Acta Linguistica Academica Vol. **64** (2017) 2, 191–212
DOI: 10.1556/2062.2017.64.2.2

Uncovering structure hand in hand

Joint Robust Interpretive Parsing in Optimality Theory

Tamás Biró
ELTE Eötvös Loránd University, Budapest
tamas.biro@btk.elte.hu

Abstract: Most linguistic theories postulate structures with covert information, not directly recoverable from utterances. Hence, learners have to interpret their data before drawing conclusions. Within the framework of Optimality Theory (OT), Tesar & Smolensky (1998) proposed Robust Interpretive Parsing (RIP), suggesting the learners rely on their still imperfect grammars to interpret the learning data. I introduce an alternative, more cautious approach, Joint Robust Interpretive Parsing (JRIP). The learner entertains a population of several grammars, which join forces to interpret the learning data. A standard metrical phonology grammar is employed to demonstrates that JRIP performs significantly better than RIP.

Keywords: genetic algorithms; hidden structure; metrical stress; Optimality Theory; Robust Interpretive Parsing

1. Information hidden from the learner

Linguistic theories usually rely on covert information, hidden from the observer – from the linguist and from the language learner alike. This information concerns important features of the theories: brackets of syntactic and metrical trees, co-indexation, thematic roles, and so forth. Does the sentence *John loves Mary*, uttered in a mutual love situation, support SVO or OVS word order? Supposing that our theories adequately mirror linguistic competence, or at least one day they will, the central role played by these abstract theoretical constructs poses a challenge to the learner.

In this article, I shall employ **metrical foot structure** as an example, while keeping in mind that the problem is far more general. Nearly all contemporary phonological theories heavily rely on the notion of **feet** in their accounts of word stress assignment (Hayes 1995). According to the standard approach, a foot must contain one stressed, as well as, optionally, at most one unstressed syllable. A word must contain at least one main foot – whence the primary stress – and may contain further feet, each with a syllable with secondary (or tertiary) stress. Thus, the word *banána* with stress on its second syllable is explained by these theories via the foot structure *ba(ná)na* or *(baná)na* or *ba(nána)*. Yet, by which one exactly? These linguistic structures correspond to the same observable (overt) form. The problem is faced both by the phonologist subscribing to a theory with metrical feet, and by the language learning child whose competence the linguist's theory supposedly describes in an adequate way. The linguist's solution is to resort to (implicit or explicit) postulates of the theory, such as monosyllabic feet are avoided whenever possible; or a language employs either iambic feet (with foot-final stress) or trochaic ones (with foot-initial stress), but never both. Even if these postulates belonged to an innate UG and were available to the child, the creativity and intuition of the theory making linguist are still hard to algorithmise as an automatic learning procedure.

In online, error-driven learning algorithms, the learner entertains a grammar, which she repeatedly updates after having heard new learning data. Whenever the input to her learning algorithm is different from what she would have produced, an "error" occurs, which triggers a change in the grammar. After the update, the grammar will correctly produce that form; or the grammar will be more likely to correctly produce that form; or the grammar will be "closer" (in some sense) to the target grammar producing that form; or, at least, to a grammar that is equivalent to the target grammar, producing the same forms.

Imagine that the target grammar, the competence of the teacher, produces the surface form *ba(nána)* – including foot structure – whereas the grammar hypothesised by the learner predicts *(bána)na* for underlying form */banana/*. The overt form – missing the hidden structure – uttered by the teacher and heard by the learner is *banána*. This piece of learning data is clearly different from the learner's own overt form *bánana*, and this error triggers a grammar update. For instance, the learner realises that replacing trochaic feet with iambic ones would do the job. So, she will predict *(baná)na* as grammatical surface form, yielding the overt form *banána* – the same as the overt form uttered by the teacher. As long as there are no other forms in the language, the learning process can be seen as successful: although arriving at a different grammar, the learner will perfectly reproduce the primary linguistic data of the teacher. Yet, languages contain more words, such as *hòcuspócus*. Upon hearing them, the learner may reconsider her hypothesised grammar, and get to the conclusion that the target language does not contain word-initial iambs (predicting *(hocús)pocus* or *(hocús)(pocùs)*) but word-final trochees. Hence, the correct parses are *(hòcus)(pócus)* and *ba(nána)*.

