# Mining Innovative Augmented Graph Grammars for Argument Diagrams through Novelty Selection

Linting Xue, Collin F. Lynch & Min Chi
North Carolina State University, Raleigh, North Carolina, USA
lxue3, cflynch, & mchi@ncsu.edu

## ABSTRACT

Augmented Graph Grammars are a graph-based rule formalism that supports rich relational structures. They can be used to represent complex social networks, chemical structures, and student-produced argument diagrams for automated analysis or grading. In prior work we have shown that Evolutionary Computation (EC) can be applied to induce empirically-valid grammars for student-produced argument diagrams based upon fitness selection. However this research has shown that while the traditional EC algorithm does converge to an optimal fitness, premature convergence can lead to it getting stuck in local maxima, which may lead to undiscovered rules. In this work, we augmented the standard EC algorithm to induce more heterogeneous Augmented Graph Grammars by replacing the fitness selection with a novelty-based selection mechanism every ten generations. Our results show that this novelty selection increases the diversity of the population and produces better, and more heterogeneous, grammars.

## Keywords

Heterogeneous Rules, Augmented Graph Grammars, Argument Diagrams, Evolutionary Computation, Novelty selection

## 1. INTRODUCTION

Intelligent tutoring systems, social-networking systems, and computer-supported collaborative platforms have grown increasingly prevalent in education (e.g. Pyrenees [15], LASAD [8], and CSCL [13]). Consequently, researchers have begun to collect large repositories of complex relational data representing student-produced conceptual or structural diagrams [8], structured user-system interaction logs [15], and personal relationships [13]. Researchers have generally analyzed this data via standard network analysis tools and gestalt relationships which allow us to assess general topological graph structures but which do not focus on individual graph features or graph rules (e.g. [15, 13]).

One of the primary goals of Graph-based Educational Data Mining is to automatically identify substructures that can reveal vital pedagogical information in graph data. These features include good sub-solutions and structural flaws in students' solutions, which can be used for automated guidance and grading [10]. Prior research has demonstrated that we can use hand-authored graph rules to evaluate student-produced argument diagrams [10]. But, hand-authored rules are expensive and time consuming to generate and do not always generalize well to novel contexts. Existing general purpose graph rule induction algorithms (e.g. [16, 2]) have limitations and are unsuited to the induction of generalized rules that use negation or other hierarchical elements [17].

Evolutionary Computation (EC), on the other hand, is both flexible and robust enough to induce complex graph structures and to deal with rich graph data. We have previously shown that EC can be used to automatically induce positive and negative graph rules for student-produced argument diagrams through fitness selection [17]. The induced rules can be used as features to provide hints for argument writing, and to detect structural flaws. Prior research also indicates that the induced graph rules from EC outperform all but one of the expert hand-authored rules and they outperform all of the rules induced by two general purpose graph grammar induction algorithms, Subdue [2] and gSpan [16]. However, prior research has shown that, while the traditional EC algorithm does converge to an optimal fitness, the premature convergence can lead to it getting stuck in local maxima, which may lead to undiscovered graph rules [6].

In this work, we augmented the standard EC algorithm to produce more heterogeneous Augmented Graph Grammars that can reflect innovative structures in student-produced argument diagrams. To that end, we incorporated a *novelty selection* mechanism into our EC system that was designed to enforce population diversity. The goal of this diversity was to explicitly retain novel *introns* and thus to reward the basic stepping stones of evolution both in the internal (*genospace*) and the external application space (*phenospace*), respectively. In this work, we experimented with two different novelty selection mechanisms: novel genotype selection and novel phenotype selection. Our research hypotheses is that *novelty selection will increase the diversity of the population and will produce better and more heterogeneous graph grammars when compared with pure fitness selection.*

1

## 2. BACKGROUND

### 2.1 Argument diagrams

Argument diagrams are graphical representations for real-world argumentation that reify the essential components of arguments such as *hypotheses* statements, *claims*, and *citations* as nodes and the *supporting*, *opposing*, and *clarification* relationships as arcs [11]. These complex elements can include text fields describing the node and arc types or free-text assertions, links to external resources and other data.

