IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 6, 2018, accepted June 1, 2018, date of publication June 25, 2018, date of current version July 25, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2846604

Optimized and Frequent Subgraphs: How Are They Related?

SAIF UR REHMAN[1], SOHAIL ASGHAR[2], (Member, IEEE), AND SIMON JAMES FONG[3]

[1]Department of Computer Science, Abasyn University, Islamabad 44000, Pakistan
[2]Department of Computer Science, COMSATS Institute of Information Technology, Islamabad 44000, Pakistan
[3]Computer and Information Science Department, University of Macau, Macau 853, China

Corresponding author: Saif Ur Rehman (saifi.ur.rehman@gmail.com)

This work was supported in part by the Nature-Inspired Computing and Metaheuristics Algorithms for Optimizing Data Mining Performance through the University of Macau under Grant MYRG2016-00069-FST and in part by the Scalable Data Stream Mining Methodology: Stream-based Holistic Analytics and Reasoning in Parallel through the FDCT Macau, under Grant FDCT/126/2014/A3.

ABSTRACT Frequent subgraph mining (FSM) is one of the most challenging tasks in graph mining. FSM consists of applying the data mining algorithms to extract interesting, unexpected, and useful graph patterns from the graphs. It also aspires to offer a richer apprehension of the given graph data. FSM has been applied to many domains, such as graphical data management and knowledge discovery, social network analysis, Bioinformatics, and security. In this context, a large number of techniques have been suggested to deal with the graph data, with the objective to extract the frequently occurring graph patterns. Such patterns are called frequent subgraph patterns (FSPs). FSPs are extracted using the traditional support threshold parameter. However, there exists no specialized scheme to decide the discovered FSPs are optimized as well. Thus, the aim of this paper is to suggest an optimization strategy to uncover the association between the frequent and the optimized subgraph patterns. For exploring the existence of the potential association between the FSPs and the optimized subgraph, a Particle Swarm Optimization technique is suggested. This relationship will be very handy to reduce the FSPs, by choosing those FSPs which were also discovered as optimized FSPs. Different experiments are performed using benchmark graph data sets to validate the existence of the aforementioned relationship between the optimized and the frequent FSPs.

INDEX TERMS Data mining, graph pattern mining, social network analysis, frequent subgraph patterns, optimized graph patterns.

I. INTRODUCTION

In numerous applications, such as the World Wide Web, Bioinformatics, Social, Technological and Communication Networks, data are usually represented by graphs [1]–[4]. With the increasing demand for the analysis of a large amount of the graph data, Graph Mining (GM) has become an active and most significant research area [5]–[7]. The goal of GM is to extract the hidden knowledge from a single large graph or a set of small graphs in an effort to comprehend its key features [8]–[10]. GM research has been further subdivided into following subareas: graph indexing [11], [12], frequent subgraph mining [13]–[18], graph classification [19]–[21], graph searching [8], [22], [23], graph clustering [24]–[26], approximate graph pattern mining [4], [27], [28], optimal graph pattern mining [29].

FSM is one of the well-known and researched topics in the graph mining domain [5], [6], [30]–[33]. In simple words, FSM aims to extract all the subgraph patterns from the given input graph dataset, whose occurrences in the graph dataset are above the user supplied threshold value. Such subgraph patterns are termed as frequent subgraphs (FSGs). Whereas the number of occurrences of the subgraph is calculated using the supports measure [34]. For example, Figure 1 shows a sample graph database.

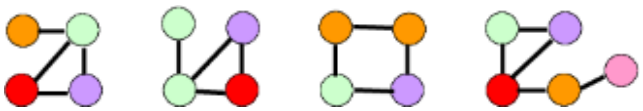

FIGURE 1. A sample graph database.

Now, if the support threshold value is assumed to be 3, then the possible FSGs which can be mined from the Figure 1, are listed in Figure 2.

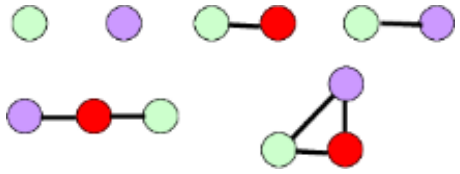

FIGURE 2. Sample mined FSGs patterns.

The problem formulations for FSM have been classified into two main approaches [2]: graph transaction based FSM and a single large graph based FSM. In the former class, the input graph data contain a set of small to medium-size graphs, which are called transactions [36] while in the later FSM class a single input graph data are used (i.e. One very large graph, graph with hundreds of thousands nodes).

It is widely accepted in the literature that FSM techniques are classified into two categories: (1) Apriori-based approaches; and (2) pattern growth-based approaches [6], [10], [30], [31], [35]. These two categories are similar in spirit to counterparts found in association rule mining, namely the Apriori algorithm and pattern-growth algorithm [34] respectively. Both of these approaches aim to identify the frequently occurring subgraph patterns from a given collection of small graph sets or within one large graph. These two approaches are different from each other in the way they mine the FSPs.

In the last few decades, numerous FSM techniques developed in both approaches such as, SPIN [20], gSpan [64], CloseGraph [65], Mofa [13], Subdigger [71], LC-mine [72], FSP [73], FS3 [74], AGM [75], Gaston [62], and Margin [76]. However, when such FSM algorithms are applied to more substantial domains, including image mining, text mining and social network mining, the computational complexity becomes critically very high due to the combinatorial explosion, encountered with respect to the number of possible FSPs [47], [76]. Therefore, many existing approaches to FSM cannot cope with large graph datasets [10], [17], [30], [31], [35], [62]. Moreover, mostly FSM techniques mine a prohibitively large number of FSPs during the mining process and there is no specialized method which can confirm that the discovered FSPs are optimized FSPs [35], [62], [74].

Optimization is one of the most challenging research areas spanning across the fields of computer science, operations research, and engineering [37], [38]. It can be defined as the process by which an optimum is achieved. The term swarm intelligence comes from the field of artificial intelligence, in which ants, insects or bird behavior are analyzed [39], [40]. This natural behavior of different swarms is adopted in the field of computer science in order to solve various optimization problems. In particular, swarm intelligence algorithms refer to the branch of optimization algorithms that simulate and model the imprecision, randomized, and stochastic features of these physical, chemical, or biological elements in arriving at marvelous solutions. Some of the known swarm intelligence techniques used in solving optimization problems are ant colony optimization (ACO) [41], [42], particle swarm optimization (PSO) [43], firefly optimization (FFA) [44], honey bees mating optimization for TSP (HBMOTSP) [45], African buffalo optimization (ABO) [46], bat algorithm (BA) [47], genetic algorithm (GA) [48], adaptive simulated annealing with greedy search (ASA-GS) [49]. In this study, we are using PSO, originally proposed in [50]. Therefore, in the next section a primer on PSO is presented in details.

TABLE 1. Key notations used in this paper.

Symbol	Definition
G(V,E)	A graph with vertices and edges represented by V and E respectively
C_B (i)	Betweenness centrality of a node *i* in G
C_C (i)	Closeness centrality of a node *i* in G
C_D (i)	Degree centrality of a node *i* in G
FSP	Frequent subgraph pattern
GD	A graph database
GM	Graph mining
Opt-SGP	Optimized subgraph pattern
PSO	Particle swarm optimization
Pos_i^t	Position of the particle *i* at time *t*
Vel_i^t	Velocity of the particle *i* at time *t*

In this exploratory study, we are interested to establish a relationship between two types of patterns, the frequent and optimized graph patterns. To the best of our knowledge, there is not a single study available on the exploration of the association between the frequent and optimized graph patterns. For the relationship between two kinds of patterns, we have proposed a PSO based optimization technique, which takes a graph dataset as an input and returns an optimized set of graphs based on the nodes. We also have performed different experiments to validate the relationship between the frequent and the optimized graph patterns. By establishing this relationship we can reduce the number of FSPs discovered (only those FSPs will be considered in the final result set which are also observed as optimized). The contribution of this study may thus be summarized as follows: (1) proposal of optimization method to graph dataset; (2) devising a novel fitness function, which is based on the basic characteristics of the graph structure; (3) discovering the potential relationship between the frequent and optimized graph patterns; (4) experimental evaluation of the proposed optimization method using different graph datasets. A list of notations used in this paper is given in Table 1.

The structure of this paper is as follow. In the next section, we have defined the basic working principle of the PSO. Problem formulation, for the FSPs discovery and optimized graph patterns, along with the relevant constraints are highlighted in Section III. Section IV discusses the Results and Discussion, which is followed by the concluding remarks.

II. A PRIMER ON PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) is now one of the most commonly adopted optimization techniques [43], [51]–[53].

