they experimented with the accuracy of the equilibrium with computerized bidders. The results showed that the complexity of the multi-object does not have much effect on the result of procurement; rather, the risk aversion has a more pronounced effect.

In game theory, Nash equilibrium has been used to analyze the outcome of strategic interaction and also how conflict may be mitigated in competitive environments. A Nash solution is a mixed strategy profile with the property that no single player can obtain a higher value of expected utility by deviating unilaterally from this profile. It means all players obtain more revenue under cooperation, and thereby promoting sustainability. The above questions are already answered by some researchers using the GA and the Nash equilibrium. However, when the number of a threshold is changed, the performance of each algorithm is different. Furthermore, there are many meta-heuristic algorithms were proposed, such as GDE3 [29], ϵ-MOEA [31], ϵ-NSGA-II [33], NSGA-III [32], SPEA2 [36], ACO [25], Water drops [28], ANGEL [38], HEA [39], (1+1) EA and RS [40], EHH [41], Hybrid ABC-PSO [42], PSO Parallel (PPSO) [46], ACO Parallel (PACO) [48].

In [49], we proposed a GA model to infer the Nash equilibrium model for a multi-round procurement problem, where the buyer is not only focused on the revenue under the first-price bidder strategy, but also considers a variety of properties for bidder values (e.g., the relationship between investor and bidder, offer for future promotion, and the business credit score of the bidder). The chromosome in the GA represents the Nash equilibrium point, which includes all characteristics of the solution for the problem. An iterative GA is said to converge when, as the iterations proceed, the fitness value gets closer to specific values; we found a balance point for all sides in procurement.

Table 2 Summary of the classification algorithms for the software project management and multi-round procurement problems.

Classes of Algorithms	Authors	Year	Ref
1. ***Local Search Meta-heuristics Algorithms***			
HL (Hill Climbing)	Kolisch *et al.*	1999	[24]
TB (Tabu search with Path Re-linking Strategy)	Nonobe and Ibaraki	2002	[25]
GRASP (Greedy randomized adaptive search procedure)	Kochetov and Stolyar	2003	[25]
LNS (Large Neighborhood Search)	Palpant *et al.*	2004	[26]
SA (Simulated Annealing)	Mika *et al.*	2005	[27]
SS (Scatter Search and Electromagnetism Theory)	Debels *et al.*	2006	[24]
RS (Random Search)	Chen *et al.*	2009	[25]
ALNS (Adaptive Large Neighborhood Search)	Muller L.F	2009	[24]
FAF (Filter-and-Fan)	He *et al.*	2016	[28]
2. ***Evolutionary and Population-based Algorithms***			
GDE3 (3rd Evolution Step of Generalized DE)	Kukkonen and Lampinen	2005	[29]
AGA (Adaptive Genetic Algorithm)	Kim *et al.*	2006	[30]
CGA (Competent Genetic Algorithm)	Yassine *et al.*	2007	[28]
GA (Genetic Algorithm)	Chang *et al.*	2008	[27]
GP (Genetic Programming)/GALib	Chang *et al.*	2008	[31]
SMPSO (Speed-Constrained Multi-objective PSO)	Nebro *et al.*	2008	[28]
MOCell (Multi-Objective Cellular Genetic)	Nebro *et al.*	2009	[28]
MOFA (Multi-Omics Factor Analysis)	Yang	2009	[28]
ϵ-MOEA (ϵ-Multi-Objective Evolutionary Algorithm)	Li and Zhang	2009	[32]
DEA (Differential Evolution Algorithm)	Wu *et al.*	2010	[28]
ϵ-NSGA-II	Vanucci *et al.*	2012	[33]
NSGA II (Adaptive$_v$, Adaptive$_s$, Adaptive$_c$, Adaptive$_a$, Adaptive$_{cv}$)	Deb *et al.*	2012	[33]
MOGA (Multi-Objective Genetic Algorithm)	Stylianou *et al.*	2013	[34]
NSGA (Non-Dominated Sorting Genetic Algorithm)	Amiri *et al.*	2013	[33]
PAES (Pareto Archived Evolution Strategy)	Wang *et al.*	2013	[27]
DEPT (Differential Evolution)	Chen and Ni	2014	[28]
NSGA-III (Non-Dominated Sorting Genetic Algorithm III)	Mkaouer *et al.*	2015	[35]
SPEA2 (Strength Pareto Evolutionary Algorithm 2)	Deb *et al.*	2015	[36]
MOEA (Multi-Objective Evolutionary Algorithm)	Hadka	2015	[34]
NSGA-II$_v$	Myszkowski *et al.*	2017	[37]
NSGA-II$_a$ (*Adaptive*)	Gholizadeh and Afshar	2018	[37]
AI (Artificial Immune)	Agarwal *et al.*	2007	[20]
EOD (Estimation of Distribution)	Fang *et al.*	2010	[20]
SFL (Shuffle Frog-Leaping)	Fang and Wang	2012	[20]
FA (Firefly Algorithm)	Sanaei *et al.*	2013	[27]
MA (Memetic Algorithm)	Wang and Liu	2013	[28]
GE (Grammatical Evolution)	Barros *et al.*	2013	[28]
PSO (Particle Swarm Optimization)	Zeighami *et al.*	2013	[27]
BCO (Bee Colony Optimization)	Nouri *et al.*	2013	[27]
ACO (Ant Colony Optimization)	Xiao *et al.*	2013	[25]
PBIL (Population-Based Incremental Learning)	Jin *et al.*	2014	[28]
IWD (Intelligent Water Drops)	Crawford *et al.*	2018	[28]
MOABC (Multi-Objective Artificial Bee Colony)	Gong *et al.*	2018	[28]
3. ***Hybrids Meta-heuristics Algorithms***			
ANGEL (Integrating ACO, GA and Local Search)	Tseng and Chen	2006	[38]
HEA (Hybrid Evolutionary Algorithm)	Rogalska *et al.*	2008	[39]
(1+1) EA and RS (Evolutionary Algorithm with Random Search)	Yannibelli and Amandi	2013	[40]
EHH (Evolutionary Hyper-Heuristic)	Wu *et al.*	2016	[41]
Tree search + GA	Zamani	2017	[42]
Hybrid ABC-PSO	Khuat *et al.*	2018	[43]
4. ***Learning and Parallel Metaheuristics Algorithms***			
PL (Population Learning)	Jedrzejowicz *et al.*	2006	[44]
NN (Neural network)	Jaberi and Jaberi	2014	[45]
PPSO (PSO Parallel)	Fahmy *et al.*	2014	[46]
Tabu Search Parallel	Bukata *et al.*	2015	[47]
PACO (ACO Parallel)	Thiruvady *et al.*	2019	[48]
5. ***Our algorithms proposed***			
GANE (Genetic Algorithm and Nash Equilibrium)	Trinh Ngoc Bao *et al.*	2017	[49]
BCPM (Bayesian Critical Path Method)	Nguyen Ngoc Tuan *et al.*	2018	[50]
UGM (Unified Game-Based Model)	Trinh Ngoc Bao *et al.*	2019	[51]
NAM (Nash Equilibrium Model)	Quyet-Thang Huynh *et al.*	2019	[52]
Probabilistic Method	Quyet-Thang Huynh *et al.*	2020	[53]
MMAS (Min-Max Ant System)	Dac-Nhuong Le *et al.*	2020	[54]

In our latest paper [50–54], we proposed a unified game-based model based on the concept of game theory and Nash equilibrium. This scientific model can help the decision-maker to recognize and define all characteristics of the conflict problems in project management. The model can be solved by a variety of multi-objective evolutionary algorithms. We introduced some important conflicts in project management and proposed a mathematical model addressing these problems. Our experimental results showed that the unified game-based model is useful to address these problems and effective in generating a near-optimal balance point.

3. MULTI-ROUND PROCUREMENT PROBLEM FORMULATION

3.1. Problem Description

Multi-round procurement is a process of self-taking advantages of the investor and the bidders. It includes several participants jointly negotiating and persuading, not only to gain the most benefit but also to maintain a long-term relationship with a sustainability strategy [8,11]; in other words, the investor's strategy in selecting bidders considers two criteria: (i) obtaining goods or services at low prices and (ii) maintaining a sustainability-oriented customer relationship by giving them a chance to win the auction. On the contrary, suppliers obviously want to maximize the profit gained through the auction. In particular, if both sides can reach an agreement, the bidders are willing to consider a special discount for goods or services in one or many stages of procurement.

The problem that occurs when participants only attempt to get the best advantage for themselves in each round of procurement; the greedy strategy does not often produce an optimal solution, in general. The investor will always put effort into constraining bidders to lower their prices. To deal with this situation, most bidders have no choice other than lowering their product prices to compete with others. This leads to a lower profit or forms a huge barrier which prevents bidders from joining the procurement process. In some situations, they may reduce their good's quality to win the deal; as a consequence, trustworthy and quality suppliers may give up. In this case, we have a lose-lose solution for the multi-round procurement.

