An Economy-based Amendment to Learning Hidden Structure with Robust Interpretive Parsing

Eunsun Jou
Massachusetts Institute of Technology

1 Introduction: Learning in the face of structural ambiguity

Structural ambiguity poses an interesting challenge for the language learner. When a surface form of a linguistic expression is compatible with multiple structural representations, how is the learner able to identify the appropriate representation(s) compatible with the grammar that she is aiming to learn? One approach to investigating this topic is to computationally model the process of learning. Computational modelling facilitates fine-tuning of the algorithm's learning process, thereby enabling a controlled investigation into how the learning result is affected by various factors of the learning process. A pertinent line of research in computational learning of structurally ambiguous expressions makes use of a learning process called Robust Interpretive Parsing (RIP; Tesar & Smolensky 1998, 2000). It is a learning method implemented in error-driven learning algorithms such as Error-Driven Constraint Demotion (EDCD; Tesar & Smolensky 1998) or the Gradual Learning Algorithm (GLA; Boersma 1997).[1] This paper focuses on problems that arise with one specific type of RIP-learning algorithm, which is the constraint-ranking GLA (OT-GLA). Following previous researchers (Boersma & Pater 2008; Jarosz 2013), I call this algorithm the *RIP/OT-GLA*.

Upon observing a linguistic expression, the RIP/OT-GLA reverse-engineers its input and compares the structural representations or *parses* linked to the input. It then identifies the parse that is optimal under its current version of the grammar. The grammar is a ranking of constraints determined by their *ranking values*, a numerical value linked with each constraint. The higher the ranking value, the higher the rank. If the identified parse is not compatible with the actual observed expression, the RIP/OT-GLA registers that it has made an error. It then searches for an alternative parse that is both compatible with the observed expression and as compatible as it can be with its current grammar. Once it has chosen an alternative parse (called the *target parse* in this paper), the RIP/OT-GLA alters the grammar by promoting (*i.e.,* increasing the ranking value of) the constraints that favor the target parse over the erroneous parse, and demoting (*i.e.,* decreasing the ranking value of) constraints that favor the erroneous parse over the target parse. The alteration happens incrementally, but with an adequate number of repetitions of this alteration – which would be triggered every time the learner runs into the expression and makes an error – the ranking between the promoted and demoted constraints will eventually be reversed.

In order to provide a more concrete introduction of RIP, and to facilitate the description of my research question, I borrow from Jarosz (2013) a hypothetical example of a RIP learning situation. The tableau in (1) reflects the grammar of a RIP/OT-GLA learning Polish stress; it is presented with the word [tɛˈlɛfɔn]. Definitions of the relevant constraints are given in (2).

(1) Learner's hypothetical grammar when presented with [tɛˈlɛfɔn] (Jarosz 2013:32)

|tɛlɛfɔn|	All-Feet-Right	Iambic	Trochaic	All-Feet-Left
a. /(ˈtɛlɛ)fɔn/	*	*		
b. /(tɛˈlɛ)fɔn/	*		*	
W c. /tɛ(ˈlɛfɔn)/		*		*
L d. /tɛ(lɛˈfɔn)/			*	*

* I thank Adam Albright, Donca Steriade, and Michael Kenstowicz for many helpful discussions and comments throughout the project. I also thank participants of AMP 2022 for their feedback.

1 See Magri (2012) for an overview of error-driven learning algorithms, and Jarosz (2013) on RIP learning in the context of error-driven learning algorithms.

Proceedings of *AMP 2022*

(2) a. ALL-FEET-RIGHT : Align each foot with the right edge of the word.
b. ALL-FEET-LEFT : Align each foot with the left edge of the word.
c. IAMBIC : The final syllable of a foot must be the head.
d. TROCHAIC: The initial syllable of a foot must be the head.

The learner reverse-engineers the input of the word, which is simply the string of segments without any stress information. In this case the input would be |tɛlɛfɔn|, presented between vertical bars at the upper left corner of the tableau. Candidates a-d in (1) are the potential parses linked to this input. The RIP/OT-GLA calculates which of the candidates best satisfies the current constraint ranking (ALL-FEET-RIGHT ≫ IAMBIC ≫ TROCHAIC ≫ ALL-FEET-LEFT). Among the four candidates, candidate (d) (/tɛ(lɛˈfon)/) best satisfies the ranking and would be identified as the optimal parse. However, there is a problem: the stress pattern of candidate (d) is incompatible with that of the observed word ([tɛˈlɛfɔn]). The former shows final stress while the latter shows penultimate stress. Since there is a discrepancy between the chosen parse and the word, the RIP/OT-GLA registers that it has made an error and tries to choose a target parse that *is* compatible with the observed word. *However, there are two candidates that are compatible with the word*: candidates (b) (/(tɛˈlɛ)fɔn/) and (c) (/tɛ(ˈlɛfɔn)/). Between the two, the RIP/OT-GLA chooses the candidate that better satisfies the current constraint ranking. In this case, that is candidate (c). Once it has chosen the target parse, the RIP/OT-GLA promotes constraints that favor candidate (c) over candidate (d) and demotes constraints that favor candidate (d) over candidate (c). Here, TROCHAIC is promoted and IAMBIC is demoted.

Notice that whether the algorithm chooses candidate (b) or candidate (c) as its target parse crucially determines the direction of learning. Had candidate (b) been chosen, the ranking values of IAMBIC and TROCHAIC would not have changed. Instead, the RIP/OT-GLA would have demoted ALL-FEET-RIGHT and promoted ALL-FEET-LEFT. Depending on which parse it thinks is correct, its direction of learning varies widely. The reason that the RIP/OT-GLA chooses candidate (c) is because it is configured to choose its target parse based on its current constraint ranking (Boersma 1997; Boersma & Hayes 2001). However, this way of choosing the target parse is problematic because the current constraint ranking must be flawed in some way. Had it not been flawed, it would not have incurred an error in the first place.