Such a radical reconsideration of the hypothesis can certainly be expected from a trained linguist. But can it be expected from a baby? Are there cognitively plausible learning mechanisms encoding this creative step? Indeed, learning algorithms in various grammatical frameworks[1] have been shown to be stuck in 'local optima', situations that are far from the target, and yet, no single learning step is capable of dislodging the system from this position.

2. Robust interpretive parsing in Optimality Theory

Here is a typical situation faced by the Robust Interpretive Parsing (RIP) algorithm proposed by Tesar & Smolensky (1998; 2000) within Optimality Theory (OT) (Prince & Smolensky 1993). In this example, oversimplified for the sake of clarity, the grammar includes only the three candidates in tableau (1) for underlying form */banana/*. (Suppose, for instance, that the rest have already been filtered out by higher ranked constraints.) The three constraints are NonFinal, requiring that the last syllable of the word be not parsed into a foot; Align(Wd, Ft, R), requiring that the right edge of the word be aligned to the right edge of some foot; and finally Trochaic, which punishes iambic feet.

(1)

/banana/	NonFinal	Align(Wd, Ft, R)	Trochaic
l 1. *(bána)na*		$*$	
w 2. *(baná)na*		$*$	$*$
☞ 3. *ba(nána)*	$*$		

Imagine now that the teacher's grammar is Trochaic $\gg$ Align(Wd, Ft, R) $\gg$ NonFinal. Reading the tableau right-to-left, we obtain this hierarchy will produce the surface form highlighted by the ☞ symbol, *ba(nána)*. Correspondingly, the teacher will utter the overt form *banána*. Having heard that, the learner recovers the underlying form */banana/*, and tests her hypothesised grammar, NonFinal $\gg$ Align(Wd, Ft, R) $\gg$ Trochaic. As the tableau shows, she concludes she would have said *(bána)na*, which corresponds to a different overt form, *bánana*. This error should trigger an

[1] Refer to Niyogi (2006) for an overview and possible failures of learning within the principles-and-parameters framework, and to Tesar & Smolensky (2000) for an introduction to learning in Optimality Theory.

update of her hypothesised hierarchy. By comparing the **loser form** she would produce to the **winner form** produced by the teacher, she promotes certain constraints and/or demotes other ones. There is no doubt, the loser form l is candidate 1. It is unclear to her, however, not having access to the teacher's mind, whether the uttered form originates from candidate 2 or from candidate 3. What should she do: which one should she use as the winner form in updating her hierarchy?

In this specific case (but not necessarily in all cases), the reader, an expert in Optimality Theory, may quickly realise that candidate 2 is harmonically bounded (Samek-Lodovici & Prince 1999) by candidate 1. Candidate 2 can never be produced, by no hierarchy will it emerge as the optimal candidate. Thus, the reader will conclude that the teacher's grammar must have produced candidate 3, and the learner ought to update her hierarchy by aiming at *ba(nána)*. However, in the case of a realistically complex grammar, the situation is not that simple, with so few candidates to be tested for boundedness. For instance, Optimality Theoretic analyses of phenomena with recursive insertion of non-overt structures will yield an infinite array of interpretations.[2] Moreover, a candidate may be bounded not only by single candidates, but also by combinations of candidates (bounding sets, Samek-Lodovici & Prince 1999), further aggravating the computational challenge. Finally and most importantly, competing structures corresponding to the same overt form need not be harmonically bounded at all (see footnote 3 below).

In traditional Robust Interpretive Parsing (RIP, Tesar & Smolensky 1998), the learner guesses the form uttered by the teacher by recurring to her own grammar. She knows that the teacher must have uttered either the surface form *(baná)na* or *ba(nána)* as the overt form *banána*. Relying on the hierarchy she has been hypothesising (NonFinal $\gg$ Align(Wd, Ft, R) $\gg$ Trochaic), she compares the candidates corresponding to the observed overt form. This **interpreting parsing** differs from standard OT production in that candidates corresponding to a differently uttered form are left out. In turn, being more harmonic than candidate 3 for the learner's hierarchy, candidate 2 – that is, *(baná)na* – emerges as winner form w. This is the candidate supposed by the learner to have been generated by the teacher. Now the learner can proceed to comparing this winner form to her loser form, *(bána)na*, and determine which constraints to promote and/or which to demote.