For clarity, the problems faced can be listed as follows:

1. *Project time*: The longer the project takes, the higher the cost. We all know that a dollar at the present time is valued more than a dollar in the future due to variables such as inflation and interest rates. As inflation increases, changes in the exchange rate lead to a decrease in the value of the original money. As a consequence, the material price will be higher or lower afterward.
2. *Material costs*: The amount of needed materials also affects the profit on both sides.
3. *Discount rate*: This depends on the strategies of the contractors; they can offer a discount of goods or services in order to gain a competitive advantage over their rivals.
4. *Selection*: Usually, we tend to choose the cheapest price. The question is how do we ensure that it is a low risk choice? As mentioned above, the quality can be low or the bidders who suffer the detriment have to set a higher price for bidding in the next stage of procurement, in order to afford their loss.

Intuitively, the problem can be formulated as perfect-information extensive-form games. Players in the game take part and offer methodologies to locate the most fitting answer to keep up their relationship and lead to the most benefit.

- There is a list of necessary materials, divided into many packages, in the plan of the bidders.
- The project will occur at a certain time. In that time, several rounds will be held and, each time, the investor will buy one or many packages necessary for the project.
- The project has many bidders, including both long-term partners and new collaborators.
- Each supplier is capable of providing some material, based on their ability. They have their own tactics with information about the price and discount after time.

In summary, in order to calculate the Nash equilibrium point for this real-world problem to achieve an appropriate sustainability strategy, we sought to provide adequate parameters for the game model; that is to say, analyzing the data from the problems mentioned above, the data could then be represented by model elements.

3.2. Calculation Functions

In most cases, any decision to gain profit results in disagreements with other involved parties. According to the Nash equilibrium [14], solving this problem creates a balanced result for all contestants. The strategy to reach this point can be modeled as a group of tactics:

$$G = \{S_o, S_c, F_o, F_c\}, \tag{1}$$

in which

- S_c and F_c are a complete plan of action and benefit function of a first player (project owner), respectively
- S_o and F_o are a complete plan of action and benefit function of a second player (project bidder)

$$F_i = B_i - C_i \text{ at time } t_i, \tag{2}$$

where

- F_i quantifies a player's net benefit
- B_i quantifies a player's net revenue
- C_i is the total of the best possible price fin round i (in a period of time t_i) in multi-round procurement

At the beginning (i.e., t_0), it is necessary to use the discount rate to calculate the decrement of the cash:

$$F_{i_0} = \frac{B_i - C_i}{(1+r)^{t_i - t_0}}, \tag{3}$$

where

- r is the discount rate
- F_i is the net benefit from t_0 to t_i.

Setting $t_0 = 0$, we have

$$F_{i_0} = \frac{B_i - C_i}{(1+r)^{t_i}}. \tag{4}$$

When the project is separated into many stages, the net benefit is calculated by

$$\begin{aligned} F &= \sum_{i=0}^{N} \frac{B_i - C_i}{(1+r)^{t_i}} \\ &= \sum_{i=0}^{N} \frac{B_i}{(1+r)^{t_i}} - \sum_{i=0}^{N} \frac{C_i}{(1+r)^{t_i}} \end{aligned} \tag{5}$$

Since $r << 1$, the string Taylor of e^x will give the approximately number

$$\begin{aligned} F &= \sum_{i=0}^{N} \frac{B_i}{(1+r)^{t_i}} - \sum_{i=0}^{N} \frac{C_i}{(1+r)^{t_i}} \\ &\approx \sum_{i=0}^{N} B_i.e^{-r.t_i} - \sum_{i=0}^{N} C_i.e^{-r.t_i} \\ &= F_1 - F2 \end{aligned} \tag{6}$$

where

- F_1 is the Benefit function
- F_2 is the Cost function

The utility function is calculated as

$$F = \sum_{i=0}^{N} B_i.e^{-r.t_i} - \sum_{i=0}^{N} C_i.e^{-r.t_i}. \tag{7}$$

3.2.1. For project owner

Assume that the estimated cost for the whole project is A and the actual cost is B (usually $B < A$). The profit of the project owner is

$$F_0 = A - B, \tag{8}$$

where B is calculated as

$$\begin{aligned} B &= \sum_{i=1}^{N} package_i \\ &= \sum_{i=1}^{N} \sum_{j=1}^{M} \left(x_j P_j\right)_i e^{-r.t_j} \end{aligned} \tag{9}$$

where

- $package_i$ is the total cost for package i
- x_j is the number of materials i at time j
- P_j is the cost of materials i at stage j after deducting the discount
- n is the number of packages
- m is the number of stages

The profit of the project owner, according to the first calculation, is

$$F_0 = \frac{A.e^{-r.t_{Finish}}}{\sum_{i=1}^{n} \sum_{j=1}^{m} (x_j P_j)_i e^{-r.t_j}} - 1. \tag{10}$$

The higher the value of F_0, the higher the benefit the project owner gets.

3.2.2. For chosen bidders

For each material, based on the sale price, original price, and discount to customers, different profit values are obtained. The benefit of each bidder follows the formula

$$\begin{aligned} F_{ci} &= \frac{\sum_{i=1}^{n} profit}{\sum_{j=1}^{m} fund} \\ &= \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} x_{jci} I_{ij} e^{-r.t_j}}{\sum_{i=1}^{n} \sum_{j=1}^{m} x_{jci} C_{ij} e^{-r.t_j}}, \end{aligned} \tag{11}$$

where

- I_{ij} is the profit for each unit of item I at the time j
- C_{ij} is the base price for each unit of item I at the time j

The profit of each contractor should be high enough to maintain their interest in the project. To balance the benefit, the total subtraction of each provider's profit is taken into account:

$$C = \sum_{\alpha=1,\beta=\alpha+1}^{\beta=h} |F_{c\alpha} - F_c\beta|, \tag{12}$$

where h is the number of contractors.

The balance is defined as, if $C = 0$, then all contractors will have the same profit in return.

3.2.3. For all bidders

The profit gained is calculated by the sum of all profits from the materials for the project

$$F_c = \sum_{\alpha=1}^{p} F_{c\alpha}, \tag{13}$$

where p is a number of bidders.

The higher the value of F_c, the bigger the benefit to the bidders.

4. A NOVEL PSO APPROACH TO SUPPORT DECISION-MAKING IN THE MULTI-ROUND OF AN AUCTION BY GAME THEORY

4.1. Nash Equilibrium Theory for Multi-Round Procurement Mixed Solution

Game theory is "*The study of mathematical models of conflict and cooperation between intelligent rational decision-makers*." It has applications in a variety of fields of social science as well as in computer science [15,16]. In this research, we present a decision-supporting approach based on game theory. Multi-round procurement as a process relating to many participants requires a balance of harms and benefits of the decision. Using game theory, real-world conflicts for such situations as pricing competition and relationship negotiation can be laid out [17].

We study infinite games of perfect information played by multiple players, all players of a game pick and join to create a blended procedure, which is a combination of beliefs about probabilities over strategies and the choices of the other player. Technically, a Nash equilibrium is a set of mutual strategies that is considered as most beneficial by all parties and no player will defer from it. As mentioned in John Nash's theory, the systems or joined moves are no mystery. The benefit value represents the quality of each player's profile of action in terms of players cooperatively trying to reach an agreement [14].

Nash equilibrium is one of the most important tools to comprehend the contention between various players. For example, in the Prisoner's dilemma, two detainees are scrutinized for the same crime. Provided that player *A* defects to the police and confesses that player *B* committed the crime, *A* gets his sentence removed and *B* gets full jail time. If both defect from each other, they get equally long prison sentences, and if both parties cooperate and remain silent, they each get reduced jail time. On the off chance that both parties blame each other, that implies a conundrum, both will be placed in prison for a long time (see Figure 1).

Nash equilibrium is a widely utilized model to comprehend the contention between numerous players. For example, the two detainees question, if two detainees are scrutinized a similar interest that on

Figure 1 Solving conflict with Nash equilibrium.

the off chance that he concedes the other individual has made the wrongdoing, he can has ten years rebuff by detainment diminished or else both will be placed in prison for a long time. On the off chance that every one of the two blames the other, that implies a conundrum, both will be placed in prison for a long time. On the off chance that one denounces, the other stay silent, the tranquil individual will get longer discipline.

The conditions of the problem are shown in Table 3.