Algorithm 1: : PSO $(S, MaxIter, Dim, \mathscr{R})$

Input:S =size of the swarm, $MaxIter$= maximum number of iterations, Dim = problem dimensions,

Output:$\mathscr{R}$, best result

```
foreach particle p in S
    foreach dimension d in Dim
x_i ← Rand(X_max , X_min)
v_i ← Rand(V_max , V_min)
    next
p_best (i) ← x_i
 if (f(p_best(i)) < f(g_best) then
g_best ← p_best(i)
 end if
next
do
   foreach particle p in S
      if (f(x_i) < f(p_best (i)) then
p_best (i) ← x_i
      end if
      if (f(p_best(i)) < f(g_best) then
g_best ← p_best(i)
      end if
     next
   foreach particle p in S
      foreach dimension d in Dim
         updatex_i using (2)
         updatev_iusing (1)
        next
   next
until MaxIter is completed
```

FIGURE 3. Steps involved in the PSO algorithm [43].

In a PSO procedure, a swarm of particles is kept and each particle in the swarm searches the optimum of a function termed as the fitness function. It keeps track of the best position it has found and the best position discovered by the complete swarm of particles. The best position discovered by a particle is called the local best position of the particle and the best position founded by the whole swarm is called the global best position. The domain of the fitness function is called the search space. Guided by the local best position and the global best position found so far, particles move over the search space in search for an optimum. The advantage of using the PSO optimization method is that it does not rely explicitly on the gradient of the optimization problem. Moreover, as a population-based metaheuristic, PSO has the advantages of robustness, effectiveness, and simplicity as compared to the other swarm intelligent approaches such as GA and ACO [54], [55]. In Figure 3, a simple PSO procedure is outlined.

Let Pos_i^t and Vel_i^t are the initial position and the velocity vector respectively, of a particle i at a given time t. Let Pos_i^{pbestt} represent the particle local best position and let Pos^{gbest} represent the global best position found over all the particles in the whole swarm up to the current time t. The velocity and position vectors of particle i at time $t+1$ can be calculated by using the equations given in (1) and (2) respectively,

$$Vel_i^{t+1} = \omega Vel_i^t + C_1 R_1 (Pos_i^{pbestt} - Pos_i^t) + C_2 R_2 (Pos^{gbestt} - Pos_i^t) \quad (1)$$

$$Pos_i^{t+1} = Pos_i^t + Vel_i^{t+1} \quad (2)$$

In Equation (1), ω represents the inertia weight, Vel_i^t denotes the velocity of the particle i at time t; C_1, C_2 are called acceleration coefficients (or the behavioral parameters) representing the weighting of the stochastic acceleration terms that haul apiece particle toward the Pos_i^{pbestt} and Pos^{gbestt} positions. R_1, R_2 random numbers between $\{0, 1\}$; Pos_i^t position of the particle$i$ at time t. In the velocity updating process, the value of the parameters $\{\omega, C_1, C_2\}$ should be determined in advance. The selection of the appropriate inertia weight may offer stability between the global and the local exploitation, and thus results in an inferior number of iterations in order to output the best possible/optimum solution. In general, to improve the convergence characteristics, the ω factor is designed to decrease linearly [50]{Shi, 1999 #312;Lee, 2008 #311}, descending from to, as follows:

$$\omega^{\mathrm{k}} = \omega_{\max} - \frac{\omega_{\max} - \omega_{\max}}{\mathrm{MaxIter}} \times \mathrm{k} \quad (3)$$

In this study, a PSO based optimization procedure is proposed for the optimization of a given graph dataset to Optimized Subgraph Patterns (opt-SGP). In the proposed optimized PSO procedure, each graph in the dataset is initialized as a swarm of random particles. Each particle stores the position information of each vertex in the graph. All particles automatically update their positions and velocities in the searching process for the optimal node set discovery, thus resulting in an opt-SGP. Each particle has a fitness value representing the importance of the node in the graph. The definition of the proposed fitness function is given on the basis of the different important characteristics of the graph node identified from the literature [35], [56], [57].

III. PROPOSED METHOD

In this section, we have described the procedure of extracting the FSPs and the opt-SGPs. For extracting the FSPs, we have used our own earlier proposed FSM framework, called A-RAFF (A-RAnked Frequent pattern-growth Framework) [58]. For the mining of the Opt-SGPs, we have used the proposed PSO based optimization strategy. These two procedures are highlighted in this section.

A. A-RAFF FRAMEWORK: EXTRACTION OF FREQUENT SUBGRAPH PATTERNS

For the extraction of FSPs, we have used our own earlier proposed FSM framework called A-RAFF, which was discussed in [58]. A-RAFF worked with the labeled undirected graph datasets. The goal of A-RAFF framework is to discover a small collection of ranked FSGs patterns from a database of labeled input graphs. A-RAFF used the basic characteristics of the pattern-growth scheme to discover the FSPs, which solely works on the divide and conquer strategy. A-RAFF is based on pattern-growth as pattern-growth discovers the entire set of FSGs patterns without involving costly operation of candidate generation during the mining process. A-RAFF

Algorithm 2: A-RAFF ($\mathbb{GD}, \sigma, \mathcal{RF}$)
Input: $\mathbb{GD}$ = a graphs dataset of the labelled undirected graphs
σ= minimum threshold
Output: $\mathcal{RF}$, a set of ranked frequent subgraphs
1 count all the features and put in the feature set **"F "**
2 **for each** feature f in the **F**-set **do**
3 Identify most informative graph features
4 **repeat** until all features are checked
5 partition the graph using KaHIP Tool based on an **F**-set [25]
6 $\mathcal{F} \leftarrow \emptyset\mathcal{F}$ denotes the frequent subgraph patterns set
7 $\mathcal{F} \leftarrow$discovered 1-frequent subgraphs patterns
8 **for each** graph $g_i\_partition \in G_i$**do**
9 FSPs ($\mathbb{GD}, g_i\_partition, \sigma, \mathcal{F}$)
10 **end**
11 FSP-Rank ($\mathcal{F}$)
12 **return** $\mathcal{RF}$

FIGURE 4. A-RAFF algorithm.

algorithm is displayed in Figure 4 and further details can be found in [58].

B. PROPOSED PSO BASED OPTIMIZATION PROCEDURE FOR THE DISCOVERY OF OPTIMIZED SUBGRAPHS

İn this section, the proposed PSO based optimization scheme to graph structure is presented. The identified graph structure characteristics are also defined which will be considered in designing the proposed fitness function.

The key to designing a good PSO procedure is to determine the structure of the particles. A good structure makes the problem simple and intuitive. In the proposed work, different graphs are used as an input to the procedure. Each graph pattern is mapped to a particle swarm, where the composition of each particle swarms is the vertex (each vertex is used as a particle). The following method is adopted for structuring the particle. Let *GD* represents the graph database, which can be mathematically written as,

$$GD = \sum_{i=0}^{n} G_i = (G_1, G_2, \ldots., G_n) \tag{4}$$

The structure of each individual G is given by.

$$G_i = (V, E, \sum\nolimits_V, \sum\nolimits_E, l) \tag{5}$$

Where V is a set of vertices, $E \subseteq V \times V$ is a set of edges, $\sum_V$ is a set of vertex labels, and $\sum_E$ is a set of edge labels respectively in the graph dataset. The l is the label defines the mappings $V \rightarrow \sum_V$ and $E \rightarrow \sum_E$. As defined earlier, in the proposed PSO procedure, each G_i corresponds to swarm and the set of vertex in each G is mapped to particles. Hence, mathematically, the structure of the particle swarm, S, can be modeled as follows,

$$S = \left(\left(P_{v1}^t, P_{v2}^t, \ldots., P_{vN}^t\right), Pos^{gBest}, f^{gBest}, N, D, t\right) \tag{6}$$

In Equation (6), P_{vi}^t represents a particle; Pos^{gBest} and f^{gBest} represents the global best position and global best value of the fitness function discovered by the swarm; N is the total number of particles in the swarm; and D represents the dimensions of the particle and t represents the current time.

In the proposed PSO based optimization procedure, as an individual vertex is assumed to be a particle, therefore, the structure of each particle, P_{vi}, is designed using different characteristics of the vertex found in the literature [28], [35], [39], [59], [60]. In the particle structure formulation, following vertex characteristics are adopted.