[2] Jason Riggle's Contenders algorithm (Riggle 2004; 2009) is able to eliminate all bounded candidates even from an infinite candidate set, but only if Gen and all the constraints can be represented as finite-state automata.

Variants of the OT online (error-driven) learning algorithm use different **update rules**. Yet, they share the general idea that constraints preferring the loser to the winner must be demoted, while constraints preferring the winner to the loser may potentially be promoted (Tesar & Smolensky 1998; Boersma & Hayes 2001; Pater 2008; Boersma 2009; Magri 2012). Looking at tableau (1), however, we see we do not have winner-preferring constraints, and the single loser-preferring constraint, Trochaic, is already at the bottom of the learner's hierarchy. The learner has thus reached a deadlock, unable to update her grammar.[3]

Observe, however, that the learner needs not identify a single candidate as the "winner" (Biró 2013). What is only made use of is the winner's **profile**. By comparing the loser's profile to it, the learner identifies the loser-preferring constraints to demote and the winner-preferring constraints to (potentially) promote. Consequently, I suggest replacing the winner's profile with a **(weighted) mean violation profile** of the potentially winner candidates. In our example, averaging the number of violations incurred by candidates 2 and 3 yields:

[3] Some reviewers have not been convinced by the toy example provided, as it was the harmonically bounded candidate that caused the deadlock. But let us also include constraint Iambic:

/banana/	NonFinal	Align (Wd, Ft, R)	Trochaic	Iambic
l 1. *(bána)na*		*		*
w 2. *(baná)na*		*	*	
☞ 3. *ba(nána)*	*			*

Repeat now the above train of thought, but with Iambic ranked below Trochaic in both the teacher's and the learner's hierarchy. The teacher would again produce *ba(nána)*, and the learner would again take *(bána)na* as the loser form (the candidate she would have produced), and *(baná)na* as the winner form (the candidate she believes she has heard). In turn, beside the loser preferring constraint Trochaic, the learner will also identify a lower ranked winner preferring constraint, Iambic. By reversing the two, the learner converges on an iambic grammar producing *(baná)na.* While this grammar is substantially different from the teacher's, there is no way of distinguishing between the two on the surface, as long as /*banana*/ is the only morpheme in the language. But as soon as we also include words with a different number of syllables, the learner may collect evidence for the target language to be trochaic, and reverses the two constraints again. Thus, she may enter an **infinite loop**, comparable to the more complex and more realistic cases discussed by Tesar & Smolensky (2000), which need not be repeated here.

(2)

		/banana/	NonFinal	Align(Wd, Ft, R)	Trochaic
l	1.	*(bána)na*		∗	
	2.	*(baná)na*		∗	∗
☞	3.	*ba(nána)*	∗		
w		(2. + 3.)/2	0.5	0.5	0.5

The last row of tableau (2) now contains the mean violation profile, with the potential winner candidates having equal weights. This last row is now used as the "winner" w, and compared to the row of the loser candidate l. A typical example of OT, this tableau contains stars in its cells, corresponding to integers as violation levels. The mean for each constraint is in general a rational number, which may be less common in OT, but nevertheless it remains possible to decide if a constraint prefers the loser or the "winner".

The comparison suggests promoting the winner preferring constraint Align(Wd, Ft, R), and demoting the loser preferring NonFinal and Trochaic. What emerges is the hierarchy Align(Wd, Ft, R) ≫ NonFinal ≫ Trochaic. Although this ranking differs from the teachers' Trochaic ≫ Align(Wd, Ft, R) ≫ NonFinal, the two grammars are equivalent, at least as far as this single underlying form is concerned: both produce surface form *ba(nána)*. The deadlock has been avoided.

However, in a realistic example, the number of potential winner candidates might be very large, if not infinite. Therefore, simply computing the average of the candidates' profiles may be unrealistic, but also not the most efficient solution. After all, as the learning process advances, the learner may rightfully suppose that her hierarchy is getting closer to the target. So, it might make sense to exploit that hierarchy, even if more cautiously than the way suggested for traditional RIP in Tesar & Smolensky (1998).

