# Building and Using an Ontology of Preference-Based Multiobjective Evolutionary Algorithms

Longmei Li[1,2](✉), Iryna Yevseyeva[3], Vitor Basto-Fernandes[4,5], Heike Trautmann[6], Ning Jing[1], and Michael Emmerich[2]

[1] School of Electronic Science and Engineering, National University of Defense Technology, Changsha 410073, Hunan, China
{longmeili,ningjing}@nudt.edu.cn
[2] Leiden Institute of Advanced Computer Science, Leiden University, 2333CA Leiden, The Netherlands
{l.li,m.t.m.emmerich}@liacs.leidenuniv.nl
[3] Faculty of Technology, School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, UK
iryna.yevseyeva@dmu.ac.uk
[4] Instituto Universitário de Lisboa (ISCTE-IUL), University Institute of Lisbon, ISTAR-IUL, Av. das Forças Armadas, 1649-026 Lisbon, Portugal
vitor.basto.fernandes@iscte.pt
[5] School of Technology and Management, Computer Science and Communications Research Centre, Polytechnic Institute of Leiria, 2411-901 Leiria, Portugal
[6] Department of Information Systems, University of Münster, 48149 Münster, Germany
trautmann@wi.uni-muenster.de

**Abstract.** Integrating user preferences in Evolutionary Multiobjective Optimization (EMO) is currently a prevalent research topic. There is a large variety of preference handling methods (originated from Multicriteria decision making, MCDM) and EMO methods, which have been combined in various ways. This paper proposes a Web Ontology Language (OWL) ontology to model and systematize the knowledge of preference-based multiobjective evolutionary algorithms (PMOEAs). Detailed procedure is given on how to build and use the ontology with the help of Protégé. Different use-cases, including training new learners, querying and reasoning are exemplified and show remarkable benefit for both EMO and MCDM communities.

**Keywords:** Preference · Evolutionary Multiobjective Optimization · Multicriteria decision making · OWL ontology · Protégé

## 1 Introduction

Evolutionary Multiobjective Optimization (EMO) [13,14] and Multiple Criteria Decision Making (MCDM) [34,50] are two research areas dealing with multiobjective optimization. While traditional MCDM aims at finding one most preferred


H. Trautmann et al. (Eds.): EMO 2017, LNCS 10173, pp. 406–421, 2017.
DOI: 10.1007/978-3-319-54157-0_28

solution based on a preference model using (mainly) mathematical programming, EMO, on the other hand, uses population-based evolutionary algorithms to obtain the whole set of Pareto optimal solutions. User selection is the second step after (an approximation of) the Pareto front (PF) has been achieved. Generally, such methods can be considered as a-posteriori methods in MCDM [34].

Since 2004 when EMO and MCDM researchers met at the Dagstuhl seminar, great attention has been drawn to a collaboration of the two fields. MCDM can help to relieve the selection burden of the Decision Maker (DM) and plays an important role in many-objective optimization (when the number of objectives is no less than three). EMO can tackle difficult problems (such as discontinuous, nondifferentiable, nonconvexity, etc.) that MCDM fails to handle.

Preference-based multiobjective evolutionary algorithms (PMOEAs) are a collaboration of EMO and MCDM. They use preference information provided by the DM to guide the search towards preferred parts of the PF, instead of approximating the whole PF. A variety of PMOEAs have been proposed, they utilize diverse kinds of preferences (reference point, desirability functions, outranking relation, etc.) at different interaction moments (a-priori, interactive, a-posteriori), integrated with several categories of MOEAs (Pareto dominance-based (e.g. PAES, NSGA-II, SPEA2), indicator-based (e.g. SMS-EMOA, HypE, POSEA [53]), decomposition-based (e.g. MOEA/D, NSGA-III), etc.), by means of various integration methods (change of objectives, constraints, or components of algorithms).

There are good review papers along the development of PMOEAs [4,7,12,41], the core value of reviews is to gain knowledge, figure out what has already been done and what are the relations between different works. Ontologies are currently the most suitable and widely used method to formally describe knowledge, by means of comprehensive notations and graphical representations, to help understand concepts and relationships in complex knowledge domains [46].

In this paper we propose an OWL ontology of PMOEAs to model and systematize the results in this domain. With the help of this ontology, researchers can easily understand, access, and analyze methods, or identify future research topics. The PMOEA ontology is made public and is extensible, it can also be reused for building ontologies for EMO and MCDM knowledge domains.

The rest of this paper is structured as follows: In Sect. 2 we give a brief introduction of PMOEAs and OWL ontology. Section 3 provides a detailed procedure of building and extending the PMOEA ontology. Section 4 exemplifies the usage of the proposed ontology. Finally, Sect. 5 concludes the work with a summarizing discussion and the outlook.

## 2 Background

### 2.1 PMOEAs

Compared to MOEAs, PMOEAs attracted wide attention in the last decade owing to some advantages. Firstly, the ultimate goal of multiobjective optimization is to assist the DM with finding a best solution. Since visualization and inspection of the whole PF is not a trivial task, a PMOEA can be used to shrink the searching space and to alleviate the selection burden of the DM. Secondly, it is challenging for MOEAs to handle many-objective optimization because most of the solutions are non-dominated. Extending the Pareto order by additional preference information is seen as a great help, if not necessity, in this context. Thirdly, focusing the search within preferred parts will reduce computational efforts spent on developing solutions in less preferred regions.

Because of these benefits, a large number of publications have appeared recently in the PMOEA area. Preference information is an essential element of PMOEAs. In the following, we classify PMOEAs by preference information and list the representative works in each category.