1) VERTEX DEGREE

Vertex degree determines the number of vertices which are adjacent to the given vertex in a graph [39], [59]. It is denoted by $d\ (v)$. For directed graphs, the vertex degree is the sum of in-degree and out-degree of a particular node. The out-degree of a vertex in a graph is the connection of a particle with other particles in the pattern. It is the sum of the row values in the adjacency matrix of a graph. Mathematically,

$$O_{di} = \sum\nolimits_j G_{ji} \tag{7}$$

Where, $j\ and\ i \in V$,

$$G_{ji} = \begin{cases} 1, & if\ vertex\ j\ is\ connected\ i \\ 0, & vertex\ j\ is\ not\ connected\ i \end{cases}$$

The in-degree of a vertex can be defined as the degree at which the other particle is communicating with a single particle in the pattern. It can be calculated as,

$$I_{di} = \sum\nolimits_j G_{ij} \tag{8}$$

Where, $i\ and\ j \in V$,

$$G_{ij} = \begin{cases} 1, & if\ vertex\ i\ is\ connected\ j \\ 0, & vertex\ i\ is\ not\ connected\ j \end{cases}$$

Since, in this paper, we are focusing on undirected graphs, therefore $G_{ji} = G_{ij}$.

2) DEGREE CENTRALITY

It is an important feature of the vertex and it refers to the ratio of the number of edges attached to the maximum number of possible edges which can be associated with a specific node [30]. It is denoted by C_D and mathematically it can be calculated as follows,

$$C_D(i) = d_i(P_{vi})/n - 1 \tag{9}$$

The value of $C_D(i)$ shows the significance of a node. The larger the value of C_D, the more significant the corresponding node is. Degree centrality refers to the ability of a specific node in the network to directly acquire network flow content and its significance and influence in the network [39], [40], [59].

3) CLOSENESS CENTRALITY

Closeness is considered as one of the fundamental models in a topological space. It can be defined as the measure of how long the information can spread from a given node to another reachable node in the network [28], [39], [60]. The smaller the value of closeness centrality, the smaller is the spread time.

Intuitively, in case of two sets, they are said to be close to each other if they are neighbors. It can be computed as,

$$C_c(i) = \frac{1}{\sum_{j=1}^{N} l(i,j)} \tag{10}$$

$l(i, j)$ is the distance between vertex *i and j* in the given graph G and it is known as a geodesic. The closeness centrality of each node takes a Boolean value, either 0 or 1. It measures how fast information spread from a given node to another reachable node in the network. The normalized version of the closeness centrality C_c can be calculated by Equation (11),

$$C_c(i) = \frac{N-1}{C_c(i)} \tag{11}$$

Where N represents the total vertex found in the graph G. The larger the value of C_c, the closer the node lies in the network or graph center. It shows that the node occupies an important position in the network graph.

4) BETWEENNESS CENTRALITY

In graph theory, betweenness centrality is a measurement of centrality on shortest paths or it is the load capacity between other nodes in the given network. For every pair of nodes in a connected graph, there exists at least one shortest path between the nodes. Mathematically,

$$C_B(i) = \frac{\sum_{j \neq k} \sigma_{jk}(i)}{\sigma_{jk}} \tag{12}$$

Where, σ_{jk} shows the total number of shortest paths from node j to node k and $\sigma_{jk}(i)$ is the number of shortest paths from node j to node k which passes through node i. The betweenness value for each node i is normalized by dividing the excluded number of node pairs,

$$C_B(i) = \left[\frac{C_B(i)}{[(n-1)(n-2)/2}\right] \tag{13}$$

This measure favors nodes that join communities (dense sub networks), rather than nodes that lie inside a community. The higher the value of C_B, the greater is the node influence in the network.

5) EIGENVECTOR CENTRALITY

For a node v, the eigenvector centrality score is proportional to the sum of the scores of all nodes which are connected to it. Let $A_{i,j}$ denotes the graph adjacency matrix and x_i represents the score of the i^{th} graph node. Thus, $A_{i,j} = 1$ if the i^{th} node is adjacent to the j^{th} node and value of $A_{i,j} = 0$ otherwise. Furthermore, the values in A can be real numbers, which represent the strength of the connection between vertices, as in a stochastic matrix. Mathematically, it can be defined as,

$$\mathrm{X_i} = \frac{1}{\lambda} \sum_{\mathrm{j} \in \mathrm{M(i)}, \mathrm{j}=1}^{\mathrm{N}} \qquad \mathrm{X_j} = \frac{1}{\lambda} \sum \mathrm{X_{i,j} X_j} \tag{14}$$

Where $M(i)$ denotes the set of nodes which are connected to i^{th} node. N is the total number of nodes in the given graph and λ are a constant. In vector notation this can be rewritten as,

$$\mathrm{X} = \frac{1}{\lambda}(\mathrm{X}) = \left(\frac{1}{\lambda}\right) \mathrm{AX} = AX = \lambda \mathrm{X} \tag{15}$$

Eigenvector centrality is like a recursive version of degree centrality. Therefore, the final structure of each particle in the swarm particles is formulated as,

$$P_{vi}^t = \left(\left(P_{vi}^t, V_{vi}^t\right), Pos_{vi}^{pBest}, f_{vi}^{pBest}\right) \tag{16}$$

$$Pos_{vi}^t = (Pos_{v1}^t, Pos_{v2}^t, \ldots\ldots, Pos_{vN}^t) \tag{17}$$

$$V_{vi}^t = (V_{v1}^t, V_{v2}^t, \ldots\ldots, V_{vN}^t) \tag{18}$$

In Equation (16), P_{vi}^t represents the position and V_{vi}^t represents the velocity vector. The P_{vi}^t and V_{vi}^t values of a particular particle at a given time t are represented in Equation (17) and (18) respectively. The flow chart of the proposed PSO based approach for the discovery of the optimized subgraph patterns (in terms of optimized nodes in the graph structure) is shown in Figure 5.

C. DESIGNING A FITNESS FUNCTION

After finalizing the structure of the particle, the next critical step is the designing of fitness functions, since the fitness function plays very important role in the success of any heuristic approach. An efficient and carefully designed fitness function is helpful for the particles in the swarm to discover better solutions rapidly [52]. In the proposed fitness function, different important characteristics of the graph node (particle) are used. These characteristics include the degree of the particle (vertex), degree centrality, closeness centrality, betweenness centrality, eigenvector centrality, and weight of the particle (in case of weighted vertices).

In graph theory, the graph patterns with high connectivity in terms of degree, having greater the centrality measure values are better as compared to the other patterns [59]. Therefore, the aggregate score of the centrality measures with high connectivity is defined by,

$$f(d_i) = \sum_{i=1}^{n} Score\{C_D(i) + C_c(i) + C_B(i) + X\} \tag{19}$$

In Equation (19), the values of the $C_D(i)$, $C_c(i)$, $C_B(i)$, $e(i)$ and X are computed from the equations, Equation (9), (11), (13), and (14) respectively. So, the fitness function of a particle at time t, P_{vi}^t, the centrality value is defined as:

$$f\left(Pos_{vi}^t\right) = [f(d)] \tag{20}$$

The proposed fitness function maps the Pos_{vi}^t of a particle i to a real number representing the aggregate score of the different particle (vertex) centrality measures characteristics. A particle with the maximum value of the fitness function is considered as a global optimum particle (vertex).

The proposed approach starts by considering the graph in the graph dataset. The velocities of the node are initialized randomly. The position of the node is defined using different identified characteristics of the node in the