This problem has been pointed out at a conceptual level by Jarosz (2013), but there have not been empirical demonstrations of the problem arising in actual learning simulations. One of the two main contributions of this paper is to address this gap in the literature. I present results from RIP learning of newly sampled artificial languages, which show evidence that choice of target parse based on constraint ranking can indeed lead to learning failure. The second contribution of the paper is to propose an alternative method of choosing the target parse. This alternative method opts for the most economical change by making use of the *Elementary Ranking Condition* representation (ERC; Prince 2002). The algorithm using this method, which I call the *RIP/ERC-GLA*, calculates for each potential target parse how many rank changes are needed for the offending constraint to be dominated. Then it chooses the target parse which involves the least amount of rank changes. Because the most economical change does not necessarily satisfy the top-ranked constraint, it avoids the pitfall that the original RIP/OT-GLA can fall into.

The remainder of the paper is organized as follows. Section 2 describes the methods used to implement the RIP/OT-GLA and RIP/ERC-GLA. It also introduces the metrical stress system created by Tesar & Smolensky (2000), which I used to create the learning problems for this study. In Section 3 I demonstrate examples of failed learning simulations where the choice of target parse based on the current constraint ranking inhibits successful learning. I propose in Section 4 the RIP/ERC-GLA as an alternative way of choosing the target parse. Section 5 compares the learning results of the RIP/OT-GLA and the RIP/ERC-GLA. It also highlights an advantage of the RIP/ERC-GLA: when the learner is generally on the right track, the RIP/ERC-GLA converges more efficiently with less errors made on the way. Section 6 concludes.

2 Methodology

2.1 *Implementation of the algorithms* For this study, I implemented the RIP/OT-GLA and the RIP/ERC-GLA in the programming language Python.[2] In order to ensure that my implementation of the

[2] The Python implementation of the algorithms, the 66 artificial languages created as learning problems, and the results of individual learning trials (offered as text files and figures) are available in the Github repository found in the following URL: https://github.com/EunsunJou/Economy_RIPGLA

RIP/OT-GLA is faithful to the original, I performed replications of GLA learning from previous studies. I was able to replicate results of learning Ilokano metathesis reported by Boersma & Hayes (2001). I was also able to replicate the learning problem illustrated by Pater (2008), whereby the GLA endlessly promotes the ranking values of the constraints of a specific artificial language. Furthermore, applying Magri's (2012) suggested solution of calibrated promotion to Pater's problem resulted in successful avoidance of the endless promotion. These results strongly suggest that the core learning algorithm first described by Boersma (1997) is successfully replicated in my implementation of the RIP/OT-GLA. Having confirmed this, I implemented the RIP/ERC-GLA by minimally altering the part of the code responsible for the choice of target parse.

2.2 *Creating learning problems* I created a set of artificial languages for the RIP/OT-GLA and RIP/ERC-GLA to learn. These languages were sampled from a system of abstract artificial languages with metrical stress, first created by Tesar & Smolensky (2000). Previous studies on RIP learning, including Boersma & Pater (2008) and Jarosz (2013), have also worked with this system. It is built on the following twelve constraints. Some of the constraint names appear abbreviated in tableaux; these abbreviations are presented in parentheses next to the constraint names.

- FootBinary (FB): Each foot must be either bimoraic or bisyllabic.
- WSP: Weight-to-stress principle. Each heavy syllable must be stressed.
- Parse: Each syllable must be footed.
- Main-Right (MR): Align the head-foot with the word, right edge.[3]
- Main-Left (ML): Align the head-foot with the word, left edge.
- All-Feet-Right (AFR): Align each foot with the word, right edge.
- All-Feet-Left (AFL): Align each foot with the word, left edge.
- Word-Foot-Right (WFR): Align the word with some foot, right edge.
- Word-Foot-Left (WFL): Align the word with some foot, left edge.
- Iambic: Align each foot with its head syllable, right edge.
- Trochaic: Align each foot with its head syllable, left edge.[4]
- Nonfinal (NF): Do not foot the final syllable of the word.

Initially, the constraints are all assigned the same ranking value and hence start out at the same tier. But after the algorithm starts making errors and performs promotions and demotions, the ranking values will change and the constraints will form a ranking. A constraint ranking is compatible with a set of languages.

A language in this system is a set of abstract words, which in turn are abstract inputs plus a stress pattern. The abstract inputs are sequences of light or heavy syllables, represented as L and H respectively. The length of an input is 2 syllables at the shortest, and 7 syllables at the longest. The 2-5 syllable inputs are all possible combinations of L and H, which result in a total of 60 kinds of inputs. In addition to these, there are the 6-syllable and 7-syllable inputs which exclusively consist of L's: |L L L L L L|, and |L L L L L L L|. Each language assigns different stress patterns to its inputs, resulting in a unique set of 62 words. Using a Praat script (Boersma & Weenink 2022), I randomly generated 66 languages that are compatible with some ranking of these constraints. They were labeled *Lang 1*, *Lang 2*, ..., *Lang 66*. With these languages, I now had ample learning problems for the algorithms to solve. For every language, the RIP/OT-GLA and the RIP/ERC-GLA each performed 30 learning trials. This resulted in a total of 3,960 learning results.

[3] What is called Main-Left and Main-Right here are also named Align-Head-Left and Align-Head-Right. See Kager (2007:210) for an example of using Align-Head-Right in explaining data from Cairene Arabic.

[4] Instead of Trochaic, Tesar and Smolensky use the constraint FootNonfinal which requires the head syllable to not only be aligned to the left edge, but to have a syllable follow after it. In other words, a foot such as (H1) would satisfy Trochaic but it would not satisfy FootNonfinal. Their reason for this choice is to reflect a typological asymmetry pointed out by Hayes (1995): trochaic languages may be either quantity sensitive or insensitive, while iambic languages are always quantity sensitive. I judged that this typological generalization is not very relevant for the artificial languages that I have been working with and thus chose the constraint Trochaic, which is the mirror image of Iambic.

