ITC 1/52
Information Technology and Control
Vol. 52 / No. 1 / 2023
pp. 25-36
DOI 10.5755/j01.itc.52.1.31949

Particle Swarm Optimization Method Combined with off Policy Reinforcement Learning Algorithm for the Discovery of High Utility Itemset

Received 2022/07/26 | Accepted after revision 2022/10/08

https://doi.org/10.5755/j01.itc.52.1.31949

HOW TO CITE: Logeswaran, K., Suresh, P., Anandamurugan, S. (2023). Particle Swarm Optimization Method Combined with Off Policy Reinforcement Learning Algorithm for the Discovery of High Utility Itemset. *Information Technology and Control*, 52(1), 25-36. https://doi.org/10.5755/j01.itc.52.1.31949

Particle Swarm Optimization Method Combined with off Policy Reinforcement Learning Algorithm for the Discovery of High Utility Itemset

K. Logeswaran
Department of Artificial Intelligence, Kongu Engineering College, Perundurai, 638060, India

P. Suresh
School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, 632014, India

S. Anandamurugan
Department of Information Technology, Kongu Engineering College, Perundurai, 638060, India

Corresponding author: klogeswaran22@outlook.com

Mining of High Utility Itemset (HUI) is an area of high importance in data mining that involves numerous methodologies for addressing it effectively. When the diversity of items and size of an item is quite vast in the given dataset, then the problem search space that needs to be solved by conventional exact approaches to High Utility Itemset Mining (HUIM) also increases in terms of exponential. This factual issue has made the researchers to choose an alternate yet efficient approaches based on Evolutionary Computation (EC) to solve the HUIM problem. Particle Swarm Optimization (PSO) is an EC-based approach that has drawn the attention of many researchers to unravel different NP-Hard problems in real-time. Variants of PSO techniques have been established in recent years to increase the efficiency of the HUIs mining process. In PSO, the Minimization of execu-

tion time and generation of reasonable decent solutions were greatly influenced by the PSO control parameters namely Acceleration Coefficient (c_1 and c_2) and Inertia Weight (ω). The proposed approach is called Adaptive Particle Swarm Optimization using Reinforcement Learning with Off Policy (APSO-RL$_{OFF}$), which employs the Reinforcement Learning (RL) concept to achieve the adaptive online calibration of PSO control and, in turn, to increase the performance of PSO. The state-of-the-art RL approach called the Q-Learning algorithm is employed in the APSO-RL$_{OFF}$ approach. In RL, state-action utility values are estimated during each episode using Q-Learning. Extensive tests are carried out on four benchmark datasets to evaluate the performance of the suggested technique. An exact approach called HUP-Miner and three EC-based approaches, namely HUPEU-MU-GRAM, HUIM-BPSO, and AGA_RLOFF, are used to relate the performance of the anticipated approach. From the outcome, it is inferred that the performance metrics of APSO-RL$_{OFF}$, namely no of discovered HUIs and execution time, outstrip the previously considered EC computations.

KEYWORDS: Reinforcement Learning, Evolutionary Computation, Particle Swarm Optimization, Execution Time.

1. Introduction

The problem of HUIs was derived from the Frequent Itemset Mining (FIM). In FIM, items having more occurrences are determined using the metrics of Support and Confidence. Many evolutionary meta-heuristic approaches proposed to mine HUIs from the given dataset in past research works. Current research focuses on mining HUIs from transactional dataset using PSO Algorithm [10]. Various parameters namely inertia weight and acceleration coefficient used in PSO determines whether optimal or near-optimal solution was found [12]. Every change in the value of these parameters (raising or lowering) has an impact on the outcome of PSO, either favourably or unfavourably. Choosing the right value for these parameters is a nontrivial task [7]. In reality, controlling values of these parameters is one of the most significant topics of research in PSO.

The current research focuses on online calibration of various control parameters used in the PSO Algorithm. This enables the feature of being adaptive to PSO approach and using the same to discover HUIs from the given dataset. To calibrate the control parameter during each iteration, the quality of solution obtained during past iteration is assessed and this feedback is given as metric to calibrate the control parameter during next iteration. Here comes the role of Reinforcement Learning. RL approaches namely Q-Learning is used to assess the quality of solution in past iterations and generates the feedback from the assessment [14]. This feedback is used as metric to calibrate the control parameters of evolutionary approaches which will impact the quality of solution generated during the next iteration. The novelty of the proposed research is to use the RL algorithm called Q-Learning to adaptively calibrate the control parameter of PSO and using it for mining HUIs from the given transactional dataset. The main contribution of the proposed research work is divided into Initialization and Particle Encoding, Discovery of Primary P_{best} and G_{best}, and APSO-RL$_{OFF}$ Q-Table Training and Testing.

1.1 Background

Transactional datasets are the one which contains the information about an any entity with predefined pattern. This pattern is used to describe the corresponding item in an transactional dataset. The importance and usefulness of an item is determined by the corresponding patterns or also called as properties of an item. Determining a correlation among the item and usefulness of an item in such a transactional dataset is an significant crucial task in the area of knowledge discovery. In past research, this problem is addressed by the FIM where it takes the frequency of an item in a transactional dataset to measure the usefulness of an item. However, this is not appropriate when an item contains other important information namely profit, self time and cost apart from the frequency component. HUI mining gives entry to this research area where it measures the utility of an item by taking into account of all information about the item. Utility denotes the importance or usefulness of an item. When search space of mining process increases then

exhaustive time is required for any conventional exact HUI mining algorithms to search the appropriate output. In recent years, many researchers have proposed different types of Evolutionary Algorithms (EA) to address this polynomial time dependent problem of HUI mining. In the proposed research work one of the EA approach called PSO algorithm is used.

2. Review of Literature

This section intends to deliver a broad, thorough, and well-organized review of the state-of-the-art evolutionary based algorithms for mining HUIs from supplied datasets.

