Subgraph Mining in Graph-based Data using Multiobjective Evolutionary Programming

Prakash Shelokar, Arnaud Quirin, Óscar Cordón,
European Centre for Soft Computing,
Mieres (Asturias), Spain.
Emails: {prakash.shelokar, arnaud.quirin, oscar.cordon}@softcomputing.es

***Abstract*—This work proposes multiobjective subgraph mining in graph-based data using multiobjective evolutionary programming (MOEP). A mined subgraph is defined by two objectives, support and size. These objectives are conflicting as a subgraph with high support value is usually of small size and *vice-versa*. MOEP applies NSGA-II's nondominated sorting procedure to evolve the population during the subgraph generation process. An experimental study on five synthetic and real-life graph-based datasets shows that MOEP outperforms Subdue-based methods, a well-known heuristic search approach for subgraph discovery in data mining community. The comparison is done using hypervolume, C and I_ϵ multiobjective performance metrics.**

***Index Terms*—Graph-based data mining, Multiobjective subgraph mining, Evolutionary programming, Multiobjective optimization, Pareto optimality.**

I. Introduction

Recently graph-based data mining (GBDM) has been applied in different fields of science and engineering, such as social or transportation network analysis, pharmaceutical or chemical molecules analysis, among others, needs to manipulate large datasets mainly structured in graphs [1]. Well-structured data like graphs are preferred in those domains over relational databases as they allow the representation of complex relationships between entities. Entities are typically represented as nodes and correspond for instance to individual people, atoms, energy producers and receivers, whereas their relationships such as people interactions, atom connections and power lines are represented as edges in a graph. One of the classical problem in this area is called subgraph mining and consists of the extraction of subgraphs or *patterns* having, in our sense, a high interest, i.e. with large sizes and/or being frequent [2].

The state-of-the-art graph-based approaches have been proposed for the task of frequent subgraph mining including, Subdue, frequent subgraph discovery, gSpan, MoFa/MoSS, JoinPath, CloseGraph, FFSM, Spin, Gaston, CLOSECUT and SPLAT, gPrune, etc [1]. All these methods work by performing search in the subgraphs lattice. Recently, evolutionary programming based Subdue (EP-Subdue) method is applied for frequent subgraph mining [3]. EP-Subdue has shown good performance over the standard Subdue method, which performs a heuristic search using the MDL principle. The superior performance of EP-Subdue over the Subdue method was due to the global search in the subgraphs lattice space.

Recently, some approaches have suggested application of Pareto-based multiobjective subgraph mining in graph database [4]–[6]. These methods have shown success in mining the approximation to the Pareto-optimal set of subgraphs on different graph datasets.

In this paper, we propose an application of multiobjective evolutionary programming for multiobjective subgraph mining in graph databases using two objectives, support and size. These objectives are conflicting in nature. MOEP uses NSGA-II's nondominated sorting to evolve the population at each generation. The proposed algorithm is compared against the different Subdue-based methods for subgraph mining [5] using five artificial and real-world datasets.

Up to our knowledge, this is the first proposal of an evolutionary multiobjective optimization (EMO) algorithm for the multiobjective graph mining task as the previous existing EMO proposal in [4] is only able to deal with tree-based structures. This is a significant advantage with respect to the existing multiobjective graph mining methods as those proposed in [5], [6] since the use of a Pareto-based EMO approach allows the proposed method to perform a global search at different levels of the subgraph lattice at the same time, thus achieving better Pareto front approximations.

The paper is organized as follows. Some preliminaries related to single and multiobjective GBDM are presented in Section II and our algorithm is presented in Section III. An experimental analysis follows in Section IV, and concluding remarks and future works are presented in the last section.

II. Preliminaries

In this section, we briefly describe the Subdue [7] and EP-Subdue [3] methods. Then multiobjective GBDM problem is depicted, as well as the MOSubdue algorithm [5], which is able to perform multiobjective subgraph mining.

A. The Subdue Algorithm

Subdue [7], [8] is a classical method in GBDM that performs different tasks such as subgraph discovery, conceptual clustering, and classification on graph-represented data. It has been successfully applied for many real-world problems including bioinformatics, counter-terrorism, web data mining, geology, aviation and chemistry among others. The search process of Subdue is guided by the minimum description length (MDL) principle [9], which assumes that the best

978-1-4244-7835-4/11/$26.00 ©2011 IEEE

subgraph is the one that minimizes the description length of the input graph when compressed by the subgraph. The description length of the subgraph S, given the input graph G, is calculated as:

$$value_{\text{MDL}}(G, S) = (DL(S) + DL(G|S))/DL(G) \quad (1)$$

where $DL(S)$ and $DL(G)$ are the description length of the subgraph S and graph G, and $DL(G|S)$ is the description length of the input graph G compressed by the subgraph S. The description length is calculated as the number of bits needed to encode an adjacency matrix representation of a graph [8]. Notice that, equation (1) jointly considers two commonly used objectives in GBDM, support and size (or complexity).

Subdue starts with a parent list of single vertex subgraphs corresponding to unique label vertices in the input graph. Child subgraphs are created from each subgraph S in the parent list by expanding each of its instances in G in all possible ways. The expansion is done by adding an edge and a vertex in G or by an edge only if both vertices are present in the instance. The child subgraphs are evaluated using equation (1). These subgraphs are stored in a child list in the ascending order of MDL values. The child list can grow exponentially as more child subgraphs are identified. In order to avoid that exponential explosion, Subdue applies a beam search [10], limiting the number of subgraphs in the child list by the choice of a threshold parameter called, beam width. For the next level of expansion, the child list becomes the parent list. The algorithm recursively applies this process until either upon reaching a user specified limit on the number of parent subgraphs to be extended, or upon exhaustion of the search space. The output is a list of the best subgraphs found.