By far the best strategy for the prisoners is cooperating to get a short jail term for them both. By contrast, receiving a long jail term is a bad strategy. As it has been pointed out, the problem draws a conflict for each individual in which they do not want to cooperate. It is a choice between competition and cooperation. The agreement to cooperate can guarantees that both win. By contrast, in a noncooperative situation, even a more attractive strategy can lead to worse results.

4.2. A Novel PSO Algorithm Applied for Multi-Round Procurement Problem

PSO [20,55–57] is an advanced computational technique which has been used previously, similar to GA [18], differential evolution (DE) [19], and ant colony optimization (ACO) [25,58,59]. The authors which proposed the algorithm were inspired by the idea of swarm intelligence, as typically seen in animal groups, such as birds, fish, and insects. PSO was first proposed by Russell Eberhart and James Kenedy in 1995 with the main idea of using individual self-experience combined with the social experiences of the entire population. However, the PSO algorithm is not the same as the GA algorithm, as it searches for optima by evaluating individual fitness values based on co-operation and competition among individuals. The PSO algorithm is the consequence of considering a herd of feathered creatures finding sustenance in a specific area. It is a metaheuristic algorithm based on the concept of swarm intelligence, which is useful in solving complex optimization problems [20].

In summary, the PSO algorithm implements a heuristic search through a technique that depends on joining every particle's conduct and the entire swarm's conduct by modeling an organic population. The distinction between the PSO and GA algorithms is extremely delicate. Both of them utilize the idea of ensuring the balance of the diversity of the population and similarly rely upon the information shared between their population to elevate their inquiry forms by joining probabilistic and deterministic standards. Therefore, they bring about similar arrangement quality. The idea which distinguishes PSO is its use of neighborhoods. Dissimilar to other heuristic search techniques, in PSO, every particle's neighborhood is kept invariant. Experimental studies have shown that the PSO algorithm exhibits better robustness and converges much closer to the global optimum than the GA algorithm [20,27,28].

Table 3 The problem circumstances.

AB	Defect	Cooperate
Defect	2.2	0.3
Cooperate	3.0	1.1

The swarm is comprehended as an accumulation containing various particles which impact the calculation productivity. Be that as it may, particles can be picked haphazardly by and large. A population of particles is initialized with arbitrary speeds and positions, and the wellness capacity is assessed by utilizing a particle's location and direction. The process initiates the population of particles (N) with some random values for position X and velocity V.

- Position: $X_i = (x_{i1}, x_{i2}, .., x_{iN}) \in R^N$.
- Velocity: $V_i = (v_{i1}, v_{i2}, .., v_{iN}) \in R^N$

Each particle maintains knowledge of its best previous evaluated position, represented as $P_i = (p_{i1}, p_{i2}, .., p_{i,N}) \in R^N$, and also has knowledge of the single global best solution found so far. Then, is it decided whether the optimization function *F(opt)* needs to be minimized or maximized. The two best values for each particle, $pBest_i = f(P_i)$ and $gBest = f(P_g)$, $P_g \in R^N$, are calculated, where *pBest* is its own best solution in its history (fitness) and *gBest* is the best value collected out of all particles in the population. The comparison of fitness values is done with the *pBest* value, and the *gBest* value is replaced when the current fitness value is found to be better than the existing *gBest* value; considering this, the particle with the best fitness value among the whole population is selected. This is not quite the same as GA, which utilizes various particles for each generation created.

4.2.1. Initialize particles

The representation step plays an important role in the successful design of PSO-MRP algorithm, which aims at finding an appropriate mapping between problem solution and PSO-MRP particle. Assuming that N particles form a population, each particle in the initial solution for the problem was generated randomly. A chromosome is the set of strategies selected for each generation. The best global chromosome found can be considered as the Nash equilibrium point, and the chromosomes in each generation are candidates for the Nash equilibrium point.

The chromosome should bring useful information to represent the solution for the multi-round procurement problem. We propose the format of the Nash equilibrium $chromosome = [4, 2, 4, 0, 3, 1]$ as follows (see Figure 2):

A generation includes the number of rounds in procurement, where each round contains bidding and player information. In the first generation, the estimation of the fitness value of every player is random.

4.2.2. Parameters selection

In each new generation, the algorithm performs the following steps:

- Step 1: Call the fitness function and provide fitness values to adjust the priority of particles.

4	2	4	0	3	1

Figure 2 Nash equilibrium's format.

- Step 2: Adjust the optimal solution (i.e., global optimum).
- Step 3: Update the speed vectors.

Parameters selection of PSO-MRP will play an important role in performance and efficiency of the algorithm. Based on the analysis of the problem, this paper propose an PSO-MRP with the following parameters as shown in Table 4.

4.2.3. Fitness function

Multi-stage procurement has a lot of cases and possibilities which can happen, considering the extra conditions given by bidders and investors in order to gain satisfactory benefits. Hence, the constraints considered were as follows:

- The whole project is divided into many stages, where the total time of each round equals the project length and each round lasts at least 1 month;
- Bidders joining must provide sufficient materials and participate until the end of the project. Moreover, they can join at any stage they want, within their capability.

The constants used are the values that maintain the balance of adaptive function factors. These constants have the role of adjusting the fitness function, in order to direct all of the particles to convergence while satisfying the criteria to get the best result. This function is used to ensure the benefit of both sides in obtaining the maximum profits. Therefore, a solution to optimize the objective function is required.

Particles are evaluated through a fitness function, the fitness value represents the quality of the solution that the particle expresses and also shows its influence in the population. In this problem, we must bring the *Fitness function* to a minimum, the fitness value is calculated by

$$Fitness = \left| A \sum_{i=1}^{p} \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} x_{jci} I_{ij} e^{-r.t_j}}{\sum_{i=1}^{n}\sum_{j=1}^{m} x_{jci} C_{ij} e^{-r.t_j}} - B \frac{A.e^{-r.t_{Finish}}}{\sum_{i=1}^{n}\sum_{j=1}^{m} \left(x_j P_j\right)_i e^{-r.t_j}} - 1 \right| + S^2_{F_C} \tag{14}$$

or

$$Fitness = |AF_c - BF_0| + S^2_{F_C} \rightarrow Min \tag{15}$$

where

Table 4 Selected control parameters in PSO-MRP algorithm for multi-round procurement problem.

Parameters Name	Variable	Value
Number of particles	N	100
Number of iterations	*Max*	1000
Inertia weight	w_{min} and w_{max}	0.4 and 0.9
Cognitive coefficient	c_1	0.5
Social component	c_2	0.5
Stopping criteria	*StopCondition*	1000

- A, B are constants
- F_c is the total profit of bidders
- F_o is the benefit of project owner
- $S^2_{F_C}$ is the variance benefit of bidders

4.2.4. Personal best, global best, and particle update

In each generation, particles are updated by comparing their fitness value to that of their best position *pBest* in all previous iterations. The *pBest* value is replaced by a new one, if the current fitness value is better. In the next step, the fitness value of each particle is compared with the global best of the population (*gBest*), where *gBest* is replaced if a better fitness value is obtained (see Figure 3).

The model to change nodes is if the *gBest* member has a lower cost, higher quality, and higher association with the venture proprietor than that of particle *i*. These criteria help to pick the best possible closeout member with the best cost while ensuring the quality of the items delivered. These models are evaluated as a point. If the criteria are met, the point is raised up; otherwise, they are dropped down by one unit. In the event that the fact is sure, the node will be replaced.

When the k^{th} particle moves to new position X_k, if a greater fitness value is achieved, its personal best $pBest_k$ is replaced with X_k. If fitness value of any *pBest* is greater than current *gBest*, *gBest* is replaced with that *pBest*. Particle *i* is updated, based on *gBest* (see Figure 4).

The velocities of particles are updated regularly based on its best experience *pBest* and the leader *gBest*. The below equation indicates how k^{th} particle updates its velocity.

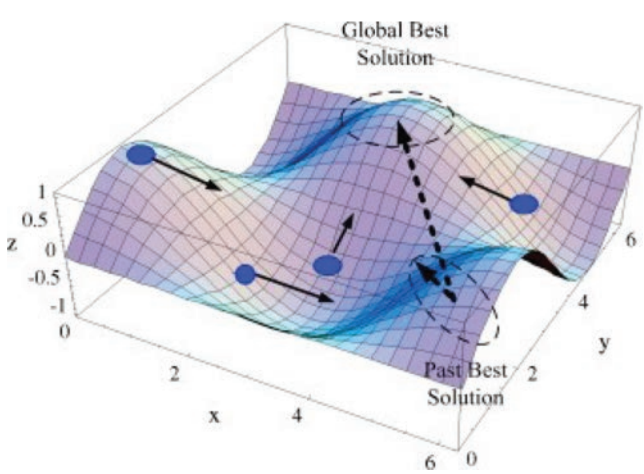

Figure 3 Outstanding particles.