A sample student-produced diagram is shown in Figure 1. The diagram includes a *hypothesis* node at the bottom right, which contains two text fields, one for a conditional or *if* field, and the other for a consequent or *then* field. Two *citations* are connected to the hypothesis via *supporting* and *opposing* arcs colored green and red, respectively. They are also connected via a *comparison* arc. Each citation contains two fields: one for the citation information and the other for a summary of the work. Each arc has a single text field explaining what purpose the relationship serves.
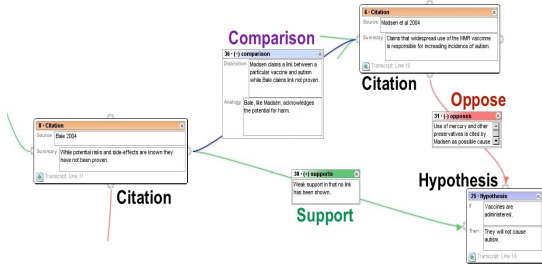


**Figure 1: A student-produced Argument Diagram.**

### 2.2 Augmented Graph Grammars

Augmented Graph Grammars (AGGs) are a graph-based rule formalism that supports rich relational structures [9]. AGGs are an extension of traditional graph grammars, which are composed of standard graph elements including *ground* nodes, *ground* arcs, and *variable* arcs which can match multiple items. In addition to these basic features, AGGs also support: complex node and arc types that contain sub-elements; negated elements which select for the nonexistence of subgraphs; generalized node and arc types which match multiple items; complex element constraints which allow us to compare individual elements; complex graph expressions which allow for universal and existential quantification; and the incorporation of NLP rules or other external constraints. As such they are an ideal rule representation for the analysis of argument diagrams.

In prior work [10, 11], we collaborated with a group of domain experts to define a set of 77 a-priori argument rules encoded as grammars. These rules were designed to identify individual features of argument diagrams or sub-graphs that were consistent with high quality argumentation or which represented common structural flaws. We have shown that these hand-authored graph rules are correlated with the student-produced argument diagram grades and essay grades and they are empirically valid and can be used as
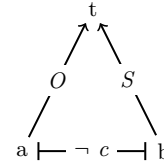


$$\left\{ \begin{array}{c} t.Type = \text{``}claim\text{''}\,or\,\text{``}hypothesis\text{''} \\ a.Type = \text{``}citation\text{''} \\ b.Type = \text{``}citation\text{''} \\ c.Type = \text{``}comparison\text{''} \end{array} \right\}$$

**Figure 2: A hand-authored Augmented Graph Grammar.**

the basis for predictive models of student grades. A sample hand-authored rule is shown in Figure 2. This rule is designed to identify cases where students use a citation $a$ to oppose a claim or hypothesis node $t$ via an opposing path $O$, and use the other citation $b$ to support the node $t$ via a supporting path $S$, however, the students do not include a comparison arc $c$ between two citations $a$ and $b$.

### 2.3 Evolutionary Computation

Evolutionary Computation (EC) is a general machine learning algorithm based upon Natural Selection. The algorithm starts with a population of candidate solutions, which may be generated at random or user-defined. The individual *solutions* are assessed by an objective measurement known as the *fitness function*. Subsequent generations are produced by a combination of *elitism* in which very fit individuals are cloned into the next generation, and fitness-proportional *reproduction* in which individuals are copied over with direct *mutations* or through *crossover* with other members in the population. The EC algorithm proceeds iteratively until a given fitness threshold is reached or until a fixed number of generations has passed. When compared with existing graph grammar induction algorithms, EC is much more flexible and robust. The behavior of the system is determined by the user-defined *solution representation*, *fitness function*, and the genetic operators including *mutation* and *crossover*.

In prior work, we applied EC to automatically induce a set of AGG rules on student-produced argument diagrams [17]. The induced rules support disjoint subgraphs, negation, and generalized elements. In that work, the solution representation was an individual graph rule. The fitness of each graph rule was accessed via Spearman's Rank Sum Correlation ($\rho$) [3] between the frequency with which a rule matches a diagram, and the argument grades. The mutation in the EC algorithm was basic point mutation that can add, delete, or modify existing nodes and arcs. Crossover was implemented using matrix crossover based upon the work of Stone, Pillmore, & Cyre [14].