Figure 1: Sample text report of a failed learning trial of Lang 55 by the RIP/OT-GLA

```
RIP/OT-GLA learning results
Algorithm: Original Configuration
Grammar changed 1743/62000 times
Plasticity: 1.0
Noise: 2.0
Constraints and ranking values
FootBin 104.8
WSP 102.91666666666666
Main-L 100.33333333333333
WFR 96.83333333333331
WFL 95.83333333333333
Main-R 88.3
Trochaic 88.06666666666666
AFR 86.54999999999994
AFL 85.79999999999993
Nonfinal 84.0
Iambic 82.89999999999998
Parse 77.66666666666664

1 words not (fully) learned in evaluation (target, learned form, learned parse):
[L L1 L] [L1 L L] /(L1 L) L/
```

Other than the method of choosing the target parse, all other parameters were set identically for the two algorithms. Plasticity was set at 1, and noise at 2. I adopted the calibrated promotion of constraints proposed by Magri (2012). In each trial, the algorithm encountered each of the 62 words 1,000 times, amounting to a total of 62,000 individual word tokens seen in each trial. The order of the 62,000 tokens was randomized for each trial. At the end of each trial, there was an evaluation phase where the algorithm produced an output for each input using its final resulting grammar. The algorithm is considered to have succeeded in learning the language if it produces the correct outputs for all 62 inputs. The learner produced a text report of each trial where it printed out the final constraint ranking, the number of grammar changes the algorithm made, whether it succeeded in learning the language, and which words it failed to learn in case it failed. Figure 1 shows the contents of a sample text report of a failed learning trial of Lang 55 by the RIP/OT-GLA.

3 Demonstration of the problem: Choosing the target parse with a losing grammar

Choosing the correct target parse is crucial in determining the trajectory of RIP learning. Among the parses that are compatible with the stress pattern of the observed word, the RIP/OT-GLA chooses the one that best satisfies the current constraint ranking. Jarosz (2013) points out that this way of choosing a target parse is problematic because it is dependent on a "losing grammar". By the point the RIP/OT-GLA registers that it has run into an error, its current constraint ranking is guaranteed to be wrong; otherwise an error would not have occurred. However, it continues to parse the observed stress pattern with this decidedly wrong grammar.

Jarosz acknowledges this problem at a conceptual level, but there have not been empirical demonstrations of this problem actually affecting the results of learning. In this section, I demonstrate actual examples of the RIP/OT-GLA failing to learn a language due to its choice of target parse. The languages of interest are Lang 55 and Lang 3. The full list of words for these languages can be found in the appendix.

3.1 *Case study 1: Lang 55* Out of the 30 learning trials for Lang 55, the RIP/OT-GLA succeeded in 9 of them. We know that Lang 55 is a language with strictly binary, trochaic feet. This is because all of the grammars resulting from the 9 successful trials showed this property: they rank FootBinary at or near the top, and rank Trochaic over Iambic (Trochaic $\gg$ Iambic). Another property that differentiated the successful and failed grammars is the ranking between Word-Foot-Right and Main-Left. In the grammars resulting from the 9 successful trials, Word-Foot-Right was ranked above Main-Left (Word-Foot-Right

≫ Main-Left). For the 21 failed trials, Word-Foot-Right was ranked below Main-Left (Main-Left ≫ Word-Foot-Right).

Now that we have a sense of what characterizes the successful and failed learning trials, let us examine what made the failed trials fail. There was a clear pattern throughout 20 of the 21 failed trials.[5] The algorithm failed to learn the correct stress pattern for one input: |L L L|. The correct realization of |L L L| in Lang 55 is [L L1 L]. (The number 1 corresponds to main or primary stress. The L1 here is the syllable bearing primary stress. Sometimes words also have secondary stress, which is marked with the number 2.) In these failed trials, the RIP/OT-GLA produced [L1 L L], with a left-aligned trochee: /(L1 L) L/. This parse is incompatible with the observed word, since the parse surfaces as initial stress while the actual word bears penultimate stress. Therefore, the RIP/OT-GLA registers that it has made an error and proceeds to choose a target parse.

Figure 2 illustrates this situation with an actual example. It reflects the grammar that emerged at the end of a *failed* learning trial. The numbers in each cell indicate the number of times the constraint of the corresponding column is violated by the candidate of the corresponding row. Candidate (a) is the parse that is wrongly identified as optimal by the RIP/OT-GLA. Candidates (b-d) are the potential target parses whose stress patterns *are* compatible with the observed word.

Figure 2: A tableau for |L L L| from a failed learning trial of Lang 55

|L L L|	FB	WSP	ML	WFR	WFL	MR	Troc	AFR	AFL	NF	Iamb	Parse
☞ a. /(L1 L) L/	0	0	0	1	0	1	0	1	0	0	1	1
b. /(L L1) L/	0	0	0	1	0	1	1	1	0	0	0	1
c. /L (L1 L)/	0	0	1	0	1	0	0	0	1	1	1	1
d. /L (L1) L/	1	0	1	1	1	1	0	1	1	0	0	2

Among candidates (b-d), candidate (b) best satisfies the current ranking. Upon choosing candidate (b) as its target parse, the RIP/OT-GLA promotes constraints that favor candidate (b) over candidate (a) and demotes constraints that favor candidate (a) over candidate (b). This means that Trochaic will be demoted and Iambic will be promoted. But we know from observing the 9 successful trials that for the learning to be successful, Trochaic needs to outrank Iambic. Furthermore, we know that Word-Foot-Right should outrank Main-Left for the learning to be successful. But since candidate (b) and candidate (a) behave identically regarding Word-Foot-Right and Main-Left, the ranking between these two constraints remain as is. In other words, the RIP/OT-GLA focuses on the wrong pair of candidates to work on while the constraints which actually hinder successful learning remain untouched. It should be emphasized that this problem occurs as a result of choosing candidate (b) as the target parse. Choosing a different candidate as the target parse leads to a different and more successful learning result, as will be demonstrated in section 4.