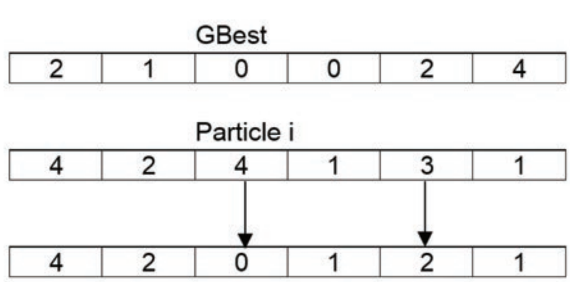

Figure 4 Example of update process.

$$V_k^{(t+1)} = w.V_k^t + c_1 r_1 (pBest_k^t - X_k^t) + c_2 r_2 (gBest_k^t - X_k^t) \quad (16)$$

in which

- c_1 and c_2 are positive acceleration constants, which control the influence of *pBest* and *gBest* on the search process.
- r_1 and r_2 follow uniform distributions in the range of $[0, 1]$.
- w is inertia weight, which usually decreases from a large value ($w_{max} = 0.9$) in the beginning to a small value ($w_{min} = 0.4$) in the last iteration to control the exploration and exploitation abilities of the swarm.

The update of the velocity has been considered an important step since the first generation of development of the PSO-MRP algorithm. Velocity regulation is carried out according to the number of particles that exceed the feasible area during the optimization process. The velocity vector for a particle *i* is updated according to three other vectors: (i) the personal best position of the i^{th} particle; (ii) the best position found over the whole swarm; and (iii) the current position of the particle. It is necessary to reduce the velocities of the particles by speeding up the attenuation rate of the inertia weight and reducing the maximum allowed velocity (see Figure 5).

The new position is updated by

$$X_k^{(t+1)} = X_k^t + max_{i=1..N}\left\{V_k^{t+1}\right\} \quad (17)$$

4.2.5. Stop conditions

For efficient implementation of the PSO-MRP algorithm, we have to determine when to stop iterating. The stop condition of the algorithm can be classified into two techniques:

- Firstly, in real-world problems in which a desirable solution range may be available, the stop condition of the algorithm is based on the closeness of the potential solution to this range;
- Secondly, the iteration can be stopped when a specific number of generations has been reached.

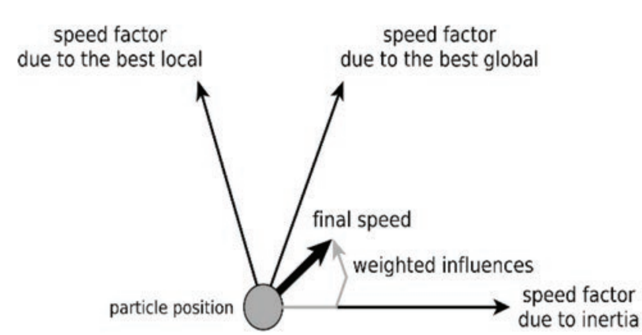

Figure 5 Velocity update process.

4.2.6. Implementation of PSO-MRP algorithm for multi-round procurement problem

The PSO-MRP algorithm is described in Algorithm 1. Details of the implementation of the PSO-MRP algorithm for multi-round procurement can be found in the supplementary material files.

5. EXPERIMENTS AND RESULTS

5.1. Data Sets

A decision support system based on PSO-MRP was developed to define the Nash equilibrium point for multi-round procurement. The system was written in Java and MySQL. In a real-world project, as most of the procurement data used are kept secret, a large-scale data set for an evolutionary algorithm such as PSO-MRP provides a big obstacle. We found three sample data sets for multi-round procurement in big Vietnamese government-funded projects. The experimental results can prove that a Game-theoretic model and PSO-MRP are suitable in solving a critical and real-world problem in this research. Table 5 shows a summary of the data sets used.

5.1.1. Data set structure

Data were organized in a .json file with the following structure:

```
{
"project_id": "",
"inflation_rate": "",
"start_date": "",

"packages": [{
"package_id": "",
"execution_time" : "",
"joined_contractors": [x, xx, xxx],
"products": [{
"product_id": "",
"quantity": ""
}],

"estimated_cost": ""
}],
"product": [{
"product_id": "",
"description": "",
"unit": ""
}],

"contractors": [{
"contractor_id": "",
"description": "",
"capacity": "",
"relationship": "",
```

Algorithm 1: A Novel PSO Algorithm Applied for Multi-Round Procurement Problem (PSO-MRP)

```
input : n is the number of packages, m is the number of stages,
        x_i^j is the number of materials i at time j,
        P_i^j is the cost of materials i at stage j after deducting the discount
        PSO parameters in Table 4;
output: The payoff parameter;
        The fitness gBest;
        The benefit of bidders, project owner;
begin
    InitializePopulation();
    t=0;                                              /* Initialize all particles */
    N=Swarm size;                                     /* iteration */
    Max=Maximum number of iterations;                 /* Number of particles */
    foreach particle_k do
        X_k^0 = Initialize the position of the k^th particle randomly;
        Compute pBest of the k^th particle by Eq.(15);
        gBest = max_{i=1..N} {pBest_i};
        V_k^0 = Initialize the velocity of the k^th particle randomly;
    do
        foreach particle_k do
            Compute fitness value of particle_k by Eq.(15);
            if fitnessvalue > pBest then
                pBest = the fitness value;            /* Update the particle's best position */
            if pBest > gBest then
                pBest = gBest;                        /* Update the global best position */
        foreach particle_k do                         /* Update particle's velocity and position */
            Update the velocity by
            V_k^(t+1) = w.V_k^t + c1 r1 (pBest_k^t - X_k^t) + c2 r2 (gBest_k^t - X_k^t);
            Update the position by
            X_k^(t+1) = X_k^t + max_{i=1..N} {V_k^(t+1)};
        t++;
    while (t > Max or StopCondition = True);
    Compute the benefit of bidders() by Eq.(12);
    Compute the benefit of project owner() by Eq.(10);
    Compute the investor payment() by Eq.(13);
end.
```

```
"products": [{
"product_id": "",
"sell_price": "",
"buy_price": ""
}],
"strategies": [{
"strategy_id": "",
"products": [{
"product_id": "",
"discounts": [{
"from": "",
"to": "",
"rate": ""
}]
}]
}]
}]
}
```

5.1.2. Data set description

The summary description of data set 1 and 2 are shown in Tables 6 and 7. The detailed information can be found in the supplementary material files.

The project of data set 3 had an total estimated cost is 27,654,400 VND. During the project duration (from 1/1/2019 to 21/6/2019), it was divided into six main packages auctioned in different stages. A total of 43 products were assessed in the unit, and five contractors joined the project. The information contains their company names, their qualities, their relationships with the investor, the discount rate and its time-line, the price contractors bought the products for, and the price they sold them for. The summary description of data set 3 is shown in Table 8; for detailed information, see the supplementary material files.

Data set 4 is a large-scale project which needs a large number of bidders, resources, items, proposals, and budgets. The project title was IT application investment project in party agencies of Haiphong city (2019–2020) with 4787 items in procurement, 45 bidders joining in the auction of 1248 rounds, each round including several items in the auction. In total, the problem contained 2584 proposals for 8715 items in packages. The summary description of data set 4 is shown in Table 9; detailed information can be found in the supplementary material files.

5.2. Testing Result

5.2.1. Testing environment

Our experiments were conducted on a computer configured as shown in Table 10.

Table 6 Detailed information of data set 1.

Parameter	Description
Project title	IT application investment project in party agencies of Nghe An province.
Project duration	From 2015–2020
Budget	Project funded with development investment from the provincial budget, investment limit is 39.8 billion
Equipment category	46 items ($H_{01} - H_{46}$)
Bidding information	6 bidders (FPT, HongPhuc, TranAnh, Hanoi ComputerJointStock, Setec, and PhucAnh)

Table 7 Detailed information of data set 2.

Parameter	Description
Project title	IT application investment project in party agencies of Haiphong city (2018–2020).
Project duration	From 2018 to 2020
Budget	Project funded with development investment from the provincial budget, investment limit is 168 billion
Equipment category	100 items ($H_{01} - H_{100}$)
Bidding information	15 bidders

Table 8 Detailed information of data set 3.

Parameter	Description
Project title	IT application investment project in party agencies of Hanoi city.
Project duration	From 1/1/2019 to 21/6/2019
Total estimated cost	27,654,400 VND
Number of products	43
Number of contractors	5

Table 9 Detailed information of data set 4.

Parameter	Description
Project title	IT application investment project in party agencies of Haiphong city (2019–2020)
Project duration	From 1/1/2019 to 31/12/2019
Budget	148,468 billion VND
Equipment category	4787 items ($H_1 - H_{4787}$)
Bidding information	45 bidders ($B_1 - B_{45}$)
Number of proposal	2584 proposals
Number of items in packages	8715 ($PackageID_1 - PackageID_{8715}$)

Table 10 Testing environment.