The drawback of Subdue is that it uses a computationally constrained beam search that discards some of the less promising subgraphs at an early stage of the exploration. Thereby terminating the possibility that of expanding these less promising subgraphs later on in order to search other promising subgraph solution space. Thus, beam search is a kind of search in the state space of subgraphs lattice not allowing backtracking. It may often end up providing suboptimal results.

B. Evolutionary Programming in GBDM

One way to overcome this problem is the use of a global search method to explore different layers of the subgraphs lattice using a population of subgraphs. One such improvement to the beam search of Subdue was an application of EP as suggested by Bandyopadhyay et al. [3]. Unlike Subdue whose beam search utilizes the *beam-width* number of subgraphs usually belonging to a single search space solution, EP-based Subdue (EP-Subdue) method utilizes a population of subgraphs in order to explore the different search space solutions in each generation. The initialization of the population is done as follows. First, single vertex subgraphs are created by using Subdue's way of initializing its parent list. Then, new subgraphs are created from each single vertex subgraph by expanding each of its instances in input graph by adding a edge and a vertex. Then, the population is initialized by random selection of these new subgraphs. The fitness of an individual in the population is evaluated as an inverse of equation (1). The maximization of fitness is thus assumed. At any generation child subgraph population Q is created using mutation. During mutation of an individual, the subgraph S that it holds is extracted. Each of its instances is expanded in all possible ways in order to create new child subgraphs. This is done by using Subdue's procedure of creating child subgraphs. Among the generated child subgraphs from the subgraph S, a child subgraph is randomly selected that becomes the mutated individual in the population Q. The fitness of this new individual is evaluated. The global best individual is updated using the individuals in Q. To yield the next generation, the population is created by copying the global best individual, and selecting the remaining individuals from $|P| \cup |Q|$ based on the fitness proportionate selection. The method continues exploring the search space for a maximum number of given generations. A single best subgraph as pointed by the global best individual is then returned.

In the following sub section a formal definition of multiobjective optimization is provided.

C. Multiobjective Optimization in Data Mining and Our Multiobjective GBDM Problem

A general multiobjective optimization (MOO) problem can be described as a vector function f that maps a tuple of l parameters (decision variables) to a tuple of o objectives [11], [12]. Formally:

$$\begin{aligned} \text{min./max. } & y = f(x) = (f_1(x), f_2(x), \ldots, f_o(x)) \\ \text{subject to } & x = (x_1, x_2, \ldots, x_l) \in X \\ & y = (y_1, y_2, \ldots, y_o) \in Y \end{aligned} \quad (2)$$

where x is called the decision vector, X is the parameter space, y is the objective vector, and Y is the objective space. To compare any two solutions, we apply the well known concept of Pareto dominance: assume, without loss of generality, a maximization problem, and consider two solutions x^1 and x^2 with vector-valued objective functions y^1 and y^2 respectively. An objective vector y^1 is said to weakly dominate another objective vector y^2 ($y^1 \succ y^2$) if no component of y^1 is smaller than the corresponding component of y^2 and at least one component is greater. Accordingly, we can say that a solution x^1 is better to another solution x^2, i.e., x^1 dominates x^2 ($x^1 \succ x^2$), if $f(x^1)$ dominates $f(x^2)$. Mathematically, the concept of Pareto optimality is defined as follows:

$$\begin{aligned} & \forall i \in \{1, 2, \ldots, o\} : f_i(x^1) \geq f_i(x^2) \wedge \\ & \exists j \in \{1, 2, \ldots, o\} : f_j(x^1) > f_j(x^2) \end{aligned} \quad (3)$$

Hence, there is not usually a single optimal solution to solve a typical MOO problem, (i.e., being better than the remainder with respect to every objective, as in single objective optimization) but a set of optimal solutions that are superior to the remainder (are not dominated by them) when all the objectives are jointly considered. These optimal solutions are known as nondominated, efficient, or Pareto optimal, and constitute the so-called nondominated, efficient, or Pareto optimal solutions set. Their set of objective vectors is called the nondominated, efficient or Pareto front (PF).

Recently, Romero-Zaliz et al. [4] introduced Evolutionary Multiobjective Optimization-based Conceptual Clustering (EMO-CC) methodology for gene ontology (GO) domain. In EMO-CC, the chromosome representation used for GO domain is a tree-like subgraph. The GO dataset mostly contains small size tree-like graphs, which allow encoding the chromosome as an instance in the input data, and also performing feasible crossover with lower time complexity. It can not handle large graphs having more general graph representation, where a node may have several parents. However, in our study, we are interested in mining general-structured subgraphs. As in the study of Romero-Zaliz et al., we are interested in subgraphs that maximize the following two objectives [5]:

- the support (the occurrence frequency of the subgraph S in the input graph or set of graphs G), computed as $\mathit{value}_{\text{support}}(G,S)$ = #subgraphs in G matching S, and
- the complexity or size (the number of vertices and edges present in the subgraph S), computed as $\mathit{value}_{\text{complexity}}(G,S)$ = #vertices(S) + #edges(S).

Both objectives are conflicting, as a large size(or highly complex) subgraph has a small support and *vice-versa*. In the following section, we briefly describe multiobjective subgraph mining using MOSubdue algorithm [5].

