# An Algorithm Based on Grammatical Evolution for Discovering SHACL Constraints

Rémi Felin, Pierre Monnin, Catherine Faron, Andrea G. B. Tettamanzi

# An Algorithm Based on Grammatical Evolution for Discovering SHACL Constraints

Rémi Felin[1 [0000−0003−2532−7555]], Pierre Monnin[1 [0000−0002−2017−8426]], Catherine Faron[1 [0000−0001−5959−5561]], and Andrea G. B. Tettamanzi[1 [0000−0002−8877−4654]]

Université Côte d'Azur, Inria, CNRS, I3S, Sophia-Antipolis, France
{name.surname}@inria.fr

**Abstract.** The continuous evolution of heterogeneous RDF data has led to an increase of inconsistencies on the Web of data (i.e. missing data and errors) that we assume to be inherent to RDF data graphs. To improve their quality, the W3C recommendation SHACL allows to express various constraints that RDF data must conform to and detect nodes violating them. However, acquiring representative and meaningful SHACL constraints from complex and very large RDF data graphs is very challenging and tedious. Consequently, several recent works focus on the automatic generation of these constraints. We propose an approach based on grammatical evolution (GE) for extracting representative SHACL constraints by mining an RDF data graph. This approach uses a probabilistic SHACL validation framework to consider the inherent errors in RDF data. The results highlight the relevance of this approach in discovering SHACL shapes inspired by association rule patterns from a real-world RDF data graph.

**Keywords:** Grammatical Evolution · Shape Mining · Web of Data

## 1 Introduction

Over the years, the Web has witnessed an increasing publication of heterogeneous data graphs forming the "Web of Data", aimed at being consumed by humans and artificial agents. These graphs are represented using the RDF (Resource Description Format) standard and queried using the SPARQL standard. RDF relies on a triple model: an RDF triple $< \mathbf{s}, \mathbf{p}, \mathbf{o} >$ expresses a relation **p** between a subject **s** and an object **o**. An **RDF data graph** is a set of interconnected RDF triples which terms are IRIs, literals and blank nodes (anonymous resources): **s** is a resource (a IRI or a blank node), **p** is a IRI, and **o** is any RDF term. RDF data can be serialized with several syntaxes among which Turtle is the simplest and most readable one. It uses qualified names with *prefixes* associated to namespaces to simplify the notation of IRIs, e.g. `foaf:name` is parsed into `<http://xmlns.com/foaf/0.1/name>`. The increase of available and heterogeneous RDF data is the result of global initiatives for producing open data,

e.g. Linked Open Data[1]. It is well known that this evolution has revealed issues of errors and incompleteness in real-world RDF data graphs and we consider it essential to recognize the principle that these inconsistencies are inherent to RDF data graphs.

To improve the quality of RDF data, the W3C recommended SHACL [11] as a standard to represent constraints on RDF data graphs. **SHACL shapes** are instances of `sh:NodeShape`[2] that allow targeting a specific set of nodes in an RDF data graph and assessing them against a set of SHACL constraints, i.e. searching possible nodes in the RDF data graph that do not conform to the shape. Overall, the evaluation process considers a shapes graph (i.e., a set of SHACL shapes) and evaluate them against an RDF graph. SHACL thus addresses the requirements for RDF data quality control, contributing to reducing the inherent inconsistencies in RDF data graphs.

We are interested in the extraction of SHACL shapes that express domain constraints from RDF data graphs. Given that SHACL is a relatively new language, real-world data graphs have a minimal set of associated SHACL shapes. This issue, extensively discussed by Rabbati et al. [28], has led the community to explore various research directions. We can distinguish between approaches aiming to extract SHACL shapes from an ontology (i.e., the "schema" associated with RDF graphs) [4, 27], approaches aiming to extract SHACL shapes by mining regularities from RDF facts [3, 8, 25], and approaches combining both [1]. In general, one of the most significant challenges in extracting SHACL shapes is scaling these methods to handle substantial data graphs. Only Fernandez-Álvarez et al. [8] address this problem by optimizing machine memory consumption. Ontology-based approaches are limited to the coverage degree of the ontologies regarding the RDF data graph, which impacts the type of constraints that can be extracted. Some approaches for extracting SHACL shapes from RDF data take errors and incompleteness into account, using a threshold of tolerance [8] or quality measures [25], but they have limitations regarding the types of SHACL shapes they can extract. For instance, the method proposed in [25] is limited to the extraction of a specific type of rules that can be later translated into SHACL shapes. Boneva et al. [3] propose a semi-automatic method to discover SHACL shapes (via a user interface), while Pandit et al. [27] suggest a manual construction of SHACL shapes using Ontology Design Patterns.

We aim to address the limitations regarding the kind of SHACL shapes that can be extracted from an RDF data graph, which motivated the following research question: *How to automatically discover SHACL shapes from RDF data?* We believe that a generative approach for the automatic construction of SHACL candidate shapes using RDF data is one of those that can achieve this ideal. To this end, we propose a mining method based on **Grammatical Evolution (GE)**, which is a particular type of genetic programming. In this approach, it is feasible to automatically generate variable-length expressions in any language [26] using well-defined grammars, composed of production rules. Gram-

[1] https://lod-cloud.net/

[2] Or `<http://www.w3.org/ns/shacl#NodeShape>`

matical Evolution has been the subject of ongoing work in recent years [30]. Recent work discusses some limitations concerning the grammar design [6], their complexity and a "poor" initialisation of individuals [9]. These limitations have been the subject of contributions intended to propose general guidelines for grammar design [23], automated techniques for finding optimal parameters [2], or new techniques for improving the population initialisation [24]. However, the two major and recurring problems with this method are *redundancy* between individuals and low *locality* [13, 16], i.e. "how well neighbouring genotypes correspond to neighbouring phenotypes" [29]. Lourenço et al. have proposed an extension to this approach called Structured Grammatical Evolution (SGE) [12, 14, 15], which enables one-to-one mapping between genes and non-terminals belonging to the grammar, with effective responses to these two problems. Other approaches aim to introduce a probabilistic approach which is based on a probabilistic selection of production rules during the genotype building phase [10, 17]. Mégane et al. extend their approach with a probabilistic SGE [18, 19] to improve their results and resolve these two issues. In these works, some well-known benchmarks are used to assess the effectiveness of the proposed models, like *Santa Fe Trail* and *Boston Housing*, but their application to tasks related to RDF data mining has not been demonstrated yet. Only Nguyen and Tettamanzi have proposed an adaptation of GE for extracting OWL disjointness axioms [21] and complex disjointness axioms [22], with some promising results.