The next section argues for an alternative to RIP, to be called **Joint Robust Interpretive Parsing** (JRIP). The loser's violation profile is compared to an averaged winner violation profile, and this average is taken over a sample of winners produced by a **population** of hypothesised grammars. A potential winner gets a higher weight when more hierarchies in the population vote for it. This approach has been inspired by genetic algorithms, a heuristic optimisation technique (Reeves 1995).[4] Maintaining a population

[4] I am indebted to Jan-Willem van Leussen, who raised the idea of using genetic algorithms during a discussion. Note also that GRIP, introduced by Biró 2013, is motivated by another heuristic optimisation technique, **simulated annealing** (cf. e.g., Reeves 1995; Biró 2006).

of hypothesised grammars has been successful in principles-and-parameters learning (Yang 2002). Genetic algorithms are not new to Optimality Theory, either (Turkel 1994; Pulleyblank & Turkel 2000), even if they have not been employed to target the covert structure problem. Note, however, that what follows is not a genetic algorithm in the strict sense.

3. JRIP: parsing together

Traditional RIP misleads the learner by suggesting that she can rely on her hypothesised grammar for the interpretation of the learning data. Yet, we know that her grammar is not correct, since an error has just been detected. An unlucky hypothesis will fatally distort the learner's interpretation of the learning input. The heuristics behind the alternative being proposed is that collective wisdom may help avoid this pitfall. The experiments presented in the next section demonstrate, however, that group influence can be both beneficial and detrimental.

Given an OT grammar and an error-driven learning algorithm (including an update rule), let p be the probability that the learning algorithm is successful: that it converges either to the target hierarchy, or at least, to a grammar that is equivalent to the target (in the sense that no error will ever prove that a different hierarchy has been arrived at). $1 - p$ is the probability on non-convergence (e.g., infinite loop) and convergence to a wrong grammar (deadlock). Depending on our theoretical assumptions, the learner may start either with a random hierarchy, or with a well-defined initial state (for instance, ranking markedness constraints above faithfulness constraints). Similarly, the nature of the target grammar and the distribution of the learning data may vary across experiments.

Take a learner entertaining a population of r hierarchies (h_1, h_2,..., h_r) and running independent learning processes in parallel. The learner can be said to have acquired the target language if (1) at least one of the hierarchies reaches the target, or (2) each of the hierarchies reaches the target. If we postulate different – such as random – initial states for the hierarchies, then each learning datum may be used to update every hierarchy.[5] Criterion (1) is simply "cheating", the multiplication of one's chances to succeed by allowing more attempts, whereas criterion (2) renders the

[5] If, however, the initial state is the same for each h_i, then we could feed them with different random samples, one piece of learning data updating a single hierarchy. Currently we shall not pursue this option. Lacking any faithfulness constraint, we will not exploit the initial faithfulness over markedness assumption, either.

task unnecessarily difficult. If the r learning processes are indeed independent, then the probability of success in case (1) will be $1-(1-p)^r$, since each hierarchy must fail for the entire project to fail; whereas the same likelihood will be p^r in case (2). Given a typical value of p, in the range of 50% to 90% (Tesar & Smolensky 2000; Boersma & Pater 2008; Biró 2013; see also the next section), a larger r makes the success in case (1) very probable, and in case (2) extremely improbable.

In the next section, however, we shall see that the r learning processes are not independent. Although the hierarchies are initialised randomly, their success also depend on the common teacher feeding them with the same data. Given various targets, the learning success of hierarchy h_i correlates with the learning success of h_j. This conclusion will follow from observing a success rate under condition (1) lower than $1-(1-p)^r$, and under condition (2) higher than p^r.

I now propose a third way of combining the r processes, the JRIP algorithm, which stands for **Joint Robust Interpretive Parsing**. In JRIP, each of the r hierarchies produces its own loser l_i and updates its ranking h_i independently from the rest. Yet, they join forces to determine the winner profile. Thereby they may help out those who would make a bad decision by themselves. Each hierarchy makes a guess for the winner candidate w_i, and the winner violation profile w will be the mean of the violation profiles of these guesses. Subsequently, each hierarchy compares its own loser l_i to the mean winner profile w to determine which constraints to demote and/or which to promote.