Parameter	Configuration
CPU	Intel Core i5 6200U (Skylake), 2.7GHz
RAM	8 GB
Operating System	Windows 10 Pro
Programming language	Java 11, json-simple-1.1.1.jar

Table 5 Summary of data sets.

Data set	Equipment Category	Budget	Number of Bidder
1	46 items	39.8 billion	6
2	100 items	168 billion	15
3	43 items	27.6544 million	5
4	4787 items, 1248 rounds, 2584 proposal	148,468 billion	45

5.2.2. Case study 1

In the first experiment, we installed and compared the performance of six algorithms, GDE3 (*The third evolution step of generalized DE*) [29], *Multi-Objective Evolutionary Algorithm* (ϵ-MOEA) [32], *Strength Pareto Evolutionary Algorithm 2* (PESA2) [33], NSGA-III [35], *Nondominated Sorting Genetic Algorithm* (ϵ-NSGA-II) [36], and our PSO-MRP algorithm on data set 1. We compared two criteria: the payoff value of the solution and the runtime. The payoff of the solution was calculated by summing the "goodness" for each corresponding weighting goal. Thus, the payoff value was within [0, 1]; the closer to 0, the better the solution.

After testing the program 10 times, the populations were randomly initialized, including 100 individuals; the maximum number of individuals created for each runtime was 10,000. The comparison of the payoff parameter and the runtime on data set 1 is given in Table 11. We compare the performance of our algorithm and other algorithms, in terms of best, worst, and average, in Figure 6.

Figure 6 shows the difference in convergence and stability of the considered algorithms. The proposed PSO-MRP algorithm met both criteria, showing good convergence in the fitness function and fast execution.

5.2.3. Case study 2

In the second experiment, we performed the same experiment as in case study 1 on data set 2. The comparison of the payoff parameter and the runtime on data set 2 is given in Table 12. We also compare the performance of our algorithm and other algorithms, in terms of the best, worst, and average, in Figure 7.

Figure 7 shows the differences in convergence and stability of the considered algorithms. The proposed algorithm had the best runtime.

5.2.4. Case study 3

After testing the program 10 times, the result of the solution is listed below:

1 Nash Equilibrium found as follows:
2 Number of particles: 10
3 The Nash solution:
4 Chromosome:
5 [4 0 4 2 1 0]

Table 11 Comparison of the payoff parameter and runtime (seconds) on data set 1.

	NSGA-III [35]		ϵ-MOEA [32]		GDE3 [29]		PESA2 [33]		ϵ-NSGAII [36]		PSO-MRP	
Ord	**Payoff**	**Time**	**Payoff**	**Time**	**Payoff**	**Time**	**Payoff**	**Time**	**Payoff**	**Time**	**Payoff**	**Time**
1	0.2961	4.197	0.2755	6.602	0.2805	5.981	0.2741	3.630	0.2744	8.954	0.2805	4.300
2	0.3322	4.178	0.2714	6.633	0.2799	5.992	0.2743	3.051	0.2765	8.921	0.2811	4.369
3	0.2911	4.209	0.2711	6.590	0.2799	5.974	0.2747	3.555	0.2801	8.933	0.2816	4.321
4	0.2966	4.166	0.2706	6.602	0.2811	6.000	0.2751	3.525	0.2809	9.102	0.2796	4.256
5	0.2979	4.178	0.2754	6.613	0.2821	5.981	0.2714	3.617	0.2712	8.990	0.2801	4.273
6	0.3025	4.184	0.2755	6.595	0.2797	6.112	0.2777	3.504	0.2708	9.051	0.2823	4.291
7	0.3105	4.189	0.2723	6.621	0.2797	5.969	0.2750	3.550	0.2714	8.915	0.2841	4.315
8	0.2923	4.171	0.2746	6.608	0.2810	5.980	0.2707	3.544	0.2721	8.956	0.2800	4.324
9	0.2940	4.190	0.2704	6.615	0.2807	5.987	0.2721	3.601	0.2725	8.937	0.2799	4.210
10	0.3025	4.178	0.2705	6.601	0.2805	5.994	0.2733	3.539	0.2708	8.929	0.2796	4.356

Note: NSGA, non-dominated sorting genetic algorithm; MOEA, multi-objective evolutionary algorithm; GDE3 3rd evolution step of generalized DE; PESA2, strength pareto evolutionary algorithm 2; PSO, particle swarm optimization; MRP, multi-round procurement.☒ Reply

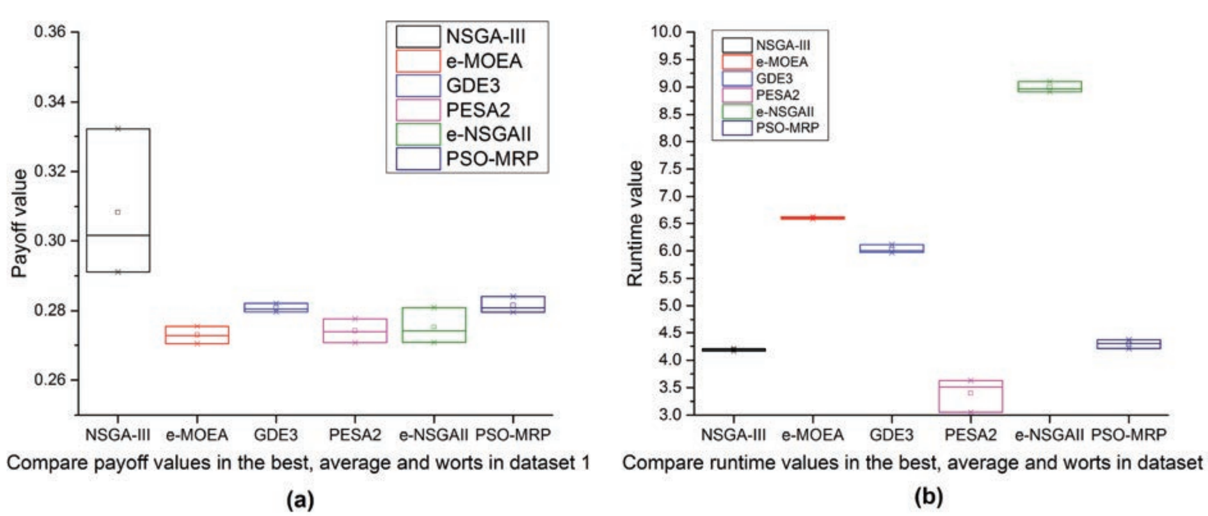

Figure 6 Comparison of payoff and runtime (second) in the best, average, and worst results in data set 1.

Table 12 Comparision of payoff parameter and runtime (seconds) on data set 2.

	NSGA-III [35]		ϵ-MOEA [32]		GDE3 [29]		PESA2 [33]		ϵ-NSGAII [36]		PSO-MRP	
Ord	Payoff	Time	Payoff	Time	Payoff	Time	Payoff	Time	Payoff	Time	Payoff	Time
1	0.2674	5.6390	0.2590	7.6990	0.2555	7.5150	0.2671	5.6690	0.2555	12.4990	0.2569	5.4960
2	0.2654	5.6360	0.2596	7.6960	0.2545	7.5170	0.2702	5.6470	0.2569	12.5010	0.2555	5.4750
3	0.2666	5.6330	0.2610	7.7210	0.2569	7.4860	0.2666	5.6210	0.2518	12.4200	0.2547	5.5020
4	0.2668	5.6230	0.2591	7.7100	0.2518	7.4970	0.2697	5.5870	0.2525	12.4250	0.2603	5.5140
5	0.2697	5.6990	0.2610	7.6550	0.2581	7.5250	0.2653	5.5920	0.2522	12.3870	0.2522	5.4770
6	0.2701	5.5870	0.2586	7.6470	0.2525	7.5290	0.2666	5.6210	0.2557	12.4650	0.2518	5.4210
7	0.2670	5.6210	0.2589	7.6960	0.2533	7.5330	0.2701	5.5880	0.2525	12.4690	0.2555	5.4330
8	0.2655	5.6520	0.2587	7.7100	0.2529	7.5290	0.2655	5.6490	0.2551	12.4700	0.2569	5.4350
9	0.2654	5.6270	0.2590	7.6990	0.2519	7.6000	0.2654	5.6760	0.2543	12.4340	0.2566	5.4150
10	0.2680	5.6190	0.2599	7.6870	0.2520	7.5550	0.2680	5.6960	0.2569	12.4530	0.2589	5.4500

Note: NSGA, non-dominated sorting genetic algorithm; MOEA, multi-objective evolutionary algorithm; GDE3 3rd evolution step of generalized DE; PESA2, strength pareto evolutionary algorithm 2; PSO, particle swarm optimization; MRP, multi-round procurement.

