Prepublication Draft, January, 2013
Slightly revised version published 2013, Phonology 30(1), 27-71.

Learning with Hidden Structure in Optimality Theory and Harmonic Grammar: Beyond Robust Interpretive Parsing*

0. Abstract

This paper explores the relative merits of constraint ranking versus weighting in the context of a major outstanding learnability problem in phonology: learning in the face of hidden structure. Specifically, the paper examines a well-known approach to the structural ambiguity problem, Robust Interpretive Parsing (RIP; Tesar and Smolensky 1998), focusing on its stochastic extension as first described by Boersma (2003). Two related problems with the stochastic formulation of RIP are revealed, rooted in a failure to take full advantage of probabilistic information available in the learner's grammar. To address these problems, two novel parsing strategies are introduced and applied to learning algorithms for both probabilistic ranking and weighting. The novel parsing strategies yield significant improvements in performance, asymmetrically improving performance of OT learners. Once RIP is replaced with the proposed modifications, the apparent advantage of HG over OT learners reported in previous work disappears (Boersma and Pater to appear).

1. Introduction

Much recent work in phonology, both theoretical and computational, has explored the consequences of replacing the strict ranking of Optimality Theory (OT; Prince and Smolensky 2004) with numerical weighting. Research on weighted grammars, including Harmonic Grammar (HG) (Legendre, Miyata and Smolensky 1990; Smolensky and Legendre 2006), Linear OT (Keller 2000), and Maximum Entropy Grammars (Johnson 2002; Goldwater and Johnson 2003; Jäger 2007) has identified a number of interesting properties of constraint weighting. One property of weighted grammars that has received much attention is their ability to model cumulative effects. On the one hand, proponents argue that constraint weighting elegantly captures gang interactions and other attested cumulative effects (Keller 2000; Keller and Asudeh 2002; Goldwater and Johnson 2003; Jäger and Rosenbach 2006; Coetzee and Pater 2008a; Pater 2009a; Pater 2009b; Potts et al. 2010). On the other hand, while some work has highlighted the ways in which weighting predicts relatively restricted typologies (Pater 2009a; Pater 2009b; Potts et al. 2010), other work has shown that the added power of weighting can result in unusual typological over-predictions (Legendre, Sorace and Smolensky 2006; Pater 2009b; Bane and Riggle to appear). There has also been a discussion of the computational properties of ranked versus weighted constraint grammars. From the perspective of acquisition modelling, Jesney and Tessier have argued that gradual learning of weighted constraint grammars can capture attested intermediate stages that ranking cannot (Jesney and Tessier 2011). Other work suggests a learnability advantage for stochastically

* This work has benefitted from discussion with a number of colleagues, including Joe Pater, Paul Boersma, Paul Smolenksy, Colin Wilson, Jason Riggle, John McCarthy, Bob Frank, and Jeff Heinz. I have also received valuable comments on portions of this work presented to audiences at NECPhon, UMass, Mayfest, U. Del. Workshop on Stress and Accent, and the Yale Computational Linguistics research group (CLAY). Finally, I would also like to thank three anonymous reviewers and the associate editor for very thorough and thoughtful comments on an earlier version of this paper.

weighted over stochastically ranked constraints (Boersma and Pater to appear) and highlights the link between weighted grammars and connectionist models in cognitive science (Legendre, Sorace and Smolensky 2006; Soderstrom, Mathis and Smolensky 2006; Pater 2009a; Goldrick 2011). However, there is also work showing that, at least for some key learning sub problems, there is no learnability advantage of weighting over ranking (Bane, Riggle and Sonderegger 2010; Magri 2012).

The present work contributes to this debate by exploring the relative merits of ranking and weighting in the context of a major outstanding learnability problem in phonology: the problem of learning in the face of hidden structure. Specifically, this paper examines a well-known approach to the structural ambiguity problem in OT, Robust Interpretive Parsing (RIP; Tesar and Smolensky 1998), focusing on its extension to probabilistic constraint-based grammars as first described by Boersma (2003). The RIP extensions of online learning algorithms for both probabilistic OT (Boersma 1997, Boersma and Hayes 2001) and for probabilistic HG (Fischer 2005; Jäger 2007; Pater 2009a; Pater 2009b; Boersma and Pater to appear) are analysed in the domain of metrical stress. Initial investigations of OT and HG learners in this context suggest an advantage for HG (Boersma and Pater to appear). These simulations, reviewed below (Section 2.4), serve as a starting point for the in-depth investigations in the present paper. The paper also introduces two novel interpretive parsing strategies that are applied to both OT and HG learning algorithms and presents in-depth analyses of the performance of all six learning algorithms[1]. The overall findings show that the advantage of HG over OT reported in previous work disappears when RIP is replaced with the proposed parsing strategies. Although the success rates of HG learners are substantially higher than that of OT learners when the original formulation of RIP is used, the proposed learning algorithms substantially and cumulatively improve the performance of OT learners, ultimately yielding comparable success rates for OT and HG learners. The analyses reveal the underlying causes of the performance differences between the ranking and weighting frameworks, and computational and theoretical implications are discussed.

The remainder of the paper is laid out as follows. Section 2 reviews previous work on learning stress and hidden structure, the OT and HG frameworks, and relevant previous work on RIP and its application in the stochastic setting. Section 3 introduces the first problem with the original formulation of RIP for the stochastic setting, parsing with a losing grammar. It also introduces a novel parsing strategy that solves this problem, Resampling RIP (RRIP), and presents simulations exploring the performance of RIP and RRIP for both OT and HG learners. Section 4 introduces the second problem with the original formulation of RIP for the stochastic setting, the parsing-production mismatch. This section also proposes the second novel parsing strategy, Expected Interpretive Parsing (EIP), and compares the performance of all three parsing strategies for both OT and HG learners. Finally, Sections 5 and 6 present final discussion and concluding remarks, respectively.

2. Background: Hidden Structure and Robust Interpretive Parsing

The present investigations of error-driven constraint-based learners falls within a rich and growing literature on the learning of stress and hidden structure in phonology. There has been work on the learning of hidden metrical structure within a principles-and-parameters (Chomsky 1981) framework (Dresher and Kaye 1990; Dresher 1999; Pearl 2011). Earlier work also includes learning of surface stress patterns using connectionist networks (Gupta and Touretzky 1994), dynamic systems (Goldsmith 1994), data-driven learning (Daelemans, Gillis and Durieux 1994), and automata-theoretic approaches (Heinz 2009). Within OT and

[1] See Biró (to appear) for forthcoming research on an alternative approach to improved interpretive parsing.

related constraint-based frameworks, there is an extensive body of work on the learning of hidden structure (Tesar 1997b; Tesar et al. 2003; Tesar 2004b; Prince and Smolensky 2004; Tesar 2004a; Alderete et al. 2005; Jarosz 2006a; Jarosz 2006b; Tesar 2006a; Tesar 2006b; Merchant 2008; Merchant and Tesar 2008; Tesar 2008; Tesar 2009; Akers 2011; Jarosz to appear a). The RIP approach to structural ambiguity investigated here is one prominent strand of learnability research from this constraint-based perspective. RIP has been applied to both strict and probabilistic variants of both ranking and weighting frameworks (Tesar and Smolensky 1998; Tesar and Smolensky 2000; Boersma 2003; Apoussidou and Boersma 2003; Apoussidou 2007; Biró to appear; Boersma and Pater to appear), and RIP has also been adapted to the learning of hidden lexical representations (Apoussidou 2006; Apoussidou 2007). Although the constraint-based learning models rely on domain-general learning strategies, many applications of RIP to the structural ambiguity problem have been tested in the domain of metrical phonology, and this is the domain used to exemplify the models developed in the present work.

RIP is an extension of constraint-based learning models that are playing an increasingly pivotal role in the literature outside of learnability proper, including much research on variation (see Coetzee and Pater 2008b for a review), gradience and phonotactics (Keller 2000; Hammond 2004; Coetzee and Pater 2008a; Hayes and Wilson 2008; Martin 2011, and many more), acquisition modelling (see e.g. Smolensky 1996; Boersma and Levelt 2000; Hayes 2004; Tessier 2009; Jarosz 2010; Jesney and Tessier 2011), and inductive bias (Hayes and Londe 2006; Wilson 2006; Hayes et al. 2008; Daland et al. 2011, among others). The prominent role that constraint-based learning has played in the phonological literature is an important motivation for the current investigations into how such learners can be extended to successfully cope with structural ambiguity. Successful strategies for handling structural ambiguity have the potential to enrich and broaden the scope of research in each of the areas above by making accessible the modelling of empirical domains that rely on hidden structure or that interact with hidden structure. The RIP approach differs from several of the earlier approaches mentioned above (Dresher and Kaye 1990; Goldsmith 1994; Dresher 1999; Heinz 2009) in providing a fully general approach to structural ambiguity that divorces the learning strategy from the substantive content of the phonological grammar. Like most other OT learning models, the RIP approach relies on the architecture of the grammar but is not tied to specific representations, constraints, or empirical domains (see Tesar 2004b for extensive discussion along these lines). This means that, while the present paper exemplifies the approach in a test system of metrical phonology, the proposed learning strategies can be applied to any case of structural ambiguity. In other words, the learning challenge undertaken by these approaches is a more general learnability problem that goes well beyond metrical phonology. Phonological theory posits various abstract representations to explain systematic regularities underlying surface patterns, including feet, syllable structure, moras, autosegments, and other prosodic structure. Metrical phonology is one domain in which such structure plays a central role, but it is important to keep in mind that further developments of constraint-based learning in this domain contribute to a deeper understanding of the challenges posed by phonological learning more generally.

The remainder of this section reviews the Stochastic OT and Noisy HG frameworks, as well as the error-driven learning approach inherent to the Gradual Learning Algorithms for ranking and weighting (Boersma 1997; Boersma and Hayes 2001; Boersma and Pater, to appear). It then explains how hidden structure poses a challenge for these learners (for related discussion see Tesar 2004b). Finally, it presents the Robust Interpretive Parsing approach to structural ambiguity, reviewing previous simulations with RIP.

2.1 Preliminaries: Ranking, Weighting, and Probability

Before turning to the problem of hidden structure learning in Stochastic OT and Noisy HG, these frameworks are briefly reviewed here (for further reading see Boersma 1997; Boersma and Hayes 2001; Pater 2009b; Boersma and Pater to appear; Pater to appear). In Stochastic OT (Boersma 1997; Boersma and Hayes 2001) constraints are associated with a ranking value along a continuous scale. At evaluation time, random noise (sampled from a normal distribution centred around zero) is added to the ranking value of each constraint independently, and the resulting relative ordering is interpreted as a strict ranking for optimization. In this way, Stochastic OT defines a probability distribution over total orderings of constraints, with re-ranking likely for constraints with similar ranking values and unlikely for constraints ranked far apart. The Gradual Learning Algorithm (GLA) for Stochastic OT (Boersma 1997; Boersma and Hayes 2001) makes use of the continuous scale by making small, repeated updates to the ranking values of constraints during learning.

(1) Example Illustrating HG Evaluation

	5	4	2	
	C1	C2	C3	*Harmony*
a. Candidate A		-1	-1	(–1)**w*(C2)+ (–1)**w*(C3) = –6
b. Candidate B	-1			(–1)**w*(C1) = –5
c. Candidate C		-2		(–2)**w*(C2) = –8

In Harmonic Grammar (Legendre, Miyata and Smolensky 1990; Smolensky and Legendre 2006; Pater 2009a; Pater 2009b), constraints are weighted rather than strictly ranked. A candidate's harmony is defined as a weighted sum of constraint violations, with each violation multiplied by the weight of the corresponding constraint. Constraint violations are generally expressed as negative integers, and the candidate with the highest (closest to zero) harmony is optimal. This is illustrated in the simple example in (1) with three candidates and three constraints. The weight of each constraint is multiplied by the number of constraint violations and summed over all constraints to yield the harmony, shown at the right. This example illustrates how in HG, in contrast to OT, lower weighted constraints can overpower higher weighted constraints: here, Candidate B is optimal with a harmony of –5 even though it violates the highest weighted constraint while the other candidates do not. In Noisy HG (Pater 2009b; Boersma and Pater to appear; Pater to appear), random noise is added to the constraint weights in the same way that it is added to the ranking values in Stochastic OT. After noise has been added to the weights in Noisy HG, the resulting weighting is used to calculate the optimal candidate. Weighted grammars can be learned using an online learning algorithm that relies on the perceptron update rule (Rosenblatt 1958; Soderstrom, Mathis and Smolensky 2006) and is otherwise identical to the Gradual Learning Algorithm for Stochastic OT. Following Boersma and Pater (to appear) and Pater (2009), this paper refers to both algorithms as GLA and uses OT-GLA and HG-GLA to distinguish the ranking and weighting versions, respectively. Thus, HG-GLA makes use of the continuous weighting scale by making small adjustments to constraint weights, just as the OT-GLA does for constraint ranking values. This paper focuses on learning in the Noisy HG version of weighted grammars; however, the findings have implications for learning in related weighted grammar frameworks such as Maximum Entropy Grammars, which are briefly discussed in Section 4.1 (Johnson 2002; Goldwater and Johnson 2003; Fischer 2005; Jäger 2007; Hayes and Wilson 2008).

2.2 Error Driven Learning and Hidden Structure

The OT-GLA and HG-GLA are both error-driven learning algorithms (Rosenblatt 1958; Wexler and Culicover 1980; Tesar 1995; Tesar and Smolensky 1998). Although the current paper focuses on learning in the stochastic setting, this section also reviews Error-Driven Constraint Demotion (EDCD) and its extensions to the problem of structural ambiguity since it is in this context that RIP was developed (Tesar 1995; Tesar and Smolensky 1998). Error-driven learning means that updates to the learner's grammar are driven by errors the learner makes while processing the learning data. Specifically, for each learning datum, the learner uses its current grammar to generate its own output for that datum. If the learner's output does not match the learning datum, the learner compares its output (the loser) to the learning datum (the winner) in order to determine how to adjust the grammar. For example, suppose the learner is presented with the form [tɛ(ˈlɛfɔn)] while learning a language much like Polish, with regular penultimate stress (Rubach and Booij 1985). Suppose the learner must correctly rank constraints preferring right and left alignment of feet, ALLFEETRIGHT and ALLFEETLEFT, respectively, and constraints preferring right and left headed feet, IAMBIC and TROCHAIC, respectively. If the learner's current ranking is ALLFEETRIGHT » IAMBIC » TROCHAIC » ALLFEETLEFT, the learner will generate the incorrect [tɛ(lɛˈfɔn)] for input /tɛlɛfɔn/, as shown in (2). As an error-driven learner, the learner will then compare the violations of the loser with the violations of the winner in order to determine which constraints favour the winner and which favour the loser. The precise update rules vary between EDCD, OT-GLA, and HG-GLA, but all involve adjusting the relative rankings or weighting of loser-preferring and winner-preferring constraints so as to increase the harmony of the winner compared to the loser. The EDCD learner demotes all undominated loser-preferring constraints to a stratum immediately below the highest ranked winner-preferring constraint (Tesar 1995; Tesar and Smolensky 1998).