In this paper, we propose an algorithm based on Grammatical Evolution for generating candidate SHACL shapes using a BNF grammar and RDF data as input. The algorithm exploits a probabilistic framework for SHACL validation, which is required given the heterogeneity and incompleteness inherent in open RDF data. The approach is validated by applying it to the extraction of SHACL shapes from a set of RDF data relating to the scientific domain. The remainder of the paper is organized as follows: in Section 2, we present the design of BNF grammars describing candidate SHACL shapes; in Section 3, we present the probabilistic SHACL validation (3.1) and how we use it to define an acceptance measure and a fitness function (3.2) in order to evaluate candidate shapes; in Section 4 we present a recombination operator responding to the redundancy problem and variation operators used to ensure a broad exploration of the solution space; in Section 5, we present the experiments carried out on a real-world RDF dataset. The results of our experiments are presented in Section 5.2, and we conclude with a discussion of future research in Section 6.

## 2 BNF Grammars of SHACL Shapes

In order to produce and exploit well-formed SHACL shapes as individuals in an evolutionary process, we defined a BNF grammar compliant with the SHACL W3C recommendation [11]. Fig. 1 presents a subset of this BNF grammar to produce shapes targeting nodes of a specified class (`sh:targetClass`) and constraining them to be linked through the predicate `rdf:type` to another specified class. It should be noted that the two classes may be the same. The grammar

provides the phenotypic and genotypic characterisation for each individual using a set of *static rules* and *dynamic rules*. The dynamic rules system, proposed by Nguyen and Tettamanzi [21, 22], allows the mapping between rules and RDF data. They wrote BNF grammars to build candidate OWL axioms and exploit them with an evolutionary algorithm based on GE. The static rules are the *immutable* components of the phenotypic character whereas the dynamic rules are the *problem instance-dependent* components of the phenotypic character, where each rule has one or many possible values depending on the considered RDF dataset, and each value is identified by a genotype. However, their dynamic rules were hard-coded in their system. In contrast, we extended the dynamic rules design by directly enabling the user to write embedded SPARQL queries as value of one or more production rules in the grammar in order to perform the mapping with the desired granularity.

```
1 <Shape>     := "a " <NodeShape>
2 <NodeShape> := "sh:NodeShape; " <ShapeBody>
3 <ShapeBody> := "sh:targetClass " <Class> "; " <ShapeProp>
4 <ShapeProp> := "sh:property [ " + <PropBody> " ] ."
5 <PropBody>  := "sh:path rdf:type; sh:hasValue " <Class> " ;"
6 <Class>     := "SPARQL ?x rdf:type ?Class"
```

Fig. 1: An extract of the BNF grammar for SHACL shapes

To illustrate, in Fig. 1, the `<Class>` non-terminal (used in lines 3 and 5) is defined by a dynamic rule to extract all possible classes from the RDF dataset using a SPARQL query. The keyword `SPARQL` in line 6 is used to specify the query graph pattern to be matched on RDF data; the result of this query is the set of nodes in the RDF data graph $\mathcal{C}$ bound to variable `?Class` in the query graph pattern: $\mathcal{C} = \{c_i, i \in [1, n]\}$. The final step is to replace the initial value of the rule `<Class>`, i.e. the SPARQL query `"SPARQL ?x a ?Class"`, by the SPARQL results $\mathcal{C}$, i.e. $c_1 \mid c_2 \mid \cdots \mid c_n$. The whole process is presented in Fig. 2.

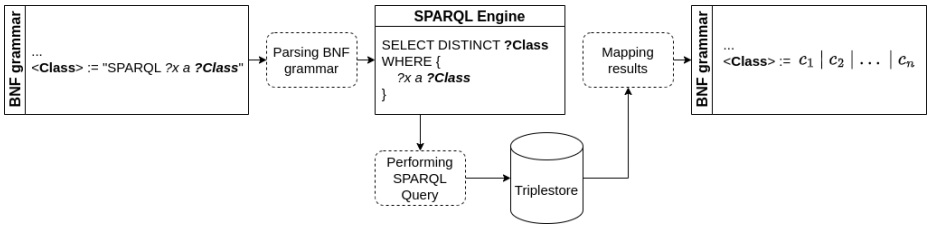



Fig. 2: Dynamic rules process based on the BNF grammar presented in Fig. 1

Using the BNF grammar presented in Fig. 1, the genotype of an individual is a pair of codons $[i, j]$, which are decoded into two classes from the dataset using

a classic genotype-phenotype mapping, and produce the following phenotype structure:

"a sh:NodeShape ; sh:targetClass $c_i$ ; sh:property [ sh:path rdf:type ; sh:hasValue $c_j$" ; ] ."

It is noteworthy that the proposed grammar can be extended to produce a wider array of SHACL shapes using a variable-length template, e.g. replacing the rule <ShapeProp> from Fig. 1 by <ShapeProp> := <Prop> <ShapeProp> | <Prop> where <Prop> := "sh:property [ " + <PropBody> " ] .". Such an extended grammar would produce SHACL shapes specifying one or more constraints (depending on the chosen length).

## 3 Probabilistic SHACL Validation as a Fitness Function

### 3.1 Preliminaries

We rely on the probabilistic framework for SHACL validation proposed by Felin et al. [7] to assess SHACL shapes considering inherent inconsistencies from RDF data. It extends the standard evaluation of RDF data against SHACL shapes by considering a *physiological* error proportion $p$ in the RDF data, i.e. a possible acceptable violation rate. *Physiological* errors in RDF data graph are inherent errors whose origins can be diverse, e.g. from collaborative building of large RDF data graphs (e.g. Wikidata) or automatically constructed RDF data graphs (e.g. DBpedia). In the rest of this section, we summarize the principles of this model.

Considering an RDF data graph $\upsilon$, the *support* of a shape $s$, $\upsilon_s$, is the set of RDF triples in $\upsilon$ targeted by $s$ (and therefore tested during the validation). The *reference cardinality* (refCard) of $s$ is the cardinality of its support: $|\upsilon_s|$. The *confirmations* and *violations* of $s$, respectively $\upsilon_s^+$ and $\upsilon_s^-$, are the sets of triples that, respectively, are consistent with $s$ and violate $s$: $\upsilon_s^+ \cap \upsilon_s^- = \emptyset$, and $\upsilon_s = \upsilon_s^+ \cup \upsilon_s^-$. The probabilistic model for SHACL validation relies on a binomial distribution $X \sim B(|\upsilon_s|, p)$ where $p$ is the physiological error proportion. The *likelihood* to observe a number of violations $|\upsilon_s^-|$ in the support of a shape $s$ considering $X \sim B(|\upsilon_s|, p)$ is defined as follows:

$$L_{|\upsilon_s^-|} = P(X = |\upsilon_s^-|) = \binom{|\upsilon_s|}{|\upsilon_s^-|} \cdot p^{|\upsilon_s^-|} \cdot (1-p)^{|\upsilon_s^+|} \tag{1}$$