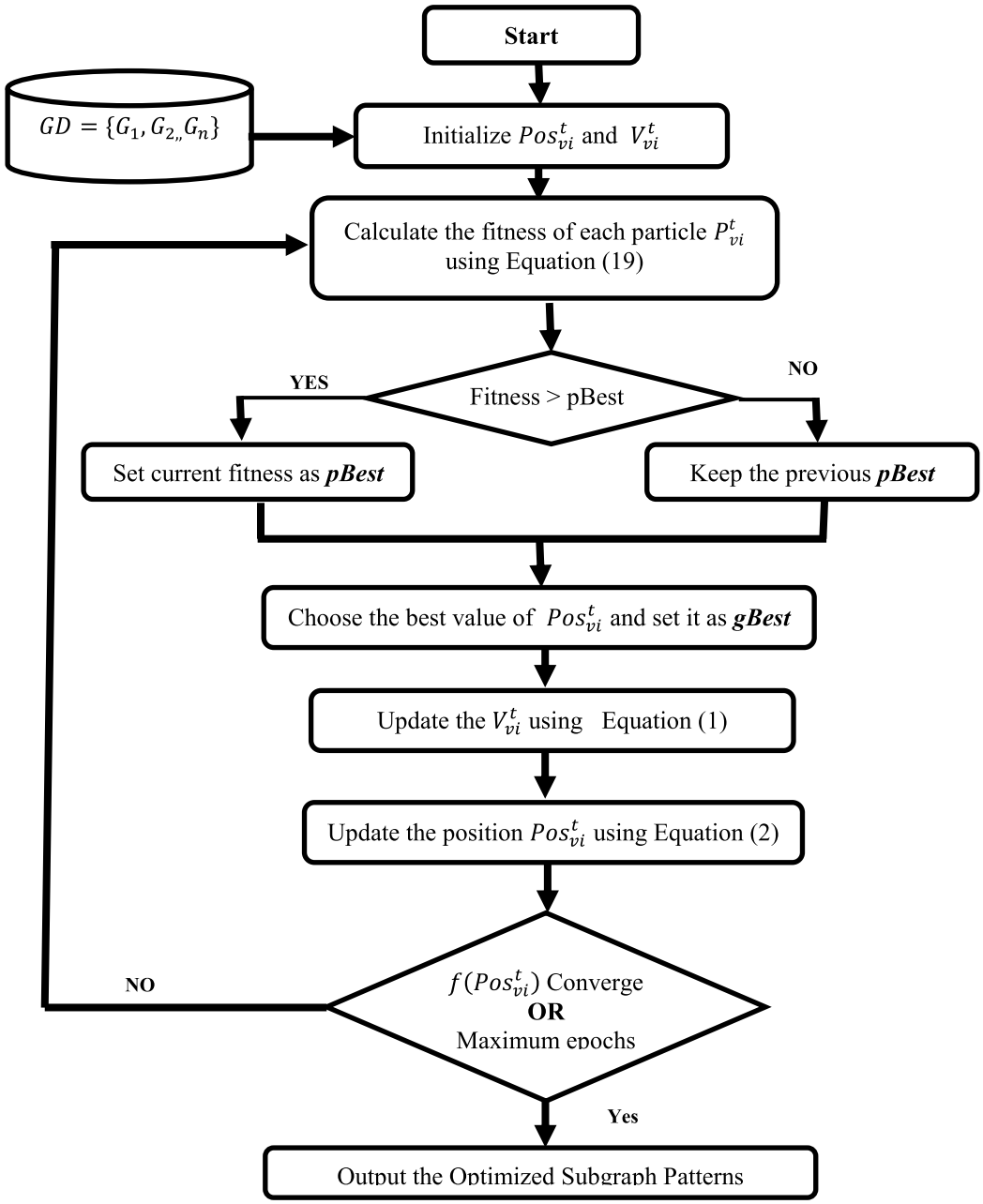

FIGURE 5. Flow chart of the proposed PSO based approach to the discovery of the opt-SGP.

graph structure. After calculating the position and velocities of the node, the fitness value of each node is computed using the Equation (20). Next, the node is evaluated to decide whether the current fitness value of the node is good or just ignoring the current fitness value by keeping the previous fitness value of the node.

Furthermore, the velocity and the position values of each node are updated with the new updated values, using Equation (1) and (2) respectively. This procedure continues until the position value of the node is converged or the maximum value of the epochs is reached.

IV. RESULTS AND DISCUSSION

To investigate the probability of the potential relationship between the mined FSPs and opt-SGPs, different experiments were conducted. Therefore, this section discusses the experimental setup and results generated by simulating our proposed technique. The FSPs were discovered using the A-RAFF framework that was discussed in details in [58]. The proposed PSO optimization approach was implemented in Java using JDK 1.8 in Netbeans 8.1. All experiments were performed on a 32-bit Intel Core i7-2600 CPU@3.40GHz machine with 8 GB RAM running on Linux operating system.

TABLE 2. Dataset and characteristics.

Dataset	Dataset Description
Chemical Compound [77]	340 chemical compounds, 24 different atoms, 66 atom types, and 4 types of bonds. On average 27 vertices per graph and 28 edges per graph. The largest one contains 214 edges and 214 vertices.
Compound Dataset [78]	The Compound dataset has 422 graphs with average graph size of 40 nodes and 42 edges. The largest graph in this dataset has 189 nodes and 196 edges.
NCI dataset [79]	The NCI dataset comprising of 237771 molecules found in the National Cancer Institute (NCI) database [79].

TABLE 3. Randomly selected dataset and characteristics.

Dataset	No of graphs	Average Size Edges	Largest Graph # of edges	Distinct node labels
GD-1	15	20	14	16
GD-2	100	28	17	40
GD-3	100	27	19	10
NCI	237,771	33	236	78

A. DATASETS

We have simulated four graph datasets with different sizes. The chosen graph datasets were benchmark graph datasets used in FSM frequently [61]–[66]. Original graph datasets description is shown in Table 2.

We have extracted the sample graphs from the above two graph datasets (See Table 2). Two sample graph datasets are extracted from Chemical Compound [77] and one sample graph dataset is extracted from the compound dataset [78]. We label the sample graph dataset as Sample_1_Chemical_340, Sample_2_Chemical_340. These two graph datasets are extracted from Chemical Compound graph dataset. The third sample graph dataset was extracted from the Compound dataset and labeled as Sample_3_Compound_422. The fourth graph dataset was NCI graph dataset. Graphs in each sample datasets are randomly selected from the original datasets. The randomly selected graph has diversified size with respect to nodes and edges. The results obtained on the sample graph datasets were also verified manually. This is due to the fact that this is the first ever study on uncovering the potential relationship between two types of graph patterns. Detailed statistics of each graph sample datasets are given in Table 3.

B. PSO PARAMETER SETTINGS

The parameters and values used in the proposed PSO optimization are presented in Table 4. Most of the existing literature on PSO optimization like [52], [67]–[70], use the same parameters setup. The error rate for was fixed to 9999.0. based on these initial values of the parameters, the proposed PSO-based optimization procedure was executed until the maximum number of iterations is exhausted or the error rate was minimized.

TABLE 4. PSO parameter setting.

Parameters	Value	Parameters	Values
Max_Iterations	1000	Dimensions	6
ω_{min}	0.4	ω_{max}	0.9
V_{max}	0.9	V_{min}	0.4
error_rate	9999.0		
Swarm Size	Depends on the total nodes in each graph. Since in the proposed approach we are mapping a graph to swarm, therefore each graph node is corresponding to a particle in the swarm. Hence, the swarm size equals to the total number of nodes in each graph.		

1) EXPERIMENT 1

In the first experiment, Sample_1_Chemical_340 was used. The FSPs discovered using the FSM framework A-RAFF from this sample graph dataset is given in Table 5. For example, at 50% threshold values, total 25 frequent subgraph patterns were discovered.

In each of these patterns set, the first two values represent the node number, next two represent the labels of the nodes and last value shows the label of the edge between two nodes. The last value in each pattern represents the edge label. For example, in Pattern # 1, (0, 1, '0', '0', '3'), 0 and 1 represents the nodes and '0' & '0' shows that node 0 and 1 has label '0' and the label of the edge between these two nodes is 3. Next, we use the proposed PSO approach to extract the opt-SGPs. For each graph, we obtained an optimized set of nodes which show opt-SGPs. The opt-SGPs are indicated in Table 6.

Next, we compare the number of the opt-SGPs and the FSPs. Here, we compute how many FSPs structures were there in the opt-SGPs. For example, at Minimum_Support = 50%, Total FSPs discovered = 25, and Total FSPs which contain the opt-SGPs = 23. Therefore, the FSPs which also represent the opt-SGPs are given as,

$$\text{opt} - \text{SGPs} = \frac{(\text{FSPs} * 100.0)}{\text{TotalOpt\_SGPs}} = (23/25) * 100.0 = 92\%.$$

Therefore, 92% FSPs were matched to the opt-SGPs. Only few FSPs do contain vertices, which were not optimized by the proposed PSO based optimization technique. This is because some graphs having many vertices and convoluted edges which were scattered having only very low values of different measures adopted for the particle structure.

Furthermore, we extracted the FSPs at different support values and then computed the matching patterns with the optimized graph patterns. Table 7 summarized the matching results of the FSPs and the opt-SGPs.

The investigation of the results described in Table 7, demonstrates that a relationship exists between the FSPs and the opt-SGPs. Maximum matching results of 92% were obtained. This indicates that a large number of the FSPs are

TABLE 5. Discovered FSPS from Sample_1_Chemical_340 at support threshold = 50%.