3.2 *Case study 2: Lang 3* Lang 3 is another language that the RIP/OT-GLA had difficulty learning. It failed in 29 out of the 30 trials for this language. Similarly to how it kept failing on the same input for Lang 55, it showed repeated failure for a handful of words. But this time, there were two different sets of words that kept causing it to fail. In 15 of the 29 trials, it failed on the inputs |H L L L| and |H H L L L|. In the remaining 14, it failed on three inputs: |L L L L L|, |L H L L L|, and the seven-syllable |L L L L L L L|. This suggests that there are two distinct groups of wrong grammars that the RIP/OT-GLA produced. In this section I illustrate a problem with the former set of grammars, which produced wrong outputs for |H L L L| and |H H L L L|. Below I present information about the two inputs in Lang 3.

(3) a. Correct output of |H L L L|: [H1 L L L2]
The learner's wrong output: [H1 L L2 L] (parse: /(H1) (L L2) L/)

b. Correct output of |H H L L L|: [H1 H2 L L L2]
The learner's wrong output: [H1 H2 L L2 L] (parse: /(H1) (H2) (L L2) L/)

[5] The one exceptional trial here was what could be called a "crashed" trial. Crashed trials are characterized by perpetual demotion of the ranking values of some constraints during the entire learning trial. At the end of this trial, the ranking values of some constraints plunged under -7,000 and the algorithm never learned 50 out of the 62 words. I do not have an explanation for why these crashes happen, but they are rare enough that they do not get in the way of observing the typical cases of failure. Out of the 29 failed trials of Lang 3 (discussed in section 3.2), there were no crashed trials.

Crucially, the 15 grammars which produced these wrong outputs rank IAMBIC over PARSE (IAMBIC ≫ PARSE), while the correct trial resulted in the opposite ranking (PARSE ≫ IAMBIC).[6] Figures 3a and 3b are tableaux from a failed learning trial of Lang 3.

Figure 3: Tableaux from a failed learning trial of Lang 3

(a) Tableau for |H L L L|

|H L L L|	FB	WSP	ML	WFL	IAMB	PARSE	MR	TROCH	NF	WFR	AFL	AFR
☞ a. /(H1) (L L2) L/	0	0	0	0	0	1	3	1	0	1	1	4
b. /(H1) L (L L2)/	0	0	0	0	0	1	3	1	1	0	2	3
c. /(H1 L) (L L2)/	0	0	0	0	1	0	2	1	1	0	2	2
d. /(H1 L) L (L2)/	1	0	0	0	1	1	2	0	1	0	3	2
e. /(H1) L L (L2)/	1	0	0	0	0	2	3	0	1	0	3	3

(b) Tableau for |H H L L L|

|H H L L L|	FB	WSP	ML	WFL	IAMB	PARSE	MR	TROCH	NF	WFR	AFL	AFR
☞ a. /(H1) (H2) (L L2) L/	0	0	0	0	0	1	4	1	0	1	3	8
b. /(H1) (H2) L (L L2)/	0	0	0	0	0	1	4	1	1	0	4	7
c. /(H1) (H2 L) (L L2)/	0	0	0	0	1	0	4	1	1	0	4	6
d. /(H1) (H2) L L (L2)/	1	0	0	0	0	2	4	0	1	0	5	7
e. /(H1) (H2 L) L (L2)/	1	0	0	0	1	1	4	0	1	0	5	6

In both tableaux, candidate (a) is the parse erroneously identified by the RIP/OT-GLA as optimal under the current constraint ranking. Candidates (b-e) are the potential target parses. Candidates (d) and (e) can be immediately ruled out due to their fatal violation of FOOTBINARY. Between candidates (b) and (c), the algorithm chooses candidate (b) because of the violation of IAMBIC by candidate (c). Once it chooses the target parse as candidate (b), it promotes constraints that favor candidate (b) over candidate (a) while demoting constraints that favor candidate (a) over candidate (b). In this case, the former are WORD-FOOT-RIGHT and ALL-FEET-RIGHT while the latter are NONFINAL and ALL-FEET-LEFT. But we know from the successful learning trial that the crucial change that needs to be made is to flip the ranking between IAMBIC and PARSE. However, because candidate (a) and candidate (b) behave identically with regards to IAMBIC and PARSE, these two constraints are never reranked. Hence Lang 3 demonstrates another example where the RIP/OT-GLA fails to find the correct constraint ranking due to its choice of target parse.

4 An alternative choice of target parse: Opting for the most economical change

In this section I propose an alternative method of choosing a target parse. This method makes use of the Elementary Ranking Condition (ERC) representation, also known as comparative tableaux (Prince 2002). Instead of choosing the target parse that satisfies the higher ranked constraints, an algorithm using this method chooses the candidate parse which, if chosen as the target, would need the least number of rerankings to accommodate for. The number of rerankings is measured using the ERC representation. I call this alternative method the *ERC method*, and I call the algorithm that uses this method the *RIP/ERC-GLA*. In section 4.1 I briefly explain the ERC representation, and how it can be used to measure the amount of rank changes needed to make the grammar favor one parse over the other. I then explain in section 4.2 how I applied this logic to my implementation of the ERC method.

4.1 *The Elementary Ranking Condition* The objective of error-driven learning is to change the grammar so that the resulting grammar does not produce the wrong output (the error) but instead produces the correct output. Let us call the wrong output the *loser*, and the correct output the *winner*. The ERC

[6] The reader may rightly be hesitant to accept that PARSE ≫ IAMBIC is a correct characterization of Lang 3, since there is only one successful learning trial to draw this conclusion from. However, the 17 successful trials with the RIP/ERC-GLA also all ranked PARSE over IAMBIC. See section 5.1.

representation classifies constraints into three classes: *loser-preferring*, *winner-preferring*, and *even*. A loser-preferring constraint is one which is violated more by the winner than the loser. It might be the case that only the winner violates this constraint, or in the case of gradiently violable constraints the winner violates the constraint more times than the loser does. A winner-preferring constraint, similarly, is a constraint that is violated more by the loser than the winner. An even constraint is violated the same amount by both constraints. The loser-preferring, winner-preferring, and even constraints are marked L, W, and e respectively.