The *acceptance* of a shape $s$ depends on the proportion of violations for $s$, i.e $\hat{p_s} = \frac{|\upsilon_s^-|}{|\upsilon_s|}$: $s$ is consistent with $\upsilon$ if $\hat{p_s}$ is smaller than the theoretical violation proportion $p$:

$$\hat{p_s} \leq p \implies KG \models s \tag{2}$$

In the case where $\hat{p_s} > p$, a Chi-Square Goodness of Fit test is performed using the *test statistic* $X_s^2$ defined as follows:

$$X_s^2 = \sum_{i=1}^{k} \frac{(n_i - T_i)^2}{T_i} \sim \chi^2_{k-1;\alpha} \tag{3}$$

where $k$ is the total number of groups, i.e. $k = 2$, $n_i$ is the observed number of individuals and $T_i$ is the theoretical number of individuals. The acceptance of the null hypothesis $H_0$ implies the acceptance of $s$:

$$X_s^2 \leq \chi^2_{k-1;\alpha} \implies KG \models s \tag{4}$$

### 3.2 Acceptability and Fitness Score

In order to iteratively produce a final population of SHACL shapes expressing some domain constraints that are implicit in an RDF dataset, we propose a fitness function based on an acceptability measure of a shape combined with the probabilistic framework presented.

**The acceptability** of a SHACL shape $s$, $A(s)$, regarding an RDF dataset, depends on the error rate when validating this RDF dataset against $s$, which depends on the theoretical error proportion $p$. The acceptability measure $A(s) \in [0, 1]$ of a SHACL shape is defined by:

$$A(s) = \begin{cases} 1 & \text{if } \hat{p_s} \leq p \text{ or } X_s^2 \leq \chi^2_{k-1;\alpha} \quad \text{(Equation (2) and 4)}, \\ \frac{L_{|\upsilon_s^-|}}{P(X=|\upsilon_s| \times p)} & \text{otherwise} \quad \text{(Equation (1))}. \end{cases} \tag{5}$$

In the computation of $A(s)$, for the hypothesis testing, we consider a margin error $\alpha = 0.05$. Therefore, the *critical value* is $\chi^2_{k-1;0.05} = 3.84$. When $A(s) = 1$, $s$ is acceptable and is (probably) selected as one of the most fit individuals, in the sense that it captures some domain knowledge extracted from the RDF data.

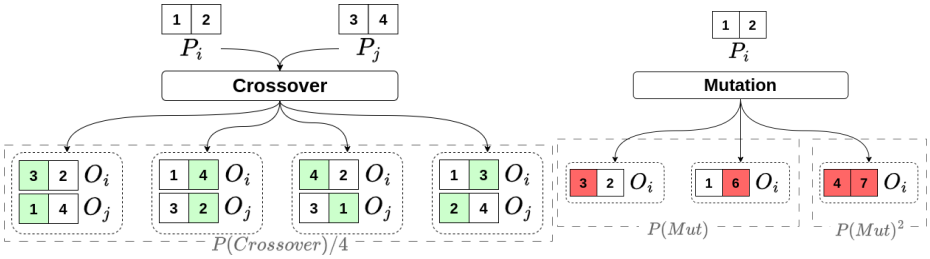
In the case where the null hypothesis is rejected, $A(s) \neq 1$ and so $s$ is not acceptable but it may be considered in the grammatical evolution algorithm for crossover or mutation operations. For this purpose, $A(s)$ is equal to the likelihood of $s$ normalized by the maximal value of the probability mass function for a binomial distribution $X \sim B(|\upsilon_s|, p)$. It ensures a better distribution of $A(s)$ values between 0 and 1 in contrast to the lonely likelihood value $L_{|\upsilon_s^-|}$ and therefore avoids excessively penalising individuals who are "close" to being acceptable but for whom the likelihood is very low.

**The fitness function** of a SHACL shape $s$, $F(s)$, regarding an RDF dataset, combines its acceptability $A(s)$ and the cardinality of its confirmations $|\upsilon_s^+|$:

$$\forall s \in P, \quad F(s) = |\upsilon_s^+| \times A(s) \tag{6}$$

## 4 Variation and Recombination Operators

In this paper, we adapt the main components of the GE variation operators to discover SHACL shapes over RDF data, considering the problem of *redundancy* and *low locality* presented as the main issues of GE by the community.




(a) *Swap Crossover* of $P_i$ and $P_j$ (b) *Int Flip Mutation* of parent $P_i$

Fig. 3: Representation of GE operators and their probabilities of occurrence

The *redundancy* is observed when many genotypes map the same phenotype expression [13]. Based on this fact, we adapt the recombination phase to filter every offspring using **a phenotypic comparison**: Algorithm 1 presents the recombination of selected individuals $\mathcal{S}$ among the whole population $\mathcal{P}$ ($\mathcal{E}$ represents the elite individuals). Line 11 describes the conditions for integrating an offspring $i$ into the replacement population $\mathcal{R}$: $i$ is integrated into $\mathcal{R}$ if the phenotypic expression of $i$ is not already observed among the elitist individuals $\mathcal{E}$ and the replacement individuals $\mathcal{R}$. As a consequence, we avoid the reflection of the redundancy in the final population: $\forall i \in \mathcal{P}, \nexists j \in \mathcal{P} : i \equiv j$.

**Algorithm 1:** Recombination of a population $\mathcal{P}$

**Data:** elite individuals $\mathcal{E}$ **and** selected individuals $\mathcal{S}$
**Result:** replacement population $\mathcal{R}$

1 $\mathcal{R} \leftarrow \{\}$;
2 **while** $|\mathcal{R}| \neq |\mathcal{P}| - |\mathcal{E}|$ **do**
3  $\mathcal{C} \leftarrow \{\}$;
4  $p_1 \leftarrow \mathcal{S}[random() * |\mathcal{S}|]$;
5  $p_2 \leftarrow \mathcal{S}[random() * |\mathcal{S}|]$; ▷ $random() * |\mathcal{S}|$ as integer;
6  **if** $p_1 \neq p_2$ **then**
7  $\mathcal{C} \leftarrow p_1 \cup p_2$;
8  $\mathcal{C} \leftarrow crossover(\mathcal{C})$; ▷ Figure 3a;
9  $\mathcal{C} \leftarrow mutation(\mathcal{C})$; ▷ Figure 3b;
10  **for** $i \in \mathcal{C}$ **do**
11  **if** $i \notin \mathcal{E} \cup \mathcal{R}$ **and** $|\mathcal{R}| \neq |\mathcal{P}| - |\mathcal{E}|$ ▷ **Phenotypic** comparison
12  **then**
13  $\mathcal{R} \leftarrow \mathcal{R} \cup i$;
14  **end**
15 **end**
16 **return** $\mathcal{R}$

In this context, the *locality* issue is dependent on the neighbourhood between the selected rules from parent to offspring. Consequently, some results from the variation operators presented in Fig. 3 can lead to a low locality, i.e. a very different offspring, but also to a fairly strong locality. Considering the grammar presented in Fig. 1, a modification of the first codon (impacting the value of the `sh:targetClass` $c_i$) significantly changes the meaning of the phenotypic trait, as SHACL validation is performed on the nodes instantiating $c_i$: a new production rule $c_i'$ replacing $c_i$ leads to a locality as low as the proximity (e.g. common instances) between them. The modification of the last codon has a lower impact on SHACL validation since the targeted nodes are the same, resulting in a fairly strong locality (even if the meaning of the phenotype is different).