(2) Learner's hypothetical grammar when presented with [tɛˈlɛfɔn]

	/tɛlɛfɔn/	ALLFEETRIGHT	IAMBIC	TROCHAIC	ALLFEETLEFT
	a. (ˈtɛlɛ)fɔn	*	*		
	b. (tɛˈlɛ)fɔn	*		*	
Winner	c. tɛ(ˈlɛfɔn)		*		*
Loser	d. tɛ(lɛˈfɔn)			*	*

The update rules for the OT-GLA (Boersma 1997; Boersma and Hayes 2001) and HG-GLA (Boersma and Pater, to appear; see also Rosenblatt 1958; Jäger 2007; Soderstrom et al 2006) learners are shown in (3) and (4), respectively. As shown in (3), for each constraint *i,* OT learners compare the violations of the winner W and loser L under constraint *i*, $c_i(W)$ and $c_i(L)$, respectively. Constraint *i*'s ranking value is increased by ε (the learning rate or plasticity) when the loser has more violations than the winner and decreased by ε when the winner has more violations than the loser (the sgn function returns -1 for negative, 1 for positive, and 0 for zero). Thus, ε is added to the ranking values of winner-preferring constraints and subtracted from those of the loser-preferring constraints. As shown in (4), the update rule for HG is identical except that the plasticity is multiplied by the difference in the number of violations assigned to the loser and the winner. The HG update rule is an adapted form of the perceptron update rule for training connectionist networks (Rosenblatt 1958), which is itself an adaptation of the standard machine learning technique gradient ascent,

whose online variant is known as stochastic gradient ascent (see Jäger 2007 for in depth discussion). In sum, both OT-GLA and HG-GLA slightly decrease the ranking or weighting of all loser-preferring constraints and slightly increase the ranking or weighting of all winner-preferring constraints. In order to calculate the update, both algorithms compare the violations assigned to the loser to the violations assigned to the winner.

(3) OT-GLA Update Rule
$\Delta r_i = \varepsilon \cdot \text{sgn}(c_i(L) - c_i(W))$

(4) HG-GLA Update Rule
$\Delta w_i = \varepsilon \cdot (c_i(L) - c_i(W))$

In the example in (2), candidate (c) is the winner for the penultimate stress language, while the loser, the learner's own output, is candidate (d). Comparing the violations incurred by candidates (c) and (d), the learner determines that IAMBIC favours the loser, and TROCHAIC favours the winner, while the remaining constraints have no preference. Based on this, the learner calculates the update rule, which indicates that IAMBIC must be demoted, and in OT-GLA and HG-GLA, that TROCHAIC must also be promoted. Despite the differences in update rules between the three algorithms, the processing required to calculate the update is the same: a comparison of the violations incurred by the winner and loser.

The learning strategy just described assumes the learner is provided with full structural descriptions of the learning data. This means the learner has access to hidden structure such as footing and syllabification as well as underlying representations, which are not available to the human learner. Access to such hidden structure is crucial for identifying the violations incurred by the winners, which, as just discussed, are crucial for calculating the update. This is because every overt form is structurally ambiguous and corresponds to numerous candidates, each with different hidden structures and therefore distinct constraint violations. In the example above, the learner was presented with a learning datum together with hidden structure, the footing: [tɛ(ˈlɛfɔn)]. It is this footing that identified candidate (c) as the winner. In a more realistic learning context wherein the learner observes only the overt [tɛˈlɛfɔn], the learning datum would be ambiguous between candidates (b) and (c). Without full structural descriptions, the learner cannot be sure about which of (b) and (c) is actually the winner in this language. The problem is that the constraint violations needed to calculate the grammar update depend on which structure, or parse, is selected. The hidden structure is what determines whether constraints like IAMBIC are violated. The example in the preceding paragraph made the simplifying assumption that this hidden structure was available to the learner. The same assumption is made by all algorithms, such as EDCD and GLA, whose performance or proofs of correctness presuppose access to full structural descriptions. This is unrealistic because children acquiring language do not have access to this information. Indeed, determining whether (b) or (c) is the winner is part of learning the grammar of the target language since different languages may parse a form like [tɛˈlɛfɔn] differently. In essence, full structural descriptions provide the learner with parses, or analyses, of all the learning data, providing the constraint violations of the learning data to which the learner's losers can be directly compared.

Thus, structural ambiguity presents a difficult learning challenge because it obscures the constraint violations incurred by the learning data, thereby obscuring the update needed to favour the winner (the learning datum) over the loser (the learner's output). In the example above, candidates (b) and (c) have drastically different consequences for the resulting grammar. As explained above, selecting candidate (c) as the winner results in an adjustment

to the constraints IAMBIC and TROCHAIC. On the other hand, if the learner mistakenly selects (b) as the winner, the update will instead involve adjustment of ALLFEETRIGHT relative to ALLFEETLEFT, leading to a grammar that is more likely to align feet with the left edge of the word. Furthermore, there is no way to definitively identify the correct parse for a form in isolation – the correct parse can ultimately be determined only by consulting other learning data. For example, in order to determine that the correct analysis involves right-aligned trochees rather than left-aligned iambs, the learner must process other forms that disambiguate between these possibilities, such as disyllabic forms with initial stress. In a more realistic setting, with many more constraints contributing to the selection of the correct hidden structure, the learning data are massively more ambiguous than in this simple example (see e.g. Tesar 2004b; Prince 2010). Nonetheless, the learner must somehow effectively navigate this huge ambiguous space of possibilities.

2.3 Robust Interpretive Parsing

Within OT, learning in the face of structural ambiguity has been a topic of on-going work since at least Tesar (1997a; 1998) and Tesar and Smolensky (1998; 2000). In order to apply error-driven learning in the presence of structural ambiguity, Tesar and Smolensky (1998) proposed Robust Interpretive Parsing (RIP), which provides an educated guess, based on the current constraint ranking, about the structure of the observed datum. Specifically, RIP uses the learner's current hierarchy to select the most harmonic candidate among the structural descriptions consistent with an overt form. That is, for a given learning datum, RIP uses standard OT evaluation but limits candidates to those that share the learning datum's overt form, thereby selecting the most harmonic among the possible structural descriptions, or parses, of the overt form according to the current grammar. The parse produced by RIP is treated as the intended winner and compared to the learner's own output, which is generated by applying the usual 'production-directed parsing': the process of mapping the underlying form to its optimal structural description. As shown in (5), this means that when the learner is presented with the unstructured overt form [tɛˈlɛfɔn], it performs interpretive parsing by finding the most harmonic candidate matching this overt form. In this example, only two candidates, (b) and (c), match the overt form. According to the current ranking, candidate (c), [tɛ(ˈlɛfɔn)], is more harmonic and is therefore selected as the RIP parse and winner. In this case, this is the correct parse for a language like Polish that has right-aligned trochees. Crucially, the learner is capable of identifying the correct parse even though the current grammar is not the target grammar for the penultimate stress language. This crucial property is why Tesar and Smolensky refer to this interpretive parsing procedure as 'robust'. If learning is to be successful, it is essential for the learner to be able to assign structure to the learning data throughout the learning process. Parsing must be possible even when the learner's grammar differs from the target grammar and does not generate the datum being processed (otherwise, the learner would only be able to parse and learn from data consistent with its initial grammar). This is exactly the solution RIP provides. Once the learner has assigned a structural description to the learning datum using RIP, production-directed parsing is applied to the underlying form /tɛlɛfɔn/, producing candidate (d) [tɛ(lɛˈfɔn)]. The winner provided by interpretive parsing, candidate (c), is compared to the result of production-directed parsing, candidate (d), yielding an error. Given the winner and the loser, the grammar update, comparing the violations of the parsed winner to that of the loser, can proceed as usual according to the update rules discussed above.

(5) Robust Interpretive Parsing for [tɛˈlɛfɔn]

	/tɛlɛfɔn/	ALLFEETRIGHT	IAMBIC	TROCHAIC	ALLFEETLEFT
	a. (ˈtɛlɛ)fɔn	*	*		
	b. (tɛˈlɛ)fɔn	*		*	
RIP parse	c. tɛ(ˈlɛfɔn)		*		*
	d. tɛ(lɛˈfɔn)			*	*

As discussed by Tesar and Smolensky (1998), however, RIP is not fool proof. It can select the wrong parse for the target language, leading the learner astray. This may happen, for example, if the learner in (5) were actually trying to learn a language with left-aligned iambs, corresponding to candidate (b). The learner can sometimes overcome such parsing mistakes given subsequent disambiguating data. However, the learner can also get stuck in a perpetual loop of mistaken parses and grammars. Tesar and Smolensky (2000) presented simulation results for a RIP version of Error Driven Constraint Demotion (RIP/EDCD) on a large metrical phonology test set with structural ambiguity. They found that RIP/EDCD learned just 60.5% of the languages in the system correctly when starting from an unranked initial hierarchy. In fact, this success rate assumes that learning produces stratified hierarchies and that pooling ties are used to deal with tied constraints, which, as discussed by Boersma (2009), can mask crucial rankings needed to uniquely select the target output forms under strict ranking. As discussed below, the performance of EDCD drops when evaluation assumes strict ranking. In general, these results indicate that RIP's potential to lead the learner astray is not just hypothetical since about half of the languages in the system cannot be learned.

RIP was later extended in several directions. It was applied to the problem of structural ambiguity using OT-GLA (Boersma 2003; Apoussidou and Boersma 2003; Apoussidou 2007) and HG-GLA (Boersma and Pater to appear). Another line of work extends Stochastic OT with lexical constraints, constraints that control the choice of underlying representations, and uses RIP/GLA in this context to learn a different kind of hidden structure, namely underlying representations (Apoussidou 2006; Apoussidou 2007). The performance of these stochastic and weighted variants of RIP is also mixed. Boersma and Pater report on simulations comparing the performance of RIP variants of EDCD, OT-GLA, and HG-GLA with the same test set used by Tesar and Smolensky (2000). Their implementation of RIP/EDCD differs from Tesar and Smolensky's (2000): rather than using pooling ties, they use permuting ties (see Boersma 2009 for discussion), and they require EDCD to learn strict rankings. This lowers the performance of RIP/EDCD to 47%. They also find that both OT and HG variants of RIP/GLA outperform RIP/EDCD, with RIP/GLA for Noisy HG getting the highest performance overall, learning almost 89% of the languages in the system on average. While the performance of RIP/HG-GLA is encouraging, Boersma and Pater find poor performance of RIP for both Classic OT and Stochastic OT, with RIP/OT-GLA learning only around 59% of the languages in the test set. Overall, therefore, Boersma and Pater's results suggest a strong advantage for the HG learners in this context.

Much other work in OT and related constraint-based frameworks has explored alternative approaches to the hidden structure problem (Tesar 1997b; Tesar et al. 2003; Tesar 2004b; Prince and Smolensky 2004; Tesar 2004a; Alderete et al. 2005; Jarosz 2006a; Jarosz 2006b; Tesar 2006a; Tesar 2006b; Merchant 2008; Merchant and Tesar 2008; Tesar 2008; Tesar 2009; Akers 2011; Jarosz to appear a). The present paper, however, takes a closer look at how RIP has been formulated for the stochastic setting, namely the applications of RIP to OT-GLA and HG-GLA. After reviewing the previous computational results more carefully,

the following sections identify two main problems with the formulation of RIP for the stochastic setting and identify adjustments to the parsing process that dramatically improve performance.

2.4 The Metrical Phonology Test Set And Previous Results

The simulations presented in this paper rely on the same metrical phonology test set used to evaluate RIP/EDCD, RIP/GLA for OT, and RIP/GLA for HG in previous work (Tesar and Smolensky 2000; Boersma 2003; Boersma and Pater to appear; Jarosz to appear a)[2]. This allows for replication of and direct comparison with previously reported results using RIP. This section presents that test set and reviews the previous results in more detail.

This test set, first defined and examined by Tesar and Smolensky (2000), consists of 124 constructed languages that can be modelled by the set of twelve metrical structure constraints shown in (6). Most of these constraints are well known from the literature, with origins in the early OT literature (McCarthy and Prince 1993; Prince and Smolensky 2004) and pre-OT metrical phonology (Prince 1990; Hayes 1995; Liberman and Prince 1977). One exception is the non-standard formulation of the constraint favouring trochees: FOOT-NONFINAL (Tesar 2000). The interaction of these twelve constraints produces a complex artificial system, inspired by natural language stress systems, capable of describing a range of diverse metrical phenomena. The test system has the crucial property of generating structural ambiguity – overt stress patterns in this system are consistent with multiple structural descriptions, and successful learning requires disentangling interdependent and ambiguous requirements made by the individual learning data[3]. Tesar and Smolensky selected 124 languages from the factorial typology generated by this constraint set to represent a wide range of metrical phenomena.

(6) Constraints (Tesar and Smolensky 2000)

FOOTBIN	Each foot must be either bimoraic or disyllabic
PARSE	Each syllable must be footed
IAMBIC	The final syllable of a foot must be the head
FOOT-NONFINAL	A head syllable must not be final in its foot
NONFINAL	The final syllable of a word must not be footed
WSP	Each heavy syllable must be stressed
WORD-FOOT-RIGHT	Align right edge of the word with a foot
WORD-FOOT-LEFT	Align left edge of the word with a foot
MAIN-RIGHT	Align head foot with right edge of the word
MAIN-LEFT	Align head foot with left edge of the word
ALL-FEET-RIGHT	Align each foot with right edge of the word
ALL-FEET-LEFT	Align each foot with left edge of the word

Each language in the system is defined by a set of surface stress patterns for sixty-two words that can be generated from this constraint set. Words are sequences of light (L) or

[2] I would like to thank Joe Pater for sharing the grammar and distribution files for the Tesar and Smolensky test set, and Paul Boersma and Bruce Tesar for creating them.