### 2.4 Novelty Selection

Absolute fitness functions of the type that we used in our prior studies, are designed to reward *individual* progress toward an absolute objective in the search space without consideration for the population as a whole. Prior studies have shown that although the fitness function is driven to con-

verge to a fitness optimum, the *objective function* sometimes suffers from the pathology of local optima [6]. This is because the objective function only rewards improvements in performance with respect to the static objective, it does not necessarily reward diversity in the search space that can ultimately lead to other solutions. One approach that EC researchers have taken to address this problem is *Novelty Selection* that is, explicitly incorporating population diversity into the fitness metric or supporting diverse solutions irrespective of the fitness value [1, 5]. The goal in doing so is to encourage the development of good sub-solutions or *stepping stones* that can support novel solutions and avoid local optima.

Current novelty selection algorithms fall into one of two broad categories: novel *genotype* selection, or novel *phenotype* selection. In EC, the genotype of a solution is the basic solution structure or code that defines the solution, which corresponds to the set of genes in a real organism. The phenotype, by contrast, is the observed behavior of the solution when it is evaluated. In the context of our work, the genotype is the AGG structure while the phenotype is the way in which the rule maps to the graphs in our dataset. Thus the genotype is fixed while the phenotype is data-driven.

The novel genotype selection is focused on finding individuals that have a unique structure relative to the remainder of the population. Prior researchers have focused on applying user-defined metrics to calculate pairwise distances between members of the population [4, 1]. The metrics are necessarily representation specific. Maximally-unique individuals are then selected for reproduction or cloning in order to maintain genetic diversity. The primary shortcoming of this method is that computing pairwise distance can be computationally intractable (e.g. comparing neural networks which is NP-Hard) [5].

While novel genotype selection seeks individuals with unique genes, novel phenotype selection rewards individuals that *behave* differently according to some separate evaluating metric. This is usually based upon some user-defined distance function based upon prior knowledge of the domain. The goal of the metrics is to enforce coverage of the solution space and, as with the genotype selection, maximally unique individuals are selected for retention. The primary disadvantage of this approach is that given two individuals with comparable behavior but distinct genes we will discard one and will potentially lose good evolvable genes in the process [5].

## 3. METHODS

In order to compare the performance of novelty selection with traditional objective fitness selection, we implemented two novelty selection methods in EC with one rewarding novel rule structures (genotype) and the other rewarding rules that match a unique set of graphs in our dataset (phenotype). For the former metric, we select the novel rules according to the *diversity score*, which is calculated using a greedy graph-matching algorithm; for the latter one, the novel rules are rewarded based on the *behavior score* using the $\chi^2$ test[3]. A large diversity or behavior score indicates that the specified rule is substantively different from the rest of the population.

### 3.1 Genotypic Distance - Diversity Score

We define the diversity score of an individual as its average genotypic distance from the remainder of the population. In order to compute this score, we developed a greedy graph matching algorithm that computes the distance based upon local-neighborhood similarity. The root intuition behind this algorithm is that if two graph grammars $G_0$ and $G_1$ are isomorphic then it should be possible to automatically align their local neighborhoods (individual nodes plus immediate neighbors). The algorithm returns a distance score between 0 and 1 inclusive. Here 0 means that the two grammars are completely isomorphic and 1 indicates they are wholly distinct from one another. The algorithm operates as follows:

First, we count the total number of nodes $n$ in both grammars on a per-type basis. For example, Figure 3 shows two graph grammars $G_0$ and $G_1$. They have a total of 6 nodes of 5 types $(A, B, C, D, E)$ and 4 arcs of 2 types $(1, 2)$. For category $A$, $G_0$ has one $A$ node $(A_0)$, while $G_1$ has two $(A_0$ & $A_1)$, so $n_a = max(2, 1) = 2$. For the remaining types $B, C, D$ and $E$, we have $n_b = n_c = n_d = n_e = 1$, and the total number of nodes $n$ is 6.
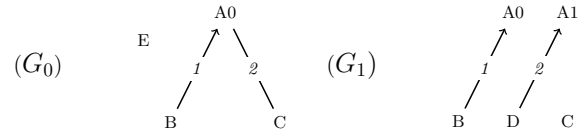