An optimization model called grey wolf optimization algorithm, which dubs the behaviour of grey wolf and applied to unravel the HUI mining problems by utilising five different Boolean operations [16]. The Boolean operators are used to convert continuous GWO to modified GWO. To maintain the uniqueness of the searching algorithm, the flow of GWO is retained, but the operators are modified to handle Boolean values. The redesigned GWO lacks a boundary check process that assures the whole search is conducted within the border region.

Many techniques follows the procedure of using the minimum utility threshold constraint at initial stage and then compute high utility itemsets. It is difficult task to determine the minimal utility threshold. Imprecise HUI is generated when the selection of minimum utility threshold is inappropriate. A strategy built on twofold particle swarm optimization (BPSO) for mining high utility itemsets by discarding the establishment a marginal utility threshold at initial stage is introduced in [3]. The technique produces itemsets using the BPSO algorithm and outputs them all as a utility-sorted list. As a post-processing step, itemset in the sorted list are evaluated based on a minimum utility criterion. In [9], HUIs will be mined using a discrete PSO called HUIM-BPSO algorithm. In this approach, traditional PSO is integrated with upgraded OR/NOR tree structure, TWU model and sigmoid updating scheme.

To extract the High Utility Itemset, the suggested approach in [6] considers a dataset of distinct advertisements that are presented on a single website. For the proposed system, the high utility itemset (HUI) are advertising through a conversion proportion greater than the predefined conversion proportion threshold value, which is utilised to anticipate the sorts of ads that generate more income.

A method to mine HUIs constructed on the MOEA/D context towards the achievement of a suitable trade-off between population convergence and variety throughout the generation process is proposed in [19]. MOEA/D is a general method that divides a multi-objective optimization issue into many single-objective optimization sub problems. Then, it employs a population-based technique to simultaneously optimise these sub complications.

An Enhanced Genetic Algorithm to extract HUIs called (HUIM-IGA) was proposed in [20] which overcomes the drawback of time consuming problem associated with conventional Genetic Algorithm for mining HUIs. In HUIM-IGA, four new unique strategies have been introduced. A neighbourhood exploration technique is offered as the first novelty to increase search efficiency for HUIs. A population mixture upkeep technique is used in the suggested HUIM-IGA to eliminate missing HUIs. In addition, an individual repair mechanism is planned to eliminate incorrect groupings for detecting HUIs. Furthermore, an exclusive technique is used to prevent the loss of HUIs. To segmenting high-value clients based on monetary worth, Online Retail dataset is used in [8]. HUIM is performed using two variants of Differential Evolutionary approach and by without involving Downward Closure Property. When the database size goes larger the performance of the existing approach deteriorates in performance. To overcome the downside, Frequent Pattern List is used to represent the portion of database. A proposed approach called Adaptive Artificial Bee Colony is applied of Frequent Pattern List to mine the frequent itemset.

Different topologies appear to make PSO a problem-associated procedure, however in [18], QLPSO, a problem-independent PSO that incorporates a RL mechanism, is developed. Popular off policy RL algorithm called Q-Learning is integrated with PSO to dynamically change the neighbourhood of a current particle. The most difficult challenge for a group of portable robots in an unfamiliar situation is to plan the best path and learn the environmental factors. For a group of portable robots to identify the best track in an unfamiliar situation and learn the situation, a

mixture of particle swarm optimization (PSO) and RL was proposed in [15]. Off policy RL algorithm called Q-Learning is used to facilitate the mobile robot to easily learn the environment in a short time.

3. Methodology

The proposed work is divided into four sections, namely Initialization and Particle Encoding Phase, Q-Table grounded parameter tuning Learning phase and online regulator of PSO Parameter Tuning and Assessment Phase. Adaptive Particle Swarm Optimization using Reinforcement Learning with Off policy (APSO_RL$_{OFF}$) approach is intended to intelligently adjust the tuning parameters of PSO namely Inertia Weight (ω) and Acceleration Coefficients C_1 and C_2. State, action, Q table, particles and reward are the five main components of the APSO_RL$_{OFF}$ algorithm. A reward function contributes to the update of Q-Table values, aids a particle in determining the appropriate action based on its current state.

Algorithm 1: Initialization

Input: $D_T \rightarrow$ Transactional Database;

$P_T \rightarrow$ Profit Table;

$\delta \rightarrow$ Minimum Utility Threshold;

$N \rightarrow$ No of Particles in each iteration;

Output:

$1-HTWUIs \rightarrow$ 1-itemset with high transaction weighted utility;

1 ***for each*** $T_r \in D_T$ ***do***

2 ***for each*** $I_t \subseteq T_r$ ***do***

3 $tu(T_r) = q(I_t, T_r) \times P_T(I_t)$

4 Compute $twu(I_t) = \sum_{I_t \subseteq T_r} tu(T_r)$

5 Compute $TU = \sum_{T_R \in D_T} ut(T_r)$

6 Compute $1-HTWUIs = \{I_t \mid twu(I_t) \geq TU \times \delta\}$

7 Set $P_{size} = |1-HTWUIs|$

// individual particle size

Initially, TWU model is used to discover the 1-itemset with high transaction weighted utility will drastically minimize the amount of invalid itemset by utilizing the property called transaction-weighted downward closure (TWDC).

Initialization and Particle Encoding Phase

To begin, the utility of each transaction's items is determined as a function of the transaction's utility ($tu(T_r)$). $q(I_t, T_r)$ and $P_T(I_t)$ represents the quantity of an item and profit of an item respectively in a database. In a transaction, the summation of utility of all item denotes the transaction-weighted utility of that particular transaction and this enables to determine the item's upper-bound value. This method is used to find the upper-lift significance of an item. Any item can be classified as $1-HTWUIs$ when that item has transaction-weighted utility fewer than the least utility count.Algorithm-1 illustrates the computation steps involved in finding and individual particle size. Considering that we are solving the HUIM problem using the PSO algorithm, it is intended to define the particles engaged in the PSO population as an itemset. To facilitate the representation of itemset as a particle, a binary vector of 0's and 1's was used in the place of particle and hence PSO is converted into binary PSO. In a binary PSO each particle is signified as a binary vector comprising 0^s and 1^s, which signal the absence of an item in an itemset and its existence, respectively.