D. Multiobjective Subdue (MOSubdue) Algorithm

The authors [5] proposed Multiobjective Subdue (MOSubdue), an extension of the single objective Subdue method for subgraph discovery. MOSubdue utilizes a Pareto dominance-based approach for selecting child subgraphs for the next level of expansion, as against single-objective Subdue method. It employs NSGA-II's nondominated sorting procedure for evaluating the fitness of child subgraphs. A nondomination level (or front) for each subgraph is identified according to dominance criteria (eq.3). Each subgraph is assigned a fitness equal to its nondomination level. The minimization of fitness (or rank) is assumed (i.e. subgraphs with rank 1 are the best; subgraphs with rank 2 are the second-best; and so on).

A beam width size of child subgraphs from a nondominated front are selected according to NSGA-II's crowding-distance measure computed for each subgraph in that front. When two subgraphs share same fitness, one located in the less crowded-region is preferred, in case of a tie one is randomly selected. The effect of diversified child subgraphs selection is that MOSubdue may perform a stochastic search in the multiobjective search space of the subgraphs.

The performance of MOSubdue was tested on synthetic and real-world graph-based datasets. The results showed that MOSubdue had found good approximations to the reference PFs of different datasets. However, because of the way the constructive process is done, by adding an edge in each step, and because of the fact beam width solutions are selected during a given step in which all of them can have the same size, the algorithm is biased towards the support only. We claim that an EMO-based strategy in which the candidates are selected at different levels can overcome this drawback.

In the present study, we benchmark MOSubdue to compare the performance of our MOEP algorithm as described in the following section.

III. MOEP based Multiobjective Subgraph Mining

In this section, our EMO-based GBDM proposal is described.

A. Graph, Subgraph and Individual Representation

The input to the EMO-based GBDM methodology is a graph-represented data, which includes feature-values that usually map to nodes, and relationships between them that map to edges. The input data G can be a single large graph or a set of graphs.

An individual in the population P is represented as a possible subgraph S in the data G. Note that in our case, the genotypic and phenotypic representations of an individual are the same, i.e., they are not binary coded. We make also a distinction between the *subgraph* S and its *instances* $I^1,\dots,I^M$ that are the concrete occurrences of S in G. P is a set of N subgraphs $S^1,S^i,\dots,S^N \in X$, where S^i is a connected subgraph within the graphical representation for all those subgraphs in G that match to the subgraph S. In the proposed algorithm, an individual (or a solution) in the population P is always composed of a subgraph S with its instances in the input data G.

B. Population Initialization

Theoretically, we could initialize the population taking the output of any other GBDM algorithm. Here we use a simpler procedure. First single vertex (i.e. one-size) subgraphs are created from unique label vertices in G. The next layer of lattice is then explored by extending all the instances of these subgraphs by an edge and a node in all possible ways. This will produce candidate subgraphs of the same complexity (i.e. size of three) but may have different values for the other objective, support. We then randomly select these subgraphs to form the initial population.

C. Candidate Subgraphs Generation

The current application of our EMO approach for GBDM relies only on mutation operator for candidate generation. Mutation yields a maximum of N new candidate subgraphs as each parent subgraph S in P is used to create a new child subgraph. All the instances in G of the parent subgraph S are extended by an edge (and a node if no cycle is closed) in order to create child instances. A child instance is then randomly selected that becomes a child subgraph in graphical representation. All of the child instances that match this child subgraph become its new instances. This child subgraph should have at least two instances in G to qualify as a child subgraph of the parent subgraph S. Otherwise a new child instance will be randomly selected to form a child subgraph. The mutation applies extending parent subgraph by an edge, which always create existing (or feasible) candidates in the input data G. This is the most commonly used candidate generation method employed by GBDM techniques [7].

D. Objective Functions Calculation

An individual S in the population P is evaluated using two objective functions (see Section II-C). The first objective (support) is the one that needs the most computational effort to be computed. To reduce the computational burden, we consider an individual is always associated with its subgraph instances in the input data G. Thus, it is enough to check these subgraph instances instead of the whole graph to compute the support.

E. The steps-by-step MOEP algorithm

To evolve and maintain diversity in the population of subgraphs, we use NSGA-II's nondominated sorting and crowding-distance based diversity methods [13]. The algorithm is given below as:

step 1: Initialize the population P of N subgraphs.
step 2: Create a population Q of child subgraphs from parent subgraphs in P by mutation.
step 3: Combine the two populations $R = P \cup Q$.
step 4: Assign fitness to subgraphs in R using the nondominated sorting procedure. The population is first sorted into different nondomination levels. Each subgraph is then assigned a fitness (or rank) equal to its nondomination level (1 is the best level).
step 5: Create a new population from R. For the new population, first choose the rank 1 subgraphs. If the number of rank 1 subgraphs are smaller than N, then choose the rank 2 subgraphs, and so on. Say that r is the last rank of subgraphs that can be accommodated. In general, the number of subgraphs belonging to rank 1 to r will be greater than N. To choose exactly N subgraphs, a crowding-distance assignment [13] is applied in the last rank to sort and select exactly N subgraphs.
step 6: Go to step 2, if termination criterion is not satisfied.
step 7: Report the nondominated subgraphs in P.

IV. Experiments

In this section, the performance of our EMO algorithm for the proposed multiobjective subgraph mining task (as defined in Section II.D) is analyzed by means of various unary and binary metrics [14], and visual representations of the obtained Pareto front (PF) approximations. For comparison purpose, we also apply single-objective Subdue and EP-Subdue using three different objective functions to produce aggregated PFs on several graph datasets, as well as MOSubdue.