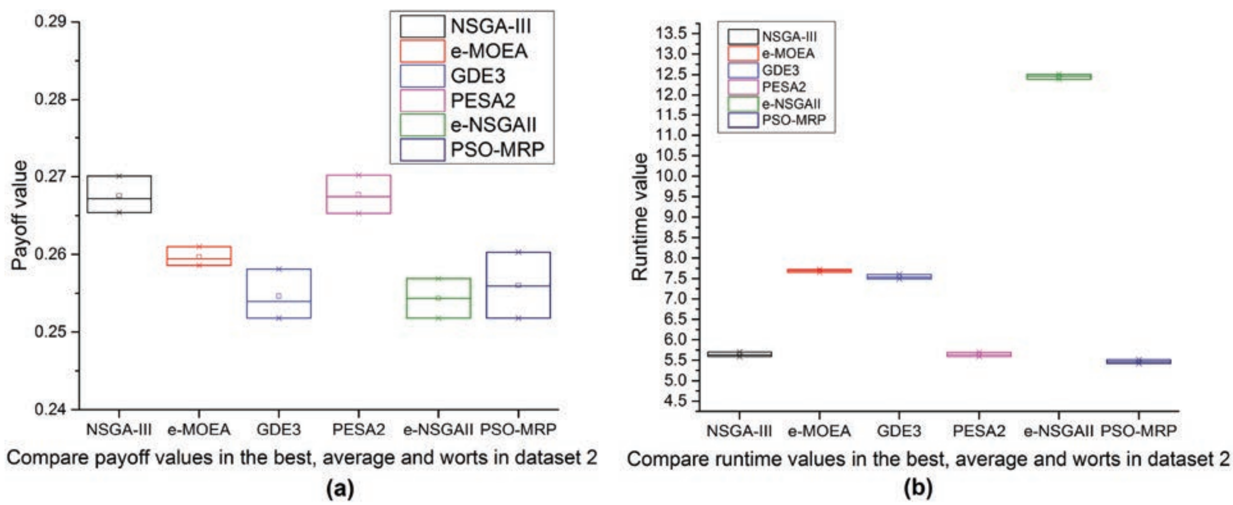

Figure 7 The comparison of payoff and runtime (second) in the best, average, and worst results in data set 2.

6
7 Fitness: 30642322
8 The benefit of bidders: 23624700
9 Bidder0: 128295300
10 Bidder1: 65719300
11 Bidder2: 3999400
12 Bidder3: 0
13 Bidder4: 38233000
14
15 The benefit of project owner: 14472956
16 Investor payment: 13181444
17 Estimated price: 27654400
18
19 Successfully obtained after 0 hours 6 minutes 5 seconds

The comparison of fitness and runtime (minutes) of our algorithm on data set 3 is shown in Table 13. Our proposed algorithm demonstrated stability and convergence of estimated cost and payment values in different experiments.

Our proposed algorithm shows the stability and convergence of values estimated cost and payment in different experiments. Through experiments shows that the values of estimated cost (27,654,400), investor payment (13,181,444), and benefit of project owner (14,472,956) are not changed. However, the distribution of the bidder values will fluctuate, resulting in a fluctuation in the fitness function (see Table 14)

5.2.5. Case study 4

To evaluate the performance and convergence of our algorithm in terms of both the fitness and time functions, we compared the performance between the different meta-heuristic algorithms on a large-scale project in data set 4, which had a large number of bidders, resources, items, proposals, and budgets. The experiment results are listed in Tables 15 and 16, which show that our algorithm had the best performance and quality of Nash equilibrium point when the fitness values were lowest.

Figure 8 shows the differences in convergence and stability of the considered algorithms. The proposed algorithm had both the best runtime and best fitness.

5.2.6. Case study 5

In this experiment, the PSO-MRP algorithm was executed 100 times with randomly initialized populations including 100 individuals, where the maximum number of individuals created for each runtime was 10000. We evaluated the stability of the algorithm

Table 13 Comparision of fitness and runtime (minutes) of our algorithm on data set 3.

			Investor (thousand VND)			Bidder (thousand VND)				
Ord	**Time**	**Fitness**	**Estimated cost**	**Payment**	**Benefit**	**0**	**1**	**2**	**3**	**4**
1	6:05	33476600	27654400	13181444	14472956	2211446	143314	79090	0	157556
2	6:08	29213000	27654400	13181444	14472956	1458516	143314	183402	0	224774
3	5:55	32380516	27654400	13181444	14472956	2211446	283406	39994	0	157556
4	5:51	29300604	27654400	13181444	14472956	1542919	138075	183402	0	157556
5	5:51	28376784	27654400	13181444	14472956	1204395	283406	18340	0	224774
6	6:13	31539874	27654400	13181444	14472956	940112	657193	39994	0	690008
7	6:08	27531258	27654400	13181444	14472956	940112	423862	183402	0	233302
8	6:10	29300604	27654400	13181444	14472956	1542919	138075	183402	0	157556
9	6:28	31750880	27654400	13181444	14472956	1960984	240098	79090	0	224774
10	6:16	28376784	27654400	13181444	14472956	1204395	283406	183402	0	224774

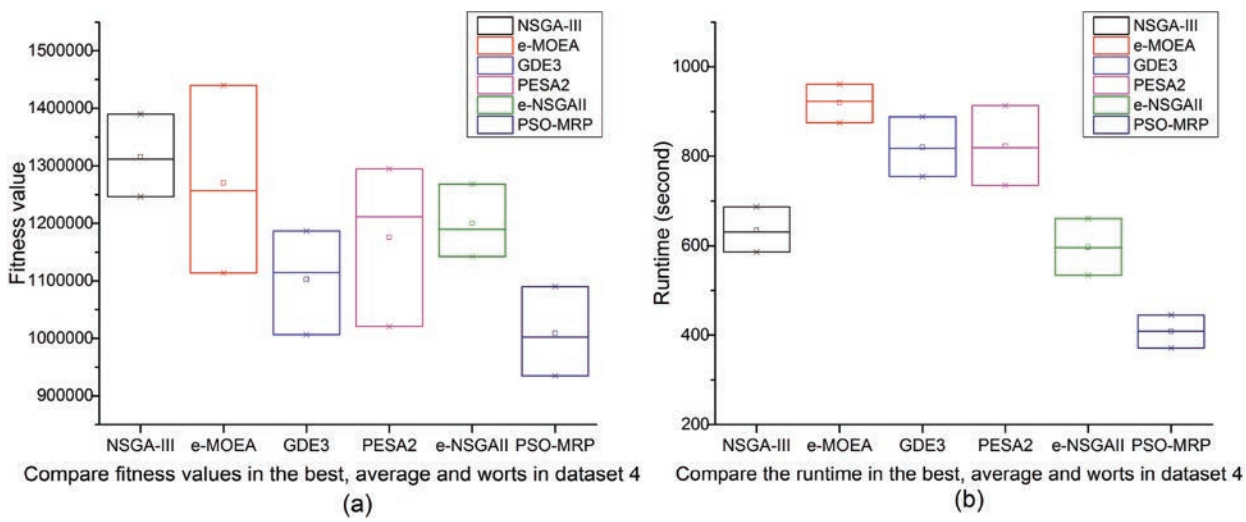

Figure 8 Comparison of fitness and runtime (second) in the best, average, and worst results in data set 4.

Table 14 Analysis and statistical evaluation of time, fitness, bidders' value of PSO-MRP algorithm in data set 3.

	Min	**Max**	**Mean**	**std**
Time	5.51	6.28	5.945	0.29789
Fitness optimal	27531258	33476600	30124690.4	1999915.156
Bidder 0	940112	2211446	1521724.4	474601.161
Bidder 4	138075	657193	273414.9	163915.868
Bidder 2	18340	183402	117351.8	71889.476
Bidder 4	157556	690008	245263	159996.683

based on certain parameters, such as the minimum (*Min*), maximum (*Max*), mean, variance, and standard deviation (*Std*) of fitness, as well as time cost, in Tables 17 and 18.

5.2.7. Discussion

From the experimental results in Tables 4 and 5, the ϵ-NSGA-II algorithm gave the best results with 2 data sets; otherwise, PESA2, NSGA-III, and PSO-MRP had the fastest running times. In addition, we have some comments and evaluations, as follows:

- The order of runtime was similar in the two data sets.
- Both the ϵ-MOEA and ϵ-NSGA-II algorithms were the worst-performing algorithms, due to the ϵ-dominance calculations taking much more time than conventional ones.
- In both data sets, the NSGA-III algorithm gave the worst results, as it very easily became stuck at local solutions.
- The ϵ-NSGA-II algorithm had good results in both data sets; so, while using ϵ-dominance is often costly, in terms of time, it had significant efficiency.
- Both ϵ-MOEA and PESA2 divided the target space into hyperboxes and, so, the algorithm efficiency depends on the width of the target space; therefore, there were differences in the result order, compared to the other algorithms, on the two data sets.
- Similarly, PSO-MRP and GDE3 had in common that the selection was not carried out through a mating process but, instead, was based on an individual transformation (mutation, or change in rules); therefore, the resulting orders had similarities in the two data sets.