## 5 Experiences

### 5.1 Shape Mining over the Covid-on-the-Web Dataset

**Setup** *Covid-on-the-Web*[3] [20] is an RDF dataset produced from the *COVID-19 Open Research Dataset (CORD-19)*. It describes articles and named entities identified in these articles and linked to *Wikidata* entities. We consider a subset containing 18.79% of the articles and 0.01% of the named entities. This dataset contains 226,647 RDF triples, 20,912 distinct articles and 6,331 distinct named entities.

We consider the mining of SHACL shapes representing **association rules** between *Wikidata entities*[4], *i.e.*, rules of the form $\mathcal{X} \rightarrow \mathcal{Y}$. We use the BNF grammar presented in Fig. 1 to generate candidate shapes. Each candidate involves a first Wikidata entity, *i.e.*, the *antecedent* called $\mathcal{X}$, and targets nodes $n$ (*i.e.*, scientific articles) typed by this entity using the `sh:targetClass` property. The proposed constraint verifies if these nodes are also typed by a second Wikidata entity, *i.e.*, the *consequent* $\mathcal{Y}$, using the `sh:hasValue` constraint applied on the `rdf:type` property.

Concerning the aim of these experiments, we considered the diversity and the discovery of a large set of acceptable shapes as the most important aspects. While it is obvious that a resource-intensive parameter setting (high population size, high effort, etc.) would provide the best results at the cost of a long computation time, we have focused on a reasonable set of parameters in order to minimize the computation time invested.

We used an implementation of the presented algorithm combined with the probabilistic SHACL validation engine implemented in the *Corese* semantic Web factory [5]. We considered a theoretical error proportion $p = 0.5$ (*i.e.*, *physiological* error), according to the experiment's results on this dataset reported by Felin et al. [7]: this value $p$ maximises the mean value of the likelihood measure $L$ (see Equation (1)).

[3] https://github.com/Wimmics/CovidOnTheWeb
[4] https://www.wikidata.org/wiki/Wikidata:Main_Page

The experiments have been performed on a server equipped with an Intel(R) Xeon(R) CPU E5-2637 v2 processor at 3.50GHz clock speed, with 172 GB of RAM, 1 TB of disk space running under the Ubuntu 20.06.4 LTS 64-bit operating system.

**Recall** In order to assess the ability of our approach to find acceptable candidates in the solution space of our problem, we define the recall $R$ of our algorithm. The recall provides the rate of distinct solutions found by our algorithm among the total number of solutions $\mathcal{A}$. Let $\Omega$ be the set of all possible (acceptable and non-acceptable) pairs ($\mathcal{X}$, $\mathcal{Y}$) of distinct named entities extracted from the *Covid-on-the-web dataset*. We may compute its cardinality using the number of distinct entities in the dataset: $|\mathbf{\Omega}| = 6,331 \times 6,330 = \mathbf{40,075,230}$.

To estimate the number of acceptable shapes in $\Omega$, we sample a random subset $\Omega' \subseteq \Omega$, representative of the solution space. To ensure the representativity of $\Omega'$, we determine its minimal size using the *Cochran* formula: $|\mathbf{\Omega'}| = \frac{z^2 \times p \times (1-p)}{m^2} = \frac{2.58^2 * 0.5^2}{0.02^2} \approx \mathbf{4,161}$, where $z$ is the standard normal z-table with a confidence level of 99% (so $z \approx 2.58$), $m$ is the tolerated margin of error (2%); $p'$ the probability that the candidate shape is acceptable (unknown in this context so $p' = 0.5$).

Consequently, we generated 4,161 distinct and random shapes which have been evaluated over the *Covid-on-the-Web* subgraph with probabilistic SHACL validation: the results show that only 2 shapes in $\Omega'$ are accepted, *i.e.*, 0.05% of the total. This allows us to estimate the total number of acceptable shapes $\mathcal{A}$: $|\mathcal{A}| = |\Omega| \times 0.0005 = \mathbf{20,037.6}$.

Finally, the recall of our algorithm $R(x)$, *i.e.*, how well the $x$ acceptable shapes cover the solution space, is defined by:

$$\mathbf{R(x)} = \frac{x}{|\mathcal{A}|} \times 100 \tag{7}$$

### 5.2 Results

**$|\mathcal{P}|/E$ choice** We assessed our approach with manually defined small population sizes ($|\mathcal{P}|$) and quite low *effort* values $E$ and we analysed the effects of the ratio $|\mathcal{P}|/E$. This corresponds to verifying if our algorithm can find credible and surprising candidate shapes using a minimum investment of CPU time. Consequently, we performed 10 executions of our algorithm using the different parameter settings presented in Table 1 (90 in total) and analyzed the final whole population $\mathcal{P}$ and the final elitist subset $\mathcal{E}$ ($\mathcal{E} \subseteq \mathcal{P}$). Each configuration has been assessed regarding the following metrics: the average fitness value $\overline{\mathbf{F}}$; the average rate of accepted shapes $\overline{\%\mathbf{A}}$; the average likelihood value $\overline{\mathbf{L}}$; the average CPU time (in ms) for evaluating an individual $\overline{\mathbf{T}}$ and the average recall $\overline{\mathbf{R}}$.