TABLE I
Description of different graph-represented datasets used.

Dataset	#Graphs	#Nodes	#Edges	#Unique Labels	Run Time (secs)	#True/ Pseudo PF
*random*1	100	2954	3009	7	1000	27
*random*2	200	5876	6015	7	1000	30
US	10	2762	2769	294	5000	9
UK	10	2732	2748	292	5000	9
Germany	10	2676	2702	284	5000	9

A. Used Graph-represented Datasets

We have used a total of 5 graph datasets during the experimental study. Table I provides some characteristics of these datasets. The first two datasets *random*1 and *random*2 were randomly generated using the random graph generator available at Subdue's website[1]. The last three are visual science maps datasets (or *Scientograms* [15]) built following De Moya-Anegón et al.'s methodology [16], [17]. The nodes of the graphs correspond to the Elsevier SCOPUS-SJR co-citation categories. Each category agglutinates the journals that were classified under that name, and likewise the documents that were published in those journals. The edges represent the strength of the co-citation measure between two categories. In this contribution, we will deal with three datasets compiled for the countries United States (*US*), United Kingdom (*UK*), and Germany over the period of 1996 to 2005. The optimal Pareto sets are not known for these datasets, so we computed the pseudo-optimal Pareto set by the fusion of all the different approximations produced by the various methods considered.

B. Parameter Setting

Subdue: The current version of Subdue[2] supports three subgraph evaluation objectives, viz. MDL, support, and size. We run Subdue with each of the three evaluation objectives in order to apply it for multiobjective GBDM. The output is an aggregated subgraph set produced from the subgraphs discovered by the three different objectives. The size of the aggregated set is fixed to 100, which contains a maximum 1/3 of aggregated set size subgraphs corresponding to each of the objectives. Finally, the nondominated subgraph set produced by the Subdue method is obtained by the application of nondominance definition (eq. 3) on the aggregated set. The algorithm parameter *beam-width* is set to three different values 5, 10, and 20 to obtain the results.

EP-Subdue: A single-objective EP-Subdue method [3] is implemented that supports three evaluation objectives, viz. MDL, support and size, in order to produce a set of nondominated subgraphs. On a dataset, with each of the objectives, the method applies fitness proportionate selection, modified mutation operator as mentioned in sub section III.C to create new subgraphs, and stores best found subgraphs per generation. We apply the same procedure as Subdue to combine the solutions

[1] http://ailab.wsu.edu/subdue/datasets/subgen.tar.gz
[2] http://ailab.wsu.edu/subdue/software/subdue-5.2.1.zip

using the different objectives. The parameters of EP are set to: population = 100, size of aggregated set = 100.

MOSubdue: As said, we have also applied the MOSubdue variant. It is a random search method and thus it has been run 10 times independently on each dataset. A final approximation to the PF obtained by MOSubdue can be obtained as a fusion of the ten different runs. It is applied with the parameter *beamwidth* set to 5, as worst results were obtained for the values 10 and 20.

MOEP: We use a population of size 100 and we run it 10 times independently on each dataset.

The results have been obtained by executing the different algorithms for some fixed duration on each dataset. Considering the complexities of the used datasets, we have allotted different execution times as given in Table I. A single run by Subdue and EP-Subdue methods on each dataset consists of an execution with each of the three objectives for 1/3 of the given execution time.

C. Metrics of performance

In this paper, we have considered the two usual kinds of multiobjective metrics [14]:

- those which measure the quality of a nondominated solution set returned by an algorithm, and
- those which compare the performance of two different algorithms.

As regards the former group, we use the hypervolume ratio (*HVR*) to compare the obtained Pareto set approximations. The hypervolume measures the volume enclosed by a Pareto set approximation with respect to a reference point. For two-dimensional objective vector, hypervolume is the summation of area covered by each member of the Pareto set. In the case of maximization problem, as ours, we define a reference point as $(0, 0)$. Here, we use the hypervolume ratio (HVR) [14]. HVR measures both diversity and closeness of the Pareto set and is calculated as:

$$\mathit{HVR} = \frac{H_1}{H_2} \tag{4}$$

where H_1 and H_2 are the volume of the Pareto set and the true Pareto set (or the pseudo-optimal Pareto set in case the latter is not known) respectively. In our case, we will use a pseudo-optimal Pareto set. A value of HVR equal to one represents that the Pareto front and the pseudo Pareto front are equal.

The unary metrics allow us to determine the absolute, individual quality of the obtained Pareto set approximation, but they cannot be used for comparison purposes. On the other hand, binary indicators have been proposed for comparing the Pareto set approximations obtained by different multiobjective algorithms. In this work, we have used the following two binary indicators C and I_ϵ to compare the Pareto set approximations two by two.

The coverage metric (*C-measure*) compares a pair of non-dominated sets by computing the fraction of each set that is covered by the other [18]:

$$C(X', X'') = \frac{|\{\forall g'' \in X''; \exists g' \in X' : g' \ \succeq \ g''\}|}{|X''|} \tag{5}$$

where $g' \ \succeq \ g''$ indicates that the subgraph g' dominates or cover the subgraph g'' in a maximization problem. A value of $C(X', X'') = 1$ means that all the subgraphs in X'' are dominated or covered by the subgraphs in X'.