This classification is based on a winner-loser pair. It is often the case that more than one potential winner is considered against the loser. In this case, the ERC representation is created for each winner-loser pairing. In figure 4 I present the ERC representations of the tableau of a failed trial of Lang 55, with the tableau itself repeated for reference. The line labels of the ERC representations always correspond to the winner's label in the tableau. For example, line (b) in figure 4 is the ERC representation of the winner-loser pair where the winner is candidate (b). In the same figure, line (c) is the ERC representation of the winner-loser pair where the winner is candidate (c). The loser is always candidate (a), the wrongly identified parse that is incompatible with the observed word.

Figure 2: A tableau for |L L L| from a failed learning trial of Lang 55 (repeated)

|L L L|	FB	WSP	ML	WFR	WFL	MR	TROC	AFR	AFL	NF	IAMB	PARSE
☞ a. /(L1 L) L/	0	0	0	1	0	1	0	1	0	0	1	1
b. /(L L1) L/	0	0	0	1	0	1	1	1	0	0	0	1
c. /L (L1 L)/	0	0	1	0	1	0	0	0	1	1	1	1
d. /L (L1) L/	1	0	1	1	1	1	0	1	1	0	0	2

Figure 4: ERC representations of the tableau of Lang 55

	FB	WSP	ML	WFR	WFL	MR	TROC	AFR	AFL	NF	IAMB	PARSE
b. ∼ /(L L1) L/	e	e	e	e	e	e	L	e	e	e	W	e
c. ∼ /L (L1 L)/	e	e	L	W	L	W	e	W	L	L	e	e
d. ∼ /L (L1) L/	L	e	L	e	L	e	e	e	L	e	W	L

A benefit of using the ERC representation is that it makes it clear which constraints should be reranked in order to accommodate the potential winner. For each winner-loser pair, it is crucial that the highest-ranked or *undominated* L (which is always the leftmost L in an ERC representation) be outranked by a W. Consider first candidate (b), or /(L L1) L/. In order to change the current ranking to one that would produce candidate (b) as the winner, IAMBIC needs to outrank TROCHAIC. Once IAMBIC outranks TROCHAIC, /(L L1) L/ would be the optimal parse under the new constraint ranking. Similarly, for candidate (c), it is important that MAIN-LEFT be outranked by some winner-preferring constraint such as WORD-FOOT-RIGHT or MAIN-RIGHT.

4.2 *Identifying the most economical change with the ERC* While any W can be promoted to outrank an L in an ERC representation, the most economical change would be to promote the highest W. For candidate (c) in figure 4 this would mean promoting WORD-FOOT-RIGHT. For the other two candidates, there is only one W. Therefore, promoting that W is automatically the most economical change. The ERC method I propose here compares the *LW distance* of the ERC representations of each winner-loser pair and chooses the pair with the shortest LW distance. I define LW distance as the number of constraints the highest W would have to overcome for it to outrank the undominated L. For candidate (b), the LW distance is 4 since IAMBIC needs to switch rankings with four constraints above it (NONFINAL, ALL-FEET-LEFT, ALL-FEET-RIGHT, TROCHAIC) in order to outrank TROCHAIC. For candidate (c), the LW distance is 1 since WORD-FOOT-RIGHT only needs to overcome MAIN-LEFT. Lastly, the LW distance for candidate (d) is 10.

Since the LW distance of candidate (c) is shortest, candidate (c) is chosen as the target parse. As a result, the RIP/ERC-GLA promotes constraints that prefer candidate (c) over candidate (a), and demotes constraints that prefer candidate (a) over candidate (c). Specifically, WORD-FOOT-RIGHT, MAIN-RIGHT, ALL-FEET-RIGHT are promoted and MAIN-LEFT, WORD-FOOT-LEFT, ALL-FEET-LEFT, and NONFINAL are demoted.

What is important is that Word-Foot-Right is promoted and Main-Left is demoted. Since Word-Foot-Right is ranked above Main-Left in all of the 9 successful learning trials of Lang 55, this is indeed a step in the right direction. Compare this result with that of the RIP/OT-GLA, which would choose candidate (b) as its target parse. Once this candidate is chosen as the target, only Iambic will be promoted and Trochaic will be demoted. Since the problematic ranking of Main-Left $\gg$ Word-Foot-Right is never fixed, the RIP/OT-GLA fails to change the grammar in a desirable direction.

A similar improvement was observed for Lang 3 with the ERC method. Figure 5 shows the ERC representations for |H L L L| and |H H L L L| from Lang 3.

Figure 5: ERC representations for |H L L L| and |H H L L L| in Lang 3

(a) ERC representations for |H L L L|

	FB	WSP	ML	WFL	Iamb	Parse	MR	Troc	NF	WFR	AFL	AFR
b. ~ /(H1) L (L L2)/	e	e	e	e	e	e	e	e	L	W	L	W
c. ~ /(H1 L) (L L2)/	e	e	e	e	L	W	W	e	L	W	L	W
d. ~ /(H1 L) L (L2)/	L	e	e	e	L	e	W	W	L	W	L	W
e. ~ /(H1) L L (L2)/	L	e	e	e	e	L	e	W	L	W	L	W

(b) ERC representations for |H H L L L|

	FB	WSP	ML	WFL	Iamb	Parse	MR	Troc	NF	WFR	AFL	AFR
b. ~ /(H1) (H2) L (L L2)/	e	e	e	e	e	e	e	e	L	W	L	W
c. ~ /(H1) (H2 L) (L L2)/	e	e	e	e	L	W	e	e	L	W	L	W
d. ~ /(H1) (H2) L L (L2)/	L	e	e	e	e	L	e	W	L	W	L	W
e. ~ /(H1) (H2 L) L (L2)/	L	e	e	e	L	e	e	W	L	W	L	W