Pattern #	Frequent Subgraph Pattern
Pattern 1	(0, 1, '0', '0', '3')
Pattern 2	(0, 1, '0', '0', '3')(1, 2, '0', '0', '3')
Pattern 3	(0, 1, '0', '0', '3')(1, 2, '0', '0', '3')(2, 3, '0', '0', '3')
Pattern 4	(0, 1, '0', '0', '3')(1, 2, '0', '0', '3')(2, 3, '0', '0', '3')(3, 4, '0', '0', '3')
Pattern 5	(0, 1, '0', '0', '3')(1, 2, '0', '0', '3')(2, 3, '0', '0', '3')(2, 4, '0', '0', '3') (3, 5, '0', '0', '3')
Pattern 6	(0, 1, '0', '0', '3')(1, 2, '0', '0', '3') (2, 3, '0', '0', '3')(3, 4, '0', '0', '3')(4, 5, '0', '0', '3')(5, 0, '0', '0', '3')
Pattern 7	(0, 1, '0', '0', '3')(1, 2, '0', '0', '3')(2, 3, '0', '0', '3')(3, 4, '0', '0', '3')(4, 5, '0', '0', '3')(5, 0, '0', '0', '3')(5, 6, '0', '1', '0')
Pattern 8	(0, 1, '0', '0', '3')(1, 2, '0', '0', '3')(2, 3, '0', '0', '3')(3, 4, '0', '0', '3')(4, 5, '0', '0', '3')(5, 0, '0', '0', '3')(5, 6, '0', '1', '0')(2, 7,'0', '1', '0')
Pattern 9	(0, 1, '0', '0', '3')(1, 2, '0', '0', '3')(2, 3, '0', '0', '3') (3, 4, '0', '0', '3')(4, 5, '0', '0', '3') (5, 6, '0', '1', '0')
Pattern 10	(0, 1, '0', '0', '3')(1, 2, '0', '0', '3') (2, 3, '0', '0', '3')(3, 4, '0', '0', '3')(4, 5, '0', '0', '3')(5, 6, '0', '1', '0')(2, 7, '0', '1', '0')
Pattern 11	(0, 1, '0', '0', '3')(1, 2, '0', '0', '3') (2, 3, '0', '0', '3')(3, 4, '0', '0', '3')(4, 5, '0', '0', '3')(4, 6, '0', '1', '0')
Pattern 12	(0, 1, '0', '0', '3')(1, 2, '0', '0', '3')(2, 3, '0', '0', '3') (3, 4, '0', '0', '3')(4, 5, '0', '0', '3') (4, 6, '0', '1', '0')(1, 6, '0', '1', '0')
Pattern 13	(0, 1, '0', '0', '3')(1, 2, '0', '0', '3')(2, 3, '0', '0', '3')(3, 4, '0', '0', '3') (4, 5, '0', '0', '3')(3, 6, '0', '1', '0')
Pattern 14	(0, 1, '0', '0', '3')(1, 2, '0', '0', '3')(2, 3, '0', '0', '3')(3, 4, '0', '0', '3')(4, 5, '0', '1', '0')
Pattern 15	(0, 1, '0', '0', '3')(1, 2, '0', '0', '3')(2, 3, '0', '0', '3')(3, 4, '0', '0', '3') (4, 5, '0', '1', '0')(1, 6, '0', '1', '0')
Pattern 16	(0, 1, '0', '0', '3')(1, 2, '0', '0', '3')(2, 3, '0', '0', '3') (3, 4, '0', '0', '3') (3, 5, '0', '1', '0')
Pattern 17	(0, 1, '0', '0', '3') (1, 2, '0', '0', '3')(2, 3, '0', '0', '3')(3, 4, '0', '0', '3') (2, 5, '0', '1', '0')
Pattern 18	(0, 1, '0', '0', '3')(1, 2, '0', '0', '3')(2, 3, '0', '0', '3')(3, 4, '0', '1', '0')
Pattern 19	(0, 1, '0', '0', '3') (1, 2, '0', '0', '3')(2, 3, '0', '0', '3')(3, 4, '0', '1', '0') (0, 5, '0', '1', '0')
Pattern 20	(0, 1, '0', '0', '3') (1, 2, '0', '0', '3')(2, 3, '0', '0', '3')(2, 4, '0', '1', '0')
Pattern 21	(0, 1, '0', '0', '3')(1, 2, '0', '0', '3')(2, 3, '0', '1', '0')
Pattern 22	(0, 1, '0', '0', '3')(1, 2, '0', '0', '3')(1, 3, '0', '1', '0')
Pattern 23	(0, 1, '0', '0', '3') (1, 2, '0', '1', '0')
Pattern 24	(0, 1, '0', '1', '0')
Pattern 25	(0, 1, '1', '9', '0')

TABLE 6. OPT-SGPS retrieved from the proposed PSO optimization procedure on Sample_1_Chemical_340.

Graphs	Optimized Subgraphs	Graphs	Optimized Subgraphs
1	0,1,2,3, 5, 10, 16, 23	8	0, 1, 2
2	0, 1, 4, 11, 4, 5, 11	9	0, 1, 2, 3, 4, 5, 6, 9, 14, 15
3	0, 1,2,3,4, 10, 11, 12, 13, 14, 15,21	10	0, 1, 2, 4, 5, 20, 22
4	0, 1, 4, 5, 6, 13,	11	0, 1, 2, 3, 10, 14, 20,
5	0, 1, 3, 4	12	0, 1, 2, 4, 9, 10, 12
6	0, 1, 3, 4	13	0,1,3,4,6
7	0, 1, 4, 5	14	0, 1, 3, 5, 9, 11, 12, 13, 14,
15	0, 2, 3, 4, 5, 6, 7, 8, 9, 15		

such that they also contain the nodes which were also output as optimized using the proposed optimization procedure. The obtained results were found encouraging. Therefore, in the subsequent sections further large sample graph datasets were simulated to explore the existence of the prospective relationship between these two types of the patterns. In addition to this, few graph patterns were extracted as opt-SGPs but these were not discovered as FSPs by the A-RAFF. Such patterns are also depicted in the last column of Table 7. Actually, such FSPs were dropped from the final set of the A-RAFF as these were duplicate patterns.

TABLE 7. Comparison of OPT-SGPS and FSPS on Sample_2_Chemical_340.

Threshold (%)	Total FSPs	Total FSPs in opt-SGPs	FSPs which are opt-SGPs (%)	opt-SGPs which are not FSPs
50	25	23	92	2
40	59	48	81.35	11
30	108	94	87.03	14
20	402	311	77.36	91
10	4615	3517	76.20	1098

TABLE 8. Comparison of OPT-SGPS and FSPS on Sample_2_Chemical_340.

Threshold (%)	Total FSPs	Total FSPs in opt-SGPs	FSPs which are opt-SGPs (%)	opt-SGPs which are not FSPs
50	47	41	87.13	6
40	57	47	82.45	10
30	117	103	88.03	220
20	200	153	75.5	353
10	651	525	80.64	126

2) EXPERIMENT 2

In this experiment, we used larger graph datasets. In this dataset, there were a total of 100 different graphs of different sizes. Likewise, in Experiment 1, here all the FSPs were generated using the proposed A-RAFF framework. After that, using the same graph dataset, the opt-SGPs were enumerated from the proposed PSO based optimization procedure.

We examine the relationship between the FSPs and the opt-SGPs to validate whether the nodes which actively partake in the FSPs are present in opt-SGPs or not. Table 8 contains the detailed analysis results which show a strong relationship between the two types of the subgraph patterns. In this graph dataset, we have achieved a maximum of 88.03% matching score, which indicates that from 117 FSPs there were total 103 FSPs which contains the nodes of the graph which were also included in the opt-SGPs. In this experiment, the opt-SGPs which were not output as FSPs by the A-RAFF are also shown in the last column.

3) EXPERIMENT 3

In the third experiment, we have used Sample_3_Compound _422 graph datasets. Graphs selected at random from this dataset were more complex and contained many edges as compared to the Sample_2_Chemical_340 graph dataset. The detailed experimental results of this investigation, at various threshold values, for mining FSPs and the discovery of the opt-SGPs are displayed in Table 9.

TABLE 9. Comparison of OPT-SGPS and FSPS on Sample_3_Compound_422.

Threshold (%)	Total FSPs	Total FSPs in opt-SGPs	FSPs which are opt-SGPs (%)	opt-SGPs which are not FSPs
50	48	41	85.41	7
40	110	91	82.72	19
30	227	175	77.09	52
20	1129	886	78.47	243
10	1400	1084	77.42	316

TABLE 10. Comparison of OPT-SGPS and FSPS on NCI graph dataset.