The I_ϵ-measure gives the minimum distance by which one set can be translated in each dimension in objective space such that the other set is weakly dominated [14]. In this paper, we consider the additive epsilon indicator, I_ϵ, given as:

$$\begin{aligned} I_\epsilon(X', X'') = & \min_\epsilon \{ f_i(g') + \epsilon \geq f_i(g''); \\ & g' \in X', g'' \in X'', \text{for } i = 1, \ldots, o \ \} \end{aligned} \tag{6}$$

where I_ϵ value is the minimal amount ϵ by which one needs to improve each objective, i.e., replace $f_i(g')$ by $f_i(g') + \epsilon$, such that it just dominates $f(g'')$, i.e., $f_i(g') + \epsilon \geq f_i(g'')$ for all $i = 1, \ldots, o$. A value of $I_\epsilon(g', g'') < I_\epsilon(g'', g')$ means that $f(g')$ dominates $f(g'')$ and $I_\epsilon(X', X'')$ indicates the minimal amount ϵ by which at least one subgraph in the Pareto set approximation X' dominates at least one subgraph in the Pareto set approximation X''.

For a Pareto set approximation, the HVR measure is better when it tends to one. In the pair-wise comparison of Pareto set approximations the C measure is better when it tends to one, while the I_ϵ measure is better when it takes lower value.

D. Experimental Analysis

Tables II presents the mean and standard deviation of the *HVR*-metric values of the final PF approximations obtained by the different algorithms. Figs. 1 and 2 show the distribution of the values of C and I_ϵ metrics. For non-deterministic algorithms (all but Subdue), 10 runs were conducted. Non-dominated fronts achieved by the different algorithms are also shown in Figs. 3 and 4. For non-deterministic algorithms, the 10 fronts were unified and the dominated solutions removed.

We can make the following remarks in view of the obtained results. With equal computational time, the single-objective algorithms (Subdue and EP-Subdue) have produced the worst Pareto front approximations to the reference PFs on all the datasets when compared to that by the multiobjective search enabled algorithms, MOSubdue and MOEP (see Table II and Figs. 1 to 4). This conclusion can be further evident from the average of *HVR*-metric values over the five datasets, which is much lower: 0.7428 for Subdue, and 0.6924 for EP-Subdue as compared to that of the multiobjective search methods. This suggests the incorporation of multiobjective search strategy enables the algorithm to explore more efficiently the multiobjective subgraph solution search space. When comparing the results with MOEP, the later obtained the best results on all datasets, with even a lower deviation.

The performance of the EMO approach is very comparable or better to that of MOSubdue method when considering the binary metrics. Fig. 1 shows that the solutions of MOEP

TABLE II

HVR-METRIC VALUES FOR THE SUBGRAPHS FOUND BY DIFFERENT METHODS. THE NUMBERS IN THE PARENTHESES REPRESENT THE STANDARD DEVIATION

Dataset	Subdue	EP	MOSubdue	MOEP
*random*1	0.9210(-)	0.9203(0.01)	0.9826(0.00)	0.9843(0.00)
*random*2	0.9260(-)	0.9264(0.01)	0.9822(0.00)	0.9828(0.00)
US	0.6099(-)	0.5285(0.10)	0.9108(0.11)	0.9826(0.02)
UK	0.5879(-)	0.5580(0.12)	0.8065(0.19)	0.9310(0.05)
Germany	0.6693(-)	0.5287(0.12)	0.8999(0.05)	0.9370(0.04)
Average	0.7428(0.17)	0.6924(0.21)	0.9164(0.07)	**0.9635**(0.03)