*Reference point* [52] is probably the most popular approach to embed preferences in PMOEAs. A reference point (or goal vector) is a user-defined point in the objective space indicating the DM's aspiration level for each objective. The closer a solution is to a reference point, the more it is preferred. It was used to change Pareto dominance relation (e.g. g-dominance [36], r-dominance [6], Chebyshev preference relation [26]), modify crowding distance (e.g. R-NSGA-II [18]), alter set quality indicator (e.g. PBEA [47], R2-EMOA [49]) and integrate with swarm-based algorithms (e.g. RPSO-SS [2]).

*Reference direction* [31] and *light beam search* [27] are extensions of reference point method in MCDM. They were also integrated with EMO (e.g. RD-NSGA-II [16], LBS-EMO [15]).

*Preference region* is a region in the objective space that is of interest to the DM. Weight functions are widely used as reflection of the DM's degree of satisfaction on objective values (within a user-defined region), such as weighted Hypervolume indicator [11], weighted NSGA-II [21]. Desirability functions are another popular means of preference articulation, which was embedded into NSGA-II [48], SMS-EMOA [51] and MOPSO [37] on both benchmark and real-world applications.

*Trade-off* is a threshold of how much the DM can sacrifice in the value of one objective in order to improve the other objective(s). It was incorporated with Pareto dominance relation (e.g. in G-MOEA [10]), front sorting (e.g. in pNSGA-II [44]) and Cone-based Hypervolume indicator (CHI) [19,43].

*Knee point* is the solution with maximal trade-off, often regarded as the most preferred when no explicit preference is given. It was introduced in EMO to guide the search to knee point, e.g. in TKR-NSGA-II [5].

*Objective comparison* is preference on objectives. For instance objectives can be qualitatively ordered as "most important, important, less important", or quantitatively assigned with different weights. Representative works include development of fuzzy preference relation and dynamic weights [28], relative importance of objectives [42], selection based on fuzzy measure and fuzzy integral [30].

*Solution comparison* is often used in interactive methods when the DM is asked to compare two alternatives, or select the best (or/and worst) among a sample set during the process of optimization. Preference elicitation is of great significance in this category. Main approaches include value function fitting-based methods (e.g. NEMO [8,9], PI-EMO-VF [17]), machine learning-based methods (e.g. Brain-Computer EMOA [3], Neural Network TDEA [40]), polyhedral cone-based methods (e.g. PI-EMO-PC [45]), rules deduction-based methods(e.g. DRSA-EMO [22]).

*Outranking* is a binary relation which expresses the degree of truth on the predicate "$x$ is at least as good as $y$". It was combined with outranking-based dominance in NOSGA [20].

Performance metrics of PMOEAs have also attracted attention in that classical EMO metrics are no longer feasible, because they measure convergence and diversity considering the whole PF, without consideration of the preference information. A new performance metric for reference point-based MOEAs was proposed in [35], a testing framework to compare different reference point-based interactive methods was also introduced in [39], which provide some suggestions on this issue.

Due to the variety of approaches and contexts where preference modelling is combined, to obtain a comprehensive view of PMOEAs becomes increasingly complex. Ontologies are currently an appropriate method to formally describe knowledge in a standard way, to help understand concepts and relationships in complex knowledge domains.

### 2.2 OWL Ontology

Ontologies are content theories about the sorts of objects, properties of objects and relations between objects that are possible in a specified domain of interest. The most widely accepted definition of ontology in this context is given by Gruber [23]: "An ontology is a formal explicit specification of a shared conceptualization for a domain of interest." It is formal and logic-based, which makes reasoning possible; it has explicit specification, which makes it easy for new learners of this domain; it is a shared conceptualization, which defines a common vocabulary for researchers who need to share information in this domain.

Web Ontology Language (OWL) [25] was approved by World Wide Web Consortium (W3C) to be one of the key Semantic Web technologies in 2004. It employs the eXtensible Markup Language (XML) for the definition of text-based documents syntax/structure, capable of reasoning (OWL-DL [33]) such as consistency and subsumption checking. OWL ontologies can be published online and may refer to or be referred from other OWL ontologies.

Reasons to develop the PMOEA ontology are the following [38]:

(1) To share a common understanding of the structure of information in the PMOEA domain. Concepts in PMOEA domain include preference information, multiobjective evolutionary algorithms and their components (selection operator, variation operator, etc.), multiobjective problems (academic and real-world), preference elicitation methods and so on. It is of high value to represent this information with machine-interpretable vocabulary and analyze it with knowledge discovery tools.

(2) To enable reuse of domain knowledge. Since PMOEA is the connection of EMO and MCDM domains, there are relevant common concepts and relations of concepts in these three domains. Many concepts and relations can be reused in EMO (also Evolutionary Computation) and MCDM ontologies.
(3) To analyze domain knowledge of PMOEA. Building the PMOEA ontology is seen as a step towards the harmonization and systematization of knowledge management in the PMOEA domain. To some extent building an ontology is like defining a set of data and their structure, after collecting information compatible with this structure, analysis can be done to provide more useful information. By analyzing the PMOEA ontology, we can find out what kind of preference information has been integrated in what kind of MOEA, through what kind of integration. We can also query for algorithms that can deal with a specific kind of problem, or find potential combination of MCDM and EMO for future research.

Ontologies are widely used in a variety of research fields, such as knowledge management, recommendation systems, e-Learning, e-Commerce, semantic web, bioinformatics and so on. As far as we know, they have not been applied in the PMOEA field, but an Evolutionary Computation Ontology for e-Learning [29] has been proposed, revealing the feasibility and suitability of using ontologies for optimization algorithms design domain.