Threshold (%)	Total FSPs	Total FSPs in opt-SGPs	FSPs which are opt-SGPs (%)	opt-SGPs which are not FSPs
50	2102	1687	80	415
40	5345	3945	73	1400
30	11,402	8451	74	2951
20	n/a	n/a	n/a	n/a
10	n/a	n/a	n/a	n/a

In Table 9, we can observe that the number of FSPs discovered is larger than the previous two graph sample datasets. From the result set, it is clear that a strong relationship between the FSPs and opt-SGPs is established. In this experiment maximum matching score was 85.41 (i.e., there were 41 FSPs) were discovered that these patterns are including the nodes which were the part of opt-SGPs).

4) EXPERIMENT 4

We have performed this experiment on larger graph dataset, the NCI organic graph dataset, which contains the graph representing the molecules from several sources [79]. See Table 3 for the dataset details.

In Experiment 4, we have achieved 80% results. In Table 10, at support threshold value of $(\sigma) = 10$ and $(\sigma) = 20$, memory was out due to a large number of extracted frequent subgraph patterns. However, this experiment on large graph dataset also confirms a potential relationship between the frequent and optimized graph patterns.

C. ANALYSIS

In the preceding section, we have discovered the FSPs using the proposed A-RAFF framework and next optimized graph structures were obtained using the proposed PSO based optimization to graphs; we executed these experiments in order to reveal any potential relationship between the two types of patterns. To the best of our knowledge, there is no other mechanism offered to decide whether the discovered FSPs are really frequent and are the most important from the entire dataset under observations, except the traditional user-specified threshold values. Therefore, in this

study, we were interested to check whether there exists any relationship or not between FSPs and opt-SGPs. From the above experimental results, we distinguished the following four different cases:

- Is there any relationship between the frequent subgraphs and optimized graphs?
- Whether a discovered frequent subgraph is also an optimized graph?
- Whether an optimized graph pattern represents a frequent subgraph pattern?
- How to decide about a significant frequent subgraph?

Next, these cases are addressed by giving suitable examples from the above mentioned experiment results.

1) CASE 1: IS THERE ANY RELATIONSHIP BETWEEN THE FREQUENT SUBGRAPHS AND OPTIMIZED GRAPHS?

In this case, the answer is ''Yes''. Results shown in Table 7, Table 8, Table 9, and Table 10 are encouraging and confer a clear picture of the relationship, which was assumed in case 1. In all the four experiments, we observed that most of the discovered FSPs include all or a maximum subset of nodes from the opt-SGPs. For example, the largest discovered FSP in the first experiment was pattern number 8, (0, 1, '0', '0', '3') (1, 2, '0', '0', '3') (2, 3, '0', '0', '3') (3, 4, '0', '0', '3') (4, 5, '0', '0', '3')(5, 0, '0', '0', '3') (5, 6, '0', '1', '0') (2, 7, '0', '1', '0'). Also, in the mined opt-SGPs, it can be observed that the set of nodes which participated in the extricated FSPs were also seen in most of the retrieved opt-SGPs (see the results in Table 7). It is also important to note that at highest values of the threshold parameter ($\sigma = 50\%$), matching score of the FSPs and opt-SGPs is very impressive (87% to 92% matching patterns, see Table 7, Table 8, Table 9 and Table 10). For example, in all the four experiments the matching score for both types of the pattern was more than 85%. As the threshold (σ) value increased the matching percentage gradually falls down. In spite of this, it is straightforward to conclude that the following relationship holds: FSPs $\approx$ opt-SGPs, which shows that both of the patterns are approximately equal to each other.

2) CASE 2: WHETHER A DISCOVERED FREQUENT SUBGRAPH IS ALSO AN OPTIMIZED GRAPH?

Again from the obtained results given in the above tables, we can say that these results are very remarkable which strengthened our assumption that a subgraph discovered as frequent subgraph pattern is also an optimized graph. For example, the patterns in FSPs {1, 2, 3, 4, 5, 6} have found their matching patterns in opt-SGPs 1, 3, 9, 10, 11, 12, 14, 15. All the four experiments at a maximum value of threshold (σ) ensured that an FSP is an opt-SGP, but with the decreasing value of the (σ) the relationship tends to weak by a magnitude of 5%. However, we believe that though there were some pessimistic results, the adopted centrality measures need further empirical exploration to validate the results of this novel research study.

3) CASE 3: WHETHER AN OPTIMIZED GRAPH PATTERNS REPRESENT A FREQUENT SUBGRAPH PATTERN?

The case 3 can be shown in a similar way to the case 2. The experimental analysis reveals that if opt-SGPs are computed using the proposed PSO approach then we can say that this subgraph pattern is also a frequent pattern in the graph dataset. However, there were some opt-SGPs which were not output as FSPs, but their fraction was very small. If we examine the results in Table 7 and Table 8, then it is comprehensible if a pattern is returned as opt-SGP then there exists an FSP with the same structure. For example, most of the nodes in the opt-SGPs {0, 1, 2, 3, 4} which were output as optimized ones were also seen in the different discovered FSPs such as, {*Pattern* − 1 *t* 6, *Pattern* − 14 *to* 25}. It was also observed that there were some opt-SGPs {*Graph* − 3, 9, 10, 11, 14} which contain the nodes {11, 12, 14, 20, 22} as optimized nodes but such nodes never occurred in the FSPs. However, as a whole, the results provide a clear picture that if a subgraph pattern is optimized then it will also be an FSP.

4) CASE 4: HOW TO DECIDE ABOUT A SIGNIFICANT FREQUENT SUBGRAPH?

In different experiments, the relationship between the FSPs and the opt-SGPs indicates that more than 85% opt-SGPs are contributing towards the FSPs (see the results in Table 7, 8, 9 and 10, at Threshold 50% and 30%). There were some FSPs which contains a strong association with the optimized patterns. As we have discovered a strong relationship in all the four graph dataset between the FSPs and the opt-SGPs, therefore, we can say that an FSP is a significant FSP if it is frequent subgraph as well as it has a strong relationship with the opt-SGPs set. Thus, *FSPs* $\approx$ opt-SGPs $\approx$ Significant FSPs, a frequent subgraph pattern which is approximately an optimized subgraph pattern is a significant frequent subgraph pattern. To further strengthen the claim, it is also important to note that the results shown in Table 7, Table 8 and Table 9 are also crossed verified manually by looking at the FSPs structure.

Computational Complexity: The computational complexity of the proposed PSO optimization strategy is computed as follows: the complexity of the proposed PSO based procedure mainly depends on the following factors: Population size (no of graphs) denoted by N and the number of iterations denoted by P. Therefore, the total computational cost for the proposed PSO based optimization procedure will be $O\ (N^*P)$.

Furthermore, the computation complexity of the frequent subgraph mining A-RAFF algorithm mainly depends on computation involved in the discovery of FSPs and ranking of the frequent subgraphs. Thus, computational complexity for the loop used for the extraction of the graph features is $O\ (n)$, where n is total of graph features. In the FSPs algorithm, there is recursion involved inside the loop. Therefore, it can easily be observed that it will compute all the frequent subgraph patterns in $O\left(2^{N^2}\right)$ time and it was provided in our earlier work [58].

V. CONCLUSION

In this exploratory study, we investigated the relationships between the FSPs and the opt-SGPs. The FSPs were discovered using the proposed FSM framework A-RAFF in our earlier work. The opt-SGPs were discovered using the proposed PSO based optimization techniques in this article. We believe that this is, in essence, the first-ever study, which is performed to identify a relationship between the two types of patterns. The PSO based optimization procedure is recommended to identify the relationship. Different graph measurements and characteristics were adopted from the literature to structure the particle in the proposed PSO procedure. Also, based on different node characteristics collected from the literature, a scoring function is designed which acts as a fitness function. One of the important advantages of this study will be the reduction of the FSPs, by choosing those FSPs in the final result set which were also proved as optimized. Different experiments were performed to validate the claim of the relationship between the FSPs and the opt-SGPs. The results are very promising and it is optimism that this study will open a new research direction in the frequent subgraph mining domain. The study concludes that the optimized nodes of the graphs in the graph dataset are also those nodes which participate in the FSPs.

Although our efforts offer a reasonable proof of the undiscovered association between the two types of patterns, however, we believe additional analysis will further strengthen this relationship. We have recommended the PSO scheme for establishing the relationship between the frequent subgraph patterns and the optimized subgraph patterns; it will be interesting to explore this relationship with another optimization algorithm including dynamic programming.

REFERENCES

[1] I. Atastina, B. Sitohang, G. A. P. Saptawati, and V. Moertini, "A review of big graph mining research," presented at the IOP Conf. Ser., Mater. Sci. Eng., 2017.

[2] C. Jiang, F. Coenen, and M. Zito, "Frequent sub-graph mining on edge weighted graphs," presented at the Int. Conf. Data Warehousing Knowl. Discovery, 2010, pp. 77–88.