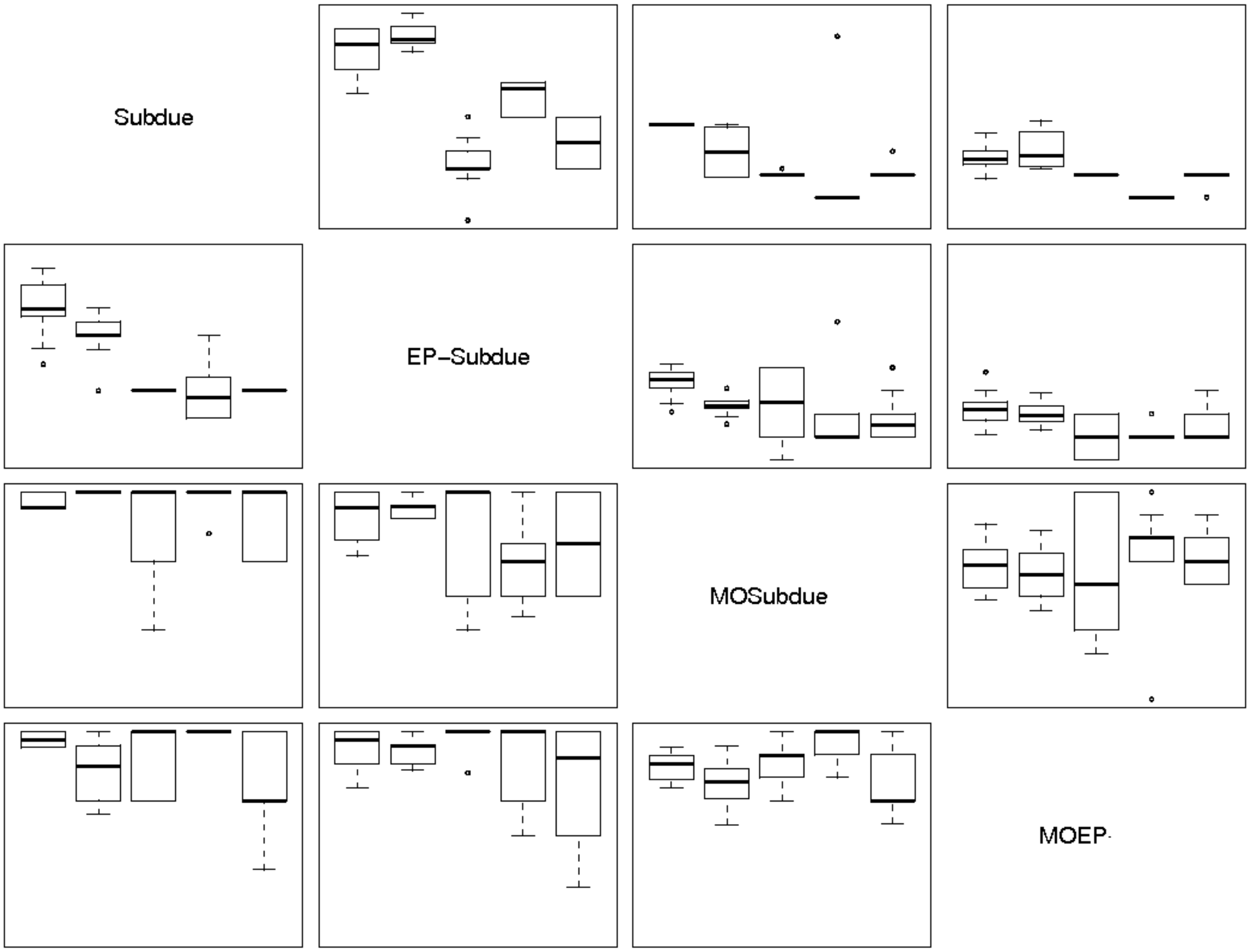

Fig. 1. Box plots based on the C-metric computed for all the methods considered. Each rectangle contains 5 box plots representing the distribution of the C-values for a certain ordered pair of algorithms; the leftmost box plot relates to *random*1 dataset, the rightmost to *Germany* dataset. The scale is 0 at the bottom and 1 at the top per rectangle. Furthermore, each rectangle refers to algorithm A associated with the corresponding row and algorithm B associated with the corresponding column and gives the fraction of B covered by A $(C(A, B))$.

dominate in general those of the other algorithms, even if some of them are dominated for the *UK* dataset by MOSubdue solutions. When looking the I_{ϵ}-metric (see Fig. 2), MOEP shows clearly is superiority over MOSubdue. We can also mention that MOEP missed only one solution of the pseudo Pareto set in only one case (see Figs. 3 and 4). The fundamental difference between the two approaches is that MOEP performs global search as the algorithm, at any generation, conforms the population of solutions belonging to different regions of the search lattice. As against, MOSubdue performs local search. At any generation, the multiobjective beam search utilizes the solutions belonging to a single region of the search lattice, thus developing a depth search only in some selected branches and consequently having the chance to miss out promising