In figure 5a, the LW distance of candidate (b) is 1, since Word-Foot-Right needs to outrank Nonfinal right above it. The LW distance of candidate (c) is also 1, although this time it is because Parse needs to outrank Iambic. For candidate (d), the LW distance is 6: Main-Right needs to overcome six constraints above it to be able to dominate the L of FootBinary. Lastly, the LW distance of candidate (e) is 7. In figure 5b, the LW distance of candidates (b) and (c) are again 1. The LW distance for candidates (d) and (e) are 7 since Trochaic needs to overcome seven constraints. Under the ERC method, both |H L L L| and |H H L L L| have two winners with the shortest LW distance: candidates (b) and (c). Whenever there is a tie like this, the RIP/ERC-GLA is configured to randomly choose one of the two. If candidate (b) is chosen, Word-Foot-Right and All-Feet-Right are promoted while Nonfinal and All-Feet-Left are demoted. If candidate (c) is chosen, the algorithm additionally demotes Iambic and promotes Parse (and Main-Right for |H L L L|).

It was pointed out in section 3.2 that all successful constraint rankings compatible with all of the words of Lang 3 rank Parse over Iambic. Notice that the RIP/ERC-GLA is able to make the correct change by choosing candidate (c) as its target parse, at least roughly half of the times it runs into an error for |H L L L| and |H H L L L|. Compare this situation with that of the RIP/OT-GLA, where the target parse will always be candidate (b). As a result, the RIP/OT-GLA never has a chance to register the ranking Iambic $\gg$ Parse as problematic, and fails to ever change it. Hence the ERC method makes it possible for the RIP/ERC-GLA to choose a target parse that actually pushes it in the right direction. In the following section I discuss how this alternative algorithm affected the learning results of the 66 artificial languages I have sampled in this study.

5 Results and discussion

5.1 *Similar overall performance, but improvement in Lang 55 and Lang 3* For the 66 artificial languages sampled in this study, the RIP/OT-GLA and the RIP/ERC-GLA each performed 30 learning trials. As presented in figure 1, each trial was followed by an evaluation phase which reported the final constraint ranking, the number of grammar changes made, whether it succeeded in learning the language, and which words it failed to learn in case it failed.

Table 1 summarizes the result of learning across all 66 languages for the two algorithms.[7] The overall performances of the RIP/OT-GLA and the RIP/ERC-GLA are similar. The RIP/OT-GLA succeeded in learning an average of 23.03 out of 30 trials per language, while the RIP/ERC-GLA succeeded in an average of 23.65 trials.

Table 1: Mean number of successful trials (out of 30) across all languages

	RIP/OT-GLA	RIP/ERC-GLA
Mean	23.03 (76.77%)	23.65 (78.83%)
Standard Deviation	9.66	6.46

While the ERC method did not improve overall performance, it clearly improved the success rate in learning the languages of interest, Lang 55 and Lang 3. The RIP/OT-GLA made 9 successful trials of learning Lang 55 and only one successful trial learning Lang 3. In contrast, the RIP/ERC-GLA made 26 successful trials for Lang 55 and 17 successful trials for Lang 3.

Table 2: Number of successful trials (out of 30) for Lang 55 and Lang 3

	RIP/OT-GLA	RIP/ERC-GLA
Lang 55	9	26
Lang 3	1	17

These results suggest that there are multiple factors hindering the success of the RIP/OT-GLA. One of these factors is the choice of target parse. While adopting the ERC method does not bring about a drastic improvement for all languages, the boost in performance for Lang 55 and Lang 3 indicates that the choice of target parse is one of the prominent problems of the RIP/OT-GLA that can be addressed by adopting this method.

5.2 *More efficient convergence with the RIP/ERC-GLA* The desideratum of the ERC method is to produce the correct stress pattern by undergoing the most economical change. It chooses as its target parse the parse that involves the least amount of change for the critical constraint (the highest W in ERC representation). It is then perhaps no surprise that the RIP/ERC-GLA converged more efficiently than the RIP/OT-GLA, where efficiency is indicated by the number of grammar changes performed until convergence. Both algorithms record how many times a grammar change occurred during a trial. A grammar change occurs anytime the algorithm registers an error, and performs promotion and demotion of ranking values. Since it encounters 62,000 words in each trial, a learner could in theory change the grammar up to 62,000 times if it incurs an error for every word it encounters. Of course, the actual number of grammar changes are much lower than that; but the numbers still range from 14 to 30,233.

As an approximate measure of efficiency, I collected for each language the median of the number of grammar changes of all successful trials by the RIP/OT-GLA and the RIP/ERC-GLA.[8] For example, the RIP/OT-GLA performed 27 successful learning trials of Lang 6. The number of grammar changes ranged from 113 to 1,465 for the 27 trials, and the median was 241. The RIP/ERC-GLA had 23 successful learning

[7] These reported performances are quite higher than the 58.95% reported by Boersma & Pater (2008) for their noisy RIP/OT-GLA. I attribute this difference to the difference in the set of languages learned. The large standard deviation indicates that the difficulty of a language varies widely for the RIP/OT-GLA – some languages are learned more easily, while others are much harder to learn. The learning results are affected by the kind of language that the learner is presented with. I was not able to obtain information about the 124 constructed languages used by Boersma and Pater, but it is quite plausible that their set of languages and my set of 66 randomly sampled languages do not have much overlap between them.

[8] The presence of outliers makes the median a more appropriate measure than the mean. Due to the stochastic nature of the GLA, it sometimes takes the algorithms a significantly larger amount of grammar changes than typical to achieve convergence. In learning Lang 62, for example, the RIP/OT-GLA typically converged after 200–300 grammar changes. But in one trial, it needed 3,363 changes before converging. In this case, the mean of the grammar changes for the 25 trials amounts to 392.64 while the median is 223, a much truer representation of the actual number of grammar changes.

trials, with grammar changes ranging from 115 to 339. The median was 153. Since the median was lower for the RIP/ERC-GLA, I deemed the ERC method as producing more efficient convergence for Lang 6.