[3] J. Wang, W. Hsu, M. Li Lee, and C. Sheng, "A partition-based approach to graph mining," presented at the 22nd Int. Conf. Data Eng. (ICDE), 2006, p. 74.

[4] S. Zhang, J. Yang, and V. Cheedella, "Monkey: Approximate graph mining based on spanning trees," presented at the IEEE 23rd Int. Conf. Data Eng. (ICDE), 2007, pp. 1247–1249.

[5] S. Thomas and J. J. Nair, "A survey on extracting frequent subgraphs," presented at the Int. Conf. Adv. Comput., Commun. Inform. (ICACCI), 2016, pp. 2290–2295.

[6] A. Dhiman and S. K. Jain, "Frequent subgraph mining algorithms for single large graphs—A brief survey," presented at the Int. Conf. Adv. Comput., Commun. Automat. (ICACCA). Dehradun, India: Spring, 2016, pp. 1–6.

[7] A. Brandstädt, "Efficient domination and efficient edge domination: A brief survey," presented at the Conf. Algorithms Discrete Appl. Math., 2018, pp. 1–14.

[8] X. Yan, H. Cheng, J. Han, and P. S. Yu, "Mining significant graph patterns by leap search," presented at the ACM SIGMOD Int. Conf. Manage. Data, 2008, pp. 433–444.

[9] T. Alam, S. A. Zahin, M. Samiullah, and C. F. Ahmed, "An efficient approach for mining frequent subgraphs," presented at the Int. Conf. Pattern Recognit. Mach. Intell., 2017, pp. 486–492.

[10] E. Abdelhamid, M. Canim, M. Sadoghi, B. Bhattacharjee, Y.-C. Chang, and P. Kalnis, "Incremental frequent subgraph mining on large evolving graphs," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 12, pp. 2710–2723, Dec. 2017.

[11] S. Ma, J. Li, C. Hu, X. Lin, and J. Huai, "Big graph search: Challenges and techniques," *Frontiers Comput. Sci.*, vol. 10, no. 3, pp. 387–398, 2016.

[12] X. Yan, F. Zhu, P. S. Yu and J. Han, "Feature-based similarity search in graph structures," *ACM Trans. Database Syst.*, vol. 31, no. 4, pp. 1418–1453, 2006.

[13] C. Borgelt and M. R. Berthold, "Mining molecular fragments: Finding relevant substructures of molecules," presented at the IEEE Int. Conf. Data Mining (ICDM), 2002, pp. 51–58.

[14] L. Dehaspe, H. Toivonen, and R. D. King, "Finding frequent substructures in chemical compounds," presented at the KDD, 1998.

[15] L. B. Holder, D. J. Cook, and S. Djoko, "Substructure discovery in the SUBDUE system," presented at the KDD Workshop, 1994, pp. 169–180.

[16] J. Huan, W. Wang, D. Bandyopadhyay, J. Snoeyink, J. Prins, and A. Tropsha, "Mining protein family specific residue packing patterns from protein structure graphs," presented at the 8th Annu. Int. Conf. Res. Comput. Mol. Biol., 2004, pp. 308–315.

[17] M. Kuramochi and G. Karypis, "Frequent subgraph discovery," presented at the IEEE Int. Conf. Data Mining (ICDM), Nov./Dec. 2001, pp. 313–320.

[18] M. Kuramochi and G. Karypis, "An efficient algorithm for discovering frequent subgraphs," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 9, pp. 1038–1051, Sep. 2004.

[19] M. Deshpande, M. Kuramochi, N. Wale, and G. Karypis, "Frequent substructure-based approaches for classifying chemical compounds," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 8, pp. 1036–1050, Aug. 2005.

[20] J. Huan, W. Wang, J. Prins, and J. Yang, "SPIN: mining maximal frequent subgraphs from graph databases," presented at the 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2004, pp. 581–586.

[21] T. Kudo, E. Maeda, and Y. Matsumoto, "An application of boosting to graph classification," presented at the Adv. Neural Inf. Process. Syst., 2005, pp. 729–736.

[22] X. Yan, F. Zhu, J. Han, and P. S. Yu, "Searching substructures with superimposed distance," presented at the 22nd Int. Conf. Data Eng. (ICDE), 2006, p. 88.

[23] C. Chen, X. Yan, P. S. Yu, J. Han, D.-Q. Zhang, and X. Gu, "Towards graph containment search and indexing," presented at the 33rd Int. Conf. Very Large Data Bases, 2007, pp. 926–937.

[24] I. Alsmadi and I. Alhami, "Clustering and classification of email contents," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 27, no. 1, pp. 46–57, 2015.

[25] P. Sanders and C. Schulz, "Think locally, act globally: Highly balanced graph partitioning," presented at the Int. Symp. Exp. Algorithms, 2013, pp. 164–175.

[26] K. Tsuda and T. Kudo, "Clustering graphs by weighted substructure mining," presented at the 23rd Int. Conf. Mach. Learn., 2006, pp. 953–960.

[27] Z. Zou, J. Li, H. Gao, and S. Zhang, "Mining frequent subgraph patterns from uncertain graph data," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 9, pp. 1203–1218, Sep. 2010.

[28] C. Chen, C. X. Lin, X. Yan, and J. Han, "On effective presentation of graph patterns: A structural representative approach," presented at the 17th ACM Conf. Inf. Knowl. Manage., 2008, pp. 299–308.

[29] W. Fan *et al.*, "Direct mining of discriminative and essential frequent patterns via model-based search tree," presented at the 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2008, pp. 230–238.

[30] C. Jiang, F. Coenen, and M. Zito, "A survey of frequent subgraph mining algorithms," *Knowl. Eng. Rev.*, vol. 28, no. 1, pp. 75–105, 2013.

[31] X. Cheng, S. Su, S. Xu, L. Xiong, K. Xiao, and M. Zhao, "A two-phase algorithm for differentially private frequent subgraph mining," *IEEE Trans. Knowl. Data Eng.*, to be published.

[32] T. Nguyen and P. Do, "Topic discovery using frequent subgraph mining approach," in *Proc. Int. Conf. Comput. Sci. Technol.*. Singapore: Springer, Nov. 2017, pp. 432–442.

[33] C. Aslay, M. A. U. Nasir, G. De Francisci Morales, and A. Gionis, "Mining frequent patterns in evolving graphs," Tech. Rep., 2018.

[34] C. C. Aggarwal and H. Wang, "Graph data management and mining: A survey of algorithms and applications," in *Managing and Mining Graph Data. Advances in Database Systems*. Boston, MA, USA: Springer, 2010.

[35] E. Estrada and J. A. Rodriguez-Velazquez, "Subgraph centrality in complex networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 71, no. 5, p. 056103, May 2005.