**Figure 3: Example of two graph grammars with five categories of nodes $(A, B, C, D, E,)$ and two categories of arcs $(1, 2)$.**

Second, we compute the individual similarity score $S = \{s_1, s_2, s_3, ..., s_i, ..., s_n\}$ for $i \in \{0, n\}$, where $s_i$ indicates the similarity score for node $N_i$. For nodes of the same type, we use greedy search to find the best match for each node and then update the maximum similarity score of the whole grammar. The value of $s_i$ is between -1 and 1, and is computed by the following formula:

$$s_i = \begin{cases} -1 & if\ N_i\ in\ G_0\ or\ G_1; \quad (1) \\ \dfrac{\#\ of\ shared\ neighbors}{total\ \#\ of\ neighbors\ in\ G_0\ and\ G1} & otherwise. \quad (2) \end{cases}$$

where $s_i = -1$ means that node $N_i$ is in either $G_0$ or $G_1$ but not both; $s_i = 0$ indicates that node $N_i$ is in both graphs, but they do not share any neighbour at all; $s_i = 1$ indicates that node $N_i$ is in both graphs and they share the same neighbor(s) with the same arc(s). Note that if two nodes share a same neighbour but with different arcs, we do not count it as the same neighbour.

In the example shown in Figure 3, we have $S = \{s_a^1, s_a^2, s_b, s_c, s_d, s_e\}$. For $A$ nodes, if we match $A_0 \in G_0$ with $A_0 \in G_1$, we have $s_a = \frac{1}{2}$; if we match $A_0 \in G_0$ with $A_1 \in G_1$, $s_a$ is 0. Thus, the best match for $A_0 \in G_0$ is $A_0 \in G_1$ and update for $s_a^1 = \frac{1}{2}$. Now for $A_1 \in G_1$, we cannot find any node to match with, so $s_a^2 = -1$ using Equation (2). For the

$B$ nodes, $B$ is present in both graphs, they share the same neighbour ($A$) with the same arc type of (1), so $s_b = 1$. Similarly $C$ nodes are present in both graphs, but they do not share any neighbours because $C \in G_1$ is isolated, so $s_c = \frac{0}{1} = 0$. For $D$ and $E$, we have $s_d = s_e = -1$ because node $D$ and $E$ is just shown in one of the two graphs. Thus we have $S = \{\frac{1}{2}, -1, 1, 0, -1, -1\}$.

Finally, we use Euclidean distance to normalize the similarity scores to a distance score within a range of $[0, 1]$ by Equation (3). Then the diversity score for an individual is the average distance score to the remaining population.

$$D = \sqrt{\frac{\sum_{n=1}^{n}(1 - s_i)^2}{n * 2^2}} \qquad (3)$$

## 3.2 Phenotypic Distance - Behavior Score

The behavior score of an individual is the average phenotypic distance between it and the remainder of the population. We use a data-driven definition of behavior. For each individual we define its *behavior signature* as a vector of positive integers representing the number of distinct subgraphs that it matches for each of the 104 graphs in our dataset. We then calculate the pairwise distance between individuals using the $\chi^2$ test of independence [12]. $\chi^2$ is a statistical test that measures divergence from the expected distribution assuming that one feature occurs independently of the others. It is often applied to evaluate the independence of two variables in mathematical statistics [7]. The null hypothesis of this test is that two variables are wholly independent. A p-value $\leq 0.05$ of $\chi^2$ test leads us to reject the null hypothesis and conclude that the variables are significantly correlated.

If two frequency sets are statistically independent from one another other according to the $\chi^2$ test then we assign a phenotypic distance score as 1 indicating that the grammars are independent. If, however they are dependent then we assign a score of 0, meaning that the grammars are substantively similar given our dataset. We then calculate the average score for each individual to indicate its relative uniqueness within the population.