Discovery of Primary P_{best} and G_{best}

In a population each particle can be characterized by the two factors called velocity ($V_i^j(t)$) and position ($X_j^i(t)$). After fitness evaluation of each particle, the $X_j^i(t)$ of each particle is initialized randomly between 0 to 1, and of each particle is initialized to either 0 or 1 (Line 3 and 4 in Algorithm-2). The Line 7 in Algorithm-1 determines the size of each particle $P_{size} = |1-HTWUIs|$. The fitness of each particle can be calculated using Equation (1). In Equation (1), represent the particle that act as a subset of Transaction Database. Find the particles having fitness greater than Minimum Utility Threshold and store them in HUIs list (Line 5 and 6 in Algorithm-2).

$$F(P) = u(P) = \sum_{P \subseteq D_T \wedge X \subseteq P} u(X, P). \quad (1)$$

Find the particle's Personnel Best in each generation $(P'_{best}(t))$ and particle's Personnel Best among all generation (P_{best}). Find the particle's Global Best in each generation $(G'_{best}(t))$ and particle's Global Best among all generation (G_{best}). Algorithm-2 in illustrate the initial computation of P_{best} and G_{best}.

Algorithm 2: Find Initial P_{best} and G_{best}

Input: $D_T \rightarrow$ Transactional Database;

$P_T \rightarrow$ Profit Table;

$\delta \rightarrow$ Minimum Utility Threshold;

$N \rightarrow$ No of particles in each iteration;

$P_{size} \rightarrow$ Individual Particle Size;

$TU \rightarrow$ Total Utility;

Output: $P_{best} \rightarrow$ Personal Best;

$G_{best} \rightarrow$ Global Best;

$HUIs \rightarrow$ High Utility Itemset;

1 ***while*** $i \leftarrow 1, N$ particles ***do***

2 ***while*** $j \leftarrow 1, P_{size}$ ***do***

3 Initialize $X_j^i(t)$ = either 1 or 0;

4 Initialize $V_i^j(t)$ = rand(0,1);

5 ***if*** $F_f(X_i(t)) \geq TU \times \delta$ ***then***

6 $HUIs \leftarrow$ GetItem $(X_i(t)) \cup HUIs$ //Find HUIs

7 Find $P'_{best}(t)$ of each N particle

8 Update P_{best}

9 Find $G'_{best}(t)$ among N P'_{best} particles

10 Update G_{best}

APSO-RL$_{OFF}$ Q-Table Training

In Q-Table training phase, the following necessary parameters have to be configured based on the problem under consideration. State Set (S_s), Action Set (A_s), Reward Function (R_F), Action Selection Strategy (A_f), and Q-Table $(Q_{table}[S_s, A_s])$ are the set of parameters that needs to be configured as described below. After which $(Q_{table}[S_s, A_s])$ is allowed to train for fixed number of iterations.

Configuration of States Set

The states of an agent depend on the environment in which the agent is working. In the proposed research work, the environment is considered as the solution space, and the state of an agent is defined by the fitness of the particle in the solution space. In our problem domain, the solution space can be assessed be repeatedly assessing the fitness of a particle, this leads the way for State set definition. State set of a Q-Table can be determined by assessing the fitness of particle using fitness function. The fitness of each particle is assessed from the itemset in solution space, and further segmented into 4 categories based on Equation(2). So only four states are derived for the proposed research. The S_s denotes the state set and contains the four states such as $Smallest(st)$, $Smaller(sr)$, $L\arg er(lr)$, and $L\arg est(lt)$. From the Equation (2), S_s can be constructed as shown in Equation (3). During each iteration the particle falls under any one these S_s based on the condition specified in Equation (2). (F_r) denotes the fitness difference between global best particle (G_{best}) and current particle. (ΔF) denotes the opportunity of objective space boundary which is formulated as fitness value variance between global worst and global best particle.

$$F(S_s) = \begin{cases} Smallest(st) & 0 \leq F_r < 0.25\Delta F \\ Smaller(sr) & 0.25\Delta F \leq F_r < 0.5\Delta F \\ L\arg er(lr) & 0.5\Delta F \leq F_r < 0.75\Delta F \\ L\arg est(lt) & F_r \geq 0.75\Delta F \end{cases} \quad (2)$$

$$S_s = \{st, sr, lr, lt\}. \quad (3)$$

Configuration of Action Set

For any problem domain, the agent's action can be determined by the type of task that needs to be performed to move from current state to next state. In the given problem space, the agent's action is configured from the control parameters of the PSO algorithm. Any agent can have more than one action for the given state and it can be denoted as Action Set A_s. The agent's actions are defined as the task of selecting the values for the control parameters. Three control parameters such as Inertia Weight, Cognitive Acceleration Coefficient and Social Acceleration Coefficient are taken in to consideration. Based on the benchmark standard values[12] and [4] of these control pa-

rameters, the values for each control parameters are classified into four range of values as shown in the Table-1. In each iteration, selecting the appropriate values for each control parameters is defined as action of the agent and an agent has a choice of choosing any one action from the A_s as shown in Equation (4) represented as A_s .

$$A_s = \{A_1, A_2, A_3, A_4\}. \quad (4)$$

Table 1
Configuration of Action set

Control Parameters \ Action Set	A1	A2	A3	A4
Inertia Weight (ω)	1-0.81	0.8-0.61	0.6-0.4	0.4<
Cognitive Acceleration Coefficient (cc)	2.5-2.1	2-1	0.99-0.5	0.5<
Social Acceleration Coefficient (sc)	0.5-0.9	1-2	2.1-2.5	2.5>

Q-Table Training

Create a $Q_{table}[n_s \times s_s]$ whose dimension is equal to $n_s \times n_a$ where n_s denotes the number of states in the problem space and n_a denotes the number of action in the problem space. Initially Q_{table} values are initialized to zero. S_s elements are designated as row label of Q_{table} and A_s elements are designated as column label of Q_{table} as shown in Figure 1.