Although complete reuse of the e-Learning ontology was not possible for the PMOEA ontology, common concepts and vocabulary were adopted when possible, in order to contribute for knowledge representation harmonization in this domain.

## 3 Building the PMOEA Ontology

*Protégé* is a free, open-source platform which provides a growing community with a suite of tools to construct domain models and knowledge-based applications with ontologies [1]. It was developed and maintained by Stanford Center for Biomedical Informatics Research (BMIR) and now has more than 300 thousand registered users. People from different background can publish, view, download, (collaboratively) edit OWL ontologies for research freely. The PMOEA ontology was built with the help of *Protégé Desktop* and made public in *WebProtégé*.

An OWL ontology comprises *Classes*, *Properties* and *Individuals*. A class describes a group of concepts with the same properties and may have necessary and sufficient conditions an individual must verify to belong to that class. A class can have subclasses that represent concepts more specific than the superclass. The hierarchy of classes, which can be represented as a tree structure that relates classes by is-a relation, defines the taxonomy adopted in the ontology. For example, **PMOEA** is the class of preference-based multiobjective evolutionary algorithms. **ReferencePoint_based** is subclass of *PMOEA* which must have reference point as preference information. There are two kinds of properties: object properties and data properties. An object property is a binary relation to relate classes or individuals. For instance, `hasPreferenceInformation` is an object property that can relate **PMOEA** and **PreferenceInformation-FromDM**, which indicates the preference information provided by the DM. A data property relates classes or individuals with a designed primitive data-type (e.g. integer, string, boolean). For example, hasDevelopingYear is a data property of **PMOEA** with datatype **integer**. Individuals represent class instances in the domain of interest, e.g. *r-NSGA-II* [6] is an individual of **PMOEA**.

The general process of building ontologies is given by Noy [38]: (1) determine the scope and domain of the ontology; (2) consider reusing existing ontologies; (3) enumerate important terms in the domain; (4) define the class hierarchy; (5) define object properties; (6) define data properties; (7) create individuals; (8) publish.

Among the above steps (4)–(7) are of core importance for the ontology design, we will give a detailed description of these steps next.

### 3.1 Class Hierarchy

A tree view of the class hierarchy is shown in Fig. 1.

**MetaHeuristic** reveals the searching method to find optimal solutions. **MOEA** (multiobjective evolutionary algorithms) and **SOEA** (single-objective evolutionary algorithms) are subclasses of **MetaHeuristic** class, **MOEA** contains **DiversityVSConvergence_based** (individuals *NSGA-II*, *SPEA2*, etc.), **Indicator_based** (individuals *SMS-EMOA*, *HypE*, etc.), **Decomposition_based** (individuals *NSGA-III*, *MOEA/D*, etc.), **Swarm_based** (individuals *MOPSO*, etc.), **Memetic** (individuals *PMA*, etc.), **Coevolution_based** (individuals *CCEA*, etc.)[1]. **PMOEA** is also a subclass of **MOEA**, whose individuals will be introduced in Sect. 3.4. **ObjectiveSpaceTransformation_based** and **ReferencePoint_based** are inferred subclasses of **PMOEA**, which will be introduced in Sect. 4.

**ImplementationLibrary** is the library or framework used by metaheuristics for implementation. It includes *jMetal*, *KanGAL*, *PISA*, *MOEAFramework*, etc. as individuals.

**InteractionTime** indicates the moment when the DM interacts with the optimization process: *a-priori*, *a-posteriori* and *progressive* are individuals of this class.

**LearningMethod** refers to the learning or preference elicitation methods used by some PMOEAs (usually interactive approaches) to mimic the DM's preferences. Subclasses include **OrdinalRegression**, **LinearProgramming**, **QuadraticProgramming**, **SupportVectorMachine**, **NeuralNetwork**.

**MOP** is the class of multiobjective optimization problems, which has **Academic_Problem** and **Realworld_Problem** as subclasses. **Academic_Problem** has subclasses **DTLZ**, **Knapsack**, **WFG**, **ZDT**.

**PreferenceInformationFromDM** indicates what the DM should provide to express his/her preferences. Subclasses include **BudgetofDMcalls**,

[1] Strictly speaking, not all of these algorithms are evolutionary algorithms, we consider to change MOEA to MOMH (Multiobjective MetaHeuristic) in the future version.