## 3.3 Dataset

For this study we used a dataset of 104 argument diagrams that was originally collected at the University of Pittsburgh in a course on Psychological Research Methods [10, 11]. The subgraph shown in Figure 1 was collected as part of this study. Students in the course were instructed to plan their written arguments graphically using LASAD, an online tool for argument diagramming and collaboration [8], and then to produce written essays. The diagramming ontology contained four types of nodes: *citation*, *claim*, *current study* and *hypothesis*; and four types of arcs: *supporting*, *opposing*, *comparison*, and *unspecified*. Current study nodes are used to represent factual information about the study such as the target population. Unspecified arcs represent cases where nodes provide clarification or concept definitions. At the end of the study, 104 paired diagrams and essays were collected. These diagrams and essays were graded by an experienced TA according to a parallel grading rubric.

## 4. EXPERIMENTS

In this work, we evaluated the impact of novelty selection on graph grammar induction by comparing the two types of novelty selection to a traditional objective-fitness approach. We ran three experiments to induce three sets of graph grammars using the different selection functions. The three experiments are Baseline, Geno, and Pheno respectively:

**Baseline**: we used traditional fitness function at each generation. The fitness function measures the correlation between the observed graph rule frequency and diagram grades.

**Geno**: we replaced the fitness function with novel genotype selection on every tenth generation. The novel genotype selection rewards grammars with novel structure for further evolution by cloning them to the next generation.

**Pheno**: we used the novel phenotype selection to reward graph grammars that have significantly different behaviours to the remaining population in every tenth generation.

For each experiment, we conducted a series of three evolutionary runs to explore the search space. In each run, we set a population size of 100 individuals and ran for 500 generations. The initial populations were composed of randomly generated grammars each of which contained between 3 and 10 elements. The nodes and arcs were all ground elements and were selected from a predefined ontology of basic types that matched the argument diagram ontology. The fitness function, crossover and mutation operators were the same as in our prior work discussed in section 2.3. On each evolutionary run, we harvested all graph grammars generated over the course of the run whose performance exceeded a threshold of ($\rho \geq 0.18$) and preserved them for later analysis. The threshold was chosen based upon a series of exploratory studies which showed that $\rho$ values at or above this threshold were statistically significant.

## 5. RESULTS & ANALYSIS

After collecting the three sets of grammars, we applied the graph matching algorithm discussed in section 3.1 to identify the isomorphic rules, we then filtered the overlapping rules to obtain the unique rule sets. Table 1 shows the number of unique rules collected from each experiment along with the $\rho$ values for the top three rules in each unique rule set. The top

**Table 1: The number of unique rules above the threshold ($\rho \geq 0.18$) and the Spearman's Correlation value $\rho$ for the top three best rules**

| Experiments | Unique rules | $\rho$ value | | |
|---|---|---|---|---|
| | | 1st | 2nd | 3rd |
| **Baseline-Only** | 37 | 0.282 | 0.279 | 0.260 |
| **Geno-Only** | **112** | **0.348** | **0.334** | **0.325** |
| **Baseline ∩ Geno** | 146 | 0.371 | 0.369 | 0.362 |
| | | | | |
| **Baseline-Only** | 26 | 0.282 | 0.260 | 0.254 |
| **Pheno-Only** | **99** | **0.348** | **0.334** | **0.333** |
| **Baseline ∩ Pheno** | 157 | 0.371 | 0.369 | 0.362 |

4

(N-U-G)

$$
\begin{array}{ccc}
\text{k0} & \text{k1} & \text{h} \\
\uparrow & \uparrow & \\
s0 & s1 & c2 \\
| & | & \\
\text{c0} & \text{c1} &
\end{array}
\qquad
\left\{
\begin{array}{l}
k*.Type = \text{``claim''} \\
h.Type = \text{``hypothesis''} \\
c*.Type = \text{``citation''} \\
s*.Type = \text{``supporting''}
\end{array}
\right\}
$$
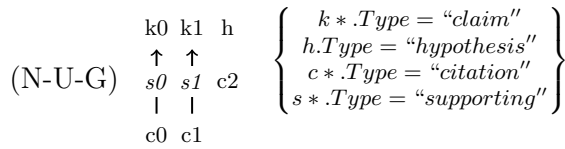
**Figure 4: Best performing graph rule in Geno Only and Pheno Only with correlation ($\rho = 0.348$).**