From overall observation, the authors drew some conclusions for case study 3:

- The result after each run converged and shared a similar solution.
- Contractors were likely to have similar interests.

Table 15 Comparison of the fitness values of different algorithms on data set 4.

Ord	NSGA-III [35]	ϵ-MOEA [32]	GDE3 [29]	PESA2 [33]	ϵ-NSGAII [36]	PSO-MRP
1	1,307,737	1,251,764	1,028,268	1,269,256	1,240,694	**1,055,274**
2	1,275,423	1,240,213	1,087,182	1,284,698	1,259,906	**953,342**
3	1,280,605	1,440,048	1,048,430	1,289,464	1,165,777	**934,703**
4	1,369,824	1,113,870	1,118,879	1,137,605	1,142,129	**998,258**
5	1,251,295	1,175,737	1,171,253	1,225,694	1,268,246	**1,064,634**
6	1,297,487	1,343,591	1,186,463	1,020,719	1,209,126	**953,237**
7	1,390,135	1,208,916	1,148,664	1,294,479	1,146,635	**1,037,284**
8	1,362,089	1,299,300	1,006,452	1,046,804	1,164,376	**935,569**
9	1,334,285	1,123,940	1,186,586	1,272,498	1,145,861	**998,714**
10	1,246,259	1,368,942	1,164,761	1,273,764	1,156,935	**1,090,015**

NSGA, non-dominated sorting genetic algorithm; MOEA, multi-objective evolutionary algorithm; GDE3 3rd evolution step of generalized DE; PESA2, strength pareto evolutionary algorithm 2; PSO, particle swarm optimization; MRP, multi-round procurement.

Table 16 Comparison of the runtimes of different algorithms on data set 4.

Ord	NSGA-III [35]	ϵ-MOEA [32]	GDE3 [29]	PESA2 [33]	ϵ-NSGAII [36]	PSO-MRP
1	10m 20s	15m 43s	13m24s	12m 59s	9m 12s	**7m 07s**
2	10m 42s	15m 48s	13m37s	14m 36s	10m 07s	**6m 23s**
3	11m 6s	15m 29s	14m 02s	15m 13s	11m 01s	**6m 11s**
4	9m 53s	14m 55s	14m 22s	14m 23s	9m 46s	**7m 25s**
5	9m 48s	15m 17s	13m 45s	12m 15s	9m 22s	**6m 48s**
6	11m 27s	14m48s	13m 51s	12m 23s	10m 56s	**6m 29s**
7	11m 1s	14m 35s	13m 11s	14m 53s	8m 54s	**7m 16s**
8	9m 46s	16m 01s	12m 35s	13m 49s	9m 36s	**7m 13s**
9	10m 31s	15m 24s	12m 41s	13m 08s	9m 57s	**6m 49s**
10	10m 33s	15m 47s	14m 48s	12m 52s	10m 24s	**6m 27s**

NSGA, non-dominated sorting genetic algorithm; MOEA, multi-objective evolutionary algorithm; GDE3 3rd evolution step of generalized DE; PESA2, strength pareto evolutionary algorithm 2; PSO, particle swarm optimization; MRP, multi-round procurement.

Table 17 Analysis and statistical evaluation of convergence of the proposed PSO-MRP algorithm in data sets 1 and 2.

	Payoff value					Time cost (second)				
Dataset	**Min**	**Max**	**Mean**	**Variance**	**Std**	**Min**	**Max**	**Mean**	**Variance**	**Std**
1	0.280	0.284	0.281	2.095E-06	0.0014476	4.210	4.369	4.302	0.0022183	0.0470992
2	0.252	0.260	0.256	7.017E-06	0.0026489	5.415	5.514	5.462	0.0012731	0.0356801

Table 18 Analysis and statistical evaluation of convergence of proposed PSO-MRP algorithm in data sets 3 and 4.

	Fitness optimal				Time cost (second)			
Dataset	**Min**	**Max**	**Mean**	**Std**	**Min**	**Max**	**Mean**	**Std**
3	27,531,258	33,476,600	28,915,472	2315.3072	5.51	6.28	5.945	0.297890733
4	934,703	1,090,015	975,453	57,217.74	371	445	409	25.667

- The choosing decision was based on many other criteria, such as the price, the relationship, and the quality of each bidder. However, more trustworthy contractors were more likely to be chosen; especially bidder 3, which did not have any relationship with the investor.
- The program took some time and depended on the size of the swarm population and the max running fold.

The proposed PSO-MRP algorithm met both criteria, demonstrating good convergence in the fitness function and fast execution. Our algorithm can, thus, provide the best performance in terms of payoff and time taken with limited iterations, as seen in case studies 4 and 5.

6. CONCLUSION

This paper at hand presents a novel approach on how to combine Game theory and PSO algorithm to support decision-making in the multi-round of an auction. In the first phase, we propose a game model which brings all needed information, criteria, and constraints of the problem. In the second phase, the PSO algorithm is utilized with a suitable set of parameters to figure out the Nash equilibrium point. We proved that conflict, such as the competition and cooperation in multi-round procurement, can be resolved by defining a sustainability strategy that brings all participants of the problem into a win-win solution. Additionally, the model was successfully applied in our experiments, demonstrating the usefulness of the Nash equilibrium point for the multi-round procurement issue.

However, restrictions have not vanished, as procurement is typically a secure project. Therefore, the data gathering process encountered many difficulties and information was ultimately lacking. Moreover, the program requires an optimized solution and may need to be run several times to get the best result. Some unexpected risks and issues during the project were not calculated. Therefore, it is still necessary to do more experiments and research to develop an optimized method.

For future research, the authors wish to improve the optimization of the algorithm, reduce the time cost to run the program, and complete the software development.

CONFLICTS OF INTEREST

The authors have declared no conflicts of interest.

ACKNOWLEDGMENTS

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.03-2019.10.

REFERENCES

[1] S. Issa, Y. Tu, A survey in the resource-constrained project and multi-project scheduling problems, J. Project Manag. 5 (2020), 117–138.

[2] S. Hartmann, D. Briskorn, A survey of variants and extensions of the resource-constrained project scheduling problem, Eur. J. Operational Res. 207 (2010), 1–14.

[3] M.A. Vega-Velazquez, A. Garcia-Najera, H. Cervantes, A survey on the software project scheduling problem, Int. J. Prod. Econ. 202 (2018), 145–161.

[4] A. Cerezo-Narvaez, A. Pastor-Fernandez, M. Otero-Mateo, P. Ballesteros-Perez, Integration of cost and work breakdown structures in the management of construction projects, Appl. Sci. 10 (2020), 1386.

[5] A.E. Akgun, Team wisdom in software development projects and its impact on project performance, Int. J. Inf. Manag. 50 (2020), 228–243.

[6] C.T. Cerdeiral, G. Santos, Software project management in high maturity: a systematic literature mapping, J. Syst. Softw. 148 (2019), 56–87.

[7] L. Ji, T. Li, Multi-round procurement auctions with secret reserve prices: theory and evidence, J. Appl. Econ. 23 (2008), 897–923.

[8] S.V. Nagabhushan, D. Sindhu Swamy, Design of two-stage bidding model for supplier selection, Int. J. Innov. Res. Sci. Eng. Technol. 2 (2013), 2676–2681.

[9] D. Wu, Estimation of Procurement Auctions with Secret Reserve Prices Doctoral dissertation, Clemson University, 2015.

[10] O.R. Koppius, M. Kumar, E. van Heck, Electronic multidimensional auctions and the role of information feedback, in Proceedings of the 8th European Conference on Information Systems (ECIS 2000), Vienna, Austria, 2000, pp. 461–468.

[11] L. Ji, Three Essays on Multi-round Procurement Auctions, Doctoral dissertation, Vanderbilt University, 2006.

[12] C. Rao, Y. Zhao, S. Ma, Procurement decision making mechanism of divisible goods based on multi-attribute auction, Electronic Commerce Res. Appl. 11 (2012), 397–406.

[13] D.C. Parkes, J. Kalagnanam, Models for iterative multiattribute procurement auctions, Manag. Sci. 51 (2005), 435–451.

[14] J.F. Nash, Equilibrium points in n-person games, Proc. Natl. Acad. Sci. 36 (1950), 48–49.