According to the results presented in Table 2, a gradual increase of the effort $E$ tends to enhance the global quality of candidate shapes into $\mathcal{P}$: all the metrics related to the individual quality ($\overline{F}$, $\overline{\%A}$, $\overline{L}$ and $\overline{R}$) are significantly better

Table 1: Used parameters to analyse the impact of $|\mathcal{P}|$/E choice.

| **Parameters** | **Value(s)** |
|---|---|
| **GE** | |
| $\|P\|$ | {100; 200; 500} |
| Effort (E) | $\{5,000; 10,000; 20,000\}$ |
| % Selection ($\mathcal{E}$) | 20% |
| % Selection ($\mathcal{R}$) | 40% |
| Selection type | Tournament |
| % Tournament | 25% |
| Crossover type - P | Swap (Fig. 3a) - 75% |
| Mutation type - P | Int Flip (Fig. 3b) - 5% |
| **Probabilistic SHACL** | |
| Confidence level $\alpha$ | 5% |
| Theoretical inconsistencies proportion $p$ | 50% |

Table 2: Results obtained using the parameters presented in Table 1. Best result for each metric is in bold and second best underlined. Highlighted columns are the best.

| | | $\|P\| = 100$ | | | $\|P\| = 200$ | | | $\|P\| = 500$ | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | E = 5,000 | E = 10,000 | E = 20,000 | E = 5,000 | E = 10,000 | E = 20,000 | E = 5,000 | E = 10,000 | E = 20,000 |
| From $\mathcal{P}$ | $\overline{F}$ | $0.79 \pm 1.4$ | $0.64 \pm 1.05$ | $\mathbf{1.52 \pm 1.7}$ | $0.53 \pm 0.81$ | $1.24 \pm 1.83$ | $\underline{1.51 \pm 1.14}$ | $0.9 \pm 0.97$ | $1.07 \pm 1$ | $1.3 \pm 0.87$ |
| | $\overline{\%A}$ | $1.9 \pm 2.77$ | $4 \pm 2.71$ | $\mathbf{10.1 \pm 4.18}$ | $2 \pm 0.75$ | $3.3 \pm 2.15$ | $\underline{8.05 \pm 3.11}$ | $1.56 \pm 0.76$ | $2.36 \pm 1.05$ | $4.98 \pm 2.33$ |
| | $\overline{L}$ | $2.55 \pm 1.77$ | $4.86 \pm 1.66$ | $\mathbf{6.74 \pm 2.07}$ | $2.42 \pm 0.97$ | $4.41 \pm 1.85$ | $\underline{6.18 \pm 1.13}$ | $1.25 \pm 0.29$ | $2.07 \pm 0.59$ | $4.47 \pm 1.29$ |
| | $\overline{T}$ | $18 \pm 2.73$ | $\underline{17.98 \pm 2.96}$ | $21.94 \pm 3.73$ | $\mathbf{16.84 \pm 2.4}$ | $19.22 \pm 3.03$ | $21.14 \pm 3.65$ | $19.21 \pm 2.28$ | $18.39 \pm 3.71$ | $19.31 \pm 3.71$ |
| | $\overline{R}$ | $0.01 \pm 0.01$ | $0.02 \pm 0.01$ | $0.05 \pm 0.02$ | $0.02 \pm 0.01$ | $0.03 \pm 0.02$ | $\underline{0.08 \pm 0.03}$ | $0.04 \pm 0.02$ | $0.06 \pm 0.03$ | $\mathbf{0.13 \pm 0.06}$ |
| From $\mathcal{E}$ | $\overline{F}$ | $3.91 \pm 7$ | $2.77 \pm 5.28$ | $\mathbf{7.41 \pm 8.48}$ | $2.65 \pm 4.06$ | $6.11 \pm 9.15$ | $\mathbf{7.41 \pm 5.74}$ | $4.41 \pm 4.9$ | $5.29 \pm 4.99$ | $\underline{6.36 \pm 4.39}$ |
| | $\overline{\%A}$ | $9.5 \pm 13.83$ | $18.5 \pm 12.92$ | $\mathbf{49.5 \pm 19.64}$ | $9.5 \pm 4.22$ | $15.75 \pm 11.31$ | $\underline{39.5 \pm 15.27}$ | $6.7 \pm 3.13$ | $11.2 \pm 4.94$ | $24.1 \pm 10.99$ |
| | $\overline{L}$ | $9.36 \pm 6.74$ | $16.18 \pm 4.47$ | $\mathbf{21.51 \pm 4.88}$ | $8.21 \pm 3.23$ | $14.96 \pm 5.84$ | $\underline{20.73 \pm 2.25}$ | $4.03 \pm 1.1$ | $7.22 \pm 1.86$ | $15.19 \pm 3.67$ |
| | $\overline{T}$ | $10.6 \pm 2.45$ | $9.35 \pm 1.43$ | $7.68 \pm 1.26$ | $9.22 \pm 1.28$ | $7.53 \pm 1.11$ | $\mathbf{6.65 \pm 0.65}$ | $10.64 \pm 2.83$ | $7.81 \pm 0.94$ | $\underline{7.21 \pm 2.66}$ |
| | $\overline{R}$ | $0.01 \pm 0.01$ | $0.02 \pm 0.01$ | $0.05 \pm 0.02$ | $0.02 \pm 0.01$ | $0.03 \pm 0.02$ | $\underline{0.08 \pm 0.03}$ | $0.03 \pm 0.02$ | $0.06 \pm 0.02$ | $\mathbf{0.12 \pm 0.06}$ |

regardless of the population size $|P|$. This is clearer regarding the elitist part of the population $\mathcal{E}$.

Globally, it appears that the smallest $|\mathcal{P}|$ with a high effort provide the best results: the results obtained with ($|\mathcal{P}| = 100$; $E = 20,000$) and ($|\mathcal{P}| = 200$; $E = 20,000$) are very similar, except the proportion of acceptable shapes in $\mathcal{E}$ (respectively 49.5% and 39.5%). For each metric, we performed a *Mann-Withney-Wilconox* (Table 3) test to highlight any differences between each obtained result: it appears that only the average recall $\overline{R}$ values (from $\mathcal{P}$ and $\mathcal{E}$) are significantly different between them ($< 0.05$) which suggests that the choice ($|\mathcal{P}| = 200$; $E = 20,000$) is the best one for this measure.

**Selection ($\mathcal{R}$) pressure** We assume that the smallest population combined with the highest effort, *i.e.*, ($|\mathcal{P}| = 100$; $E = 20,000$), is the best choice for analysing selective pressure and learning about its impact on metrics. Conse-

Table 3: *Mann-Whitney-Wilcoxon* test: comparison between the results obtained for ($|P| = 100$; $E = 20,000$) and ($|P| = 200$; $E = 20,000$) with $\alpha = 0.05$.

| From $\mathcal{P}$ | | From $\mathcal{E}$ | |
|---|---|---|---|
| Metrics | P-value | Metrics | P-value |
| $\overline{F}$ | 0.528 | $\overline{F}$ | 0.529 |
| $\overline{\%A}$ | 0.198 | $\overline{\%A}$ | 0.210 |
| $\overline{L}$ | 0.684 | $\overline{L}$ | 0.796 |
| $\overline{T}$ | 0.631 | $\overline{T}$ | 0.076 |
| $\overline{R}$ | **0.037** | $\overline{R}$ | **0.028** |

quently, we have studied different selection types: *Scaled Roulette Wheel* and *Tournament* with the different settings presented in Table 4.