Figure 1
Q-Table

S_s \ A_s	A_1	A_2	A_3	A_4
st	...			...
sr		$Q(sr, A_2)$		
lr				
lt	...			...

During the training phase of Q-Table, the initial values are updated in every episode of RL algorithm. Select a particle from N particles and calculate the Relative Fitness (F_r) of particle using Equation (5). Calculate the Scope of Objective space (ΔF) using Equation (6).

$$F_r = f(y^t) - f(G_{best}) \quad (5)$$

$$\Delta F = f(G_{best}) - f(G_{worst}). \quad (6)$$

By applying Equation (2) constraints on F_r and ΔF , calculate the state of the particle. For the calculated state, choose an appropriate action from the Q-Table using $\varepsilon - greedy$ Action Selection Strategy $(\varepsilon - ASS)$ as shown in Equation (7). In Equation (7), p is chosen from the range of random numbers between 0 to 1. ε value always range between 0 to 1. To promote exploration at the beginning of training phase ε is initialized with higher degree of value between 0 to 1. Later the ε value is annealed to small constant between the range 0 to 1. Figure-1 illustrate the Q-Table.

$$\pi(s_t, a_t) = \begin{cases} \max_a Q(s_t, a) & \varepsilon \geq p \\ Random(a) & \varepsilon < p \end{cases}. \quad (7)$$

After using $\varepsilon - ASS$, the agent will be provided with appropriate values for the control parameters of PSO. By using this control parameter values, the velocity and position of each item in a particle is updated using Equation (8) and Equation (9) respectively.

$$V_j^i(t+1) = \omega \times V_i^j(t) + [C_1 \times rand() \times [{}_j^i P_{best} - y_j^i(t)]] + [C_2 \times rand() \times [{}^j G_{best} - y_j^i(t)]] \quad (8)$$

$$X_j^i(t+1) = \begin{cases} 1 & random() < sigmoid(v_j^i(t+1) = \dfrac{1}{1+e^{-v_j^i(t+1)}} \\ 0 & otherwise \end{cases}. \quad (9)$$

For an any action taken by the agent calculate the reward (R_f) by assessing the fitness of the particle as shown in Equation (10). In Equation (10), $\max_f f(y^t)$ denote the individual best fitness of t^{th} generation and $\max_f f(y^{t-1})$ denotes the individual fitness of $(t-1)^{th}$ generation.

$$R_f = \frac{\max_f f(y^t) - \max_f f(y^{t-1})}{\max_f f(y^{t-1})}. \quad (10)$$

Algorithm 3: Q-Table Training

Input: $D_T \rightarrow$ Transactional Database;
$\delta \rightarrow$ Minimum Utility Threshold;
$N \rightarrow$ No of particles in each iteration;
$P_{size} \rightarrow$ Individual Particle Size;
$TU \rightarrow$ Total Utility;
$\pi \rightarrow \varepsilon - GreedyPolicy$;
$HUIs \rightarrow$ High Utility Itemset;
$S_s \rightarrow$ State Set;
$A_s \rightarrow$ Action Set;
$\mathrm{I}_{\max} \rightarrow$ Maximum Iteration;
Output: $Q_{table}[S_s, A_s]$

```
set t=0;// current iteration
for i = 0,3 do
    for j = 0,3 do
        set Q_table[S_s^i, A_s^j] =0
while t < I_max do
    while i ← 1,N particles do
        calculate the F_r of X_i(t)
        //using Equation (5)//
        F_r - the relative fitness of
        //current particle
        calculate the ΔF
//using Equation (6)
        // ΔF represent the fitness
        //difference
        choose a state ( S_t ) from S_s
        //using Equation (2)
        determine the action ( A_t )
        //using Equation (7)
        while j ← 1,P_size do
        update the velocity V_i^j (t+1)
        //using Equation (8)
        update the velocity X_i^j (t+1)
        //using Equation (9)
        compute the reward ( R_f )
        //using Equation (10)
        estimate the Q value
//using Equation (11)
        apprise the Q value in Q_table
    t = t+1
```

Compute the Q value for the current iteration using Equation (11) and update it in the Q-table corresponding to the current iteration's state and action taken. In Equation (11), α denotes the learning rate that profits a random value in the assortment between 0 to 1 and γ denotes the discount rate that profits a random value in the assortment between 0 to 1. Algorithm-3 illustrate the Q-Table train phase.

$$Q(s_{t+1}, a_{t+1}) = Q(s_t, a_t)(1-\alpha) + \alpha[R_f + \gamma \max_a Q(s_{t+1}, a_t)] \quad . \tag{11}$$

Q-Table Testing

In Q-Table testing phase, mining of HUIs using PSO with control parameters being calibrated using Q-Learning algorithm is performed. Select a particle from N particles and calculate the Relative Fitness (F_r) of particle using Equation (5). Calculate the Scope of Objective space (ΔF) using Equation (6). By applying Equation (2) constraints on F_r and ΔF, calculate the state of the particle. For the calculated state, choose an appropriate action with highest Q value from the Q-Table. After action selection, the agent will be provided with appropriate standards for the control parameters of PSO.