[3] As discussed earlier, the learning problem undertaken here is the general problem of learning weightings and rankings given structurally ambiguous data, and this metrical phonology test set provides one domain in which to examine this learning challenge. While there are some theories of stress (Gordon 2002) and stress learning (Daelemans, Gillis and Durieux 1994; Gupta and Touretzky 1994; Heinz 2009) that do not rely on hidden structure, the problem of hidden structure in other domains, such as syllabification, remains to be dealt with.

heavy (H) syllables ranging in length between two and seven syllables (e.g. [H L H L]). Each word is associated with a surface stress pattern (e.g. [H1 L0 H2 L0]), indicating for each syllable whether it has primary stress (1), secondary stress (2), or no stress (0). Any given ranking or weighting of the constraints assigns a particular foot structure and pattern of stress (e.g. [(H1 L0) (H2) L0]). Indeed, it is the footing that underlies the systematic stress patterns in the system. The learner, however, is exposed only to the overt stress patterns (e.g. [H1 L0 H2 L0]) and must infer a ranking or weighting of constraints (and an associated footing) capable of generating the observed surface stress patterns. The learner is considered successful when it has acquired a grammar that is consistent with all the learning data it is exposed to, that is, when it assigns the correct surface stress patterns to all the words of the language.[4]

In addition to examining the learning of various metrical patterns, this system presents a more fundamental challenge to learning models. As discussed earlier, much work in OT decouples the learning strategies from the specific phenomena learners must cope with, focusing on developing learners that are successful regardless of the exact constraints and representations used (Tesar 1995; Tesar 2004b; Pater 2008; Magri 2012; Boersma and Pater to appear). This consideration has played an important role in OT learnability from the beginning: Tesar's (1995) foundational work on Constraint Demotion proved that, in general, CD is guaranteed to converge on any target language in its hypothesis space given fully structured data. One can ask an analogous question about learning in the context of structural ambiguity (see e.g. Tesar 2004b): can a given learning algorithm successfully learn a grammar for any structurally ambiguous language in its hypothesis space, regardless of the constraint set? Testing learners on this test set therefore also addresses this more general question. Viewed from this perspective, the test set provides a relatively large and diverse set of challenging target languages to the learners, which compares favourably with previous work in terms of the number of systems tested. For example, Heinz (2009) considers 109 stress patterns, Gupta and Touretzky (1994) considered 19, and a number of previous studies examine only a handful of stress systems. Thus, from this computational perspective, good performance on this system reflects the ability to handle a wide range of patterns.

Another perspective, and one that is an important direction for future work, is that of typology and its interaction with learnability. While it is important to determine whether a fully general and feasible solution to structural ambiguity exists, the possibility that successful learnability relies on substantive or formal restrictions on the hypothesis space must also be investigated. Accordingly, it is important to investigate the learnability of attested stress systems and to determine whether structural ambiguity in natural language has particular formal properties that are crucial to its learnability. Metrical phonology is an especially promising domain for such research since stress typology is very well studied, large typologies of stress systems have been developed, and large stress databases are available (Hayes 1995; Gordon 2002; van der Hulst, Goedemans and van Zanten 2010). A number of the previous studies discussed earlier examine learning of attested languages (Dresher and Kaye 1990; Goldsmith 1994; Gupta and Touretzky 1994; Heinz 2009), and this is an important direction for future work with constraint-based learners as well. While the Tesar and Smolensky system was inspired by stress typology, it is nonetheless an artificial system with no direct correspondence to attested languages. In addition, recent theoretical and typological developments have identified some problematic predictions made by the constraints it assumes (McCarthy 2003; Hyde 2007; Pruitt, Kathryn 2010). Development of

[4] If there are multiple weakly equivalent grammars consistent with the learning data, learning is deemed successful when the learner converges on any one of them (since the learning data does not disambiguate been them).

large tests sets for constraint learning based on natural language stress typologies will require the integration of theoretical, computational, and typological work in order to create a computational implementation of a theory (or theories) of constraints that can be used to model the entire typology. Given the significant recent developments in all of these areas, this presents an exciting opportunity for integration of work from these diverse perspectives.

Thus, the test set is not without limitations, but it allows for the primary concerns of the present work to be addressed: namely, the relative performance of different learning models on structurally ambiguous learning data, and the finding from previous work suggesting an advantage for HG on the basis of experiments with this test set. As discussed above, Tesar and Smolensky (2000) and Boersma and Pater (to appear) report on RIP simulations with this test set. The results for RIP/EDCD, RIP/OT-GLA, and RIP/HG-GLA are summarized in (7)[5]. Boersma and Pater (to appear) allowed each run of each algorithm a maximum of 1,000,000 iterations, where an iteration corresponds to the processing of one overt form. The reported performance is the average of 10 separate runs for each algorithm. Performance of RIP/EDCD in both studies was deemed successful when the algorithm had converged to a hierarchy that correctly predicted the stress patterns for each of the sixty-two forms in the language, making no further errors on the data. As shown in the table, RIP/EDCD learns roughly between 47% and 60% of the languages in the system correctly, depending on whether pooling or permuting ties are used. As discussed above, the permuting ties simulations make the more standard assumption that target languages must be total rankings. For RIP/OT-GLA and RIP/HG-GLA, Boersma and Pater (to appear) set the learning rate (plasticity) to 0.1, the initial weights/ranking values to 10, and the evaluation noise to 2.0. They define a language to be successfully learned if, when evaluation noise is set to zero, the resulting ranking/weighting correctly generates the stress patterns for all sixty-two forms in the language. Using these evaluation criteria, they find performance of RIP/OT-GLA to be about 59% and performance of RIP/HG-GLA to be about 89%.

(7) Performance of RIP Algorithms Reported in Previous Work

Algorithm	Languages Learned	Citation
RIP/EDCD (pooling ties)	60.48%	Tesar and Smolensky (2000)
RIP/EDCD (permuting ties)	46.94%	Boersma and Pater (to appear)
RIP/OT-GLA	58.95%	Boersma and Pater (to appear)
RIP/HG-GLA	88.63%	Boersma and Pater (to appear)

As Boersma and Pater discuss, these results look promising for stochastic approaches and for weighted constraints. The following sections, however, take a closer look at the RIP/GLA algorithms for OT and HG, identifying two main problems with how RIP has been formulated in the stochastic setting. These problems turn out to have significant consequences for the performance of the stochastic RIP learning algorithms.

3. Problem 1: Parsing with a Losing Grammar

To see the first problem with the original formulation of RIP for GLA, first formulated for the stochastic setting by (Boersma 2003; Apoussidou and Boersma 2003), consider the more explicit formulation of the algorithm given in (8). Learning begins by selecting an initial Stochastic OT (or in later work, Noisy HG) grammar, G_0, as shown in step 1. For example,

[5] Boersma and Pater (2008) also report on results of applying RIP to non-noisy OT-GLA and HG-GLA (which perform worse than their noisy counterparts), as well as a variant of Noisy HG they call exponential HG (whose best performance is similar to HG-GLA). The focus in the present work is on the more widely used noisy variants of GLA, and therefore only these results are reviewed here.

the ranking or weighting values of all constraints may initially be set to 10. Then, as indicated in step 2, the learner iterates over the data *D*, one learning datum *d* at a time, by randomly sampling from the set of learning data. Each learning datum *d* is an overt form, without abstract structure. For each datum *d*, the learner samples a ranking/weighting, G´, from the current stochastic grammar G_i by adding evaluation noise to the ranking/weighting values of G_i (step 2.a). The learner then uses G´ to perform Robust Interpretive Parsing on *d* (step 2.b) to produce the Parse. The Parse is a fully structured candidate from which the learner extracts the underlying representation to arrive at the Input (step 2.c). Note, however, that, in the context of grammatical stress, this amounts to stripping away abstract structure and stress: parsing is not necessary in order to arrive at the underlying form since it is directly recoverable from the learning datum (for example, the underlying form of [L0 H1 L0] is /L H L/). Next, the learner uses G´ again to generate its own Output (step 2.d). Finally, the learner compares the fully structured Output it generated to the structured Parse (step 2.e). If they do not match, the learner compares their violations and updates the grammar G_i (2.e.i) using the update rules defined earlier in (3) and (4), yielding the updated grammar, G_{i+1}.

(8) Robust Interpretive Parsing for GLA
1. Initialize Stochastic Grammar: G_0
2. Iterate over *d* in *D*:
    a. Sample $G' \sim G_i$
    b. Parse = $RIP_{G'}$ (*d*)
    c. Input = uf(Parse)
    d. Output = $Optimise_{G'}$ (Input)
    e. If Output ≠ Parse:
        i. G_{i+1} = Update (G_i, Parse, Output)

What is crucial about the above formulation is that the same grammar G´ is used for production (step d) and for interpretive parsing (step b). Boersma and Apoussidou do not provide the explicit formulation of the algorithm in (8), but it is clear from their discussion of the procedure that the algorithm uses the same grammar for parsing and production. In particular, Apoussidou and Boersma (2003; p111) write:

> In RIP/GLA, the interpretation step is done within Stochastic OT, i.e. after adding a bit of evaluation noise to the constraint rankings, and this same temporary ranking is then used for the generation of the learner's own form; the adjustment step proceeds as usual, i.e. with reranking of all the constraints that prefer the adult form or the learner's form.

As explained in the following section, the use of the same grammar for production and interpretive parsing turns out to be problematic.

3.1 Reformulating RIP: Resampling

The problem with the original RIP application in the stochastic setting is highlighted when the algorithm is slightly reformulated as in (9). The original formulation defines an error in terms of a mismatch between the fully structured Output and the fully structured Parse. It may therefore appear that forms matching in their overt stress patterns may be counted as errors if their footing (structure) differs. However, in the present learning context, where underlying representations are fixed for each learning datum (since they are recoverable from the learning datum), this situation can never arise during learning – whenever the Output and Parse match in their overt stress pattern, they necessarily match in their full structural descriptions. If the Output matches the stress contour of the learning datum, this means the most harmonic candidate according to G´ is a candidate with the observed stress contour. In

this context, where underlying forms are held fixed for interpretive parsing, the set of candidates competing with one another for interpretive parsing will be a subset of the candidates competing with one another in production, namely the subset whose overt portion matches the learning datum. Thus, the candidate selected as optimal in production is also a candidate for interpretive parsing. If G′ selects a particular candidate as optimal among the full set of candidates, it will also select the same candidate as optimal in a subset containing that candidate. This is true regardless of whether weighting or ranking is used.

Consequently, in the original formulation of RIP, errors only occur when the Output and Parse differ in their overt forms. Therefore, the comparison between the winner and loser can be converted from applying to fully structured forms to applying to the overt forms without affecting the behaviour of the algorithm. This is exactly what the reformulation in (9) does. Note that in the original formulation, the RIP step was needed in order to generate the Parse to which the Output could be compared and from which the Input was derived. In the reformulation, RIP parsing is not necessary for checking whether an error has been produced since only the overt portion of the learning datum is consulted (step 2.d). Also, as discussed above, the Input can be recovered directly form the learning datum without parsing. This means the parsing step can therefore be moved inside the conditional, as shown in step 2.d.i of the reformulated algorithm.

(9) Reformulated Robust Interpretive Parsing for GLA
1. Initialize Stochastic Grammar: G_0
2. Iterate over *d* in *D*:
    a. Sample $G' \sim G_i$
    b. Input = uf(*d*)
    c. Output = $\text{Optimise}_{G'}$ (Input)
    d. If overt(Output) ≠ *d*:
        i. Parse = $\text{RIP}_{G'}$ (*d*)
        ii. G_{i+1} = Update (G_i, Parse, Output)

This reformulation does not affect the behaviour of the algorithm, but it does serve to highlight a peculiar aspect of interpretive parsing. Specifically, from the reformulation it is clear that parsing is only relevant in case the selected grammar G′ generates an error. At the point when parsing is required, the learner knows G′ is the wrong grammar since G′ generated an error for this form. What is odd about this use of interpretive parsing in the stochastic setting, then, is that the learner nonetheless uses the known-to-be-incorrect G′ for interpretive parsing. In the original formulation of RIP for Classic OT (Tesar and Smolensky 1998), the learner has no better alternative. In Classic OT, the learner's current grammar is a single, categorical ranking, and the learner has no choice but to use this ranking, its current best guess about the target language, for RIP. In the stochastic setting, however, each grammar defines a distribution over rankings (or weightings), and the learner's current knowledge about the target language is richer. The learner has a whole distribution of rankings or weightings available for parsing and does not need to rely on a ranking or weighting that produced an error.

A simple modification to RIP, Resampling RIP (RRIP), solves this problem and is defined in (10). The only difference between the algorithm in (10) and the one in (9) is that the new algorithm samples a new grammar for interpretive parsing: this is shown in step 2.d.i. Specifically, if the learner's output fails to match the learning datum, the learner simply samples another grammar G″ from G_i and uses it for interpretive parsing. This is the simplest possible way for the learner to reference its stochastic grammatical knowledge, relying only on mechanisms (sampling a random grammar) that are needed anyway. To see the effect this

resampling has, consider a learner which has not yet settled on the target grammar but whose current stochastic grammar generates the target ranking or weighting with some probability. This is the situation the learner starts out in if all constraints are ranked or weighted equally. Initially, all rankings or weightings are equally likely, and, as learning progresses, the distribution over rankings or weightings narrows, focusing gradually on more apt grammars. If the learner samples an incorrect G′ for production and makes an error, the original RIP algorithm will nonetheless consistently use G′ for parsing. In contrast, RRIP will select another grammar according to its current knowledge about which rankings or weightings are likely. Whereas RIP is doomed to use an incorrect ranking or weighting for parsing, RRIP's resampling makes it possible for the learner to select a correct ranking or weighting to use for interpretive parsing, increasing the likelihood that the resulting parse will be correct for the target language. Put differently, when the learner's ranking or weighting fails to produce the observed stress pattern, the learner falls back on its rich stochastic grammatical knowledge to select another ranking or weighting it has some confidence in. It uses its stochastic grammar to select with high probability those rankings or weightings that provide the current best guess about the hidden structure. There is a chance resampling will end up selecting the same parse[6], but, unlike RIP, RRIP also has a chance of selecting a different grammar, and it does so in proportion to its confidence in different grammars. In this way, the learner takes advantage of the rich information represented by its stochastic grammar.