(EC-Best)

$$
\begin{array}{ccc}
\text{k0} & & \text{h} \\
\nwarrow\nwarrow & & \\
s0 \ \ s1 & & \text{k1} \\
\swarrow \quad \searrow & & \\
\text{c0} & & \text{c1}
\end{array}
\qquad
\left\{
\begin{array}{l}
k*.Type = \text{``claim''} \\
h.Type = \text{``hypothesis''} \\
c*.Type = \text{``citation''} \\
s*.Type = \text{``supporting''}
\end{array}
\right\}
$$

**Figure 5: Best performing rule in EC experiment with the correlation ($\rho = 0.371$).**

(B-U-G)

$$
\begin{array}{ccc}
\text{k0} - s2 \rightarrow \text{k1} \\
\uparrow \qquad\qquad \uparrow \\
s0 \qquad\qquad s1 \\
| \qquad\qquad\quad | \\
\text{c0} \qquad \text{h} \qquad \text{c1}
\end{array}
\qquad
\left\{
\begin{array}{l}
k*.Type = \text{``claim''} \\
c*.Type = \text{``citation''} \\
s*.Type = \text{``supporting''}
\end{array}
\right\}
$$

(G-U-G)

$$
\begin{array}{ccc}
c - s0 \rightarrow \text{k0} \\
| \qquad\qquad | \\
u \qquad\qquad s1 \\
\downarrow \qquad\qquad \downarrow \\
\text{k1} \quad \text{k3} \quad \text{h}
\end{array}
\qquad
\left\{
\begin{array}{l}
k*.Type = \text{``claim''} \\
c.Type = \text{``citation''} \\
h.Type = \text{``hypothesis''} \\
s*.Type = \text{``supporting''} \\
u.Type = \text{``unspecified''}
\end{array}
\right\}
$$

(P-U-G)

$$
\begin{array}{ccc}
\text{h} \\
\uparrow \\
s3 \\
| \\
\text{k} \qquad \text{c3} \\
\nwarrow\nwarrow \\
s1 \quad s2 \\
\swarrow \quad\ \searrow \\
\text{c0} - s0 \rightarrow \text{c1}
\end{array}
\qquad
\left\{
\begin{array}{l}
k*.Type = \text{``claim''} \\
c.Type = \text{``citation''} \\
h.Type = \text{``hypothesis''} \\
s*.Type = \text{``supporting''}
\end{array}
\right\}
$$

**Figure 6: Example graph rules with unique structures. B-N-G: unique rule in Baseline with correlation ($\rho = 0.280$); G-N-G: unique rule in Geno experiment with correlation ($\rho = 0.197$); P-N-G: unique rule in Pheno experiment with correlation ($\rho = 0.182$).**
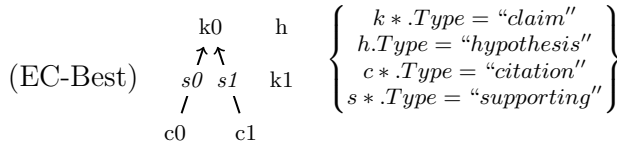
three rows display the rules that are unique to the Baseline and Geno experiments along with the the overlapping rules shared between them (Baseline ∩ Geno ). The bottom three rows show the rules that are unique to the Baseline and Pheno experiments, and the overlapping rules between them (Baseline ∩ Pheno).

As Table 1 indicates, after removing the isomorphic rules, the Geno and Pheno experiments still produced a large number of high-performing rules with Geno-Only having 112 unique rules and Pheno-Only having 99. The top three performing rules in Geno- and Pheno-Only outperform the rules in both the Baseline-Only. After examining these rules, we found that the top two rules in Geno- and Pheno-Only are isomorphic with the same performance and the best rule is shown in Figure 4. This rule contains 6 nodes with two citations ($c0$ & $c1$) supporting two claims ($k0$ & $k1$) and two isolated nodes, one hypothesis ($h$) and one citation ($c2$), which may or may not be connected to the remaining structure. This reflects an argument diagram where the students have two solid claims supported by different citations and where they include both a hypothesis and at least one other additional supporting citation. This rule captures another highly correlated feature in the student-produced argument diagrams that two claims are supported by two different citations.