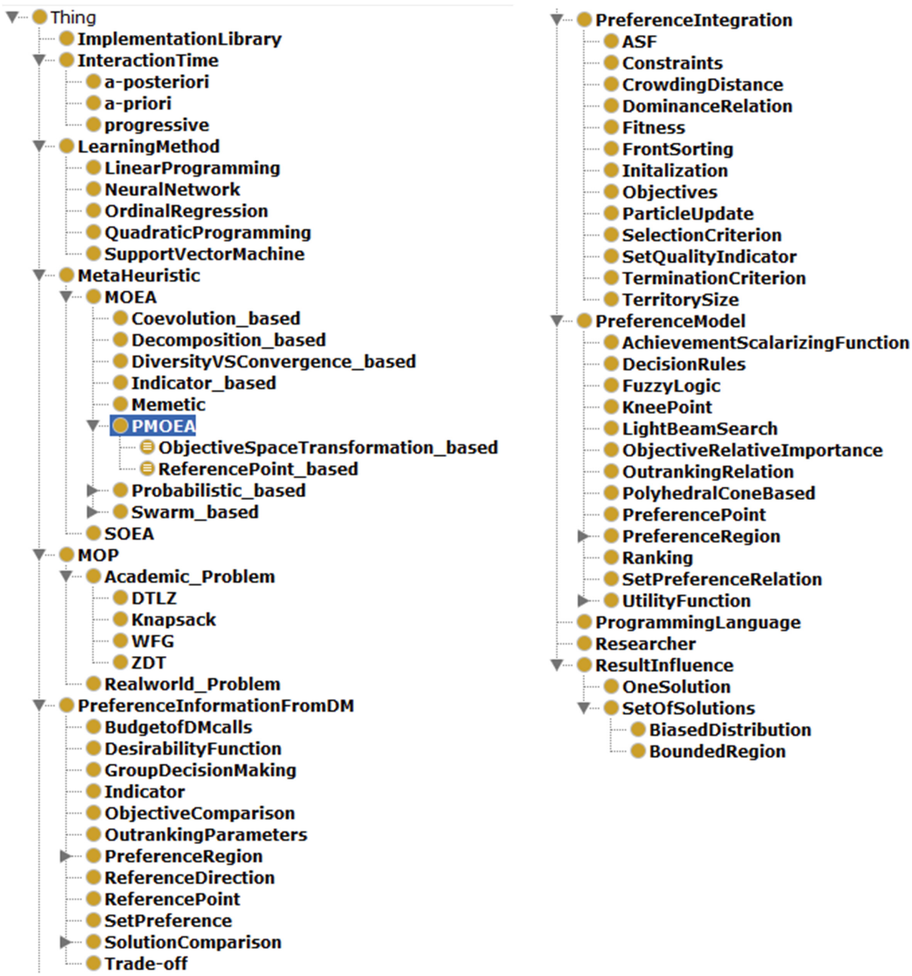


**Fig. 1.** Class hierarchy

**DesirabilityFunction**, **GroupDecisionMaking**, **Indicator**, **ObjectiveComparison**, **OutrankingParameters**, **PreferenceRegion**, **ReferenceDirection**, **ReferencePoint**, **SolutionComparison** (where **PairwiseComparison**, **SampleRanks** and **SampleSorts** are subclasses), **Trade-off**.

**PreferenceIntegration** defines how the preference information is integrated in the search method, i.e. what is modified in the optimization process to incorporate preferences. Subclasses include the following: **ASF** (achievement scalarizing function), **Constraints**, **CrowdingDistance**, **DominanceRelation**, **Fitness**, **FrontSorting**, **Initialization**, **Objectives**, **ParticleUpdate**, **SelectionCriterion**, **SetQualityIndicator**, **TerminationCriterion**, **TerritorySize**.

**PreferenceModel** specifies the preference model applied in the PMOEA. It is strongly related to **PreferenceInformationFromDM**, but it focuses on the internal model utilized by the algorithm, about which the DM does not care or know. Its subclasses include **AchievementScalarizingFunction**, **DecisionRules**, **FuzzyLogic**, **KneePoint**, **LightBeamSearch**, **ObjectiveRelativeImportance**, **OutrankingRelation**, **PolyhedralConeBased**, **PreferencePoint**, **PreferenceRegion**, **UtilityFunction** (which has **Linear**, **AdditivePiecewiseLinear**, **GeneralAdditive**, **Quasiconcave**, **ChoquetIntegral**, **Polynomial**, **SetPreferenceRelation**, **DesirabilityFunction** as subclasses).

**ProgrammingLanguage** refers to the language used for implementation of Metaheuristics.

**Researcher** is the class of researchers in this domain, who are authors of academic papers.

**ResultInfluence** shows the type of the result, which can be classified as **OneSolution** and **SetOfSolutions**. **BiasedDistribution** and **BoundedRegion** are subclasses of **SetOfSolutions**.

### 3.2 Object Properties

Object properties are binary relations on individuals, they may be functional, transitive, symmetric and reflexive. Object properties may have a domain and a range specified. For example, *R-NSGA-II* `canSolve` *ZDT1*, `canSolve` is an object property whose domain is *MetaHeuristic* and range is *MOP*. The main

**Table 1.** Object properties

| Object property | Domain | Range |
|---|---|---|
| `hasResultInfluence` | **MetaHeuristic** | **ResultInfluence** |
| `hasPreferenceModel` | **PMOEA** | **PreferenceModel** |
| `canSolve` | **MetaHeuristic** | **MOP** |
| `hasSearchAlgorithm` | **PMOEA** | **MetaHeuristic** |
| `hasInteractionTime` | **PMOEA** | **InteractionTime** |
| `hasAuthor` | **MetaHeuristic** | **Researcher** |
| `hasPreferenceInformationFromDM` | **PMOEA** | **PreferenceInformationFromDM** |
| `hasPreferenceIntegration` | **PMOEA** | **PreferenceIntegration** |
| `hasLearningMethod` | **PMOEA** | **LearningMethod** |
| `isInteractiveVersionOf` | **PMOEA** | **PMOEA** |
| `hasInteractiveVersion` | **PMOEA** | **PMOEA** |
| `hasComparison` | **MetaHeuristic** | **MetaHeuristic** |
| `isExtensionOf` | **MetaHeuristic** | **MetaHeuristic** |
| `hasExtension` | **MetaHeuristic** | **MetaHeuristic** |
| `useLibrary` | **MetaHeuristic** | **ImplementationLibrary** |
| `useLanguage` | **MetaHeuristic** | **ProgrammingLanguage** |

object properties in our ontology are listed in Table 1. Note that one individual can be related to several individuals with the same object property, such as *R-NSGA-II* `hasPreferenceInformationFromDM` *ReferencePoint* and *R-NSGA-II* `hasPreferenceInformationFromDM` *weights* hold at the same time.