(10) Resampling Robust Interpretive Parsing (RRIP) for GLA

1. Initialize Stochastic Grammar: G_0
2. Iterate over *d* in *D*:
    a. Sample $G' \sim G_i$
    b. Input = uf(*d*)
    c. Output = $\text{Optimise}_{G'}$ (Input)
    d. If overt(Output) ≠ *d*:
        i. Sample $G'' \sim G_i$
        ii. Parse = $\text{RIP}_{G''}$ (*d*)
        iii. G_{i+1} = Update (G_i, Parse, Output)

Clearly, the computational differences between RIP and RRIP are minimal. The computational cost of resampling is negligible since selecting a random ranking or weighting from the current stochastic grammar can be done very efficiently. In fact, there is a potential computational advantage to RRIP since it performs parsing (and resampling) only when there is an error, whereas RIP automatically parses each overt form. It is worth noting that the reformulation in (9) is crucial for the resampling modification. Simply resampling from the grammar in (8) prior to production would not work since, in that formulation, errors are defined in terms of fully structured candidates. Such a learner would count as errors not only outputs that fail to match the datum but also outputs that match the datum but, due to a random and arbitrary difference between G′ and G″, differ from the parse with respect to their structure. The randomly occurring differences in grammars between the two samplings

[6] An anonymous reviewer asks whether it might be possible to constrain resampling further by requiring that G″ be distinct from G′. This is an appealing possibility worth exploring; however, it is not immediately clear how 'distinct' would be defined for continuous weightings, and even distinct rankings are often weakly equivalent, selecting the same optima. Therefore the expected gain of such a restriction is not obvious, while implementing it would come with an additional computational cost that is not trivial. In any case, the present focus is on exploring the consequences of a simple and minimal modification of RIP, and Section 4 shows there is a more fundamental problem with RIP that resampling on its own cannot address.

would therefore lead to many spurious errors and grammar updates. Of course, such a variant would be equivalent in behaviour to RRIP (albeit less efficient) if in addition to resampling, the error was defined in terms of overt forms only, as in (9) and (10). In sum, there are two crucial computational differences between RIP and RRIP: 1) RRIP selects a new random ranking or weighting for parsing rather than using the one used in production, and 2) RRIP defines errors as mismatches in overt forms rather than fully structured forms.

The next section explores the effect these differences have on performance. The simulations focus on RIP and RRIP versions of GLA for OT and HG. The modifications in RRIP would have no effect on the performance of EDCD since in EDCD there only is one ranking that can be used for parsing. However, it is useful to keep in mind the performance of RIP/EDCD reported in previous work and summarized in (7) as it provides a point of reference for performance of RIP in Classic OT.

3.2 Resampling Robust Interpretive Parsing Simulations and Results

Simulations were performed in order to compare the performance of HG and OT variants of RIP/GLA and RRIP/GLA. As discussed earlier, all simulations were performed with the grammatical stress test system developed by Tesar and Smolensky (2000). One goal of the simulations is to replicate Boersma and Pater's (to appear) results and to determine the robustness and variability of RIP for OT and HG. Thus, simulations of RIP for both OT-GLA and HG-GLA were performed with the parameters used by Boersma and Pater: learning rate (plasticity) of 0.10, evaluation noise of 2, and initial ranking/weighting values of 10. Following Boersma and Pater, all runs were allowed a maximum of 1,000,000 iterations for learning. Also following Boersma and Pater, success was defined as correct generation of stress patterns for all sixty-two overt forms in a language using the learner's grammar with evaluation noise set to zero. Also, since the algorithms are non-deterministic, ten runs of each RIP/OT-GLA and RIP/HG-GLA with these parameter settings were performed in order to calculate average performance and variation in performance from run to run.

In addition, simulations with a range of other parameter settings were performed to examine sensitivity of the algorithms to the learning parameters. Specifically, in addition to the learning rate of 0.10, both OT and HG variants of RIP were tested with learning rates of 0.05, 0.25 and 0.50. Finally, all the same manipulations were performed for the new RRIP algorithm for both the OT and HG versions. The results of these 160 simulated learners on the 124 languages in the system (a total of 19840 separate runs) are summarized in (11), which shows the average percentage of languages learned correctly and the sample standard deviation for each algorithm/plasticity combination.

Consider first the simulations, shown in the shaded cells of the table, that replicate the results reported by Boersma and Pater (to appear), summarized earlier in (7). The performance for RIP/HG-GLA shown here (88.71) is nearly identical to the results reported by Boersma and Pater (88.63). The performance of RIP/OT-GLA, 56.13, is a few percentage points lower than the earlier result of 58.95. This difference of 2.82 is suspiciously high for the sample standard deviation of 1.62. The discrepancy may be due to a difference in terminating conditions. All performance results presented here assume learning continues for the maximum number of iterations as long as the learner's stochastic grammar continues to make errors, even if the grammar with noise removed correctly generates the data. Occasionally, learners temporarily enter a grammar that makes no errors with evaluation noise removed, but because their stochastic grammars have not actually converged, they later change their grammars to ones that do produce errors on subsequent iterations. In the present results, learners were only counted as successful if they made no errors on subsequent iterations. If, however, these learners were counted as successful and the subsequent iterations ignored, the performance of RIP/OT-GLA in the current simulations would

increase to 59.84, comparable to Boersma and Pater's result of 58.95. Thus, other than what may be a difference in terminating conditions, the present simulations do replicate Boersma and Pater's reported results. Importantly, the present results replicate the major effects identified by Boersma and Pater.

(11) Success Rate (SD) of RIP and RRIP Across Various Parameter Settings

Algorithm	Learning Rate (plasticity)			
	.05	.10	.25	.50
RIP/OT-GLA	55.81 (1.82)	56.13 (1.62)	56.21 (2.15)	**57.50** (2.28)
RIP/HG-GLA	**88.79** (0.97)	88.71 (0.66)	85.48 (1.57)	82.90 (2.92)
RRIP/OT-GLA	**84.19** (1.91)	82.58 (1.91)	81.13 (2.29)	80.08 (3.09)
RRIP/HG-GLA	**89.44** (0.71)	89.27 (0.94)	87.58 (1.79)	82.98 (1.84)

The results confirm Boersma and Pater's finding that RIP/HG-GLA dramatically outperforms RIP/OT-GLA. Furthermore, the results in (11) show performance of RIP/OT-GLA and RIP/HG-GLA at several additional learning rates. These results indicate that the learning rate does affect the performance of the algorithms, with the HG learners showing a bit more sensitivity to learning rate. RIP/HG-GLA performs better at the lower learning rates, while RIP/OT-GLA performs somewhat better with higher learning rates. In general, however, the variation in performance for runs of RIP/OT-GLA and runs of RIP/HG-GLA is very small in comparison to differences in performance between the OT-GLA and HG-GLA algorithms. RIP/HG-GLA performs significantly better than RIP/OT-GLA across all learning rates tested ($p < .001$; Welch two sample t-test), confirming that the substantial difference in performance between RIP/OT-GLA and RIP/HG-GLA is meaningful.

The results most central to the present work concern the performance of RRIP relative to RIP. As shown in (11), the difference in performance between RIP/OT-GLA and RRIP/OT-GLA is substantial: the gain in performance is apparent across parameter settings and is about 25 percentage points. Whereas RIP/OT-GLA learns only 56-58% of the languages correctly, RRIP/OT-GLA learns between 80% and 84% of languages correctly, depending on learning rate. This difference in performance between RIP/OT-GLA and RRIP/OT-GLA is substantial and highly significant across all learning rates ($p < .001$; Welch two sample t-test). RRIP/OT-GLA appears to be a bit more sensitive than RIP/OT-GLA to learning rate, with better performance at lower learning rates. Despite the increase in performance for OT-GLA, RRIP/OT-GLA nonetheless performs worse than RRIP/HG-GLA; this difference is significant at all learning rates ($p < .05$; Welch two sample t-test). Finally, the effect of using RRIP rather than RIP with HG-GLA is less clear. Performance of RRIP/HG-GLA is numerically higher than RIP/HG-GLA, but the difference is small and significant only at learning rate .25 ($p < 0.05$; Welch two sample t-test). Based on these parameter settings, therefore, it is not entirely clear whether RRIP/HG-GLA is an improvement over RIP/HG-GLA. However, Section 4.4 explores additional parameter settings for HG-GLA where the difference in performance between RIP and RRIP becomes clearer.

3.3 Discussion

This section has identified a problem with Robust Interpretive Parsing as first formulated for RIP/OT-GLA by (Boersma 2003; Apoussidou and Boersma 2003). It has also introduced a simple modification to the algorithm, Resampling RIP, and illustrated the effects of this modification for both OT-GLA and HG-GLA at a variety of settings. Resampling RIP dramatically affects the performance of OT-GLA, improving learning success rates as compared to RIP by about 25-30%, depending on learning rate. RRIP also results in some

minimal improvement for the HG-GLA, although this improvement is not significant at all learning rates. In addition to improving performance, RRIP is also potentially more efficient than RIP because it eliminates the need to parse every observed datum, instead parsing only in case of errors. It does have to resample a new grammar when there is an error, but the added computational cost of resampling is very small. Overall, RRIP presents significant advantages over RIP, especially where success rates of OT-GLA are concerned.

With respect to the computational merits of HG relative to OT, the disproportionate improvement of RRIP over RIP for OT means that RRIP substantially narrows the performance gap between Stochastic OT and Noisy HG. Although HG-GLA still outperforms OT-GLA under the new interpretive parsing formulation, the difference is much smaller, shrinking from a gap of about 30 percentage points under RIP to one of about 5 or 6 under RRIP. Thus, RRIP significantly weakens the advantage for HG over OT suggested by Boersma and Pater's results.

These results raise the question of whether there is some parsing strategy that could yield even greater improvements in performance – is RRIP making full use of the learner's stochastic grammatical knowledge? It turns out the answer is no: despite its dramatic improvements over RIP, RRIP is not making full use of the learner's stochastic grammatical knowledge. This is the main topic of the next section, which introduces a more general problem with RIP and RRIP and a new algorithm that addresses it. Section 4 also addresses several additional questions, including why RRIP/HG-GLA shows so little improvement over RIP/HG-GLA.

4. Problem 2: Parsing – Production Mismatch

The previous section has shown that the original formulation of RIP uses a suboptimal procedure for interpretive parsing. The section introduced an alternative to RIP, one that capitalizes on the richness of the learner's stochastic grammatical knowledge by indirectly referencing the distribution of rankings or weightings represented by the learner's stochastic grammar. This distribution provides a rich source of information about learner's grammatical knowledge. It is more than a single ranking or weighting – it represents the learner's confidence in each of the possible rankings or weightings, a confidence that shifts gradually during learning towards the target grammar. At any given point during learning, this distribution provides the most complete information about the learner's grammatical knowledge, the learner's current best guess about the relative likelihood of various rankings or weightings. The rich information contained in this distribution is exploited by the interpretive parsing component of the algorithm by (re)sampling rankings or weightings and using these to select the most harmonic parse. Sampling makes use of the distribution indirectly without ever explicitly calculating the probability with which parses are generated.

Viewing interpretive parsing in this way, as sampling from a probability distribution over parses under the current grammar, reveals a more general problem with RIP. Put simply, Robust Interpretive Parsing is problematic because it creates a mismatch between production and interpretive parsing. The probability with which RIP selects parses is distinct from the probability of those parses under the grammar, and this parsing-production mismatch is the source of the second problem for RIP. This section first discusses this mismatch in more depth, and then proposes an alternative parsing strategy that samples parses according to their relative probability. Finally, the section presents the results of OT-GLA and HG-GLA simulations using the alternative parsing strategy.

4.1 The Mismatch

The parsing-production mismatch is equally relevant for RIP and RRIP. Both procedures rely on selecting a ranking or weighting randomly according to the stochastic grammar, and then using that ranking or weighting to identify the most harmonic of the possible parses. The problem is that the probability with which this two-step parsing process generates parses does not match the relative probabilities of producing those parses under the stochastic grammar.

To see this mismatch, consider the example in (12), in which stress and footing is assigned to the input /L L L/. Suppose the learner's current Stochastic OT grammar has a tie of ALLFEETLEFT and ALLFEETRIGHT, for example by ranking both constraints at 300, and the next highest-ranked constraint is TROCHAIC at 200, and finally IAMBIC at 100. Such a Stochastic OT grammar will generate two rankings with equal probability: ALLFEETLEFT » ALLFEETRIGHT » TROCHAIC » IAMBIC and ALLFEETRIGHT » ALLFEETLEFT » TROCHAIC » IAMBIC. Assuming a standard noise setting of 2, all other rankings will receive probability so low as to be zero for practical purposes. The two rankings generate two distinct outputs. If ALLFEETLEFT is highest ranked, the output will be [(L1 L) L], candidate (a), whereas if ALLFEETRIGHT is highest ranked, the output will be [L (L1 L)], candidate (c). Thus, this grammar generates each of [(L1 L) L] and [L (L1 L)] approximately 50% of the time. Note that these outputs have different stress patterns and that each stress pattern is generated with a unique structure. In particular, the grammar generates the pattern [L L1 L] using right-aligned trochees: [L (L1 L)].

(12) Example Illustrating Parsing-Production Mismatch in OT

/L L L/	ALLFEETLEFT	ALLFEETRIGHT	TROCHAIC	IAMBIC
a. (L1 L) L		*		*
b. (L L1) L		*	*	
c. L (L1 L)	*			*
d. L (L L1)	*		*	

Suppose the learning datum is [L L1 L] and the learner must use its stochastic grammar to parse this datum. To perform Robust Interpretive Parsing, the learner restricts optimization to shaded candidates (b) and (c), those that match the overt stress pattern. Given [L L1 L], what parses does RIP generate? The generated parse depends on the relative ranking of ALLFEETLEFT and ALLFEETRIGHT, with both relative rankings occurring about 50% of the time under the current grammar. Thus, about 50% of the time RIP will select the parse [(L L1) L], candidate (b), and about 50% of the time RIP will select the parse [L (L1 L)], candidate (c). In other words, RIP identifies two possible parses for [L L1 L] and generates them with equal probability. The problem is that, as discussed above, the grammar only generates [L L1 L] using one of these parses, namely [L (L1 L)]. In other words, according to the learner's current grammar, [L (L1 L)] is the only possible parse of [L L1 L], but RIP fails to reflect this categorical restriction imposed by the grammar. Put differently, the learner's current grammar already reflects the ranking conditions inherent to selecting trochees. After all, the grammar ranks TROCHAIC above IAMBIC, indicating its preference for trochees. RIP, however, fails to capitalize on this knowledge during interpretive parsing by allowing incompatible parses to be generated.[7]

The failures of RIP to rely on accumulated grammatical knowledge are not restricted to categorical ranking information. Suppose that TROCHAIC, instead of being strictly ranked above IAMBIC, is partially tied with it so that the probability of TROCHAIC » IAMBIC is 90%

[7] The example discussed in this paragraph also holds for the version of RIP/EDCD for Classic OT with 'permuting ties' (Boersma 2009).

while the opposite ranking occurs 10% of the time. This grammar, which is otherwise the same as the one above, has some probability of generating candidates (b) and (d). In particular, this grammar generates each of (a) and (c) with 45% probability and each of (b) and (d) with 5% probability. Thus, the new grammar generates [L L1 L] using two different structures, but it considers the right-aligned parse to be 9 times more likely than the left-aligned parse. All the same, RIP is oblivious to this statistical preference. When choosing between candidates (b) and (c), the relative ranking of ALLFEETLEFT and ALLFEETRIGHT is decisive, making the relative ranking of TROCHAIC and IAMBIC irrelevant. Since ALLFEETLEFT is highest ranked 50% of the time, parse [(L L1) L], candidate (b), is selected 50% of the time. Thus, although in production the grammar generates [(L L1) L] 9 times less often than it generates [L (L1 L)], in parsing RIP generates both structures equally often. In this example, RIP fails to reflect the soft preferences inherent to the learner's current grammar in the same way that it failed to reflect the categorical restrictions imposed by the grammar in the example above.