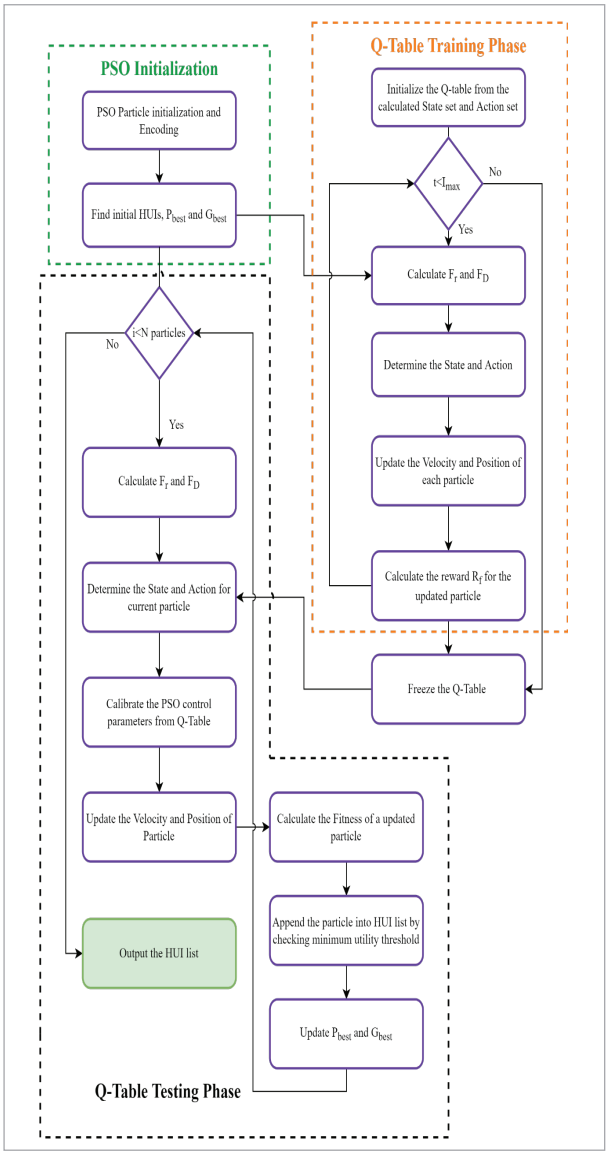
Algorithm 4: Proposed APSO-RL$_{\text{OFF}}$

Input: $D_T \rightarrow$ Transactional Database;
$\delta \rightarrow$ Minimum Utility Threshold;
$N \rightarrow$ No of particles in each iteration;
$P_{size} \rightarrow$ Individual Particle Size;
$TU \rightarrow$ Total Utility; $\pi \rightarrow \varepsilon - GreedyPolicy$;
$HUIs \rightarrow$ High Utility Itemset;
$S_s \rightarrow$ State Set; $A_s \rightarrow$ Action Set;
$\mathrm{I}_{\square} \rightarrow$ Maximum Iteration;
$P_{best} \rightarrow$ Particles Personal Best;
$G_{best\rightarrow}$ Particle Global best;
$Q_{table}[S_s, A_s] \rightarrow$ Q-Table;
Output: $HUIs \rightarrow$ High Utility Itemset;

```
while t < I_max do
    while i ← 1,N particles do
        F ← F_f(X_i(t))
        calculate the F_r of X_i(t)
//using Equation (5)
        choose a state ( S_t ) for X_i(t)
        //using Equation (2)
        determine the action ( A_t )
//using Equation (7)
        while j ← 1,P_size do
        update the velocity V_i^j (t+1)
        //using Equation (8)
```

update the position $X_i^j(t+1)$ //*using Equation (9)*
if $F_f(x_i(t)) \geq TU \times \delta$ *then*
$HUIs \leftarrow GetItem(X_i(t+1)) \cup HUIs$
find $P_{best}^{'}(t)$ of each M particle
apprise P_{best}
find $G_{best}^{'}(t)$ between M $P_{best}^{'}(t)$ -particles
update G_{best}
set $t \leftarrow t+1$

Figure 2
APSO_RL$_{OFF}$work flow



By using this control parameter values, the velocity and position of each item in a particle is updated using Equation (8) and Equation (9) respectively. Compute the fitness of the updated particle using Equation (12). In a transaction L_j , the utility of an item X_i with quantity Q_{ij} and unit profit ρ_i can be represented as $\upsilon(X_i, L_j)$ and it can be calculated as shown in Equation (13).

$$F_f(x_i(t)) = \upsilon(X_i, L_j) \tag{12}$$

$$\upsilon(X_i, L_j) = \rho_i \times Q_{ij}\,. \tag{13}$$

By applying constraint in below Equation (14), current particle is checked for the HUIs. If the condition in Equation (14) is satisfied then the particle is appended to HUI list. Update the P_{best} and G_{best} . Algorithm-4 illustrate the Q-Table testing phase in the proposed APSO-RL$_{OFF}$ algorithm. Work flow of APSO_RL$_{OFF}$ is shown in Figure-2. The proposed method starts with PSO initialization phase in which P_{best} and G_{best} are updated using Q-Table training phase. Later HUIs are mined using Q-Table testing phase.

$$HUI = \{X \mid \upsilon(X) \geq TU \times \alpha\}\,. \tag{14}$$

4. Results and Discussion

This section contains a description of the data sets, the setup and results of a series of experiments, as well as debates and interpretations based on the findings. To examine the performance of the suggested APSO_RL$_{OFF}$ approach, a series of tests were carried out.

Dataset

Four real-world data sets were utilised to evaluate efficiency: Connect, Chess, Accident, and Mushroom. For data mining, these databases are accessible from the SPMF repository[2]. The accident dataset only accounts for 10% of the total due to its size [17], [11]. All datasets are text documents with the.txt suffix. Chess dataset is compiled from enumerated complicated areas of chess endgames. The domain elements of stated game theoretical values are described as table in Endgame database. The gain or loss of a current move and the number of move still required to assure minimum optimal game can be deter-

mined from the theoretical saved value of the game.

Connect-4 game has 8-ply position that is permissible when succeeding move is not forced and none of the players have yet won.