Table 4: Parameters used to analyse the impact of the selective pressure on $\mathcal{R}$.

| **Parameters** | **Value(s)** |
|---|---|
| **GE** | |
| $\|P\|$ | 100 |
| Effort (E) | $20,000$ |
| % Selection ($\mathcal{E}$) | 20% |
| Selection type | {Scaled Roulette Wheel; Tournament} |
| % Selection ($\mathcal{R}$) | {20%; 40%; 60%} |
| % Tournament | {10%; 25%; 50%} |
| Crossover type - P | Swap (Fig. 3a) - 75% |
| Mutation type - P | Int Flip (Fig. 3b) - 5% |
| **Probabilistic SHACL** | |
| Confidence level $\alpha$ | 5% |
| Theoretical inconsistencies proportion $p$ | 50% |

The results obtained with the *Scaled Roulette Wheel* selection are presented in Table 5. They highlight that the metrics are enhanced with a high selection rate ($S = 60\%$) even though no very good candidates have been found. It appears that a high selection rate enhances the exploration of the solution space, resulting in a relatively strong difference between the results observed for $\mathcal{P}$ and the elitist subset $\mathcal{E}$.

The results obtained with the *Tournament* selection are presented in Table 6 and we identify the same trend. For $S = 20\%$, the global difference of results between $\mathcal{P}$ and $\mathcal{E}$ is quite low, enhancing the homogeneity of the population $\mathcal{P}$ whereas a high selection rate ($S = 60\%$) reflects the heterogeneity of the population because the global difference of results between $\mathcal{P}$ and $\mathcal{E}$ is high. We can see from the difference between the average time for the whole population

Table 5: Results obtained using the *Scaled Roulette Wheel* selection and parameters presented in Table 4: best result for each metric is in bold and second best underlined.

| | | $S = 20\%$ | $S = 40\%$ | $S = 60\%$ |
|---|---|---|---|---|
| From $\mathcal{P}$ | $\overline{F}$ | $\underline{0.86 \pm 1.06}$ | $\mathbf{1.78 \pm 2.99}$ | $0.56 \pm 0.16$ |
| | $\overline{\%A}$ | $\underline{8.7 \pm 3.83}$ | $7.3 \pm 5.33$ | $\mathbf{11.7 \pm 3.06}$ |
| | $\overline{L}$ | $7.85 \pm 2.46$ | $\mathbf{8.85 \pm 1.89}$ | $\underline{8.25 \pm 2.6}$ |
| | $\overline{T}$ | $20.25 \pm 6.26$ | $\underline{19.86 \pm 8.61}$ | $\mathbf{19.83 \pm 5.78}$ |
| | $\overline{R}$ | $\underline{0.04 \pm 0.02}$ | $\underline{0.04 \pm 0.03}$ | $\mathbf{0.06 \pm 0.02}$ |
| From $\mathcal{E}$ | $\overline{F}$ | $\underline{4.27 \pm 5.29}$ | $\mathbf{8.87 \pm 14.97}$ | $2.74 \pm 0.8$ |
| | $\overline{\%A}$ | $\underline{43.5 \pm 19.16}$ | $36 \pm 25.47$ | $\mathbf{58 \pm 14.94}$ |
| | $\overline{L}$ | $\underline{24.33 \pm 4.36}$ | $\mathbf{25.25 \pm 5.5}$ | $23.94 \pm 4.31$ |
| | $\overline{T}$ | $8.49 \pm 1.29$ | $\underline{7.77 \pm 1.46}$ | $\mathbf{7.38 \pm 1.19}$ |
| | $\overline{R}$ | $\underline{0.04 \pm 0.02}$ | $0.04 \pm 0.03$ | $\mathbf{0.06 \pm 0.02}$ |

vs the average time for the elite that a high selection rate $S$ tends to make the population vary considerably while maintaining a very good elite population which ensures shapes of good quality in the elite with a wide exploration in the global population, favouring the discovery of heterogeneous and potentially interesting shapes.

Table 6: Results obtained using the *Tournament* selection and parameters presented in Table 4: best result for each metric is in bold, second best underlined. The highlighted column corresponds to the *reference* results presented in Table 2.

| | | $S = 20\%$ | | | $S = 40\%$ | | | $S = 60\%$ | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | $Tour = 10\%$ | $Tour = 25\%$ | $Tour = 50\%$ | $Tour = 10\%$ | $Tour = 25\%$ | $Tour = 50\%$ | $Tour = 10\%$ | $Tour = 25\%$ | $Tour = 50\%$ |
| From $\mathcal{P}$ | $\overline{F}$ | $1.78 \pm 1.93$ | $\underline{1.82 \pm 2.44}$ | $1.08 \pm 1.98$ | $1.2 \pm 1.54$ | $1.52 \pm 1.7$ | $1.71 \pm 2.42$ | $0.99 \pm 0.83$ | $\mathbf{2.32 \pm 2}$ | $1.78 \pm 2.33$ |
| | $\overline{\%A}$ | $\mathbf{11.1 \pm 5.2}$ | $8.9 \pm 5.45$ | $8.1 \pm 5.43$ | $9.9 \pm 4.51$ | $10.1 \pm 4.18$ | $9.7 \pm 4.27$ | $8.3 \pm 3.97$ | $\underline{10.7 \pm 4.99}$ | $8.4 \pm 3.27$ |
| | $\overline{L}$ | $5.55 \pm 2.6$ | $6.84 \pm 2.81$ | $\underline{6.98 \pm 2.58}$ | $6.11 \pm 1.14$ | $6.74 \pm 2.07$ | $6.03 \pm 2.02$ | $6.12 \pm 1.32$ | $6.36 \pm 2.1$ | $\mathbf{7.1 \pm 1.32}$ |
| | $\overline{T}$ | $22.92 \pm 9.12$ | $\mathbf{16.4 \pm 3.4}$ | $\underline{18.88 \pm 6.41}$ | $23.93 \pm 5.93$ | $21.94 \pm 3.73$ | $21.66 \pm 7.37$ | $28.44 \pm 7.5$ | $27.89 \pm 6.46$ | $27.9 \pm 6.62$ |
| | $\overline{R}$ | $\mathbf{0.06 \pm 0.03}$ | $0.04 \pm 0.03$ | $0.04 \pm 0.03$ | $\underline{0.05 \pm 0.02}$ | $\underline{0.05 \pm 0.02}$ | $\underline{0.05 \pm 0.02}$ | $0.04 \pm 0.02$ | $\underline{0.05 \pm 0.03}$ | $0.04 \pm 0.02$ |
| From $\mathcal{E}$ | $\overline{F}$ | $8.83 \pm 9.64$ | $\underline{9.02 \pm 12.21}$ | $5.34 \pm 9.94$ | $5.93 \pm 7.69$ | $7.41 \pm 8.48$ | $8.47 \pm 12.13$ | $4.9 \pm 4.14$ | $\mathbf{11.55 \pm 10.02}$ | $8.86 \pm 11.68$ |
| | $\overline{\%A}$ | $\mathbf{55.5 \pm 25.98}$ | $43.5 \pm 26.46$ | $39.5 \pm 25.65$ | $49 \pm 21.96$ | $49.5 \pm 19.64$ | $48.5 \pm 21.35$ | $41 \pm 19.12$ | $\underline{52.5 \pm 24.41}$ | $42 \pm 16.36$ |
| | $\overline{L}$ | $18.96 \pm 6.49$ | $19.1 \pm 3.73$ | $21.6 \pm 4.95$ | $21.22 \pm 2.5$ | $21.51 \pm 4.88$ | $19.77 \pm 3.69$ | $22.68 \pm 4.24$ | $\underline{24.25 \pm 5.67}$ | $\mathbf{24.46 \pm 3.09}$ |
| | $\overline{T}$ | $7.54 \pm 1$ | $8.09 \pm 1.07$ | $7.76 \pm 0.79$ | $7.4 \pm 1.52$ | $7.68 \pm 1.26$ | $7.25 \pm 1.51$ | $\mathbf{6.93 \pm 0.43}$ | $\underline{7.06 \pm 1.18}$ | $8.02 \pm 1.38$ |
| | $\overline{R}$ | $\mathbf{0.06 \pm 0.03}$ | $0.04 \pm 0.03$ | $0.04 \pm 0.03$ | $\underline{0.05 \pm 0.02}$ | $\underline{0.05 \pm 0.02}$ | $\underline{0.05 \pm 0.02}$ | $0.04 \pm 0.02$ | $\underline{0.05 \pm 0.02}$ | $0.04 \pm 0.02$ |