In this way RIP fails to reflect the probability distribution the grammar assigns to the data. This occurs because by restricting the set of candidates, RIP changes the constraint interactions that are relevant for selecting candidates. The set of candidates consistent with an overt form is generally a small subset of the full space of candidates. This smaller space changes the relative probabilities of the parses because fewer constraint interactions are relevant for selecting optima among this reduced set of candidates. In the example above, the presence or absence of candidates (a) and (d) alters the constraint interactions. In production, the relative ranking of lower-ranked foot form constraints is crucial, but in parsing, the absence of (a) and (d) means foot form constraints are irrelevant. Ranking conditions crucially rely on the set of alternatives, and when RIP alters this set, it alters how likely the various parses are *relative to one another*. Clearly, any parsing procedure must alter the *absolute* probabilities of candidate parses if the production grammar assigns any probability to forms with other stress contours. The problem, however, concerns the *relative* probability assigned to competing parses when mismatching candidates are added or removed from the competition. In the second example above, candidates (b) and (c) are generated with 5% and 45% probability, respectively, in production, a relative proportion of 1 to 9, but RIP generates both parses with equal probabilities of 50%, a proportion of 1 to 1. In this way, reducing the candidate set can dramatically alter the relative probabilities of candidate parses from their relative probabilities in production.

This mismatch in the *relative* probabilities is the problem and occurs in RIP versions of both Stochastic OT and Noisy HG because in both frameworks the relative probability of two candidates depends on what else is in the candidate set. Interestingly, the mismatch appears to be less severe for probabilistic weighting than for probabilistic ranking, at least for the kinds of constraint interactions and hidden structures typically found in stress systems. Consider again the example in (12), this time in a weighted grammar. That is, suppose the learner's current Noisy HG grammar weights ALLFEETLEFT and ALLFEETRIGHT at 300, TROCHAIC at 200, and IAMBIC at 100 as shown in (13). In contrast to the analogous example in OT, the Noisy HG grammar does not exhibit a severe mismatch. Just as in OT, this grammar generates candidates (a) and (c) with equal probability, roughly 50%, in production. The weightings of the foot form constraints limit candidates to those parsed with trochees, while noise allows ALLFEETLEFT and ALLFEETRIGHT to vary in relative weighting, producing variation between left- and right-aligned trochees. Unlike in OT, however, RIP selects candidate (c) with near certainty as the parse for [L L1 L]. This is because in HG, what matters is the overall difference in the weighted sum of violations between the two candidates. The foot form constraints, weighted 100 points apart, are decisive in parsing, just as they are in production. Although the alignment constraints are highly weighted, their

relative weighting is of little consequence because it is the difference in overall harmony that determines the winner. Noise cannot overwhelm the 100 point harmony difference between candidates (b) and (c). Therefore, in this example RIP succeeds in making use of categorical grammatical knowledge favoring trochees.

(13) Example Illustrating Lack of Parsing-Production Mismatch in HG

	300	300	200	100	
/L L L/	AFL	AFR	TROCH	IAMB	
a. (L1 L) L		–1		–1	(–1)**w*(AFR) + (–1)**w*(IAMB) = –400
b. (L L1) L		–1	–1		(–1)**w*(AFR) + (–1)**w*(TROCH) = –500
c. L (L1 L)	–1			–1	(–1)**w*(AFL) + (–1)**w*(IAMB) = –400
d. L (L L1)	–1		–1		(–1)**w*(AFL) + (–1)**w*(TROCH) = –500

Probabilistic weighting is not totally immune to the parsing-production mismatch problem, however. Mismatches do occur, especially when decisive constraints are variably weighted or when variably weighted constraints have multiple violations. For example, in a case identical to (13) except with the weight of IAMB set to 198 so foot form constraints can vary in weight, there is a statistical mismatch[8]. This weighting produces candidates (a), (b), (c), and (d) with approximately 42%, 8%, 42%, and 8% probability, respectively. Therefore, in production, candidate (c) is about seven times as likely as candidate (b). RIP, however, selects candidate (c) about 76% of the time, which is about three times as often as candidate (b). This example is qualitatively similar to the second OT example above: a strong statistical preference for trochees in production is not fully respected in parsing. Quantitatively, the mismatch is not as severe as in OT, however, since RIP merely weakens the statistical preference without eliminating it entirely. In general, strict ranking creates more opportunities for the parsing-production mismatch because crucial relative rankings can be made completely irrelevant when the candidate set is reduced (as in example (12)). On the other hand, in HG all differences in weighting, even for low-weighted constraints, are always relevant for both the full and reduced set of candidates. Nevertheless, statistical mismatches occur in both frameworks because the presence or absence of alternative candidates affects the relative probabilities with which parses are selected as optimal.

The above discussion has implications for the learning of structural ambiguity in a variant of stochastic weighted grammars known as Maximum Entropy Grammars (Johnson 2002; Goldwater and Johnson 2003; Jäger 2007). As discussed above, the relative probability of two candidates in Stochastic OT and Noisy HG depends on what other candidates are considered. This is not so in Maximum Entropy Grammars. In Maximum Entropy Grammars the probability of a candidate is calculated directly from the harmony: specifically, it is proportional to the exponential of the candidate's harmony. Since the harmony does not depend on other candidates, the relative probabilities of two candidates are fixed, regardless of what other candidates are considered. This means that, by the very definition of candidate probability in these models, Maximum Entropy Grammars are not subject to the parsing-production mismatch problem. RRIP for Maximum Entropy Grammars will therefore select parses according to their production probabilities. The consequences of this observation for learning in the face of structural ambiguity are not explicitly explored here, but it means that future work must also consider RRIP versions of Maximum Entropy Grammars.

RIP is attractive because it provides a simple way to use the learner's current grammar for both production and parsing. However, as this section has shown, using the

[8] These probabilities were estimated by simulating production and parsing for this tableau 1,000,000 times.

same procedure for parsing and production actually results in incompatible parsing and production distributions for both OT and HG and a failure to fully exploit existing grammatical knowledge during interpretive parsing. The following sections explore the consequences of this mismatch in more depth by introducing and testing learning models without a parsing-production mismatch.

4.2 Expected Interpretive Parsing

How could the learner better exploit its current stochastic grammatical knowledge during parsing? A general solution to this question comes from the statistical learning literature. The well-known Expectation-Maximization (EM) algorithm (Dempster, Laird and Rubin 1977) is a general procedure for learning in the face of hidden structure and has been applied to learning with a variety of language models. Formally, learning in EM involves maximizing *likelihood*, which is a measure of fitness, quantifying the grammar's ability to generate the language data. A major advantage of EM is that, with each iteration of learning (where an iteration is a pass through the learning data), the learner is guaranteed to improve its ability to generate the learning data (or to converge). Thus, EM provides a general approach to hidden structure that indicates how to learn from ambiguous data so as to improve the overall fitness of the grammar based on the most complete information available in the current stochastic grammar. Informally, EM uses its current grammar to parse the learning data, assigning parses in proportion to their probability under the grammar. It then updates the grammar based on these 'expected' parses, rewarding grammars that are capable of generating the data with high likelihood. The essential insight of EM is that parses should be weighted by the learner in proportion to the probability with which the current grammar generates them.

In fact, EM was Tesar and Smolensky's original inspiration for RIP, which they re-interpreted for a non-probabilistic Classic OT setting. The stochastic grammar setting provides the opportunity to make this connection tighter since probability is an essential component of EM's solution to hidden structure. However, as the previous section shows, the original formulation of RIP for the stochastic setting does not fully exploit the connection with EM since in RIP/OT-GLA and RIP/HG-GLA parses are not selected according to their probability under the grammar. Other work within the probabilistic constraint ranking setting does explore this connection: Jarosz (2006a; 2006b) adapted EM to the problem of learning probabilistic OT grammars. Jarosz illustrated the capacity of these models to deal with hidden structure in the general case, learning both structural ambiguity and underlying representations in several computational case studies. However, the full EM calculations are computationally costly, involving explicit enumeration of and calculation of the expected probabilities of possible parses (or their components) for all the learning data in batch. The present work investigates an alternative solution that harnesses the basic insight underlying Jarosz's model and EM's solution to hidden structure but relies on the sampling strategy inherent to the online error-driven learners, OT-GLA and HG-GLA.

EM's basic insight can be applied to OT-GLA and HG-GLA simply by selecting parses in proportion to their probability under the learner's current grammar. The formulation in (14) defines this new parsing algorithm for GLA, Expected Interpretive Parsing (EIP). This formulation is identical to the one for RRIP in (10), except that resampling and Robust Interpretive Parsing steps are replaced with sampling from the conditional probability of the parse given the current grammar and the learning datum: P(parse | G_i, *d*) (step 2.d.i.). Thus, EIP defines the probability according to which parses should be sampled as the probability of those parses under the current grammar, the relative production probabilities discussed in the previous section. Consider again the example in (12) and the grammar, call it G_A, with FOOTNONFINAL » IAMBIC. For this grammar and the overt datum [L L1 L], the probability of the trochaic parse, P([L (L1 L)] | G_A, [L L1 L]), is 100%, while the probability of the iambic

parse, P([(L L1) L] | G_A, [L L1 L]) is 0%. For the second grammar discussed earlier, with FOOT-NONFINAL » IAMBIC only 90% of the time, these parsing probabilities would be 90% and 10%, respectively.

(14) Expected Interpretive Parsing (EIP) for GLA

1. Initialize Stochastic Grammar: G_0
2. Iterate over *d* in *D*:
    a. Sample $G' \sim G_i$
    b. Input = uf(*d*)
    c. Output = $\text{Optimise}_{G'}$ (Input)
    d. If overt(Output) ≠ *d*:
        i. Parse ~ P(parse | G_i, *d*)
        ii. G_{i+1} = Update (G_i, Parse, Output)

In EIP, interpretive parsing is defined as sampling from these distributions. Whereas in true EM parses (or their components) are enumerated, and the grammar update is weighted by this probability, in EIP this weighting is accomplished indirectly by repeated sampling from this distribution. More probable parses are sampled more often, causing their respective updates to the grammar to be made more often. In sum, EIP addresses the parsing-production mismatch problem discussed in the last section by defining the parsing procedure in terms of the current production grammar. In this way, EIP resurrects the connection with EM that originally inspired RIP, applying EM's approach to hidden structure to two online, sampling learning algorithms for stochastic constraint-based grammars.

4.3 Expected Interpretive Parsing Simulations and Results

The previous sections have shown that (R)RIP does not parse according to the current (production) grammar and have introduced an algorithm, Expected Interpretive Parsing, that does parse according to the current grammar. This section presents simulations illustrating how parsing consistently with the learner's production affects performance.

As discussed earlier, an advantage of RIP is that parsing can be accomplished by using standard optimization, albeit with a different candidate set. Sampling parses from the conditional probability P(parse | G_i, d) requires another computational approach. The simulations presented here use a Monte Carlo method of rejection sampling relying on the learner's production module. Specifically, in case of an error on datum *d*, the learner's current grammar is used to generate output forms for the underlying form of *d*. As soon as one of the output forms matches the overt stress pattern of *d*, it is used as the parse. Outputs that do not match the surface stress pattern of *d* are discarded. By using only those samples that match the overt stress pattern, this method effectively samples from the conditional probability of various parses of *d*, given its overt stress pattern and the current grammar, exactly the probability distribution required for EIP.

Consider once again the example in (12) and the grammar with FOOTNONFINAL » IAMBIC. If the learner needs to generate a parse for [L L1 L], the learner will simply use the current grammar to generate outputs for /L L L/ until one matches [L L1 L], in which case that output will be used as the parse. Recall that this grammar generates each of [(L1 L) L] and [L (L1 L)] about 50% of the time. Since only [L (L1 L)] matches the stress pattern of the datum and only outputs matching the overt stress pattern are used as parses, the sampler generates outputs until it generates [L (L1 L)], and then uses this output as its parse. In this example, the chance of parsing [L L1 L] as [L (L1 L)] is therefore 100%, as required by EIP. In the example grammar with FOOTNONFINAL » IAMBIC only 90% of the time, the grammar again has a 50% of generating [L L1 L], but this grammar generates [L L1 L] with two

different parses. It generates [L (L1 L)] with 45% probability and [(L L1) L] with 5% probability. Since the sampler only keeps those outputs that match the stress pattern, its chance of parsing [L L1 L] as [L (L1 L)] is 45/50, or 90%, and its chance of parsing [L L1 L] as [(L L1) L] is 5/50, or 10%. In both examples, since the probability of selecting the overt stress pattern is 50%, the sampler has a 50% chance each time it generates an output of keeping it as the parse.

In general, the number of outputs that need to be generated until a match is found depends directly on the probability of the overt form under the grammar. The more likely the overt form is according to the current grammar, the more quickly a parse for it will be found on average. This parsing method thus has the advantage that parsing gets more efficient as learning continues and the learner's grammar becomes more and more likely to generate the surface stress pattern of the target language. However, if the learner's grammar generates the overt form with very low probability, it can take many samples before a match is found, meaning that selecting a parse can require more effort when the learner's current grammar is very far from the target grammar.

One further detail is crucial in implementing this sampling method for EIP. When the learner converges on a grammar that is incorrect for the target language, that is, when learning is unsuccessful, the learned grammar often fails to generate one or two of the data forms in the language entirely, assigning (near) zero probability to them. Without some terminating condition, the sampler will loop indefinitely in such cases since it cannot generate a matching stress pattern for such data forms. In the simulations reported here, this problem was avoided by allowing each form a maximum of 1000 samples from the grammar to generate a parse. If 1000 samples failed to provide a parse of the datum, the learner did not make an update to their grammar and did not learn from this error[9]. The remaining simulation details are identical to the earlier set-up for RIP and RRIP. All learning rate/algorithm combinations were repeated 10 times for each of the 124 languages (constituting a total of 9920 separate runs of EIP), as before, with average success rates and sample standard deviations calculated.