`isInteractiveVersionOf` and `hasInteractiveVersion` are inverse of each other, which means "A `isInteractiveVersionOf` B" infers "B `hasInteractiveVersion` A" and vise versa. `isExtensionOf` and `hasExtension` are also inverse of each other. They are both transitive, which means "A `isExtensionOf` B and B `isExtensionOf` C" infers "A `isExtensionOf` C".

### 3.3 Data Properties

Data properties relate an individual to an XML Schema Datatype value. For example, hasDevelopingYear is a data property of *R-NSGA-II* with datatype **integer**, which specifies the year when *R-NSGA-II* was proposed. The main data properties defined in our ontology are listed in Table 2.

**Table 2.** Data properties

| Data property | Domain | Range |
|---|---|---|
| isContinuousProblem | **MOP** | **boolean** |
| isDiscreteProblem | **MOP** | **boolean** |
| isMixedIntegerProblem | **MOP** | **boolean** |
| isManyObjectiveProblem | **MOP** | **boolean** |
| isMultimodalProblem | **MOP** | **boolean** |
| isNoisyProblem | **MOP** | **boolean** |
| hasExpensiveEvaluation | **MOP** | **boolean** |
| hasNumberOfObjectives | **MOP** | **interger** |
| hasDevelopingYear | **MetaHeuristic** | **interger** |
| hasReference | **MetaHeuristic, MOP** | **string** |
| hasMultipleRegionOfInterest | **PMOEA** | **boolean** |
| hasSpreadControl | **PMOEA** | **boolean** |
| preservesParetoDominance | **PMOEA** | **boolean** |

Three data properties present the characteristics of PMOEAs. hasMultipleRegionOfInterest describes whether the algorithm can obtain more than one Region of Interest (ROI) in one run. hasSpreadControl indicates whether it allows the DM to control the spread of the obtained ROI, preservesParetoDominance shows whether it preserves the order induced by Pareto dominance. These are important properties of PMOEAs which were also examined by Bechikh [4].

### 3.4 Creating Individuals

Here we focus on the individuals of **PMOEA** class and use DF-SMS-EMOA [51] as an example, as shown in Fig. 2.

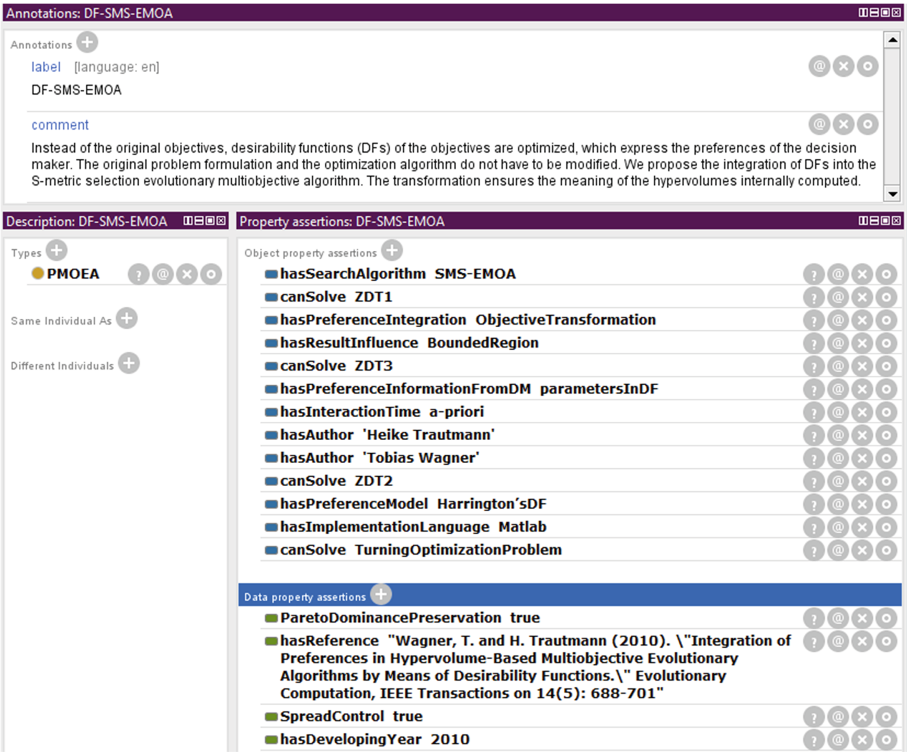


**Fig. 2.** Individual of DF-SMA-EMOA

Firstly, we create an individual of **PMOEA** named "DF-SMS-EMOA" (*label*) and give a brief introduction as *comment*.

Secondly, basic information of the paper where this algorithm was proposed (authors, reference and publish year) is used to create properties `hasAuthor`, hasReference, hasDevelopingYear.

Thirdly, we consider the interaction time, preference information, preference model, integration method, searching algorithm and result influence of this method, and create properties relating to the corresponding individuals (if there is no such individual, we should create it first).

Then, in the experiment part of the DF-SMS-EMOA paper we set the test problems using property `canSolve`. We should also create **MOP** individuals and fill their properties (isContinuousProblem, isDiscreteProblem, isMixedIntegerProblem, isManyObjectiveProblem, isMultimodalProblem, hasExpensiveEvaluation, hasNumberOfObjectives). Implementation language and library mentioned in the paper are set with `useLibrary` and `useLanguage` properties if they are given.

At last, we create hasMultipleRegionOfInterest, hasSpreadControl and preservesParetoDominance by checking these characteristics. If we are not sure about the property, we can leave it out. Not all the properties are required for an individual to belong to **PMOEA** class.

The PMOEA ontology is published on *WebProtégé* (http://webprotege.stanford.edu/#Edit:projectId=79e2fcdc-b58a-443f-8447-67ce5a388f84). Now it has 92 classes and 507 individuals (including 75 PMOEA individuals). Guests can access, visualize, comment and download the ontology, it also allows collaborative edit after sharing with specified users.