The Mushroom dataset provides information about imaginary samples of mushrooms in North America and it was collected from Society of Audubon. The dataset comprises of family of mushroom from Lepiota and Agaricus which has 23 variety of species. From the dataset, probability of edibility and harmfulness of mushroom has to inferred.

Accident 10% is statistics from the National Institute of Statistics for the years 1991-2000 which contains the injuries sustained in Flanders as a result of traffic accidents (NIS). The traffic accident data contains a wealth of information regarding the many circumstances that led to the injuries. Dataset is described in Table 2.

Table 2
Dataset Characteristics

S. No.	Dataset	Connect	Chess	Mushroom	Accidents_10%
1	Items Count	130	76	119	469
2	Tran. Avg. Length	43	37	23	34
3	Transaction Count	67557	3196	8124	34018

Experimental Configuration

All studies were performed on the 64-bit Windows 11 operating system. with an Intel® Core™ i7-10750H CPU running at 2.6 GHz with Turbo Boost upto 5.0 GHz and 12GBDDR3L 204-Pin RAM memory with data speed of 2933 MHz. Python language in Anaconda-Spyder framework was used for deployment purpose.

For all experimentation, population size at initial stage was set at 30 and the termination criterion was set at 2500 iterations. Ten separate runs are used to collect experimental results, notably the execution time necessary to identify HUIs and the number of discovered HUIs. The learning parameter values for Q-Learning namely discount rate (γ) and learning rate (α) are obtained from the [1]and set to 0.2 and 0.75 respectively.

Performance Metrics

Two performance metrics namely Number of Discovered HUIs and Execution time are used to evaluate the robustness and competency of the proposed APSO_RL$_{OFF}$ approach. Substantial experiments are conduction and the obtained results are assessed against the above said performance metrics by comparing the proposed APSO_RL$_{OFF}$ approach with existing EC based HUIM approach. AGA_RL$_{OFF}$[14], HUIM-BPSO [9], and HUPE$_{UMU}$-GRAM [5] are the three existing state-of-the-art HUIM methods based on meta heuristic EC.

Execution Time

Figure 3 compares the execution time of the proposed method with existing approaches. Execution time determines the efficiency of the approach by measuring the time taken to accomplish the expected output. The differing proportion of HUIs that can be determined in a stipulated execution time can be dignified by adjusting the minimal utility threshold (δ). The execution time is dignified in millisecond, and δ is expressed as a percentage (%). According to Figure 3, APSO_RL$_{OFF}$ performs poorly compared to the other three algorithms by spending a large amount of runtime for the Connect, chess, Mushroom and Accident 10%, datasets. The reason for the poor performance is because the combination of RL with POS causes the algorithm to spend time for training the Q-table.

In relations to average runtime on chess dataset, proposed APSO_RL$_{OFF}$ was compared to the existing approach, and it was discovered that APSO RL$_{OFF}$ underperforms HUPE$_{UMU}$-GRAM by 26% and HUIM-BPSO by 23%. However, APSO_RL$_{OFF}$ outperforms AGA_RL$_{OFF}$ by more than 18%.

Number of HUIs Revealed

By default, discovery of exact number of all HUIs cannot be guaranteed by the any EC based HUIM algorithms. Enactment of any EC based HUIM approach can be determined by the evaluation metrics namely number of HUIs revealed. It shows a crucial part in it. Actual number of genuine and complete HUIs can be extracted for the given dataset using precise approach called Two-Phase method [13].

Figure 3
Execution in relation to minimum utility threshold versions for Connect (a), Chess (b), Mushroom (c) and Accident_10% (d)