The performance results of EIP for OT-GLA and HG-GLA at the four learning rates are shown in (15). The results for RIP and RRIP are repeated from (11) for convenience. Comparing EIP/OT-GLA to the other OT-GLA variants reveals that Expected Interpretive Parsing dramatically improves performance over RIP and RRIP. Whereas RRIP's performance of roughly 81-84% is dramatically higher than RIP's performance of roughly 56-58%, EIP's performance of about 93-94% is substantially higher than both RIP and RRIP. On average EIP improves upon the performance of RRIP by about 10 percentage points, and this difference is highly significant at all learning rates ($p < .001$; Welch two sample t-test). For the HG learners, examination of EIP/HG-GLA's performance at the parameter settings tested indicates EIP/HG-GLA performs comparably to RRIP/HG-GLA and RIP/HG-GLA. Comparing performance of EIP/OT-GLA and EIP/HG-GLA suggests that HG-GLA no longer has an edge over OT-GLA. Indeed, at the parameters tested here, EIP/OT-GLA's performance is better than EIP/HG-GLA's performance by several points. However, the following sections explore HG-GLA's performance with different parsing strategies at a wider range of parameter settings, showing that the above results do not represent HG-GLA's best performance.

[9] 1000 was chosen based on initial testing indicating that this was a sufficient sample to reliably produce parses for the learning data. In general, a higher cut-off increases the chances that the learner will be able to parse and learn from each learning datum, but it also increases processing time early in learning and for unsuccessful runs.

(15) Success Rate (SD) of RIP, RRIP, and EIP Across Various Parameter Settings

Algorithm	Learning Rate (plasticity)			
	.05	.10	.25	.50
RIP/OT-GLA	55.81 (1.82)	56.13 (1.62)	56.21 (2.15)	**57.50** (2.28)
RIP/HG-GLA	**88.79** (0.97)	88.71 (0.66)	85.48 (1.57)	82.90 (2.92)
RRIP/OT-GLA	**84.19** (1.91)	82.58 (1.91)	81.13 (2.29)	80.08 (3.09)
RRIP/HG-GLA	**89.44** (0.71)	89.27 (0.94)	87.58 (1.79)	82.98 (1.84)
EIP/OT-GLA	93.87 (0.78)	**93.95 (0.57)**	93.71 (1.69)	92.82 (1.29)
EIP/HG-GLA	88.23 (0.56)	**88.31 (1.02)**	85.56 (1.96)	83.23 (2.57)

4.4 Parameter Settings and Probabilistic Ranking Versus Weighting

One puzzling aspect of the above results is the apparent lack of effect of parsing strategy on the HG-GLA. The best performance of HG-GLA reported above is about 88-89% regardless of parsing strategy. One aspect of this puzzle is why EIP/HG-GLA fails to improve over RRIP/HG-GLA. A possible answer to this question was alluded to in the earlier discussion of the parsing-production mismatch. The examples of the mismatches discussed earlier were much more severe for OT than for HG. If these examples are representative of the kinds of constraint interactions that typically occur in stress systems such as this one, mismatches may be dramatically less pervasive in HG than in OT on typical learning trials. Weak mismatches would cause RRIP to perform similarly to EIP since the only difference between these two algorithms is the distribution according to which parses are sampled. Since RIP and EIP perform comparably, the results are largely consistent with this possibility. However, if the parsing-production mismatch in HG is weak, rather than non-existent, some improvement would be expected from EIP. Furthermore, the parsing-production mismatch does not explain why RIP/HG-GLA and RRIP/HG-GLA perform comparably. Why does parsing strategy seem to have no effect whatsoever on HG-GLA? This section considers this question more carefully and reveals that part of the explanation lies in the differences between how the parameter settings affect learning in HG-GLA as opposed to OT-GLA.

The simulations presented here include a range of parameter settings for all the algorithms. These settings are a reasonable starting point for exploring how parameter settings affect performance in the various algorithms. Although the results presented here consider the same range of parameter setting for Stochastic OT and Noisy HG algorithms, this range provides a more complete picture of the performance range for Stochastic OT than it does for Noisy HG. There are three parameters to set, initial weights or ranking values, learning rate, and noise. If all the constraints are initially tied, as they are in these simulations, the initial ranking values for the OT learners are irrelevant since all that matters is whether one constraint ranks higher or lower than another – its numerical value is not important. In other words, if noise is held constant, it does not matter whether the ranking values of two constraints are 1 and 10 or 1001 and 1010. Therefore, for Stochastic OT, if constraints start out tied, there are only two meaningful parameters: noise and learning rate. It is possible to hold one parameter constant and vary the other to get a good idea of the performance range. This is exactly what the current results reflect. The same range of performance would be expected for higher/lower noise values, with best performance shifted to higher/lower learning rate values. Since the best performance for EIP/OT-GLA falls at one of the intermediate parameter values tested, it is likely that this combination of parameter settings is close to the best possible performance EIP/OT-GLA can get on this test set. The best performance of RIP/OT-GLA shown in the results above falls at the highest learning rate tested. To determine whether performance of RIP/OT-GLA improves at even higher plasticity values, two additional parameter settings were tested with 10 runs each: plasticity of 1.0 and plasticity of 2.0. The results indicate higher learning rates cannot save RIP/OT-

GLA – performance at these higher learning rates is on average 57.18 and 56.35, respectively. It is possible that some plasticity value intermediate between the range of .25 and 2.0 would slightly improve performance, but the results indicate that 57% is a good estimate of the best performance for RIP/OT-GLA on this test set. Finally, performance of RRIP/OT-GLA is highest at the lowest learning rate so it is worth checking whether even lower learning rates would improve performance. In fact, they do not – performance at a learning rate of .01 drops to 83.17%. Overall, the results for RIP, RRIP, and EIP versions of the OT-GLA presented here provide good estimates of the best performance that can be expected from these algorithms on this test set. This means we can be reasonably confident that performance gains of EIP/OT-GLA over RRIP/OT-GLA and RRIP/OT-GLA over RIP/OT-GLA are robust and representative of the range of possible parameter settings.

In contrast, the performance range for the variants of the HG-GLA algorithms presented here is likely not to be similarly comprehensive. This is because the current results represent a smaller portion of the space of crucial parameter combinations for HG-GLA simply because more parameter settings are crucial for the HG-GLA. Indeed, all three parameter settings – initial weights, noise, and plasticity – have consequences for the HG-GLA, and their possible combinations create a large space of possibilities. A weighting difference of 10 and 1 is not equivalent to a weighting difference of 1010 and 1001. This is because harmony in HG depends on a multiplicative weighting and constraint violation relationship. Consider the simple example in (16). Each violation of C2 incurred by candidate A is added to the candidate's total score. This means in order for candidate A to win, the weight of C1 must be more than three times that of the weight of C2. Therefore, if the weights of C1 and C2 are 10 and 1, respectively, candidate A is optimal. However, if the weights of C1 and C2 are 1010 and 1001, respectively, candidate B is optimal. Thus, because optimality is determined in terms of these multiplicative weighting interactions in HG, the absolute weight assigned to a constraints initially during learning matters. If the learner starts out with C1 and C2 weighed at 1, the weights will only have to shift by a total of about 2 for candidate A to be selected as optimal. In contrast, if the learner begins with both constraints weighted at 100, it will take many more iterations (or a much higher learning rate) to reach a weighting with C1 weighted three times higher than C2. The metrical stress system investigated here involves forms with multiple feet, stresses, and syllables, and therefore candidates with multiple violations for constraints are very common, often yielding multiplicative weighting interactions.

(16) Example Illustrating HG's Sensitivity to Absolute Weights

		3	1	
		C1	C2	
	a. Candidate A		-3	$0*w(C1) + (-3)*w(C2) = -3$
	b. Candidate B	-1		$(-1)*w(C1) + 0*w(C2) = -3$

A further consequence of this multiplicative interaction is that Noisy HG is less sensitive to evaluation noise, and the interaction between noise and weighting is more complex. Consider again the example above and suppose both C1 and C2 are weighted or ranked equally at 10. In Stochastic OT, regardless of the noise setting, candidates A and B are equally likely to be generated. Not so in Noisy HG. In Noisy HG with a standard noise setting of 2, the chances that the weight of C1 will randomly be selected to be three times higher than C2 are very low. This means Noisy HG will almost always generate candidate B. Note that this interacts in a nontrivial way with the weighting value: if the constraints are both tied at 100 rather than 10 with noise still set to 2, the probability of generating candidate A becomes vanishingly small.

Since the test system investigated here does involve a number of gradient constraints that frequently assign multiple violations, the effect of a given level of noise should be weaker for Noisy HG grammars than for Stochastic OT grammars. If this is true, it is possible that the lack of an effect of the parsing mechanism for HG-GLA could be due to a lack of sufficient noise. If, despite the noise, the Noisy HG grammars are consistently selecting the same candidates as the winners, there would be no effect of resampling and no effect of parsing strategy. In other words, if Noisy HG is so insensitive to noise as to assign nearly all probability to a single candidate, the parsing strategy should be largely inconsequential.

This possibility was investigated in two ways. First, the expected interpretive parsing outcomes of Stochastic OT and Noisy HG were qualitatively examined at the initial weighting/ranking of 10 and noise of 2, with learning updates turned off. This was done in order to determine whether the choice of OT vs. HG affected the variability in parses. A grammar with all constraints tied generates a wide range of rankings and weightings and provides an opportunity to observe the variability in parsing outcomes that operates in all target languages before any learning has taken place. In Stochastic OT, there was indeed much variation between the parses selected for the overt forms. In contrast, parses selected by the Noisy HG grammar were much more consistent, with most overt forms being consistently parsed in only one way. Thus, it is clear that the same level of noise results in much more parsing variation in Stochastic OT than in Noisy HG. This raises the question of how performance of HG-GLA would be affected if noise were increased. To investigate this, additional runs of each of the HG-GLA variants were performed at noise settings of 4 and 8. All runs used the learning rate 0.1 and were set up as before with the initial weights of 10. As before, each run was repeated 10 times with the average and sample standard deviation calculated.

(17) Success Rate (SD) of RIP, RRIP, and EIP Across Noise Settings

Algorithm	Noise		
	2	4	8
RIP/HG-GLA	88.71 (0.66)	**91.05** (1.11)	90.89 (1.14)
RRIP/HG-GLA	89.27 (0.94)	**92.42** (0.78)	90.97 (1.60)
EIP/HG-GLA	88.31 (1.02)	**94.19** (0.74)	91.94 (1.26)

The results are summarized in (17), with the previous results of HG-GLA with noise set to 2 repeated for convenience. These results are consistent with the hypothesis discussed earlier, namely, that it is the lack of noisiness causing the HG-GLA algorithms to show no effect of parsing strategy. When noise is increased to 4, EIP does yield an improvement in performance over RRIP, and RRIP does yield an improvement over RIP. These gains in performance are relatively small, but they are both highly significant ($p < .01$; Welch two sample t-test). This means that better parsing can yield better performance for both HG-GLA and OT-GLA. Furthermore, the results also indicate that, for a learning rate of 0.1 and initial weights of 10, overall performance is best at noise set to around 4, with lower performance when noise is increased to 8. These results increase the upper range of the performance for the HG-GLA algorithms presented in the previous section (and in previous work). Importantly, the performance of EIP/HG-GLA at noise set to 4 is comparable to the best performance of EIP/OT-GLA ($p > .1$; Welch two sample t-test).

In sum, the discussion and simulations in this section showed that HG-GLA is highly sensitive to the combinations of parameter settings. While varying one parameter in the OT-GLA provides a fairly comprehensive view of its range of performance, the space of crucial parameter settings is more complex for the HG-GLA and requires more extensive exploration. Additional exploration of this space revealed that the amount of variability

produced at a noise setting of 2 in this stress system is quite limited. This explains why the choice of parsing strategy has little effect at this noise setting. At higher noise settings, differences between the parsing strategies emerge and indicate that EIP does improve performance over RRIP, which in turn improves over RIP. The gains in the performance are small compared to the gains for OT-GLA, but they nonetheless indicate that parsing strategy does matter for HG-GLA when the noise level is high enough to produce some variability in parses. Parsing with a known loser (RIP) leads to poorer performance than resampling (RRIP). Likewise, the parsing-production mismatch problem does exist for Noisy HG, albeit weakly, and leads to poorer performance of RRIP compared to EIP. As discussed above, the interactions of parameter settings for HG-GLA are complex, and further work is needed to provide a comprehensive picture of how the performance of HG-GLA in the domain of hidden structure depends on the various parameter settings, the types of constraints (gradient or binary), and their interactions.

4.5 Discussion

The preceding sections introduced a parsing strategy, Expected Interpretive Parsing, and explored the consequences of parsing consistently with production for both Stochastic OT and Noisy HG. The results of the simulations revealed that EIP/OT-GLA is a substantial improvement over RRIP/OT-GLA. Recall that the only difference between RRIP and EIP is the distribution according to which parses are sampled. Therefore, the extent to which EIP performs differently from RRIP is an indication of the extent to which RRIP is not doing expected parsing. Since there is substantial difference in performance, it is clear that on average parses generated by RRIP for the OT-GLA are very different from parses generated by EIP for OT-GLA. Therefore, these simulations indicate that the parsing-production mismatch is a pervasive and serious problem for RIP versions of the OT-GLA, and not just artefacts observed in rare and artificially constructed examples, such as the ones discussed in Section 4.1. The results also revealed that, due to a difference in constraint interaction, Noisy HG produces less variability during parsing (and production) than Stochastic OT for the same level of noise. When noise is increased to produce some variation in parsing, EIP/HG-GLA improves significantly over RRIP/HG-GLA. This indicates that the parsing-production mismatch problem also exists for Noisy HG, just as it does for Stochastic OT, although the severity of the mismatch appears to be weaker in HG, at least for the kinds of constraint interactions that occur in the stress system examined here. The preceding sections have identified the parsing-production mismatch problem, demonstrated that it has significant consequences for learning in Stochastic OT and Noisy HG, and identified some ways in which OT and HG may differ with regard to this problem. A complete understanding of the ways in which the severity of the mismatch depends on choice of framework, the kinds of constraints, and the kinds of hidden structure is an important question for future work.