The following pseudo-code summarises the JRIP algorithm:

```
– For each $i = 1\ldots r$, initialise hierarchy $h_i$ randomly.
– For each overt form $o$ (piece of learning data) produced by the teacher,
    1. Recover underlying form $u$ from $o$.
    2. For each hierarchy $h_i$ ($i = 1\ldots r$), and given $u$, produce loser form $l_i$.
    3. Determine the set $W$ of potential winner forms, namely those candidates whose corresponding overt form coincides with $o$.
    4. For each hierarchy $h_i$ ($i = 1\ldots r$), find winner form $w_i$ in $W$ (most harmonic element of $W$ with respect to $h_i$).
```

5. Determine **mean violation profile $\boldsymbol{w}$**:

   $$C(w) := \frac{1}{r}\sum_{i=1}^{r} C(w_i)$$

   where $C(w_i)$ is the violation of constraint C by candidate w_i.
6. For each hierarchy h_i $(i = 1 \ldots r)$,
   - C is a **winner-preferring** constraint if $C(l_i) - C(w) > \beta$.
   - C is a **loser-preferring** constraint if $C(w) - C(l_i) > \lambda$.
7. For each hierarchy h_i $(i = 1 \ldots r)$, if l_i is not in W, then demote the loser preferring constraints and/or promote the winner preferring constraints, depending on the update rule of the learning algorithm.

- Run until every hierarchy h_i $(i = 1 \ldots r)$ reproduces all learning data.

In JRIP, the failure of a learning process going astray should be prevented by community wisdom. Being initiated randomly, some hierarchies in the population will be closer to the target, and so forestall the rest from drawing the wrong conclusions. In other cases, as we shall see, even the few lucky hierarchies "decline after many to wrest judgement". Following the multitude is a surprisingly strong drive: most successful experiments terminated with all hierarchies generating the same surface forms, not only the same overt forms.

The role of the β and λ parameters, borrowed from Biró 2013, is to introduce a margin between winner-preferring and loser-preferring constraints, and thereby to prevent inadvertent mistakes. Traditionally, they are both equal to zero. A constraint is winner-preferring with respect to learner's hierarchy h_i, if $C(l_i) - C(w) > 0$; that is, if the loser for hierarchy h_i incurs more violations than the average of the violations incurred by candidates w_1, w_2,..., w_r. Similarly, a loser-preferring constraint traditionally satisfies the inequality $C(w) - C(l_i) \;\; > \;\; 0$. By introducing positive β and λ parameters, the procedure becomes more conservative. Less constraints are promoted or demoted, and more are left intact. To promote or demote a constraint, the case must be made stronger: the loser must violate it very differently from the average of the winners.

Here is a plausible scenario. Some update rules focus on demoting the highest ranked loser preferring constraint, and so they can be led astray by mistakenly categorising a highly ranked constraint as loser preferring. But this is exactly what happens in situations similar to the one depicted in the tableaux of section 2. A hierarchy h_i may pick a harmonically bounded candidate as the winner, violating a constraint satisfied by the "real winner" (the surface form produced by the teacher). After averaging, the mean winner profile will therefore incur more violations for this constraint, possibly turning it into "falsely loser preferring" – exactly as it happened to Trochaic in tableau (2). Imagine now that another hierarchy, h_j, has correctly ranked it high. (Remember that Trochaic dominated the teacher's grammar in section 2.) Therefore, update rules demoting the highest ranked loser preferring constraint will erroneously demote it in h_j. Yet, a sufficiently large λ will refrain the learner from categorising this constraint as loser preferring. Demoting a lower ranked constraint instead will usually have less dramatic consequences, and so it is a safer move.

The conservative margin introduced by positive β and λ parameters has significantly increased the hierarchies' tendency to "choose the good" in the experiments which we now turn to.

4. Learning metrical stress

4.1. The linguistic model

I ran a series of experiments in order to assess the learnability of contemporary theories of stress based on the abstract notion of metrical feet. Similarly to Tesar & Smolensky 2000 and the ensuing literature (Boersma 2003; Boersma & Pater 2008; Biró 2013), the Generator function of the OT grammar added a foot structure – including monosyllabic and bisyllabic, main and non-main feet – to the underlying series of light and heavy syllables. Their twelve constraints, widespread in metrical phonology, were also adopted, and the demurring voices against some of them (Eisner 1997; McCarthy 2003; Biró 2003) were ignored:

- FootBinarity: Each foot must be either bimoraic or bisyllabic. Thus, assign one violation mark per foot composed of a single light syllable.
- Weight-to-Stress Principle (WSP): Each heavy syllable must be stressed. Thus, assign one violation mark per every heavy syllable that is not stressed.
- ParseSyllable: Each syllable must be footed. Thus, assign one violation mark per syllable unparsed into some foot.