[15] M.J. Osborne, An Introduction to Game Theory, vol. 3, no. 3, Oxford University Press, New York, NY, USA, 2004.

[16] R.B. Myerson, Game Theory, Harvard University Press, 2013.

[17] G. Schuh, S. Runge, Applying game theory in procurement. An approach for coping with dynamic conditions in supply chains, Contrib. Game Theory. Manag. 7 (2014), 326–340.

[18] E.D. Goodman, Introduction to genetic algorithms, in Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation, ACM, 2014, pp. 205–226.

[19] A. Ouarda, M. Bouamar, A comparison of evolutionary algorithms: PSO, DE and GA for fuzzy c-partition, ?Int. J. Comput. Appl. 91 (2014), 32–38.

[20] A. Nayyar, D.N. Le, N.G Nguyen, (eds.), ?Advances in swarm intelligence for optimizing problems in computer science, CRC Press, 2018.

[21] R. Hassan, B. Cohanim, O. De Weck, G. Venter, A comparison of particle swarm optimization and the genetic algorithm, in 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, 2005, p. 1897.

[22] N. Yang, X. Liao, W.W. Huang, Decision support for preference elicitation in multi-attribute electronic procurement auctions through an agent-based intermediary, Decis. Support Syst. 57 (2014), 127–138.

[23] M. Bichler, K. Guler, S. Mayer, Split-award procurement auctions-can Bayesian equilibrium strategies predict human bidding behavior in multi-object auctions? Prod. Operations Manag. 24 (2015), 1012–1027.

[24] S. Hartmann, R. Kolisch, Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem, Eur. J. Operational Res. 127 (2000), 394–407.

[25] J. Xiao, X.T. Ao, Y. Tang, Solving software project scheduling problems with ant colony optimization, Comput. Operations Res. 40 (2013), 33–46.

[26] R.Zamani, An evolutionary implicit enumeration procedure for solving the resource-constrained project scheduling problem, Int. Trans. Operational Res. 24 (2017), 1525–1547.

[27] B. Roy, A.K. Sen, A novel metaheuristic approach for resource constrained project scheduling problem, in Soft Computing: Theories and Applications, Springer, Singapore, 2020, pp. 535–544.

[28] R. Pellerin, N. Perrier, F. Berthaut, A survey of hybrid metaheuristics for the resource-constrained project scheduling problem, Eur. J. Operational Res. 280 (2020), 395–416.

[29] S. Kukkonen, J. Lampinen, GDE3: the third evolution step of generalized differential evolution, in 2005 IEEE Congress on Evolutionary Computation, IEEE, Vol. 1, 2005, pp. 443–450.

[30] K. Kim, M. Gen, M.H. Kim, Adaptive genetic algorithms for multi-resource constrained project scheduling problem with multiple modes, Int. J. Innov. Comput. Inf. Control. 2 (2006), 41–49.

[31] M. Wall, Lib: a C++ library of algorithm components. http://lancet.mit.edu/ga/

[32] H. Li, Q. Zhang, Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II, IEEE Trans. Evol. Comput. 13 (2008), 284–302.

[33] K. Deb, A. Pratap, S. Agarwal, T.A.M.T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2012), 182–197.

[34] W. Mkaouer, M. Kessentini, A. Shaout, P. Koligheu, S. Bechikh, K. Deb, A. Ouni, Many-objective software remodularization using NSGA-III, ACM Trans. Softw. Eng. Methodol. 24 (2015), 17.

[35] C. Stylianou, A.S. Andreou, A multi-objective genetic algorithm for intelligent software project scheduling and team staffing, Intell. Decis. Technol.7(2018), 59–80.

[36] K. Deb, M. Mohan, S. Mishra, Evaluating the ε−domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions, Evol. Comput.13(2015), 501–525.

[37] D. Hadka, Moea framework-a free and open source java framework for multiobjective optimization. version 2.11, 2015. http://www.moeaframework.org.

[38] E. Gholizadeh, N.B. Afshar, Solving a Bi-objective resource constrained multi-mode project scheduling problem with regard to payment planning using the NSGA-II algorithm, 2018.

[39] L.Y. Tseng, S.C. Chen, A hybrid metaheuristic for the resource-constrained project scheduling problem, Eur. J. Oper. Res. 175 (2006), 707–721.

[40] M. Rogalska, W. Bozejko, Z. Hejducki, Time/cost optimization using hybrid evolutionary algorithm in construction project scheduling, Automation in Constr.18 (2008), 24–31.

[41] V. Yannibelli, A. Amandi, Project scheduling: a multi-objective evolutionary algorithm that optimizes the effectiveness of human resources and the project makespan, Eng. Optimization, 45 (2013), 45–65.

[42] X. Wu, P. Consoli, L. Minku, G. Ochoa, X. Yao, An evolutionary hyper-heuristic for the software project scheduling problem, in International Conference on Parallel Problem Solving from Nature, Springer, Cham, 2016, pp. 37–47.

[43] F. Zaman, S.M. Elsayed, T. Ray, R.A. Sarkerr, Evolutionary algorithms for finding nash equilibria in electricity markets, IEEE Trans. Evol. Comput. 22 (2017), 536–549.

[44] T.T. Khuat, M.H. Le, A novel hybrid abc-pso algorithm for effort estimation of software projects using agile methodologies, J. Intell. Syst. 27 (2018), 489–506.

[45] P. Jedrzejowicz, A. Skakovski, A population learning algorithm for discrete-continuous scheduling with continuous resource discretisation, In Sixth International Conference on Intelligent Systems Design and Applications, vol. 2, IEEE, 2006, pp. 1153–1158.

[46] M. Jaberi, M. Jaberi, A multi-objective resource-constrained project-scheduling problem using mean field annealing neural networks, J. Math. Comput. Sci. 9 (2014), 228–239.

[47] A. Fahmy, T.M. Hassan, H. Bassioni, Improving RCPSP solutions quality with Stacking Justification - application with particle swarm optimization, Expert Syst. Appl. 41 (2014), 5870–5881.

[48] L. Bukata, P. Šůcha, Z. Hanzálek, Solving the resource constrained project scheduling problem using the parallel tabu search designed for the CUDA platform, J. Parallel. Distrib. Comput. 77 (2015), 58–68.

[49] D. Thiruvady, C. Blum, A.T. Ernst, Maximising the net present value of project schedules using CMSA and parallel ACO, in International Workshop on Hybrid Metaheuristics, Springer, Cham, Switzerland, 2019, pp. 16–30.

[50] B.N.T.Q.T. Huynh, T.L. Nguyen, Research on Genetic Algorithm and Nash Equilibrium in Multi-Round Procurement, in New Trends in Intelligent Software Methodologies, Tools and Techniques: Proceedings of the 16th International Conference SoMeT_17, IOS Press, 297 (2017).pp. 51.

[51] N.T. Nguyen, Q.T. Huynh, T.H.G. Vu, A Bayesian critical path method for managing common risks in software project scheduling, in Proceedings of the Ninth International Symposium on Information and Communication Technology, 2018, pp. 382–388.

[52] B.N. Trinh, H.Q. Thang, X.T. Nguyen, P.C. Luong, N.K. Ho, Applying a unified game-based model in a payment scheduling problem and design of experiments using MOEA Framework, in SoMeT, 2019, pp. 55–68.

[53] T.Q. Huynh, N.B. Trinh, T.X. Nguyen, Nash equilibrium model for conflicts in project management, J. Comput. Sci. Cybern. 35 (2019), 167–184.

[54] Q.T. Huynh, N.T. Nguyen, Probabilistic method for managing common risks in software project scheduling based on program evaluation review technique, Int. J. Inf. Technol. Project Manag. 11 (2020), 77–94.

[55] H. Garg, Reliability, availability and maintainability analysis of industrial systems using PSO and fuzzy methodology, Mapan. 29 (2014), 115–129.

[56] H. Garg, M. Rani, S.P. Sharma, Predicting uncertain behavior and performance analysis of the pulping system in a paper industry using PSO and Fuzzy methodology, in Fuzzy Systems: Concepts, Methodologies, Tools, Applications, IGI Global, 2017, pp. 1070–1109.

[57] H. Garg, A hybrid PSO-GA algorithm for constrained optimization problems, Appl. Math. Comput. 274 (2016), 292–305.

[58] D.N. Le, A New Ant Algorithm for optimal service selection with end-to-end QoS constraints, J. Internet Technol. 18 (2017), 1017–1030.

[59] D.N. Le, G.N. Nguyen, V. Bhateja, S.C. Satapathy, Optimizing feature selection in video-based recognition using Max-Min Ant System for the online video contextual advertisement user-oriented system, J. Comput. Sci. 21(2017), 361–370.