The top three rules in Baseline ∩ Geno and Baseline ∩ Pheno outperform the rules in both Baseline-Only and the rules in Geno- and Pheno-Only. We also found that these three best rules are isomorphic with the same performance, meaning that all three fitness models are capable of identifying the best performing rules on our dataset. Figure 5 shows the best graph rule with the correlation ($\rho = 0.371$). It represents a rule with 5-nodes, two of which are citations ($c0$ & $c1$) that support a shared claim node ($k0$). The remaining nodes consist of a single claim ($k1$) and hypothesis ($h$) which may or may not be connected to the other elements. This reflects a graph where the authors identified at least two related citations that can be synthesized to support a single
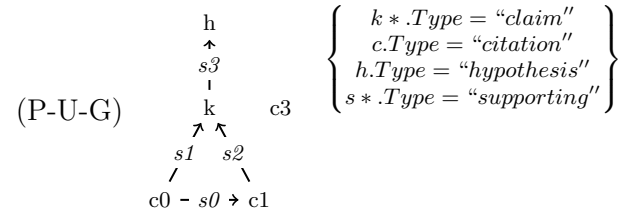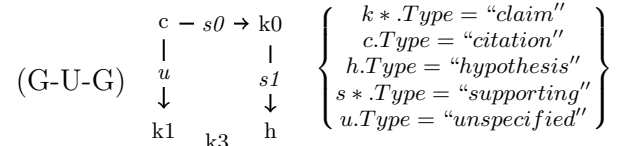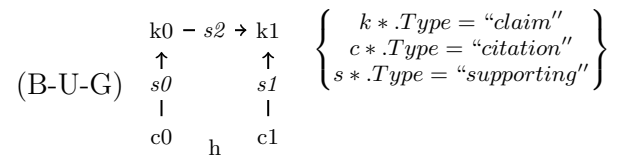
claim and where they included both a hypothesis and another claim. This is one of the structures that students have been encouraged to make in their arguments as it shows an ability to synthesize cited work to form a complex claim.

We also investigated the unique structures that were specific to each experiment. The structure refers to the sub-graph within a graph rule but without isolated node(s). When comparing the Baseline and Geno experiments, we found three unique structures that only show up in the Baseline experiment and six in Geno. When comparing the Baseline and Pheno experiments, we identified three unique structures in the Baseline experiment and four in the Pheno experiment respectively.

Figure 6 shows three example graph rules with unique structures in each experiment. B-U-G is a unique rule induced in the Baseline experiment, it matches cases where two citations ($c0$ & $c1$) support two claims ($k0$ & $k1$) and are connected via a supporting arc ($S2$) and where an isolated hypothesis ($h$) may or may not be connected to the remaining structure. This rule reflects a very interesting argument structure where the student used one citation to directly support a claim and the other citation to support this claim with another intermediate claim. G-U-G shows rule that was induced in the Geno experiment. It has one citation ($c$) that supports a claim ($k0$) which in turn supports a hypothesis ($h$). This citation is also connected to a claim ($k1$) with an unspecified arc ($u$). And it has an isolated claim ($k3$) which may or may not be connected to the remainder of the structure. This rule indicates another innovative use

5

of chaining support which students were encouraged to use and which is comparable to B-N-G.

P-U-G shows a graph rule from the Pheno experiment, it contains a connected structure with four arcs, and is the *most* complex rule above the threshold. This connected structure has two citations with one supporting another ($c0$ & $c1$) and then jointly supporting a shared claim ($k$) which in turn directly supports a hypothesis ($h$). The rule also contains an isolated citation ($c3$) which may or may not connect to the remaining structure. Conceptually this indicates a case where a grounded claim supports a research hypothesis. In the real word, it indicates that the author sought out closely-related sources of literature or noted important connections between them, then used this well-supported claim to support a research hypothesis, something which they had been encouraged to do in class.

## 6. CONCLUSION AND FUTURE WORK

In this work, we augmented the standard EC with two novelty section methods to induce Augmented Graph Grammars on student-produced argument diagrams by replacing the fitness function with a novelty selection function every ten generations. This novelty selection promotes diversity in the population by explicitly encouraging the production and maintenance of novel stepping stones or partial solutions in the genotypic and phenotypic spaces. Our experimental results indicate that, when compared to pure objective-fitness selection, the novelty-selection functions produced more heterogeneous and better-performing graph grammars. The unique rules that were induced by each experiment reflect some novel features in student-produced argument diagrams. The significance of this work is that the novelty selection can enhance EC to produce more empirically-valid rules that can be used for automatic grading.