Table 3 shows the results of the comparison. There are 60 languages accounted for here. Languages 13, 34, and 35 are excluded because either one or both algorithms did not have any successful trials, thus making a comparison impossible. Languages 14, 19, and 59 are excluded because the two algorithms showed the same median value for these languages.

Table 3: Comparison of median number of grammar changes

(a) Languages that showed lower median of changes by the RIP/ERC-GLA ($n = 41$)

Language	OT	ERC	Difference	Language	OT	ERC	Difference
Lang 01	119	118.5	0.5	Lang 33	5426	100	5326
Lang 02	414	147	267	Lang 37	1729	148	1581
Lang 03	1279	145	1134	Lang 38	2915	157	2758
Lang 05	160	158	2	Lang 41	451.5	43	408.5
Lang 06	241	153	88	Lang 42	50	49.5	0.5
Lang 07	189.5	187.5	2	Lang 44	9723	44	9679
Lang 09	141	133.5	7.5	Lang 45	97.5	97	0.5
Lang 10	105	97	8	Lang 46	90	86	4
Lang 11	131	89	42	Lang 48	248	229.5	18.5
Lang 12	57	54	3	Lang 49	2963	103	2860
Lang 15	3500	129	3371	Lang 50	96	91	5
Lang 17	53	51	2	Lang 51	51.5	49	2.5
Lang 18	146.5	90.5	56	Lang 52	274	104.5	169.5
Lang 20	170	107	63	Lang 54	280	88	192
Lang 21	5478	159	5319	Lang 55	231	105	126
Lang 22	5348.5	39	5309.5	Lang 57	184	168	16
Lang 24	93	83	10	Lang 58	232	221.5	10.5
Lang 25	16.5	16	0.5	Lang 62	223	214	9
Lang 27	954	189.5	764.5	Lang 63	141	129	12
Lang 29	3061	106.5	2954.5	Lang 64	117	100	17
Lang 32	132	129	3	**Mean difference**			**1039.09**

(b) Languages that showed lower median of changes by the RIP/OT-GLA ($n = 19$)

Language	OT	ERC	Difference	Language	OT	ERC	Difference
Lang 04	147	159	12	Lang 40	62.5	63	0.5
Lang 08	82.5	84	1.5	Lang 43	145	155	10
Lang 16	187	198.5	11.5	Lang 47	89	102	13
Lang 23	130	180	50	Lang 53	142.5	151.5	9
Lang 26	50	52	2	Lang 56	75	79	4
Lang 28	177	184	7	Lang 60	109	121	12
Lang 30	115	122	7	Lang 61	49	54.5	5.5
Lang 31	133.5	146	12.5	Lang 65	100	104	4
Lang 36	63	67.5	4.5	Lang 66	140.5	190	49.5
Lang 39	62	68	6	**Mean difference**			**11.66**

Among the 60 languages compared, 41 of them showed a lower median under the ERC method. The mean difference of the median grammar changes for these languages is 1039.09. Roughly speaking, it took the RIP/OT-GLA about 1039.09 more grammar changes than the RIP/ERC-GLA to converge on the correct grammar. It is clear that for these languages, the ERC method led to more efficient convergence. The situation is quite different for the 19 languages which showed a smaller median grammar change with the RIP/OT-GLA. It may seem like in these languages, the RIP/OT-GLA converges more efficiently than the RIP/ERC-GLA. However, the mean difference of the median grammar changes for these languages is 11.66. In other words, the it took the RIP/ERC-GLA on average 11.66 more grammar changes to converge than the RIP/OT-GLA. This seems a rather insignificant difference against a backdrop of 62,000 potential opportunities for grammar change, and against the much larger difference of 1039.09 in the other group. Given that the number of grammar changes can differ on the order of 100 even within the same language under the same configuration, this difference seems more like a tie rather than a significantly more efficient convergence.

The ERC method by itself does not enhance the success rate of RIP learning, but it does help the algorithm reach its goal more quickly when the learner is generally on the right track. Since the ERC method is designed to opt for efficiency, it is not surprising that it leads to efficient convergence. However, the magnitude of the difference is telling. When the ERC method leads to more efficient convergence, it does so by an average of 1039 grammar changes. The median grammar change of Languages 21, 22, and 33 are above 5,000 for the RIP/OT-GLA but 159, 39, and 100 respectively for the RIP/ERC-GLA. This drastic decrease in the number of grammar changes suggests that there is a factor which contributes to inefficient learning for the RIP/OT-GLA that is avoided by the RIP/ERC-GLA.

6 Conclusion

This paper demonstrated a problem that arises under the original configuration of the RIP/OT-GLA. When the learner registers that it has made an error, it needs to choose an alternative parse of the word that it is currently faced with. The choice of the target parse determines which constraints are promoted and demoted, hence affecting the direction of learning. In choosing the target parse, the learner chooses the one which best satisfies the current constraint ranking. In other words, the parse which is favored by the high-ranked constraints is chosen as the target. But this is problematic because the constraint ranking is necessarily flawed – if it was not flawed, it would not have incurred an error. This problem was previously recognized by Jarosz (2013), but at a conceptual level. This paper demonstrates that the problem is real. It discusses the case of Lang 55 and Lang 3, where the OT-based choice of target parse hinders the learner from changing the constraint rankings that crucially need to be changed in order to reach success. As an alternative method of choosing the target parse, I suggested the ERC method. Implemented in the algorithm RIP/ERC-GLA, the ERC method rewards economical change. Among the potential target parses, the learner chooses the parse which needs the least amount of rerankings for it to surface as the winner. Since the most economical change does not always mean satisfying the high-ranked constraints, this method is less dependent on the current, flawed constraint ranking. While the ERC method is not a cure-all for the problems of RIP learning, it certainly solves the problem that it is designed to solve. The learner showed significantly improved learning results for Lang 55 and Lang 3. Furthermore, it contributes to more efficient convergence. The RIP/ERC-GLA reaches convergence after a much smaller number of grammar changes.