## 4 Using the PMOEA Ontology

Building the PMOEA ontology is the first step towards its beneficial uses. By checking the information given in the PMOEA Ontology, new learners can quickly get familiar with the domain, because concepts and relations are explicitly defined. Researchers from EMO and MCDM as well as experts and domain engineers can also look for the information they need and comment on the controversial information with the help of *WebProtégé*.

*DL Query* is a powerful Protégé Desktop plugin for querying and extracting knowledge. It can search for classes or individuals that satisfy certain conditions. The query language is based on the Manchester OWL syntax [24], a user-friendly syntax for OWL-DL.

Next we briefly provide a set of query examples, which shall serve as an overview of possible analysis of the ontology and show how query statements and syntax look like:

- Find all PMOEAs that use a specified kind of preference information, e.g., reference point:
  (`hasPreferenceInformationFromDM value ReferencePoint`)
- Find all PMOEAs that use a specified kind of preference integration method, e.g., objective space transformation:
  (`hasPreferenceIntegration value ObjectiveTransformation`)
- Find all PMOEAs that have been used to solve problems that belong to a certain class, e.g., many objective optimization:
  (`canSolve some (isManyObjectiveProblem value true)`)
- Find all PMOEAs that have been applied to a certain engineering problem (e.g. airfoil optimization), and use a specific implementation language (e.g. Java):
  (`canSolve some AirfoilOptimization` and `hasImplementationLanguage value Java`)
- Find all PMOEAs by authors or developing year, e.g. Juergen Branke and after 2012:
  (`hasAuthor value 'Juergen Branke'` and `hasDevelopingYear some integer[> "2012"^^integer]`)

- Find all PMOEAs that are extensions of a certain algorithm, e.g., R-NSGA-II:
(`isExtensionOf value R-NSGA-II`)
Note that *reasoning* supported by a Protégé Desktop plugin *Reasoner* (e.g. Pellet, HermiT, etc.) is required here for description logics inference. For example, *KR-NSGA-II* is extension of *R-NSGA-II*, *TKR-NSGA-II* is extension of *KR-NSGA-II*, are explicitly defined (asserted) in the ontology by the designer. "*TKR-NSGA-II* `isExtensionOf` *R-NSGA-II*" will be inferred by the *Reasoner* because `isExtensionOf` is a transitive property.
- Find all PMOEAs that use a certain algorithm in comparative experiments, e.g., r-NSGA-II:
(`hasComparison value r-NSGA-II`)
- Find all PMOEAs that use a specified type of benchmark, so that new algorithms can compare with them, e.g., ZDT1:
(`canSolve value ZDT1`)

Another important feature of OWL ontology modeling is the use of *Reasoner*. The result of DL Query can be defined as new (inferred) classes. For instance, the result of the first DL Query can be defined as a new class **ReferencePoint-based**, which is a subclass of **PMOEA**. Any PMOEA to be added in the future that has reference point as preference information will be automatically sorted into this subclass. This is systematically guaranteed by the inference engine implemented by the *Reasoner*.

The PMOEA Ontology defines the structure of this domain and provides further information by various queries. One promising advantage of the ontology is its extensibility. New individuals of **PMOEA** can be added following the procedure in Sect. 3.4, as well as new individuals of **MOP**. The value of PMOEA Ontology will increase with the progressive accumulation of information. Moreover, the classes and properties can also be extended and improved by means of *WebProtégé*, which offers a shared Ontology edit platform for collaborative editing. Further use cases of the PMOEA Ontology, including visualization tool, finding appropriate method for an application, discovering opportunities for new research, can be found in [32].

## 5 Conclusions and Outlook

This paper presents an Ontology of Preference-based Multiobjective Evolutionary Algorithms (PMOEA). Background information of PMOEA and Ontology is given, after which a detailed structure of the ontology is discussed. Then How to build/extend the PMOEA ontology is introduced, simple and practical examples for various use cases of the proposed ontology are described and explained. It demonstrates the utility of web semantic technologies within the PMOEA (EMO and MCDM) communities. It is believed that the more useful and comprehensive data PMOEA Ontology contains, the more valuable information it will provide, for both the research community in EMO and MCDM fields, as well as practitioners who want to use PMOEAs. In the future, on the one hand, new individuals of PMOEA, MOEA and MCDM methods will be added to the ontology, on the other hand, additional properties will be created such as hasBenchmark of real-world problems and hasSourceCode of MetaHeuristic individuals. Moreover, the PMOEA ontology might be reused and serve as inspiration for building more detailed ontologies for larger domains, such as the EMO domain and the MCDM domain.

**Acknowledgement.** Longmei Li acknowledges financial support from China Scholarship Council (CSC). Heike Trautmann and Michael Emmerich acknowledge support by the European Research Center for Information Systems (ERCIS).