In future work, we plan to work with domain experts to determine whether the rules are semantically valid, and whether or not they can serve as the basis for automatic hinting. We will also build an intelligent argument grading system to automatically grade and provide feedback on student-produced argument diagrams based on the induced graph grammars and other argument diagram features.

## 7. REFERENCES

[1] L. T. Bui, H. A. Abbass, and J. Branke. Multiobjective optimization for dynamic environments. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 3, pages 2349–2356. IEEE, 2005.

[2] D. J. Cook, L. B. Holder, and N. Ketkar. Unsupervised and supervised pattern learning in graph data. In *Mining Graph Data*, chapter 7, pages 159–180. John Wiley & Sons, Inc, Hoboken, New Jersey, 2007.

[3] P. Dalgaard. *Introductory Statistics with R*. Springer Verlag New York Inc., 2002.

[4] E. D. De Jong, R. A. Watson, and J. B. Pollack. Reducing bloat and promoting diversity using multi-objective methods. In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, pages 11–18. Morgan Kaufmann Publishers Inc., 2001.

[5] S. Doncieux and J.-B. Mouret. Behavioral diversity measures for evolutionary robotics. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8. IEEE, 2010.

[6] L. J. Fogel, A. J. Owens, and M. J. Walsh. *Artificial intelligence through simulated evolution*. John Wiley, 1966.

[7] H. Liu and R. Setiono. Chi2: Feature selection and discretization of numeric attributes. In *Tools with artificial intelligence, 1995. proceedings., seventh international conference on*, pages 388–391. IEEE, 1995.

[8] F. Loll and N. Pinkwart. Lasad: Flexible representations for computer-based collaborative argumentation. *International Journal of Human-Computer Studies*, 71:91–109, Januart 2013.

[9] C. F. Lynch. Agg: Augmented graph grammars for complex heterogeneous data. In S. G. Santos and O. C. Santos, editors, *Proceedings of the 7th International Conference on Educational Data Mining (EDM 2014)*, volume 1183 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2014.

[10] C. F. Lynch and K. D. Ashley. Empirically valid rules for ill-defined domains. In J. Stamper and Z. Pardos, editors, *Proceedings of The $7^{th}$ International Conference on EDM 2014*. IEDMS, 2014.

[11] C. F. Lynch, K. D. Ashley, and M. Chi. Can diagrams predict essay grades? In S. Trausan-Matu, K. E. Boyer, M. E. Crosby, and K. Panourgia, editors, *Intelligent Tutoring Systems*, Lecture Notes in Computer Science, pages 260–265. Springer, 2014.

[12] C. F. Lynch, L. Xue, and M. Chi. Evolving augmented graph grammars for argument analysis. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, pages 65–66. ACM, 2016.

[13] O. Noroozi, H. Biemans, M. C. Busstra, M. Mulder, V. Popov, and M. Chizari. Effects of the drewlite cscl platform on studentsâĂŹ learning outcomes. *Collaborative and distributed e-research: Innovations in technologies, strategies and applications*, pages 276–289, 2012.

[14] S. Stone, B. Pillmore, and W. Cyre. Crossover and mutation in genetic algorithms using graph-encoded chromosomes. *Unpublished*, March 2011.

[15] K. VanLehn, D. Bhembe, M. Chi, C. Lynch, K. Schulze, R. Shelby, L. Taylor, D. Treacy, A. Weinstein, and M. Wintersgill. Implicit versus explicit learning of strategies in a non-procedural cognitive skill. In *International Conference on Intelligent Tutoring Systems*, pages 521–530. Springer, 2004.

[16] Y. Xifeng and H. Jiawei. gspan: Graph-based substructure pattern mining. In *Proceedings of the IEEE International Conference on Data Mining (ICDM 2002)*, pages 721–724. IEEE, 2002.

[17] L. Xue, C. Lynch, and M. Chi. Unnatural feature engineering: Evolving augmented graph grammars for argument diagrams. In T. Barnes, M. Chi, and M. Feng, editors, *EDM*, pages 255–262. IEDMS, 2016.

6