- MainFootRight: Align the main foot with the word, right edge. Assign one violation mark per each syllable intervening between the right edge of the main foot and the right edge of the word.

- MainFootLeft: Align the main foot with the word, left edge. Assign one violation mark per each syllable intervening between the left edge of the word and the left edge of the main foot.

- AllFeetRight: Align each foot with a word, right edge. For each foot, assign one violation mark per each syllable intervening between the right edge of that foot and the right edge of the word.

- AllFeetLeft: Align each foot with a word, left edge. For each foot, assign one violation mark per each syllable intervening between the left edge of the word and the left edge of that foot.

- WordFootRight: Align the word with some foot, right edge. Assign one violation mark to the candidate iff the right edge of the word does not coincide with the right edge of some foot.

- WordFootLeft: Align the word with some foot, left edge. Assign one violation mark to the candidate iff the left edge of the word does not coincide with the left edge of some foot.

- Iambic: Align each foot with its head syllable, right edge. Assign one violation mark per foot whose last (single or second) syllable is not stressed (that is, per binary trochees).

- FootNonfinal: Each head syllable must not be final in its foot. Assign one violation mark per foot whose last (single or second) syllable is stressed (that is, per monosyllabic feet and binary iambs).

- NonFinal: Do not foot the final syllable of the word. Thus, assign one violation mark to the candidate iff the last syllable of the word is footed.

The model was implemented in the *OTKit* software package (Biró 2010).

4.2. Experimental setup

At the beginning of an experiment, the teacher's grammar and the r hierarchies entertained by the learner were randomly initialised: each of the twelve constraints received a random floating point rank between 0 and 50.[6] The following update rules were compared: Paul Boersma's GLA, demoting loser preferring constraints and promoting winner preferring constraints by the same (and unreduced) plasticity of 1 (Boersma & Hayes 2001); Giorgio Magri's earliest variant thereof, demoting the highest ranked loser preferring constraint by 1, and promoting all n_w winner preferring constraints by $1/n_w$ (Magri 2011; 2012); **alldem**, demoting all loser preferring constraints by 1; and **topdem** or Minimal GLA, demoting the highest ranked loser preferring constraint by 1 (Boersma 1998).[7]

The first two are promotion-demotion algorithms, while the last two are demotion-only algorithms. GLA and alldem are rank-insensitive, while Magri and topdem refer to the loser preferring constraint ranked highest in the learner's hierarchy h_i. According to the experiments to be presented, it is mostly rank-sensitiveness that influences the behaviour of the learning algorithm. We shall see that the alldem update rule yields results similar to GLA, and topdem follows Magri's update rule.

Feeding the learner, the teacher cyclically generated overt forms from a pool of four underlying forms: a sequence of four heavy syllables, a sequence of five light syllables, and two sequences of mixed syllables. The exact order of presentation was: */ab.ra.ka.dab.ra/*, */a.bra.ka.da.bra/*, */ho.cus.po.cus/* and */hoc.cus.poc.cus/*. The learner's hierarchies were expected to reproduce all of them. The learning was considered unsuccessful if the learner could not learn the target language after 500 cycles of presentation.

Most of the experiments did not require more than 100 cycles, and 99% of the learning terminated within 200 or 250 cycles, depending on the parameters. Hence, the reported success rates hardly underestimate those that could be obtained with a more severe – but computationally much more demanding – stopping condition (more cycles, or testing for various forms of failure: real divergence, infinite loops, etc.). Auxiliary experiments employed a 5000-cycle-limit, improving the success rate by less than 0.1%. Using one of the parameter combinations yielding the lowest

[6] A systematic study of the OT factorial typology predicted by the twelve constraints, as suggested by a reviewer, has been deferred to future work.

[7] Please note that Boersma's GLA, unlike the other algorithms, was devised to learn Stochastic OT, and is the only algorithm of the four that can learn variation at all (Boersma, p.c.). Variation in language is, however, not covered in this paper.

success rates ($r = 10$, $\beta = \lambda = 0.0$), not more than 4 out of 10,000 experiments terminated after 500 and before 5000 learning cycles – and most of them within 700 cycles – for any update rule: this is the number of learning successes that would have been missed, if learning were stopped after 500 cycles. This increase of less than 0.1% should be compared to the error margins otherwise falling in the magnitude of 1%.