## References

1. Protege webpage (2016). http://protege.stanford.edu/
2. Allmendinger, R., Li, X., Branke, J.: Reference point-based particle swarm optimization using a steady-state approach. In: Li, X., et al. (eds.) SEAL 2008. LNCS, vol. 5361, pp. 200–209. Springer, Heidelberg (2008). doi:10.1007/978-3-540-89694-4_21
3. Battiti, R., Passerini, A.: Brain-computer evolutionary multiobjective optimization: a genetic algorithm adapting to the decision maker. IEEE Trans. Evol. Comput. **14**(5), 671–687 (2010)
4. Bechikh, S., Kessentini, M., Said, L.B., Ghédira, K.: Chapter four-preference incorporation in evolutionary multiobjective optimization: a survey of the state-of-the-art. Adv. Comput. **98**, 141–207 (2015)
5. Bechikh, S., Said, L.B., Ghédira, K.: Searching for knee regions of the Pareto front using mobile reference points. Soft Comput. **15**(9), 1807–1823 (2011)
6. Ben Said, L., Bechikh, S., Ghédira, K.: The r-dominance: a new dominance relation for interactive evolutionary multicriteria decision making. IEEE Trans. Evol. Comput. **14**(5), 801–818 (2010)
7. Branke, J.: MCDA and multiobjective evolutionary algorithms. In: Greco, S., Ehrgott, M., Figueira, J.R. (eds.) Multiple Criteria Decision Analysis, vol. 233, pp. 977–1008. Springer, New York (2016)
8. Branke, J., Corrente, S., Greco, S., Słowiński, R., Zielniewicz, P.: Using Choquet integral as preference model in interactive evolutionary multiobjective optimization. Eur. J. Oper. Res. **250**(3), 884–901 (2016)
9. Branke, J., Greco, S., Slowinski, R., Zielniewicz, P.: Learning value functions in interactive evolutionary multiobjective optimization. IEEE Trans. Evol. Comput. **19**(1), 88–102 (2015)
10. Branke, J., Kaußler, T., Schmeck, H.: Guidance in evolutionary multi-objective optimization. Adv. Eng. Softw. **32**(6), 499–507 (2001)
11. Brockhoff, D., Bader, J., Thiele, L., Zitzler, E.: Directed multiobjective optimization based on the weighted hypervolume indicator. J. Multi-Criteria Dec. Anal. **20**(5–6), 291–317 (2013)
12. Coello, C.A.C.: Handling preferences in evolutionary multiobjective optimization: a survey. In: Proceedings of the 2000 Congress on Evolutionary Computation, vol. 1, pp. 30–37. IEEE (2000)
13. Coello, C.C., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-objective Problems. Springer, New York (2007)
14. Deb, K.: Multi-objective Optimization Using Evolutionary Algorithms, vol. 16. Wiley, New York (2001)

15. Deb, K., Kumar, A.: Light beam search based multi-objective optimization using evolutionary algorithms. In: IEEE Congress on Evolutionary Computation, 2007. CEC 2007, pp. 2125–2132. IEEE (2007)
16. Deb, K., Kumar, A.: Interactive evolutionary multi-objective optimization and decision-making using reference direction method. In: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, pp. 781–788. ACM (2007)
17. Deb, K., Sinha, A., Korhonen, P.J., Wallenius, J.: An interactive evolutionary multiobjective optimization method based on progressively approximated value functions. IEEE Trans. Evol. Comput. **14**(5), 723–739 (2010)
18. Deb, K., Sundar, J.: Reference point based multi-objective optimization using evolutionary algorithms. In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, pp. 635–642. ACM (2006)
19. Emmerich, M., Deutz, A., Kruisselbrink, J., Shukla, P.K.: Cone-based hypervolume indicators: construction, properties, and efficient computation. In: Purshouse, R.C., Fleming, P.J., Fonseca, C.M., Greco, S., Shaw, J. (eds.) EMO 2013. LNCS, vol. 7811, pp. 111–127. Springer, Heidelberg (2013). doi:10.1007/978-3-642-37140-0_12
20. Fernandez, E., Lopez, E., Lopez, F., Coello, C.A.C.: Increasing selective pressure towards the best compromise in evolutionary multiobjective optimization: the extended NOSGA method. Inf. Sci. **181**(1), 44–56 (2011)
21. Friedrich, T., Kroeger, T., Neumann, F.: Weighted preferences in evolutionary multi-objective optimization. In: Wang, D., Reynolds, M. (eds.) AI 2011. LNCS (LNAI), vol. 7106, pp. 291–300. Springer, Heidelberg (2011). doi:10.1007/978-3-642-25832-9_30
22. Greco, S., Matarazzo, B., Slowinski, R.: Interactive evolutionary multiobjective optimization using dominance-based rough set approach. In: IEEE Congress on Evolutionary Computation (CEC), pp. 1–8. IEEE (2010)
23. Gruber, T.R.: A translation approach to portable ontology specifications. Knowl. Acquis. **5**(2), 199–220 (1993)
24. Horridge, M., Drummond, N., Goodwin, J., Rector, A.L., Stevens, R., Wang, H.: The Manchester OWL syntax. In: OWLed, vol. 216 (2006)
25. Horridge, M., Jupp, S., Moulton, G., Rector, A., Stevens, R., Wroe, C.: A practical guide to building OWL ontologies using Protégé 4 and CO-ODE tools edition 1.2. The University of Manchester (2009)
26. Jaimes, A.L., Montano, A.A., Coello, C.A.C.: Preference incorporation to solve many-objective airfoil design problems. In: 2011 IEEE Congress on Evolutionary Computation (CEC), pp. 1605–1612. IEEE (2011)
27. Jaszkiewicz, A., Słowiński, R.: The light beam search approach-an overview of methodology applications. Eur. J. Oper. Res. **113**(2), 300–314 (1999)
28. Jin, Y., Sendhoff, B.: Incorporation of fuzzy preferences into evolutionary multiobjective optimization. In: GECCO, p. 683 (2002)
29. Kaur, G., Chaudhary, D.: Evolutionary computation ontology: E-learning system. In: 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions), pp. 1–6. IEEE (2015)
30. Kim, J.H., Han, J.H., Kim, Y.H., Choi, S.H., Kim, E.S.: Preference-based solution selection algorithm for evolutionary multiobjective optimization. IEEE Trans. Evol. Comput. **16**(1), 20–34 (2012)
31. Korhonen, P.J., Laakso, J.: A visual interactive method for solving the multiple criteria problem. Eur. J. Oper. Res. **24**(2), 277–287 (1986)