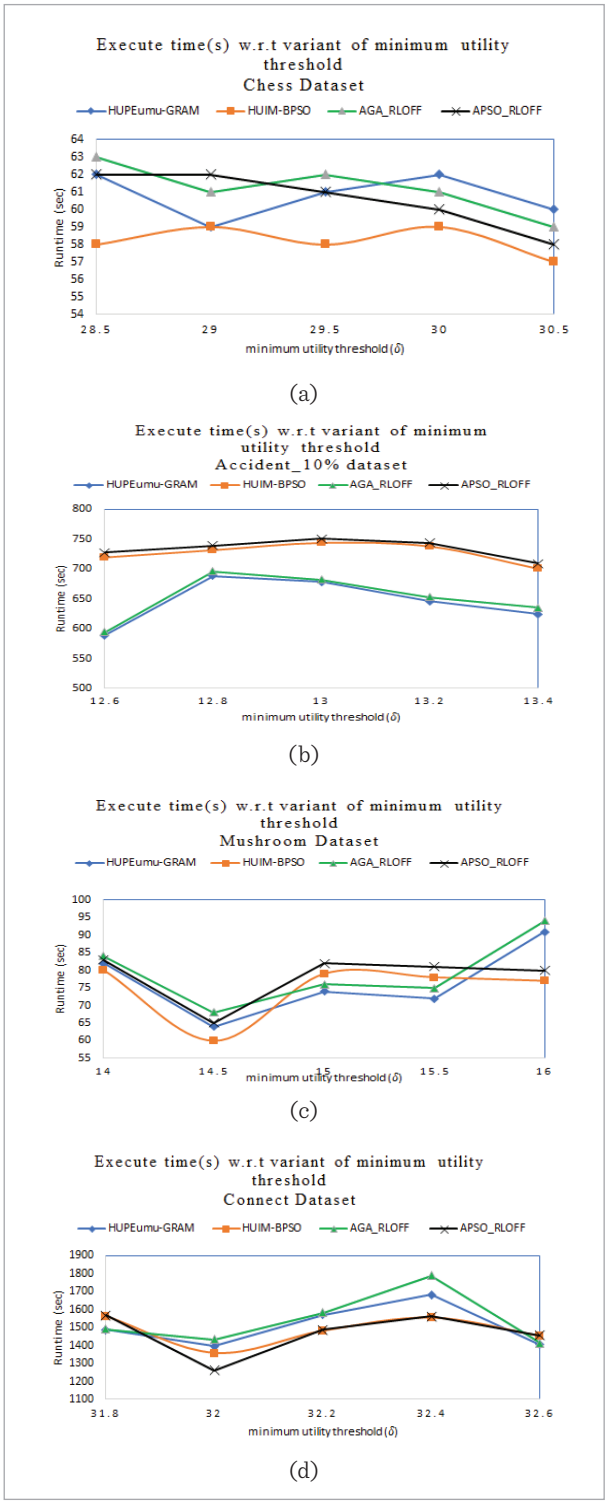
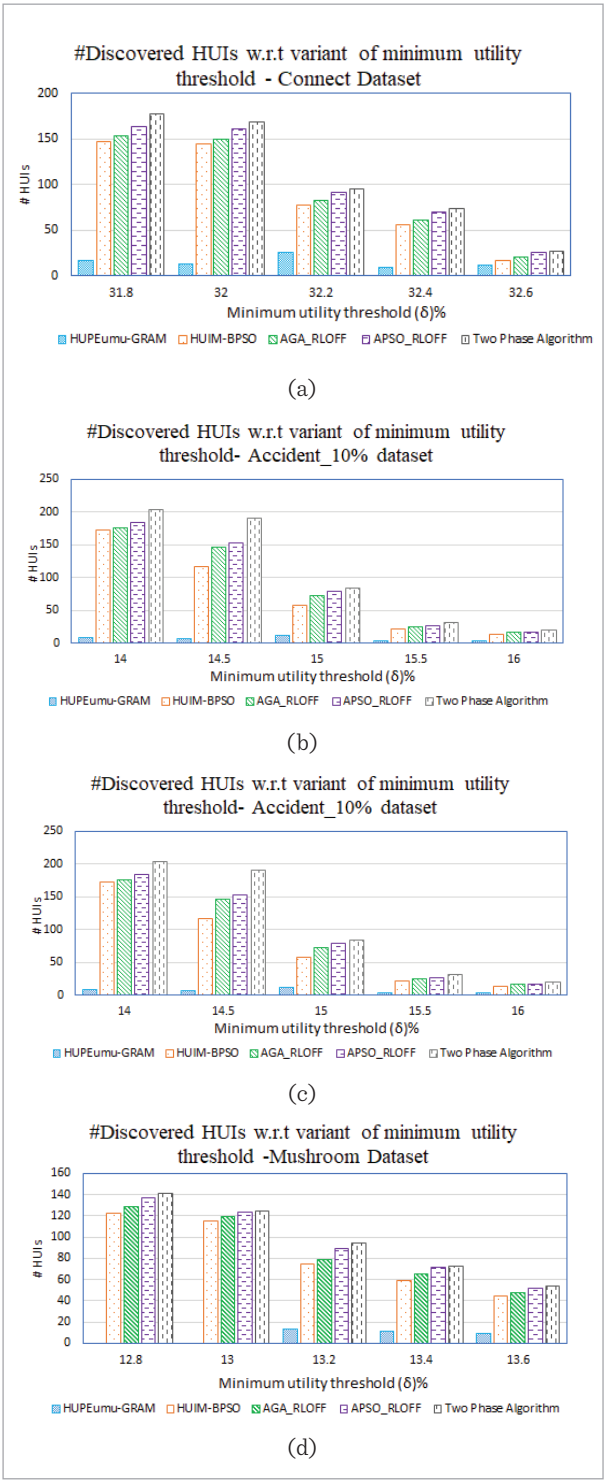


Figure 4
The number of exposed HUIs in relation to the minimum utility threshold for Connect (a), Chess (b), Mushroom (c) and Accident_10% (d)



The effectiveness of the recently proposed EC based APSO RL_{OFF} and three existing bio-inspired $HUPE_{UMU}$-GRAM [16], HUIM-BPSO [7], and AGA_RL_{OFF} was evaluated in this section by comparing the quantity of HUIs identified in Two-Phase method.

Experiment is done with each dataset to evaluate the HUIs discovering capability of proposed APSO_RL_{OFF} by the comparing with existing algorithm. Along horizontal axis, the δ is varied by increasing with 0.2 unit. From the trial, it is examined that quantity of discovered HUIs has been significantly improved in the APSO_RL_{OFF}. (see Figure 4). The APSO_RL_{OFF} discovers 89.6 percent, 85.4 percent, 98.7 percent, and 96.8 percent of the total number of HUIs on the chess, Accident 10%, Mushroom, and Connect datasets, respectively.

5. Conclusion

To address the problem of HUIM, numerous conventional meta-heuristic EC have been proposed in past recent years. However, in those approaches the control parameters were not appropriately initialized to constant value using standard benchmark range and those control parameters values are not problem specific. In the current research, problem specific and feedback based calibration of PSO control parameters called PSO_RL_{OFF} is deployed to mine the HUIs from the given dataset. In APSO_RL_{OFF}, control parameters namely ω, $_1$ and C_2 of PSO were carefully calibrated using off policy RL algorithm called Q-Learning. From the experimental results, it was observed that HUIM problem was efficiently handled by APSO_RL_{OFF}. In the learning phase of APSO_RL_{OFF}, Q-table values are updated during each episode and in testing phase, PSO control parameters are calibrated using Q-Table values.

Yardstick dataset namely Mushroom, Chess, Connect and Accident_10% were used. Experiment was performed on those dataset using stat-of-the-art EC methods namely $HUPE_{UMU}$-GRAM, HUIM BPSO, AGA_RL_{OFF} and the proposed technique APSO_$RL_{OFF}$to examine the performance of proposed APSO_RL_{OFF}. From the result, it was examined that APSO_RL_{OFF} can achieve significant improvement in the accuracy solution for most of the problem space. Even though APSO_RL_{OFF} performs better, it is having a limitation of consuming more time for execution process. To resolve this limitation alternated approach to Q-leering training phase will be introduced in future work. In the future, we will perform a thorough examination to uphold RL and further algorithms for diverse combinatorial optimization problems.