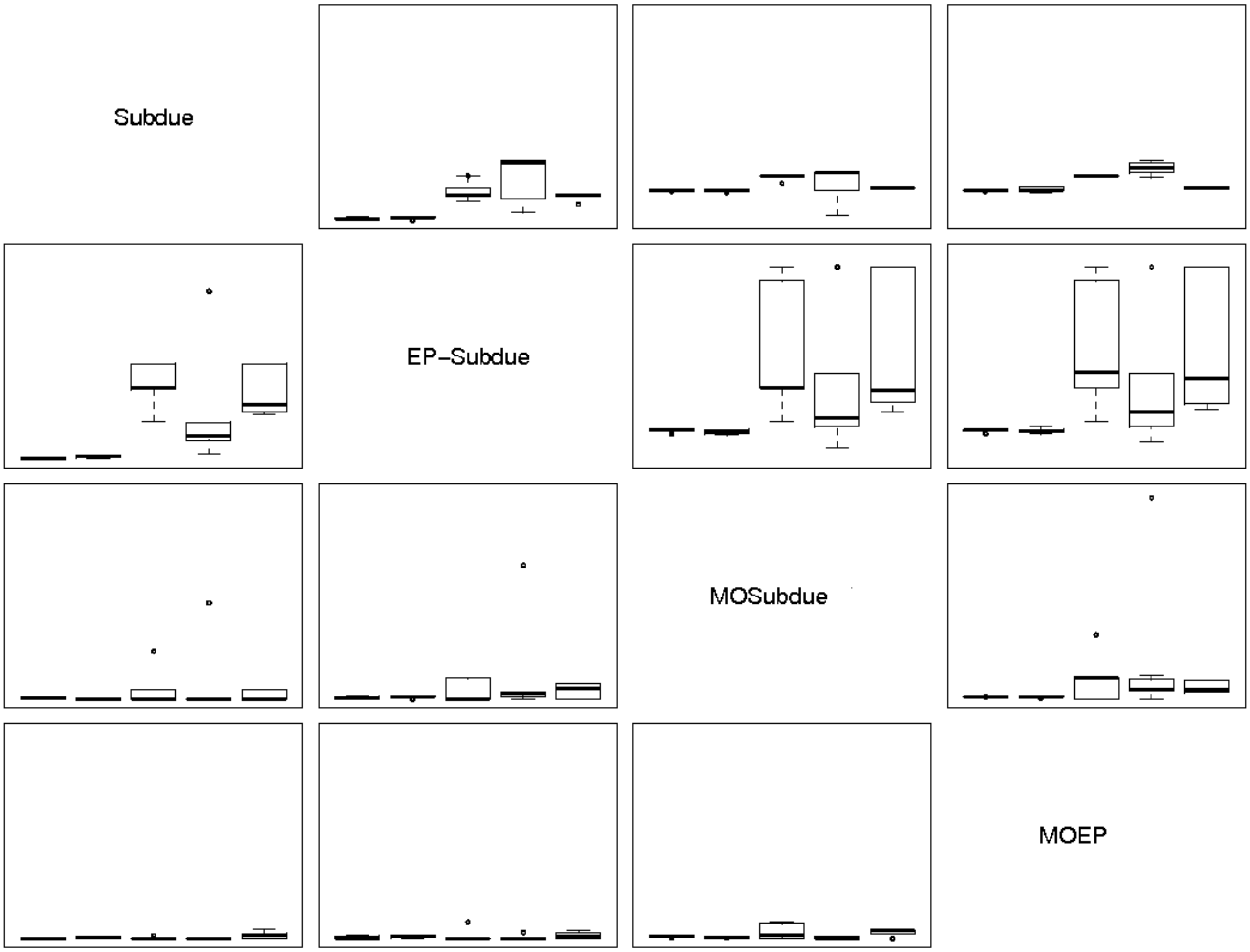

Fig. 2. Box plots based on the I_ϵ-metric computed for all the methods considered. Each rectangle contains 5 box plots representing the distribution of the I_ϵ-values for a certain ordered pair of algorithms; the leftmost box plot relates to *random*1 dataset, the rightmost to *Germany* dataset. The scale is 0 at the bottom and 11 at the top per rectangle. Furthermore, each rectangle refers to algorithm A associated with the corresponding row and algorithm B associated with the corresponding column and gives the minimum distance by which the nondominated subgraph set produced by A can be translated in each dimension in the objective space such that the nondominated subgraph set produced by B is weakly dominated $I_\epsilon(A, B)$.

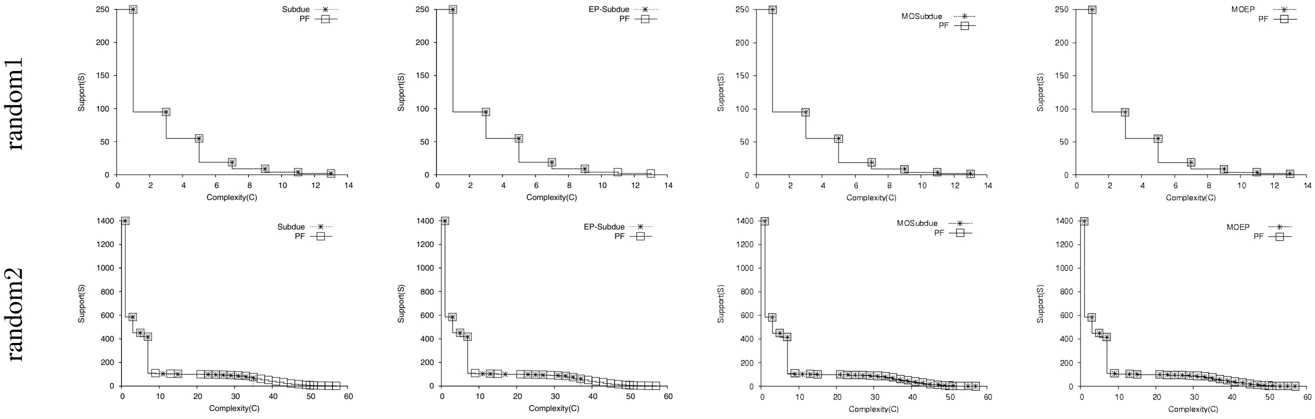

Fig. 3. Plot of the nondominated front found by different algorithms for the random datasets. For comparison the reference Pareto front (PF) is also shown.

solutions.

V. Conclusion

In the present study, we have successfully shown how our MOEP algorithm can tackle the challenging problem of