[36] R. Agarwal and R. Srikant, "Fast algorithms for mining association rules," presented at the 20th VLDB Conf., 1994.
[37] A. P. Agrawal and A. Kaur, "A comprehensive comparison of ant colony and hybrid particle swarm optimization algorithms through test case selection," in *Data Engineering and Intelligent Computing*. Singapore: Springer, 2018, pp. 397–405.
[38] Y. Jin, K. Miettinen, and H. Ishibuchi, "Guest editorial evolutionary many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 1–2, Feb. 2018.
[39] N. Deo, *Graph Theory With Applications to Engineering and Computer Science*. New York, NY, USA: Dover, 2017.
[40] A. Gibbons, *Algorithmic Graph Theory*. Cambridge, U.K.: Cambridge Univ. Press, 1985.
[41] G. Di Caro and M. Dorigo, "AntNet: Distributed stigmergetic control for communications networks," *J. Artif. Intell. Res.*, vol. 9, pp. 317–365, Dec. 1998.
[42] M. Shojafar, Z. Pooranian, P. G. V. Naranjo, and E. Baccarelli, "FLAPS: bandwidth and delay-efficient distributed data searching in Fog-supported P2P content delivery networks," *J. Supercomput.*, vol. 73, no. 12, pp. 5239–5260, 2017.
[43] F. Huilian, "Discrete particle swarm optimization for TSP based on neighborhood," *J. Comput. Inf. Syst.*, vol. 6, no. 10, pp. 3407–3414, 2010.
[44] S. Arora and S. Singh, "The firefly optimization algorithm: Convergence analysis and parameter selection," *Int. J. Comput. Appl.*, vol. 69, no. 3, pp. 48–52, 2013.
[45] Y. Marinakis, M. Marinaki, and G. Dounias, "Honey bees mating optimization algorithm for the Euclidean traveling salesman problem," *Inf. Sci.*, vol. 181, no. 20, pp. 4684–4698, 2011.
[46] M. A. K. Azrag, T. A. A. Kadir, J. B. Odili, and M. H. A. Essam, "A global African buffalo optimization," *Int. J. Softw. Eng. Comput. Syst.*, vol. 3, no. 3, pp. 138–145, 2017.
[47] E. Osaba, X.-S. Yang, F. Diaz, P. Lopez-Garcia, and R. Carballedo, "An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems," *Eng. Appl. Artif. Intell.*, vol. 48, pp. 59–71, Feb. 2016.
[48] G. Dong, X. Fu, and H. Xue, "An ant system-assisted genetic algorithm for solving the traveling salesman problem," *Int. J. Advancements Comput. Technol.*, vol. 4, no. 5, pp. 165–171, 2012.
[49] X. Geng, Z. Chen, W. Yang, D. Shi, and K. Zhao, "Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search," *Appl. Soft Comput.*, vol. 11, no. 4, pp. 3680–3689, 2011.
[50] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," presented at the 6th Int. Symp. Micro Mach. Hum. Sci. (MHS), 1995, pp. 39–43.
[51] B. Khan and P. Singh, "Selecting a meta-heuristic technique for smart micro-grid optimization problem: A comprehensive analysis," *IEEE Access*, vol. 5, pp. 13951–13977, 2017.
[52] J. Odili, M. N. M. Kahar, A. Noraziah, and S. F. Kamarulzaman, "A comparative evaluation of swarm intelligence techniques for solving combinatorial optimization problems," *Int. J. Adv. Robot. Syst.*, vol. 14, no. 3, pp. 1–11, 2017.
[53] J.-B. Park, Y.-W. Jeong, J.-R. Shin, and K. Y. Lee, "An improved particle swarm optimization for nonconvex economic dispatch problems," *IEEE Trans. Power Syst.*, vol. 25, no. 1, pp. 156–166, Feb. 2010.
[54] J.-B. Park, K.-S. Lee, J.-R. Shin, and K. Y. Lee, "A particle swarm optimization for economic dispatch with nonsmooth cost functions," *IEEE Trans. Power Syst.*, vol. 20, no. 1, pp. 34–42, Feb. 2005.
[55] F. Valdez, P. Melin, and O. Castillo, "Fuzzy logic for combining particle swarm optimization and genetic algorithms: Preliminary results," presented at the Mexican Int. Conf. Artif. Intell., 2009, pp. 444–453.
[56] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, p. 026113, Feb. 2004.
[57] T. P. Newman, "Tracking the release of IPCC AR5 on Twitter: Users, comments, and sources following the release of the Working Group I Summary for Policymakers," *Public Understand. Sci.*, vol. 26, no. 7, pp. 815–825, 2017.
[58] R. U. Saif and A. Sohail, "A-RAFF: A ranked frequent FP-growth subgraph pattern discovery approach," *J. Internet Technol.*, vol. 20, no. 2, Mar. 2019.
[59] J. L. Gross and J. Yellen, *Handbook of Graph Theory*. Boca Raton, FL, USA: CRC Press, 2004.
[60] K. Okamoto, W. Chen, and X.-Y. Li, "Ranking of closeness centrality for large-scale social networks," presented at the Int. Workshop Frontiers Algorithmic, 2008, pp. 186–195.
[61] D. J. Cook and L. B. Holder, *Mining Graph Data*. Hoboken, NJ, USA: Wiley, 2006.
[62] S. Nijssen and J. N. Kok, "A quickstart in frequent structure mining can make a difference," presented at the 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2004, pp. 647–652.
[63] F. Qiao, X. Zhang, P. Li, Z. Ding, S. Jia, and H. Wang, "A parallel approach for frequent subgraph mining in a single large graph using spark," *Appl. Sci.*, vol. 8, no. 2, p. 230, 2018.
[64] X. Yan and J. Han, "gSpan: Graph-based substructure pattern mining," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2002, pp. 721–724.
[65] X. Yan and J. Han, "CloseGraph: Mining closed frequent graph patterns," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2003, pp. 286–295.
[66] M. Wörlein, T. Meinl, I. Fischer, and M. Philippsen, "A quantitative comparison of the subgraph miners MoFa, gSpan, FFSM, and Gaston," in *Proc. Eur. Conf. Princ. Data Mining Knowl. Discovery*. Berlin, Germany: Springer, Oct. 2005, pp. 392–403.
[67] O. Ertenlice and C. B. Kalayci, "A survey of swarm intelligence for portfolio optimization: Algorithms and applications," *Swarm Evol. Comput.*, vol. 39, pp. 36–52, Apr. 2018.
[68] M. Mavrovouniotis, C. Li, and S. Yang, "A survey of swarm intelligence for dynamic optimization: Algorithms and applications," *Swarm Evol. Comput.*, vol. 33, pp. 1–17, Apr. 2017.
[69] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," presented at the Int. Conf. Evol. Program., 1998, pp. 591–600.
[70] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," presented at the Congr. Evol. Comput. (CEC), 1999.
[71] S. Shahrivari and S. Jalili, "High-performance parallel frequent subgraph discovery," *J. Supercomput.*, vol. 71, no. 7, pp. 2412–2432, 2015.
[72] B. Douar, M. Liquiere, C. Latiri, and Y. Slimani, "LC-mine: A framework for frequent subgraph mining with local consistency techniques," *Knowl. Inf. Syst.*, vol. 44, no. 1, pp. 1–25, 2015.
[73] S. Han, W. K. Ng, and Y. Yu, "FSP: Frequent substructure pattern mining," in *Proc. 6th Int. Conf. Inf., Commun. Signal Process.*, Dec. 2007, pp. 1–5.
[74] T. K. Saha and M. Al Hasan, "FS^3: A sampling based method for top-k frequent subgraph mining," *Stat. Anal. Data Mining, ASA Data Sci. J.*, vol. 8, no. 4, pp. 245–261, 2015.
[75] A. Inokuchi, T. Washio, and H. Motoda, "An apriori-based algorithm for mining frequent substructures from graph data," in *Proc. European Conf. Principles Data Mining Knowl. Discovery*. Berlin, Germany: Springer, Sep. 2000, pp. 13–23.
[76] L. T. Thomas, S. R. Valluri, and K. Karlapalem, "MARGIN: Maximal frequent subgraph mining," *ACM Trans. Knowl. Discovery Data*, vol. 4, no. 3, 2010, Art. no. 10.
[77] *The Predictive Toxicology Evaluation Challenge*. Accessed: Apr. 2, 2018. [Online]. Available: http://www.cs.ox.ac.uk/activities/machlearn/PTE/
[78] *Compound Graph Dataset*. Accessed: Apr. 2, 2018. [Online]. Available: https://github.com/Jokeren/DataMining-gSpan/tree/master/extern/data
[79] *National Cancer Institute (NCI) Database is a Database of 2,300+ Agents That are Being Used in the Treatment of Patients With Cancer or Cancer-Related Conditions*. Accessed: Apr. 2, 2018. [Online]. Available: http://cactus.nci.nih.gov/ncidb2.1/

SAIF UR REHMAN received the M.C.S. degree (Hons.) from the Institute of Computing and Information Technology, Gomal University, Dera Ismail Khan, Pakistan, in 2005, and the M.S. degree from SZABIST, Islamabad, Pakistan. He is currently pursuing the Ph.D. degree in computer science with Abasyn University, Islamabad. He is currently an Assistant Professor with UIIT, PMAS Arid Agriculture University, Rawalpindi, Pakistan. His research interests include data mining, graph mining, social graph analysis, and big data analytics.

SOHAIL ASGHAR (M'06) received the degree (Hons.) in computer science from the University of Wales, U.K., in 1994, and the Ph.D. degree from the Faculty of Information Technology, Monash University, Melbourne, Australia, in 2006. He is currently a Professor of computer science with the COMSATS Institute of Information Technology, Islamabad. He has taught and researched in data mining. He is a member of ACS.

SIMON JAMES FONG received the B.Eng. degree (Hons.) in computer systems and the Ph.D. degree (Hons.) in computer science from La Trobe University, Australia, in 1993 and 1998, respectively. He is currently an Associate Professor with the Computer and Information Science Department, University of Macau. He is also one of the founding members of the Data Analytics and Collaborative Computing Research Group, Faculty of Science and Technology. Prior to the academic career, he took up various managerial and technical posts, such as a systems engineer, an IT consultant, and the e-commerce director in Australia and Asia. He has published over 300 international conference and peer-reviewed journal papers, mostly in the areas of data mining and optimization algorithms.