**Acceptable shapes** Among all the conducted experiments, we have discovered a set of $1,766$ distinct and acceptable shapes. An overview of these results is provided in Table 7. Some of these shapes have been accepted with a very high violation rate ($> 50\%$) but can be easily validated, e.g. the candidate implying the following rule (`gene expression profiling` $\rightarrow$ `gene expression`) is easily understandable and acceptable even if it implies 52.6% of violations. Moreover, 46.38% of the whole has been accepted after performing hypothesis testing which

shows it has a valuable impact on the acceptance of shapes and so on the mining process.

However, some of these shapes require final validation from experts due to the complexity, e.g. the following rule (`chemokine` → `cytokine`) has been automatically accepted and validated after some research: "*Chemokines [...] are a family of small cytokines*"[5]. Rule (`tlr9` → `toll-like receptor`) has been accepted but must be validated by experts.

Table 7: Overview of the distinct and acceptable shapes discovered from all the performed experiments.

| Metrics | $\overline{F}$ | $\overline{L}$ | $\overline{T}$ | $\overline{R}$ |
|---|---|---|---|---|
| Values | 19.49 | 19.14 | 7.93 | 8.91 |

The discovery of very good candidates from some of these experiments impacts the standard deviation of many values $\overline{F}$ and $\overline{\%A}$ (some of these are higher than the mean value). We also note some trivial shapes with identical classes for the `sh:targetClass` and the constraint `sh:hasValue` which implies a perfect acceptance of these candidates, *i.e.*, without any violations, and so a very good fitness value. These can be generated because of production rules selection (with a modulo operator) and a *quasi-infinite* range for codon definition. However, this is a fairly rare occurrence: we observe it among (only) 132 candidates from the 1,766 acceptable shapes, *i.e.*, 7.47%. We suggest accepting a low occurrence of these shapes being discovered (even if they are uninteresting) in order to avoid any negative impact on the exploration of the solution space.

The $\overline{T}$ value presented in Table 7 and the correlation between the number of violations and the CPU time presented in Fig. 4 suggest that the CPU time required to invest in an evolutionary process is maximal at the beginning, then decreases as the average number of violations decreases (and therefore when the shapes become more and more acceptable). Considering this expected evolution and that the average time is low, this demonstrates the relevance of this evolutive approach to the discovery of SHACL shapes over an RDF data graph and appears to be suitable for scalability.

## 6 Conclusion

In this paper, we have proposed a framework using a grammatical evolution method to extract candidate SHACL shapes from an RDF dataset based on a manually defined BNF grammar. The proposed algorithm provides an effective response to the redundancy problem whereas the low locality appears to be *problem-dependent*. Additionally, the generative evolution allows to tackle the

[5] https://en.wikipedia.org/wiki/Chemokine

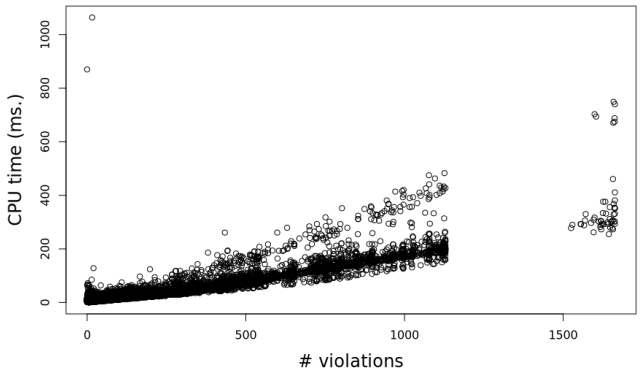



Fig. 4: CPU time spent for the SHACL validation of each discovered shape considering the number of violations

requirement of a broad exploration of the huge search space of possible SHACL shapes to discover acceptable ones. The framework uses a probabilistic SHACL validation process with an acceptability measure and a fitness function to evaluate candidate forms and retain the best ones. Experiments show that our approach captures interesting SHACL shapes describing domain constraints from a real-world RDF dataset. In addition, it provides an effective way to discover a large set of heterogeneous shapes in RDF data, weighting the errors that these may imply and adapting the mining of these shapes. Our future research will focus on studying the scalability of this approach to larger RDF datasets such as DBpedia ($> 50M$ triples). In addition, we plan to extend the approach to the exploration of complex shapes, e.g. shapes with multiple constraints. Finally, it appears important to study the application (and adaptation) of various algorithms proposed by the community, e.g. Structured Grammatical Evolution [15], to the SHACL shapes mining.

*Supplemental Material Statement* The source code, RDF datasets and obtained results are available in a public repository.[6]

**Acknowledgements** This work has been partially founded by the 3IA Côte d'Azur "Investments in the Future" project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002.