Appendix: Full word list of Lang 55 and Lang 3

The number 1 corresponds to primary stress, and the number 2 to secondary stress.

Table 4: Full list of words for Lang 55

[L1 L]	[H1 H2 H2]	[H1 L H2 L]	[L1 L H2 L H2]	[H1 L L L2 L]	[H1 H2 L H2 H2]
[L H1]	[L1 L L2 L]	[H1 L H2 H2]	[L1 L H2 H2 L]	[H1 L L L H2]	[H1 H2 H2 L2 L]
[H1 L]	[L1 L L H2]	[H1 H2 L2 L]	[L1 L H2 H2 H2]	[H1 L L H2 L]	[H1 H2 H2 L H2]
[H1 H2]	[L1 L H2 L]	[H1 H2 L H2]	[L H1 L L2 L]	[H1 L L H2 H2]	[H1 H2 H2 H2 L]
[L L1 L]	[L1 L H2 H2]	[H1 H2 H2 L]	[L H1 L L H2]	[H1 L H2 L2 L]	[H1 H2 H2 H2 H2]
[L1 L H2]	[L H1 L2 L]	[H1 H2 H2 H2]	[L H1 L H2 L]	[H1 L H2 L H2]	[L1 L L L L2 L]
[L H1 L]	[L H1 L H2]	[L1 L L L2 L]	[L H1 L H2 H2]	[H1 L H2 H2 L]	[L1 L L L L L2 L]
[L H1 H2]	[L H1 H2 L]	[L1 L L L H2]	[L H1 H2 L2 L]	[H1 L H2 H2 H2]	
[H1 L2 L]	[L H1 H2 H2]	[L1 L L H2 L]	[L H1 H2 L H2]	[H1 H2 L L2 L]	
[H1 L H2]	[H1 L L2 L]	[L1 L L H2 H2]	[L H1 H2 H2 L]	[H1 H2 L L H2]	
[H1 H2 L]	[H1 L L H2]	[L1 L H2 L2 L]	[L H1 H2 H2 H2]	[H1 H2 L H2 L]	

Table 5: Full list of words for Lang 3

[L L1]	[H1 H2 H2]	[H1 L H2 L]	[L L1 H2 L H2]	[H1 L L2 L L2]	[H1 H2 L H2 H2]
[L H1]	[L L1 L L2]	[H1 L H2 H2]	[L L1 H2 H2 L]	[H1 L L2 L H2]	[H1 H2 H2 L L2]
[H1 L]	[L L1 L H2]	[H1 H2 L L2]	[L L1 H2 H2 H2]	[H1 L L2 H2 L]	[H1 H2 H2 L H2]
[H1 H2]	[L L1 H2 L]	[H1 H2 L H2]	[L H1 L L2 L]	[H1 L L2 H2 H2]	[H1 H2 H2 H2 L]
[L L1 L]	[L L1 H2 H2]	[H1 H2 H2 L]	[L H1 L L2 H2]	[H1 L H2 L L2]	[H1 H2 H2 H2 H2]
[L L1 H2]	[L H1 L L2]	[H1 H2 H2 H2]	[L H1 L H2 L]	[H1 L H2 L H2]	[L L1 L L2 L L2]
[L H1 L]	[L H1 L H2]	[L L1 L L2 L]	[L H1 L H2 H2]	[H1 L H2 H2 L]	[L L1 L L2 L L2 L]
[L H1 H2]	[L H1 H2 L]	[L L1 L L2 H2]	[L H1 H2 L L2]	[H1 L H2 H2 H2]	
[H1 L L2]	[L H1 H2 H2]	[L L1 L H2 L]	[L H1 H2 L H2]	[H1 H2 L L L2]	
[H1 L H2]	[H1 L L L2]	[L L1 L H2 H2]	[L H1 H2 H2 L]	[H1 H2 L L2 H2]	
[H1 H2 L]	[H1 L L2 H2]	[L L1 H2 L L2]	[L H1 H2 H2 H2]	[H1 H2 L H2 L]	

References

Boersma, Paul (1997). How we learn variation, optionality, and probability. *Proceedings of the Institute of Phonetic Sciences* 21, 43–58.

Boersma, Paul & Bruce Hayes (2001). Empirical tests of the gradual learning algorithm. *Linguistic Inquiry* 32:1, 41–86.

Boersma, Paul & Joe Pater (2008). Convergence properties of a gradual learning algorithm for harmonic grammar. *Ms.* .

Boersma, Paul & David Weenink (2022). Praat: doing phonetics by computer [Computer program]. URL `http://www.praat.org/`. Version 6.2.12, retrieved 17 April 2022.

Hayes, Bruce (1995). *Metrical Stress Theory: Principles and Case Studies*. University of Chicago Press.

Jarosz, Gaja (2013). Learning with hidden structure in optimality theory and harmonic grammar: Beyond robust interpretive parsing. *Phonology* 30, 27–71.

Kager, René (2007). Feet and metrical stress. de Lacy, Paul (ed.), *The Cambridge Handbook of Phonology*, Cambridge University Press, 195–227.

Magri, Giorgio (2012). Convergence of error-driven ranking algorithms. *Phonology* 29, 213–269.

Pater, Joe (2008). Gradual learning and convergence. *Linguistic Inquiry* 39:2, 334–345.

Prince, Alan (2002). Entailed ranking arguments. Ms. Available as ROA-500 from the Rutgers Optimality Archive.

Tesar, Bruce & Paul Smolensky (1998). Learnability in Optimality Theory. *Linguistic Inquiry* 29:2, 229–268.

Tesar, Bruce & Paul Smolensky (2000). *Learnability in Optimality Theory*. MIT Press.