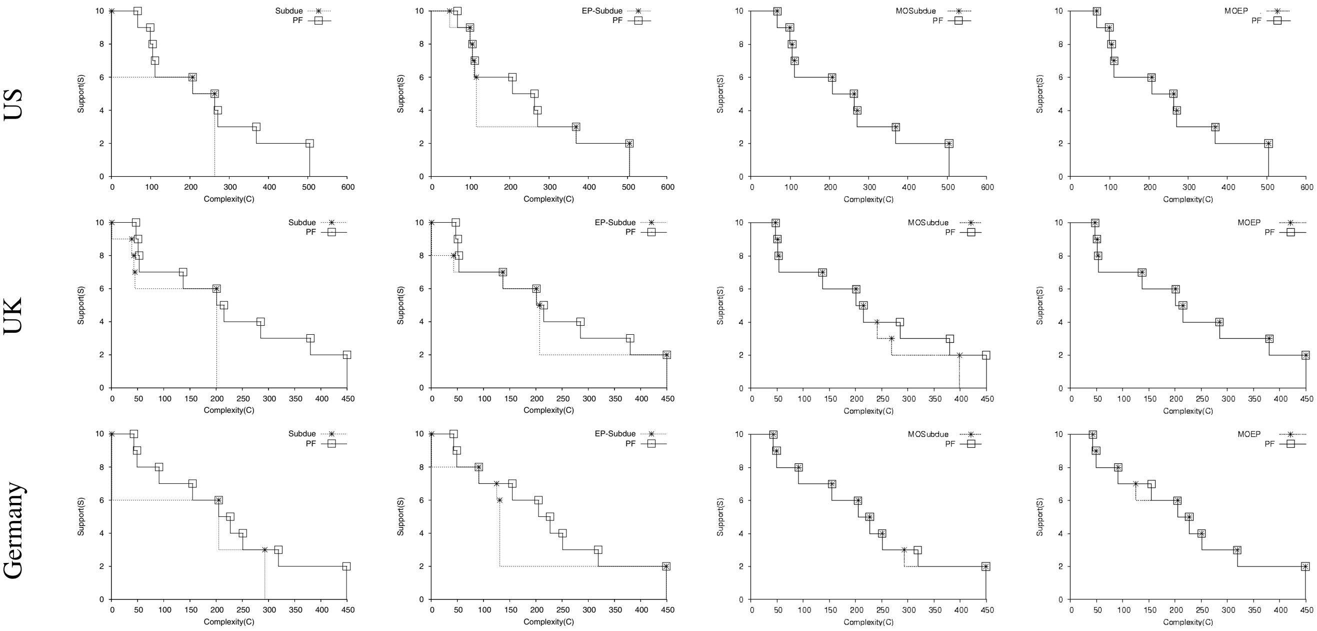

Fig. 4. Plot of the nondominated front found by different algorithms for the real datasets. For comparison the reference Pareto front (PF) is also shown.

multiobjective subgraph discovery in graph-represented data. We consider that the appearance of such kind of data and applications highlights the need of efficient graph mining methods. Compared to single-objective Subdue and EP-Subdue, and MOSubdue, we demonstrated that our EMO-based algorithm succeeds in generating the best approximation to the pseudo Pareto-optimal set.

As future work, we will consider the implementation of new MOEP variants, the development of a crossover operator for subgraph generation, and the comparison with more diverse graph datasets.

Acknowledgment

This work has been partially supported by the Spanish Ministry of Science and Innovation (MICINN) under project TIN2009-07727, including EDRF fundings.

References

[1] D. Cook and L. Holder, Eds., *Mining Graph Data*. London: Wiley, 2007.

[2] T. Washio and H. Motoda, "State of the art of graph-based data mining," *ACM SIGKDD Explor Newsl*, vol. 5, no. 1, pp. 59–68, 2003.

[3] S. Bandyopadhyay, U. Maulik, D. J. Cook, L. B. Holder, and Y. Ajmerwala, "Enhancing structure discovery for data mining in graphical databases using evolutionary programming," in *Proc Florida Artif Intell Res Symp*, 2002, pp. 232–236.

[4] R. C. Romero-Zaliz, C. Rubio-Escudero, J. P. Cobb, F. Herrera, Ó. Cordón, and I. Zwir, "A multiobjective evolutionary conceptual clustering methodology for gene annotation within structural databases: A case of study on the *gene ontology* database," *IEEE Trans Evol Comput*, vol. 12, no. 6, pp. 679–701, 2008.

[5] P. Shelokar, A. in, and Ó. Cordón, "A multiobjective variant of the subdue graph mining algorithm based on the NSGA-II selection mechanism," in *Proc IEEE World Congr Comput Intell*, 2010, pp. 463–470.

[6] A. Papadopoulos, A. Lyritsis, and Y. Manolopoulos, "SkyGraph: An algorithm for important subgraph discovery in relational graphs," *Data Min Knowl Disc*, vol. 17, pp. 57–76, 2008.

[7] D. Cook and L. Holder, "Graph-based data mining," *IEEE Intell Syst*, vol. 15, pp. 32–41, 2000.

[8] D. J. Cook and L. B. Holder, "Substructure discovery using minimum description length and background knowledge," *J Artif Intell Res*, vol. 1, pp. 231–255, 1994.

[9] J. Rissanen, *Stochastic Complexity in Statistical Inquiry Theory*. River Edge: World Scientific Publishing Co., Inc., 1989.

[10] B. T. Lowerre, "The HARPY speech recognition system," Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, 1976.

[11] V. Chankong and Y. Y. Haimes, *Multiobjective Decision Making Theory and Methodology*. Amsterdam: North-Holland, 1983.

[12] T. Gal, T. Stewart, and T. Hanne, Eds., *Multicriteria Decision Making: Advances in MCDM Models, Algorithms, Theory and Applications*. Dordrecht: Kluwer Academic, 1999.

[13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans Evol Comput*, vol. 6, pp. 182–197, 2002.

[14] K. Deb, *Multi-objective Optimization using Evolutionary Algorithms*. Chichester, UK: Wiley, 2001.

[15] A. Quirin, Ó. Cordón, B. Vargas-Quesada, and F. Moya-Anegon, "Graph-based data mining: A new tool for the analysis and comparison of scientific domains represented as scientograms," *J Informetr*, vol. 4, no. 3, pp. 291–312, 2010.

[16] F. D. Moya-Anegón, B. Vargas-Quesada, V.Herrero-Solana, Z. Chinchilla-Rodríguez, E. Corera-Álvarez, and F. J. Munoz-Fernández, "A new technique for building maps of large scientific domains based on the cocitation of classes and categories," *Scientometrics*, vol. 61, no. 1, pp. 129–145, 2004.

[17] B. Vargas-Quesada and F. D. Moya-Anegón, *Visualizing the Structure of Science*. Secaucus: Springer-Verlag New York, 2007.

[18] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans Evol Comput*, vol. 3, pp. 257–271, 1999.