The performance gains of EIP as compared to RRIP highlight the severity of the parsing-production mismatch problem, revealing that learning success rates can be substantially improved when parses are sampled according to their conditional probability given the (production) grammar. The preceding sections also introduced a concrete method for generating samples from the conditional probability in practice. As explained earlier, the amount of effort needed to generate a parse using this method of rejection sampling varies, and parsing can be expensive when the grammar generates the overt portion of the datum with low probability. However, the results presented here illustrate that this method can be used successfully in practice. For this fairly complex stress system, a cut off of 1000 output samples is sufficient to demonstrate the substantial gains in performance provided by EIP and at the same time makes it feasible to test tens of thousands of learning runs, as reported above. Overall, the procedure yields a learning algorithm that allows for extensive empirical

testing in its current form. Nonetheless, an important avenue for future work is determining whether it is possible to improve on the computational efficiency of this method, for example by identifying a way to sample directly from the conditional probability. This would require a method that i) uses the current stochastic ranking or weighting, and ii) generates only output forms that match a given overt form in such a way that iii) the relative probabilities of the output forms correspond to their relative production probabilities. A method with all three properties that improves upon the one used here is not readily apparent. Note that RIP satisfies the first two properties, but as the preceding sections have shown, violates the third property. One approach would be to rely on elementary ranking conditions (Prince 2002, Riggle 2009) to constrain parsing. While this possibility is appealing, it is important to keep in mind that in the context of hidden structure, elementary ranking conditions are themselves hidden. A fully structured winner is needed in order to define a ranking condition, and the learner does not know the full structure. Furthermore, even if ranking conditions could be used in some way to limit generation of parses to those that match a given overt form, it is not obvious that property (iii) would be satisfied. In sum, further investigation into alternative procedures for performing EIP is warranted; however, the current rejection sampling method provides a concrete and empirically testable implementation of the algorithm that can be used in practice.

An important motivation for the present work was the comparison between Stochastic OT and Noisy HG with respect to their computational properties. The current simulations explore a sizable portion of the parameter space for both Stochastic OT and Noisy HG using three different learning algorithms. Although previous work and the simulations with RIP presented in Section 3.2 suggest an advantage for Noisy HG, the new algorithms proposed here narrow the gap in performance between OT and HG. Indeed, the results of simulations with EIP do not support a performance advantage for either HG-GLA or OT-GLA. Overall, the results indicate that the suboptimal parsing strategy of RIP affects the OT-GLA disproportionately more than HG-GLA and that once a parsing strategy that takes advantage of the learner's stochastic grammatical knowledge is used, the performance of OT-GLA learners rivals that of HG-GLA learners.

Although the new parsing strategies significantly improve the performance of both HG and OT learners, none of the learners succeed in learning all the languages in the system. A complete explanation for the learners' imperfect performance is beyond the scope of this paper, but a few observations relevant to this question can be made about their failures. While all the learners explored here are non-deterministic, the performance of the individual learning models exhibits a good deal of consistency across multiple runs. That is, for a given combination of framework, parsing strategy, and other parameter settings, the individual runs tend to produce similar results, with some languages consistently learned, some consistently not learned, and others exhibiting variable outcomes. In other words, different target languages appear to be more or less difficult to learn by these models. However, which languages are more or less difficult varies significantly by learner, especially between the OT and HG learners. The HG learners are less consistent in their learning outcomes than the OT learners, and there is no single language that all learners fail to learn[10]. Understanding the

[10] One example language that shows up frequently in the set of failures is a target language that all OT learners fail on. In this language, stress is always penultimate except for disyllabic forms with light initial syllables, in which case stress is final. An analysis of this pattern makes use of non-finality and iambic feet to account for the general pattern of penultimate stress (e.g. […σσ(σ ˈ σ)σ]), and a ranking of FOOTBIN ≫ NONFINALITY to account for the final stress in disyllabic light-initial forms (e.g. [(L ˈ σ)] ≻ [(ˈ L)σ]). Further work is needed to determine the ultimate source of failures on languages such as this one, but one possibility that should be investigated is that the learners can be

formal properties responsible for making some languages harder to learn than others is an important question for future research and could be the key to further improvements in performance.

5. General Discussion

This paper has focused primarily on success rates of learning algorithms in the limit, but there are other computational considerations that should be explored in future work. One important consideration, and one that has played a prominent role in the literature on learnability within Classic OT from the very beginning, is one of algorithmic efficiency. Starting with the earliest work on OT learnability, the learnability results for the Constraint Demotion family of learning algorithms included proofs not only of their correctness but also of their data complexity (Tesar 1995; Tesar and Smolensky 1998). While the data complexity of the CD family of learning algorithms is well-understood, further work is needed to better understand the data complexity of learning models in the face of structural ambiguity. Another important consideration is how much computation time or memory is required to process each data form. The discussions in Sections 3.1 and 4.3 addressed this aspect of the proposed parsing strategies, RRIP and EIP, respectively. Given the more complete picture of these algorithms' performance in the later sections of the paper, the balance between these various considerations is now clearer. As discussed earlier, since RRIP parses data forms and resamples only when there is an error, it requires less effort on average to process each datum than RIP. Furthermore, the results of simulations in Sections 3 and 4 show that RRIP/OT-GLA has better performance than RIP/OT-GLA across the board. In this case, therefore, there appear to be no drawbacks to RRIP over RIP, only significant performance advantages. For HG, the improvement of RRIP over RIP is minimal and not consistently present, but there again seem to be no drawbacks to RRIP and only the possibility of some performance gains, again a win-win situation. Things are not quite as straightforward in the case of EIP, since there are some computational trade-offs with the rejection sampling implementation of EIP parsing. While the procedure potentially requires more processing time for each datum, this added effort is balanced by substantial gains in performance for the OT learners. The processing time per datum can be restricted with good results on this test set; nonetheless, it is an open question whether a more efficient parsing method could push the balance even further in favour of EIP. Because the performance gains of EIP for HG are minimal, the advantage of EIP/HG-GLA over RRIP/HG-GLA is less clear.

In general, this paper has argued that the performance gains afforded by RRIP and EIP are due to these parsing strategies' better reliance on the learner's accumulated grammatical knowledge. If RRIP is making better use of available information than RIP, and EIP is making better use of available information than RRIP, is there some strategy that could make even better use of information than EIP and improve performance even further? This is a question that requires further work, but there are reasons to think the answer is yes. There is often a trade-off between the amount of computation or memory spent on each learning datum and the amount of data needed for successful learning. One way to view what EIP is doing is that it is using its current grammar to estimate the distribution over possible parses of each overt form, and making updates to its grammar based on this distribution. In EIP, the estimation of this distribution is spread out over time since only one sample parse is used to calculate the update to the grammar each time there is an error. It would be possible to extend EIP to take multiple sample parses for each overt form and calculate an update from this

---

led astray by an overwhelming amount of evidence consistent with a preferred (but incorrect) analysis. In this case, 60 of the 62 learning data have penultimate stress, which may be leading the learner to prefer right-aligned trochees.

more complete estimate of the distribution over parses. Such a learner would bring EIP even closer to the original inspiration for RIP, Expectation-Maximization. However, such a learner would also require much more effort to process each overt form. An interesting avenue for further work is developing a model that makes fuller use of the distribution over parses when processing each overt form and exploring the consequences for accuracy and data complexity. Another computational avenue to explore for potential performance improvements is to investigate the role that OT-GLA's known non-convergence (Pater 2008) plays in the context of hidden structure. While OT-GLA works well in most cases (for simulations, see e.g. Boersma and Pater 2008), it may be that performance of RRIP/OT-GLA and EIP/OT-GLA could be further improved if the OT-GLA update rule were replaced with the provably convergent update rule proposed by Magri (2012). In fact, there is forthcoming related work that indicates Magri's update rule does yield improvements in the context of structural ambiguity (Biró to appear). A promising direction for further research is exploring whether the parsing improvements proposed here can yield cumulative improvements in conjunction with Magri's update rule for OT-GLA.

Another important area for further work is developing a better understanding of how general properties of constraints and representations affect learnability in the context of hidden structure. This paper has the examined computational properties of six learning algorithms in a particular stress system. Using this test system allows for direct comparison with previous work (Tesar and Smolensky 2000; Boersma 2003; Boersma and Pater to appear; Jarosz to appear a). It also allowed for further investigation into Boersma and Pater's initial findings suggesting an advantage for HG learners, which were determined on the basis of simulations with this test set. Furthermore, the findings in this paper have highlighted several formal properties of this system that dramatically affect the results and whose consequences should be explored more generally. One of these properties is the parsing-production mismatch problem and how it differs between HG and OT. This paper has identified the mismatch problem and shown that its presence has significant consequences for successful learning. The discussion and simulation results also suggest that the mismatch problem may be more severe for OT than for HG. More generally, however, what properties of constraints, representations, or mode of constraint interaction underlie the parsing-production mismatch problem? Are there restrictions on constraints or representations that could eliminate or reduce the severity of the mismatch? If so, the performance differences between RRIP and EIP would be minimized, bringing the performance of RRIP closer to EIP, eliminating or weakening the need for the more costly parsing computations of EIP. Likewise, if it can be shown that the mismatch problem is less severe in HG than in OT as a direct consequence of weighting, this would be a potential advantage for HG-GLA, and HG, on the problem of learning with structural ambiguity. As discussed earlier in Section 4.1, the parsing production mismatch does not, by definition, exist in Maximum Entropy Grammars. The present findings therefore motivate exploration of learning with structural ambiguity within the Maximum Entropy Grammar framework. Since ME Grammars are not subject to the parsing-production mismatch problem, the application of RRIP to ME Grammars will result in expected parsing, making the more costly computations of EIP unnecessary.

While the parsing-production mismatch clearly plays an important role in the success of these learners, it is not the only obstacle to successful learning (as evidenced by EIP's imperfect performance). Likewise, OT-GLA's non-convergent update rule cannot explain all the unsuccessful runs of EIP/OT-GLA since the HG learners do not succeed on all the languages either. This means there are other obstacles to learnability that must be considered. As discussed earlier, an important question for further work on learning hidden structure is whether successful learning depends on some formal or substantive restrictions on the hypothesis space. While much work on learnability in OT has relied on general learning

strategies independent of the constraints and representations, some recent work within the Classic OT setting explores formal restrictions on the hypothesis space that enable more effective strategies for hidden structure learning (Tesar 2008, 2009). An important avenue for further work is determining whether formal restrictions such as those considered by Tesar can shed light on hidden structure learning in the stochastic setting as well. Another important direction is to explore the relationship between learnability and the properties of structural ambiguity in natural language typology. As discussed in Section 2, the constraints and mode of constraint interaction underlying the Tesar and Smolensky stress system are not uncontroversial (McCarthy 2003; Hyde 2007; Pruitt, Kathryn 2010), and it is important to determine the consequences for learnability of different theoretical assumptions. Thus, further exploration of formal learnability restrictions on the hypothesis space must proceed in conjunction with typological and theoretical work examining the nature of the formal system underlying the kinds of structural ambiguity found in the phonological systems of natural languages. In general, the computational, typological, and theoretical considerations are interconnected, and integrating these approaches promises to contribute to a deeper understanding of the formal and empirical consequences of HG and OT as well as to more successful strategies for dealing with hidden structure and other learnability challenges.

6. Conclusion

In sum, the focus of this work is on an outstanding learnability problem, the problem of learning phonology in the face of structural ambiguity, and how models of learning in this context can be improved and compared. This paper has identified two problems with the formulation of Robust Interpretive Parsing in the stochastic setting, proposed two new parsing strategies, and presented in-depth explorations of the consequences of the parsing strategies for learning algorithms in both HG and OT. The first problem, parsing with a known loser, has an easy solution, RRIP, which dramatically and disproportionately improves performance of OT-GLA. The second problem, the parsing-production mismatch, is a more general issue with RIP in the stochastic setting that appears to affect Stochastic OT more severely than Noisy HG. The proposed solution to this problem, EIP, again substantially and disproportionately improves the performance of OT-GLA. With both parsing improvements, OT-GLA and HG-GLA yield comparable end-state success rates. This in-depth analysis, therefore, indicates that once a broad range of computational approaches is considered, there is no evidence of an advantage for HG in this learning context. These results have theoretical implications since basic questions of learnability are an important consideration in the evaluation of the theoretical frameworks of HG and OT.

This paper also identifies several properties of the stress system that have learnability consequences, which must be investigated further in order to better understand the relative computational merits of OT and HG more generally. In addition, the simulation results highlight differences between Stochastic OT and Noisy HG with respect to the modelling of variation. Variation is a major area of research in theoretical phonology (see Coetzee and Pater 2008b for a review), and these results motivate further exploration of the consequences of these differences for theories of variation. Overall, the results contribute to a deeper understanding of the relative merits of the HG and OT theoretical frameworks and the nature of the learning problem in the context of structural ambiguity.

References

Akers, Crystal. 2011. Simultaneous Learning of Hidden Linguistic Structures. *Simultaneous Learning of Hidden Linguistic Structures*. PhD Dissertation, Rutgers University, New Brunswick, N.J.

Alderete, John, Brasoveanu, Adrian, Merchant, Nazarré, Prince, Alan and Tesar, Bruce. 2005. Contrast Analysis Aids the Learning of Phonological Underlying Forms. In Chung-hye Han and Alexei Kochetov (editor.), *Proceedings of the 24th West Coast Conference on Formal Linguistics*, 34-42. Somerville, MA: Cascadilla Proceedings Project.

Apoussidou, Diana. 2006. On-line learning of underlying forms. *On-line learning of underlying forms*. University of Amsterdam, ms.

Apoussidou, Diana. 2007. The learnability of metrical phonology. *The learnability of metrical phonology*. PhD Dissertation, University of Amsterdam.

Apoussidou, Diana and Boersma, Paul. 2003. The learnability of Latin Stress. *IFA Proceedings* 25. University of Amsterdam. 101-148.

Bane, Max and Riggle, Jason. to appear. The typological consequences of weighted constraints. In *Proceedings of the Forty-Fifth Meeting of the Chicago Linguistic Society (2009).*

Bane, Max, Riggle, Jason and Sonderegger, Morgan. 2010. The VC dimension of constraint-based grammars. *Lingua* 120(5). 1194-1208.

Biró, Tamás. to appear. Towards a Robuster Interpretive Parsing, Learning from overt forms in Optimality Theory. *Journal of Logic, Language and Information.*

Boersma, Paul. 1997. How we learn variation, optionality, and probability. *IFA Proceedings* 21. University of Amsterdam. 43-58.

Boersma, Paul. 2003. Review of Tesar & Smolensky (2000): Learnability in Optimality Theory. *Phonology* 20(3). 436-446.

Boersma, Paul. 2009. Some Correct Error-Driven Versions of the Constraint Demotion Algorithm. *Linguistic Inquiry* 40(4). 667-686.

Boersma, Paul and Hayes, Bruce. 2001. Empirical Tests of the Gradual Learning Algorithm. *Linguistic Inquiry* 32(1). 45-86.

Boersma, Paul and Levelt, Claartje. 2000. Gradual Constraint-Ranking Learning Algorithm Predicts Acquisition Order. In *Proceedings of 30th Child Language Research Forum*, 229-237. Stanford, California: CSLI.