## References

1. shaclgen 0.2.5.2, https://github.com/uwlib-cams/shaclgen

[6] https://github.com/RemiFELIN/RDFMining/tree/eurogp_2024

2. Ali, M.S., Kshirsagar, M., Naredo, E., Ryan, C.: Autoge: A tool for estimation of grammatical evolution models. In: International Conference on Agents and Artificial Intelligence (2021), https://api.semanticscholar.org/CorpusID:232106265
3. Boneva, I., Dusart, J., Fernández Alvarez, D., Gayo, J.E.L.: Shape Designer for ShEx and SHACL Constraints. ISWC 2019 - 18th International Semantic Web Conference (Oct 2019), https://hal.science/hal-02268667, poster
4. Cimmino, A., Fernández-Izquierdo, A., García-Castro, R.: Astrea: Automatic generation of SHACL shapes from ontologies. In: ESWC. Lecture Notes in Computer Science, vol. 12123, pp. 497–513. Springer (2020)
5. Cérès, R., Corby, O., Demairy, E.: Corese (Mar 2023), https://github.com/Wimmics/corese
6. Dick, G., Whigham, P.A.: Initialisation and grammar design in grammar-guided evolutionary computation (2022)
7. Felin, R., Faron, C., Tettamanzi, A.G.B.: A Framework to Include and Exploit Probabilistic Information in SHACL Validation Reports. In: ESWC (2023)
8. Fernandez-Álvarez, D., Labra-Gayo, J.E., Gayo-Avello, D.: Automatic extraction of shapes using shexer. Knowledge-Based Systems **238**, 107975 (2022). https://doi.org/10.1016/j.knosys.2021.107975
9. Harper, R.: Ge, explosive grammars and the lasting legacy of bad initialisation. In: IEEE Congress on Evolutionary Computation. pp. 1–8 (2010). https://doi.org/10.1109/CEC.2010.5586336
10. Kim, H.T., Ahn, C.W.: A new grammatical evolution based on probabilistic context-free grammar. In: Handa, H., Ishibuchi, H., Ong, Y.S., Tan, K.C. (eds.) Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems - Volume 2. pp. 1–12. Springer International Publishing, Cham (2015)
11. Kontokostas, D., Knublauch, H.: Shapes constraint language (SHACL). W3C recommendation, W3C (2017)
12. Lourenço, N., Assunção, F., Pereira, F., Costa, E., Machado, P.: Structured grammatical evolution: A dynamic approach, pp. 137–161 (01 2018). https://doi.org/10.1007/978-3-319-78717-6_6
13. Lourenço, N., Ferrer, J., Pereira, F.B., Costa, E.: A comparative study of different grammar-based genetic programming approaches. In: McDermott, J., Castelli, M., Sekanina, L., Haasdijk, E., García-Sánchez, P. (eds.) Genetic Programming. pp. 311–325. Springer International Publishing, Cham (2017)
14. Lourenço, N., Pereira, F., Costa, E.: Unveiling the properties of structured grammatical evolution. Genetic Programming and Evolvable Machines **17** (09 2016). https://doi.org/10.1007/s10710-015-9262-4
15. Lourenço, N., Pereira, F.B., Costa, E.: Sge: A structured representation for grammatical evolution. In: Bonnevay, S., Legrand, P., Monmarché, N., Lutton, E., Schoenauer, M. (eds.) Artificial Evolution. pp. 136–148. Springer International Publishing, Cham (2016)
16. Medvet, E.: A comparative analysis of dynamic locality and redundancy in grammatical evolution. In: McDermott, J., Castelli, M., Sekanina, L., Haasdijk, E., García-Sánchez, P. (eds.) Genetic Programming. pp. 326–342. Springer International Publishing, Cham (2017)
17. Mégane, J., Lourenço, N., Machado, P.: Probabilistic grammatical evolution (2021)
18. Mégane, J., Lourenço, N., Machado, P.: Co-evolutionary probabilistic structured grammatical evolution. In: Proceedings of the Genetic and Evolutionary Computation Conference. ACM (jul 2022). https://doi.org/10.1145/3512290.3528833

19. Mégane, J., Lourenço, N., Machado, P.: Probabilistic structured grammatical evolution. In: 2022 IEEE Congress on Evolutionary Computation (CEC). IEEE (jul 2022). https://doi.org/10.1109/cec55065.2022.9870397
20. Michel, F., Gandon, F., Ah-Kane, V., Bobasheva, A., Cabrio, E., Corby, O., Gazzotti, R., Giboin, A., Marro, S., Mayer, T., Simon, M., Villata, S., Winckler, M.: Covid-on-the-web: Knowledge graph and services to advance COVID-19 research. In: ISWC (2). Lecture Notes in Computer Science, vol. 12507, pp. 294–310. Springer (2020)
21. Nguyen, T.H., Tettamanzi, A.G.B.: An Evolutionary Approach to Class Disjointness Axiom Discovery. In: Barnaghi, P.M., Gottlob, G., Manolopoulos, Y., Tzouramanis, T., Vakali, A. (eds.) WI 2019 - IEEE/WIC/ACM International Conference on Web Intelligence. pp. 68–75. ACM, Thessaloniki, Greece (Oct 2019). https://doi.org/10.1145/3350546.3352502
22. Nguyen, T.H., Tettamanzi, A.G.B.: Grammatical Evolution to Mine OWL Disjointness Axioms Involving Complex Concept Expressions. In: CEC 2020 - IEEE Congress on Evolutionary Computation. pp. 1–8. IEEE, Glasgow, United Kingdom (Jul 2020). https://doi.org/10.1109/CEC48606.2020.9185681
23. Nicolau, M., Agapitos, A.: Understanding grammatical evolution: Grammar design, pp. 23–53 (01 2018). https://doi.org/10.1007/978-3-319-78717-6_2
24. Nicolau, M., O'Neill, M., Brabazon, A.: Termination in grammatical evolution: grammar design, wrapping, and tails. pp. 1–8 (06 2012). https://doi.org/10.1109/CEC.2012.6256563
25. Omran, P., Taylor, K., Rodríguez Méndez, S., Haller, A.: Learning shacl shapes from knowledge graphs. Semantic Web **14**, 1–21 (09 2022). https://doi.org/10.3233/SW-223063
26. O'Neill, M., Ryan, C.: Grammatical evolution. IEEE Trans. Evol. Comput. **5**(4), 349–358 (2001)
27. Pandit, H., O'Sullivan, D., Lewis, D.: Using ontology design patterns to define shacl shapes. In: WOP@ISWC. pp. 67–71. Monterey California, USA (2018)
28. Rabbani, K., Lissandrini, M., Hose, K.: SHACL and shex in the wild: A community survey on validating shapes generation and adoption. In: WWW (Companion Volume). pp. 260–263. ACM (2022)
29. Rothlauf, F., Oetzel, M.: On the locality of grammatical evolution. In: Collet, P., Tomassini, M., Ebner, M., Gustafson, S., Ekárt, A. (eds.) Genetic Programming. pp. 320–330. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
30. Ryan, C., O'Neill, M., Collins, J.: Introduction to 20 Years of Grammatical Evolution, pp. 1–21. Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-319-78717-6_1