References

1. Alipour, R., Feizi, D., Balafar, M. A. A Hybrid Algorithm Using a Genetic Algorithm and Multiagent Reinforcement Learning Heuristic to Solve the Traveling Salesman Problem. Neural Computing and Applications, 2018, 30(9), 2935-2951. https://doi.org/10.1007/s00521-017-2880-4
2. Fournier-Viger, L., Gomariz, G., Soltani, D., Lam, H. T. The SPMF Open-Source Data Mining Library Version 2. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2016, 9853 LNCS, 36-40. https://doi.org/10.1007/978-3-319-46131-1_8
3. Gunawan, W., Pulungan, R. A BPSO-based Method for High-Utility Itemset Mining Without Minimum Utility Threshold. Knowledge-Based Systems, 2020, 190(xxxx), 105164. https://doi.org/10.1016/j.knosys.2019.105164
4. Harrison, E., Ombuki-Berman, B. M. An Adaptive Particle Swarm Optimization Algorithm Based on Optimal Parameter Regions. In 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Nov. 2017, 2018 January, 1-8. https://doi.org/10.1109/SSCI.2017.8285342
5. Kannimuthu, S., Premalatha, K. Discovery of High Utility Itemsets Using Genetic Algorithm with Ranked Mutation. Applied Artificial Intelligence, 2014, 28(4), 337-359. https://doi.org/10.1080/08839514.2014.891839
6. Keerthi, M., Anitha, J., Agrawal, S., Varghese, T. Mining High Utility Itemset for Online Ad Placement Using Particle Swarm Optimization Algorithm. In Advances in Intelligent Systems and Computing, 1108 AISC, Springer, 2020, 555-562. https://doi.org/10.1007/978-3-030-37218-7_63
7. Krevelen, R. V. Genetic Algorithm Control Using Reinforcement Learning--Introducing the Auto-Tune and Auto-Control ({ATAC}) framework, 2007, doi: 10.13140/RG.2.1.3351.1446.

8. Krishna, G. J., Ravi, V. High Utility Itemset Mining Using Binary Differential Evolution: An Application to Customer Segmentation. Expert Systems with Applications, 2021, 181(March 2020), 115122. https://doi.org/10.1016/j.eswa.2021.115122
9. Lin, J. C.-W., Yang, L., Fournier-Viger, P., Hong, T.-P., Voznak, M. A Binary PSO Approach to Mine High-Utility Itemsets. Soft Computing, 2017, 21(17), 5103-5121. https://doi.org/10.1007/s00500-016-2106-1
10. Lin, J. C.-W., Yang, L., Fournier-Viger, P., Wu, J. M.-T., Hong, T.-P., Wang, L. S.-L., Zhan, J. Mining High-Utility Itemsets Based on Particle Swarm Optimization. Engineering Applications of Artificial Intelligence, 2016, 55, 320-330. https://doi.org/10.1016/j.engappai.2016.07.006
11. Lin, J. C.-W., Yang, L., Fournier-Viger, P., Hong, T.-P., Voznak, M. A Binary PSO Approach to Mine High-Utility Itemsets. Soft Computing, 2017, 21(17), 5103-5121. https://doi.org/10.1007/s00500-016-2106-1
12. Liu, Y., Lu, H., Cheng, S., Shi, Y. An Adaptive Online Parameter Control Algorithm for Particle Swarm Optimization Based on Reinforcement Learning. In 2019 IEEE Congress on Evolutionary Computation (CEC), June 2019, 815-822. https://doi.org/10.1109/CEC.2019.8790035
13. Liu, Y., Liao, W.-K., Choudhary, A. A Two-Phase Algorithm for Fast Discovery of High Utility Itemsets. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2005, 3518 LNAI, 689-695. https://doi.org/10.1007/11430919_79
14. Logeswaran, K., Suresh, P. High Utility Itemset Mining Using Genetic Algorithm Assimilated with off Policy Reinforcement Learning to Adaptively Calibrate Crossover Operation. Computational Intelligence, 2021. https://doi.org/10.1111/coin.12490
15. Meerza, S. I. A., Islam, M., M. Uzzal, Q-Learning Based Particle Swarm Optimization Algorithm for Optimal Path Planning of Swarm of Mobile Robots. In 1st International Conference on Advances in Science, Engineering and Robotics Technology 2019, ICASERT 2019, May 2019, 1-5. https://doi.org/10.1109/ICASERT.2019.8934450
16. Pazhaniraja, N., Sountharrajan, S., Sathis Kumar, B. High utility itemset mining: a Boolean operators-based modified grey wolf optimization algorithm, Soft Computing, 2020, 0123456789(21), 16691-16704. https://doi.org/10.1007/s00500-020-05123-z
17. Song, W., Huang, C. Mining High Utility Itemsets Using Bio-Inspired Algorithms: A Diverse Optimal Value Framework, IEEE Access, 2018, 6, 19568-19582. https://doi.org/10.1109/ACCESS.2018.2819162
18. Xu, Y., Pi, D. A Reinforcement Learning-Based Communication Topology in Particle Swarm Optimization. Neural Computing and Applications, 2020, 32(14), 10007-10032. https://doi.org/10.1007/s00521-019-04527-9
19. Zhang, L., Fu, G., Cheng, F., Qiu, J., Su, Y. A Multi-Objective Evolutionary Approach for Mining Frequent and High Utility Itemsets. Applied Soft Computing, 2018, 62, 974-986. https://doi.org/10.1016/j.asoc.2017.09.033
20. Zhang, Q., Fang, W., Sun, J., Wang, Q. Improved Genetic Algorithm for High-Utility Itemset Mining. IEEE Access, 2019, 7, 176799-176813. https://doi.org/10.1109/ACCESS.2019.2958150