32. Li, L., Yevseyeva, I., Basto-Fernandes, V., Trautmann, H., Jing, N., Emmerich, M.: An ontology of preference-based multiobjective evolutionary algorithms. arXiv:1609.08082, submitted to IEEE Transactions on Evolutionary Computation
33. McGuinness, D.L., Van Harmelen, F., et al.: Owl web ontology language overview. W3C Recomm. **10**(10), 77–213 (2004)
34. Miettinen, K.: Nonlinear Multiobjective Optimization, vol. 12. Springer, New York (2012)
35. Mohammadi, A., Omidvar, M.N., Li, X.: A new performance metric for user-preference based multi-objective evolutionary algorithms. In: 2013 IEEE Congress on Evolutionary Computation, pp. 2825–2832. IEEE (2013)
36. Molina, J., Santana, L.V., Hernández-Díaz, A.G., Coello, C.A.C., Caballero, R.: g-dominance: reference point based dominance for multiobjective metaheuristics. Eur. J. Oper. Res. **197**(2), 685–692 (2009)
37. Mostaghim, S., Trautmann, H., Mersmann, O.: Preference-based multi-objective particle swarm optimization using desirabilities. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN 2010. LNCS, vol. 6239, pp. 101–110. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15871-1_11
38. Noy, N.F., McGuinness, D.L., et al.: Ontology development 101: a guide to creating your first ontology (2001)
39. Ojalehto, V., Podkopaev, D., Miettinen, K.: Towards automatic testing of reference point based interactive methods. In: Handl, J., Hart, E., Lewis, P.R., López-Ibáñez, M., Ochoa, G., Paechter, B. (eds.) PPSN 2016. LNCS, vol. 9921, pp. 483–492. Springer, Cham (2016). doi:10.1007/978-3-319-45823-6_45
40. Pedro, L.R., Takahashi, R.H.C.: Decision-maker preference modeling in interactive multiobjective optimization. In: Purshouse, R.C., Fleming, P.J., Fonseca, C.M., Greco, S., Shaw, J. (eds.) EMO 2013. LNCS, vol. 7811, pp. 811–824. Springer, Heidelberg (2013). doi:10.1007/978-3-642-37140-0_60
41. Rachmawati, L., Srinivasan, D.: Preference incorporation in multi-objective evolutionary algorithms: a survey. In: IEEE Congress on Evolutionary Computation, 2006. CEC 2006, pp. 962–968. IEEE (2006)
42. Rachmawati, L., Srinivasan, D.: Incorporating the notion of relative importance of objectives in evolutionary multiobjective optimization. IEEE Trans. Evol. Comput. **14**(4), 530–546 (2010)
43. Shukla, P.K., Emmerich, M., Deutz, A.: A theoretical analysis of curvature based preference models. In: Purshouse, R.C., Fleming, P.J., Fonseca, C.M., Greco, S., Shaw, J. (eds.) EMO 2013. LNCS, vol. 7811, pp. 367–382. Springer, Heidelberg (2013). doi:10.1007/978-3-642-37140-0_29
44. Shukla, P.K., Hirsch, C., Schmeck, H.: A framework for incorporating trade-off information using multi-objective evolutionary algorithms. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN 2010. LNCS, vol. 6239, pp. 131–140. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15871-1_14
45. Sinha, A., Korhonen, P., Wallenius, J., Deb, K.: An Interactive Evolutionary Multi-objective Optimization Method Based on Polyhedral Cones. In: Blum, C., Battiti, R. (eds.) LION 2010. LNCS, vol. 6073, pp. 318–332. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13800-3_33
46. Staab, S., Studer, R.: Handbook on Ontologies. Springer, Heidelberg (2013)
47. Thiele, L., Miettinen, K., Korhonen, P.J., Molina, J.: A preference-based evolutionary algorithm for multi-objective optimization. Evol. Comput. **17**(3), 411–436 (2009)
48. Trautmann, H., Mehnen, J.: Preference-based Pareto optimization in certain and noisy environments. Eng. Optim. **41**(1), 23–38 (2009)

49. Trautmann, H., Wagner, T., Brockhoff, D.: R2-EMOA: focused multiobjective search using R2-indicator-based selection. In: Nicosia, G., Pardalos, P. (eds.) LION 2013. LNCS, vol. 7997, pp. 70–74. Springer, Heidelberg (2013). doi:10.1007/978-3-642-44973-4_8
50. Vira, C., Haimes, Y.Y.: Multiobjective decision making: theory and methodology. No. 8, North-Holland (1983)
51. Wagner, T., Trautmann, H.: Integration of preferences in hypervolume-based multiobjective evolutionary algorithms by means of desirability functions. IEEE Trans. Evol. Comput. **14**(5), 688–701 (2010)
52. Wierzbicki, A.P.: The use of reference objectives in multiobjective optimization. In: Fandel, G., Gal, T. (eds.) Multiple Criteria Decision Making Theory and Application. LNE, vol. 177, pp. 468–486. Springer, Heidelberg (1980)
53. Yevseyeva, I., Guerreiro, A.P., Emmerich, M.T.M., Fonseca, C.M.: A portfolio optimization approach to selection in multiobjective evolutionary algorithms. In: Bartz-Beielstein, T., Branke, J., Filipič, B., Smith, J. (eds.) PPSN 2014. LNCS, vol. 8672, pp. 672–681. Springer, Cham (2014). doi:10.1007/978-3-319-10762-2_66