Boersma, Paul and Pater, Joe. to appear. Convergence Properties of a Gradual Learning Algorithm for Harmonic Grammar. In John McCarthy (ed.), *Harmonic Grammar and Harmonic Serialism*. London: Equinox Press.

Chomsky, Noam. 1981. *Lectures on Government and Binding*. *Lectures on Government and Binding*. Dordrecht: Foris.

Coetzee, Andries and Pater, Joe. 2008a. Weighted constraints and gradient restrictions on place co-occurrence in Muna and Arabic. *Natural Language & Linguistic Theory* 26(2). 289-337.

Coetzee, Andries and Pater, Joe. 2008b. The place of variation in phonological theory. In John Goldsmith, Jason Riggle and Alan Yu (editor.), *The Handbook of Phonological Theory*. 2nd. Blackwell.

Daelemans, Walter, Gillis, Steven and Durieux, Gert. 1994. The acquisition of stress: A data-oriented approach. *Computational Linguistics* 20(3). MIT Press. 421-451.

Daland, Robert, Hayes, Bruce, White, James, Garellek, Marc, Davis, Andrea and Norrmann, Ingrid. 2011. Explaining sonority projection effects. *Phonology* 28(02). Cambridge: Cambridge University Press. 197-234.

Dempster, A P, Laird, N M and Rubin, D B. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* 39(1). Blackwell Publishing for the Royal Statistical Society. 1-38.

Dresher, Bezalel Elan. 1999. Charting the Learning Path: Cues to Parameter Setting. *Linguistic Inquiry* 30(1). 27-67.

Dresher, B Elan and Kaye, Jonathan D. 1990. A computational learning model for metrical phonology. *Cognition* 34(2). 137 - 195.

Fischer, Marcus. 2005. A Robbins-Monro type learning algorithm for an entropy maximizing version of Stochastic Optimiality Theory. *A Robbins-Monro type learning algorithm for an entropy maximizing version of Stochastic Optimiality Theory*. Master's Thesis, Humboldt University, Berlin.

Goldrick, Matthew. 2011. Linking Speech Errors and Generative Phonological Theory. *Language and Linguistics Compass* 5(6). Blackwell Publishing Ltd. 397-412.

Goldsmith, John. 1994. A dynamic computational theory of accent systems. In Jennifer Cole and Charles Kisseberth (editor.), *Perspectives in phonology*, 1-28. Stanford: CSLI.

Goldwater, Sharon and Johnson, Mark. 2003. Learning OT constraint rankings using a Maximum Entropy model. In *Proceedings of the Workshop on Variation within Optimality Theory*. Stockholm University.

Gordon, Matthew. 2002. A Factorial Typology of Quantity-Insensitive Stress. *Natural Language & Linguistic Theory* 20(3). 491-552.

Gupta, Prahlad and Touretzky, David S. 1994. Connectionist models and linguistic theory: Investigations of stress systems in language. *Cognitive Science* 18(1). Elsevier. 1-50.

Hammond, Michael. 2004. Gradience, Phonotactics, and the Lexicon in English Phonology. *International Journal of English Studies* 4. 1-24.

Hayes, Bruce. 1995. *Metrical stress theory : principles and case studies*. *Metrical stress theory : principles and case studies*. Chicago: University of Chicago Press.

Hayes, Bruce. 2004. Phonological Acquisition in Optimality Theory: the Early Stages. In René Kager, Joe Pater and Wim Zonneveld (editor.), *Fixing Priorities: Constraints in Phonological Acquisition*, 245-291. Cambridge: Cambridge University Press.

Hayes, Bruce and Londe, Zsuzsa Cziraky. 2006. Stochastic phonological knowledge: the case of Hungarian vowel harmony. *Phonology* 23(01). 59-104.

Hayes, Bruce and Wilson, Colin. 2008. A Maximum Entropy Model of Phonotactics and Phonotactic Learning. *Linguistic Inquiry* 39(3). 379-440.

Hayes, B, Zuraw, K, Siptár, P and Londe, Z. 2008. Natural and unnatural constraints in Hungarian vowel harmony. *Language.*

Heinz, Jeffrey. 2009. On the role of locality in learning stress patterns. *Phonology* 26(02). 303-351.

Hyde, Brett. 2007. Non-finality and weight-sensitivity. *Phonology* 24(02). Cambridge: Cambridge Univ Press. 287-334.

Jarosz, Gaja. 2006a. Rich Lexicons and Restrictive Grammars - Maximum Likelihood Learning in Optimality Theory. *Rich Lexicons and Restrictive Grammars - Maximum Likelihood Learning in Optimality Theory*. PhD Dissertation, the Johns Hopkins University, Baltimore, MD.

Jarosz, Gaja. 2006b. Richness of the Base and Probabilistic Unsupervised Learning in Optimality Theory. In *Proceedings of the Eighth Meeting of the ACL Special Interest Group on Computational Phonology at HLT-NAACL*, 50–59. New York City, USA: Association for Computational Linguistics.

Jarosz, Gaja. 2010. Implicational markedness and frequency in constraint-based computational models of phonological learning. *Journal of Child Language. Special Issue on Computational Models of Child Language Learning* 37(3). Cambridge University Press. 565-606.

Jarosz, Gaja. To appear a. Naive Parameter Learning for Optimality Theory - The Hidden Structure Problem. In *Proceedings of the Fortieth Conference of the North East Linguistics Society.*

Jäger, Gerhard and Rosenbach, Anette. 2006. The winner takes it all - almost. Cumulativity in grammatical variation. *Linguistics* 44. 937-971.

Jäger, Gerhard. 2007. Maximum Entropy Models and Stochastic Optimality Theory. In Annie Zaenen, Jane Simpson, Tracy Holloway King, Jane Grimshaw, Joan Maling and Chris Manning (editor.), *Architectures, rules, and preferences: variation on themes by Joan Bresnan*, 467-479. Stanford: CSLI Publications.

Jesney, Karen and Tessier, Anne-Michelle. 2011. Biases in Harmonic Grammar: the road to restrictive learning. *Natural Language and Linguistic Theory* 29(1). 251-290.

Johnson, Mark. 2002. Optimality-Theoretic Lexical Functional Grammar. In Suzanne Stevenson and Paula Merlo (editor.), *The lexical basis of suntactic processing: Formal, computational and experimental issues*, 59-73. Amsterdam: John Benjamins.

Keller, Frank. 2000. Gradience in Grammar: Experimental and Computational Aspects of Degrees of Grammaticality. *Gradience in Grammar: Experimental and Computational Aspects of Degrees of Grammaticality*. PhD Dissertation, University of Edinburgh.

Keller, Frank and Asudeh, Ash. 2002. Probabilistic Learning Algorithms and Optimality Theory. *Linguistic Inquiry* 33(2). 225-244.

Legendre, Géraldine, Miyata, Yoshiro and Smolensky, Paul. 1990. Can connectionism contribute to Syntax? Harmonic Grammar, with an application. In M Ziolkowski, M Noske and K Deaton (editor.), *Proceedings of the Twenty-Sixth Regional Meeting of the Chicago Linguistic Society*, 237-252. Chicago: Chicago Linguistic Society.

Legendre, Géraldine, Sorace, Antonella and Smolensky, Paul. 2006. The Optimality Theory – Harmonic Grammar Connection. In *The harmonic mind : from neural computation to optimality-theoretic grammar*. Cambridge, Mass.: MIT Press.

Liberman, Mark and Prince, Allen. 1977. On stress and linguistic rhythm. *Linguistic Inquiry* 8. 249-336.

Magri, Giorgio. 2012. Convergence of error-driven ranking algorithms. *Phonology* 29(02). 213-269.

Martin, Andrew. 2011. Grammars leak: Modeling how phonotactic generalizations interact within the grammar. *Language* 87(4). 751-770.

McCarthy, John J. 2003. OT constraints are categorical. *Phonology* 20. 75-138.

McCarthy, John J and Prince, Alan. 1993. Generalized alignment. In Geert E Booij and J van Marle (editor.), *Yearbook of morphology*, 79-153. Dordrecht: Kluwer.

Merchant, Nazarré. 2008. Discovering Underlying Forms: Contrast Pairs and Ranking. *Discovering Underlying Forms: Contrast Pairs and Ranking*. PhD Dissertation, Rutgers University, New Brunswick, NJ.

Merchant, Nazarré and Tesar, Bruce. 2008. Learning underlying forms by searching restricted lexical subspaces. In *Proceedings of the Forty-First Conference of the Chicago Linguistic Society*, 33-47. Chicago Linguistics Society.

Pater, Joe. 2008. Gradual Learning and Convergence. *Linguistic Inquiry* 39(2). 334-345.

Pater, Joe. 2009a. Review of Paul Smolensky and Géraldine Legendre (2006). The harmonic mind: from neural computation to optimality-theoretic grammar. *Phonology* 26(01). 217-226.

Pater, Joe. 2009b. Weighted Constraints in Generative Linguistics. *Cognitive Science* 33. 999-1035.

Pater, Joe. to appear. Canadian raising with language-specific weighted constraints. *Language.*

Pearl, Lisa S. 2011. When Unbiased Probabilistic Learning Is Not Enough: Acquiring a Parametric System of Metrical Phonology. *Language Acquisition* 18(2). 87-120.

Potts, Christopher, Pater, Joe, Jesney, Karen, Bhatt, Rajesh and Becker, Michael. 2010. Harmonic Grammar with Linear Programming: From linear systems to linguistic typology. *Phonology* 27(1). 1-41.

Prince, Alan. 2002. Entailed ranking arguments. *Entailed ranking arguments,* ***ROA-500***. Rutger University, ms.

Prince, Alan. 2010. Counting Parses. *Counting Parses*. Rutgers University, New Brunswick, NJ, ms.

Prince, Allen. 1990. Quantitative Consequences of Rhythmic Organization. (Editor.) K Deaton, M Noske and M Ziolkowski. *CLS26-II: Papers from the Parasession on the Syllable in Phonetics and Phonology.*

Prince, Alan. 2002. Anything Goes. In Takeru Honma, Masao Okazaki, Toshiyuki Tabata and Shin-Ichi Tanaka (editor.), *New century of phonology and phonological theory*, 66-90. Tokyo: Kaitakusha.

Prince, Alan and Smolensky, Paul. 2004. Optimality Theory : Constraint Interaction in Generative Grammar. Malden, MA: Blackwell Pub.

Pruitt, Kathryn. 2010. Serialism and locality in constraint-based metrical parsing. *Phonology* 27(03). Cambridge: Cambridge University Press. 481-526.

Riggle, Jason. 2009. The Complexity of Ranking Hypotheses in Optimality Theory. *Computational Linguistics* 35(1). 47-59.

Rosenblatt, Frank. 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review* 65. 386-408.

Rubach, Jerzy and Booij, Geert E. 1985. A grid theory of stress in polish. *Lingua* 66(4). 281 - 320.

Smolensky, Paul. 1996. The Initial State and 'Richness of the Base'. *The Initial State and 'Richness of the Base'*. Johns Hopkins University, Baltimore, MD.: Technical Report JHU-CogSci-96-4, ms.

Smolensky, Paul and Legendre, Géraldine. 2006. (The Harmonic Mind : From Neural Computation to Optimality-theoretic Grammar). Cambridge, Mass.: MIT Press.

Soderstrom, Melanie, Mathis, Don and Smolensky, Paul. 2006. Abstract genomic encoding of Universal Grammar in Optimality Theory. In *The harmonic mind: from neural computation to optimality-theoretic grammar*, 403-471.

Tesar, Bruce. 1995. Computational Optimality Theory. *Computational Optimality Theory*. PhD Dissertation, University of Colorado, Boulder, CO.

Tesar, Bruce. 1997a. An Iterative Strategy for Learning Metrical Stress in Optimality Theory. In *The Proceedings of the 21st Annual Boston University Conference on Language Development*, 615-626. Boston University, Mass.

Tesar, Bruce. 1997b. Multi-Recursive Constraint Demotion. *Multi-Recursive Constraint Demotion*. Rutgers University, New Brunswick, NJ, ms.

Tesar, Bruce. 1998. An Iterative Strategy for Language Learning. *Lingua* 104. 131-145.

Tesar, B. 2000. Using Inconsistency Detection to Overcome Structural Ambiguity in Language Learning. *Using Inconsistency Detection to Overcome Structural Ambiguity in Language Learning*. Technical Report RuCCS-TR-58, Rutgers Center for Cognitive Science, Rutgers University., ms.

Tesar, Bruce. 2004a. Contrast Analysis in Phonological Learning. *Contrast Analysis in Phonological Learning*. Rutgers University, NJ, ms.

Tesar, Bruce. 2004b. Using Inconsistency Detection to Overcome Structural Ambiguity. *Linguistic Inquiry* 35(2). 219-253.

Tesar, Bruce. 2006a. Faithful Contrastive Features in Learning. *Cognitive Science* 30(5). 863 - 903.

Tesar, Bruce. 2006b. Learning from Paradigmatic Information. In *Proceedings of the Thirty-Sixth Conference of the North East Linguistics Society*, 619-638.

Tesar, Bruce. 2007. A comparison of lexicographic and linear numeric optimization using ldots. *A comparison of lexicographic and linear numeric optimization using ldots*. Rutgers University, New Brunswick, N.J., ms.

Tesar, Bruce. 2008. Output-Driven Maps. *Output-Driven Maps*. Rutgers University, New Brunswick, NJ, ms.

Tesar, Bruce. 2009. Learning Phonological Grammars for Output-Driven Maps. In *Proceedings of the Thirty-Ninth Conference of the North East Linguistics Society.*

Tesar, Bruce, Alderete, John, Horwood, Graham, Merchant, Nazarré, Nishitani, Koichi and Prince, Alan. 2003. Surgery in Language Learning. In *Proceedings of the 22nd West Coast Conference on Formal Linguistics*, 477-490.

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

Tesar, Bruce and Smolensky, Paul. 2000. Learnability in Optimality Theory. Cambridge, Massachusetts: MIT Press.

Tessier, Anne-Michelle. 2009. Frequency of violation and constraint-based phonological learning. *Lingua* 119(1). 6-38.

van der Hulst, Harry, Goedemans, Rob and van Zanten, Ellen. 2010. *A survey of word accentual patterns in the languages of the world*. *A survey of word accentual patterns in the languages of the world*. De Gruyter Mouton.

Wexler, Kenneth and Culicover, Peter. 1980. (Formal Principles of Language Acquisition). Cambridge, MA: MIT Press.

Wilson, Colin. 2006. Learning Phonology With Substantive Bias: An Experimental and Computational Study of Velar Palatalization. *Cognitive Science* 30(5). 945-982.