Theoretically, learning can fail not only due to the hidden structure problem, but also as a result of the non-convergence of some learning algorithms (e.g., Pater 2008). Therefore, we also have to test the cases when the teacher provides the full surface forms, including foot brackets, for the learner. Three of the four learning approaches always converged in tens of thousands of experiments, whereas GLA converged in 98.8% of the cases, setting an upper limit to the expected success in the covert information case. (Given 16,000 experiments, the 95% confidence interval was 98.78%±0.17%.) Increasing the number of data presentation cycles from 500 to 5000 did not significantly improve the learning rate. In fact, even then all successful learning experiments (9879 out of 10,000) terminated within less than 200 cycles, suggesting that these 1.2% are genuine failures, not due to stopping the learning process too early.[8]

4.3. Results

Figure 1 presents the success rates of different update rules, as a function of the number of hierarchies r. The $r = 1$ case corresponds to traditional RIP, with a probability $p \approx 0.77$ that the randomly initialised learner acquires the language of the randomly initialised teacher. The JRIP algorithm with a few hierarchies markedly improves the learning rate, even though the

[8] This success rate for GLA is lower by 1% than the corresponding value in Biró 2013 for learning the same four forms. The difference must be due to the fact that the data presentation order was randomised in Biró 2013. Incidentally, the same difference, about 1%, reappears in the main experiments: GLA with RIP was successful in less than 77% of the cases with cyclic presentation order (section 4.3), and in more than 78% of the cases with random order (Biró 2013). Interestingly, Boersma (2003, 440) reports an opposite observation: when he repeated the experiments of Tesar and Smolensky (124 target languages with 62 overt forms each), cyclic presentation order performed better than randomised order.

Note, moreover, that the success rate significantly decreased as more hierarchies were learning in parallel. When ten hierarchies were presented with full-information surface forms, the success rate diminished to 95.54%±0.32%. This score is still much higher than $0.9878^{10} = 0.8845$, the expected success rate if the ten learning processes were independent.

success diminishes with a larger r. Interestingly, the curves for the rank-sensitive update rules (Magri and topdem) get much closer to the 100% success rate than the rank-insensitive rules (GLA and alldem). Although, statistically speaking, the rise is significant thanks to the large number of experiments, the effect size is small: the later two algorithms cannot improve their scores by more than a few percents.

Figure 1 also contains the success rates for independently learning hierarchies, under the conditions (1) and (2) discussed above. If all hierarchies must succeed, then the learning rate quickly diminishes, but much slower than p^r. Similarly, if the success of a single hierarchy suffices, then the rate grows fast, but not as fast as $1 - (1 - p)^r$. As mentioned earlier, this observation suggests that the learning success also depends on the target. Certain languages (sets of learning data) are hard to learn, while others are easy; therefore the r learning processes, which are run in parallel and independently from each other, tend to succeed or fail on the same datasets.

At the same time, as an anonymous reviewer remarks, the fact that these two graphs do not coincide affirms that the learning dynamics of standard RIP is sensitive to initialisation. Otherwise, parallel learning processes fed on the same learning data would fail or succeed in tandem. The reviewer then notes how close the at-least-one-must-succeed condition (△) falls on each graph to the 100% success rate for $r = 5$. In other words, in a random sample of hierarchies of size $r = 5$, there will almost always be at least one ranking from the – hence, apparently large – region of successful initial hierarchies.

Figures 2, 3 and 4 illustrate the role of the parameters β and λ. Increasing the value of at least one parameter helps us avoid the performance drop observed for larger r values. Figures 2 and 3 contain the same data, but organised in different ways. The former demonstrates that given a learning method and a set value of β, the middle ranges of λ yield the best results: $\lambda = 0.4$ for Boersma's GLA and $\lambda = 0.2$ for Magri's alternative version of it. The later figure testifies to the weaker influence of the choice of the β parameter. Magri's update rule, which promotes the winner preferring constraints with small steps, only marginally benefits from a positive β. Thanks to the 6000 experiments per data point, the statistical significance can be often demonstrated, and still, the effect size is very small.

Figure 4 displays the influence of λ on learning with the demotion-only update rules. Their behaviours are comparable to those of the corresponding promotion-demotion algorithms. As parameter β is employed

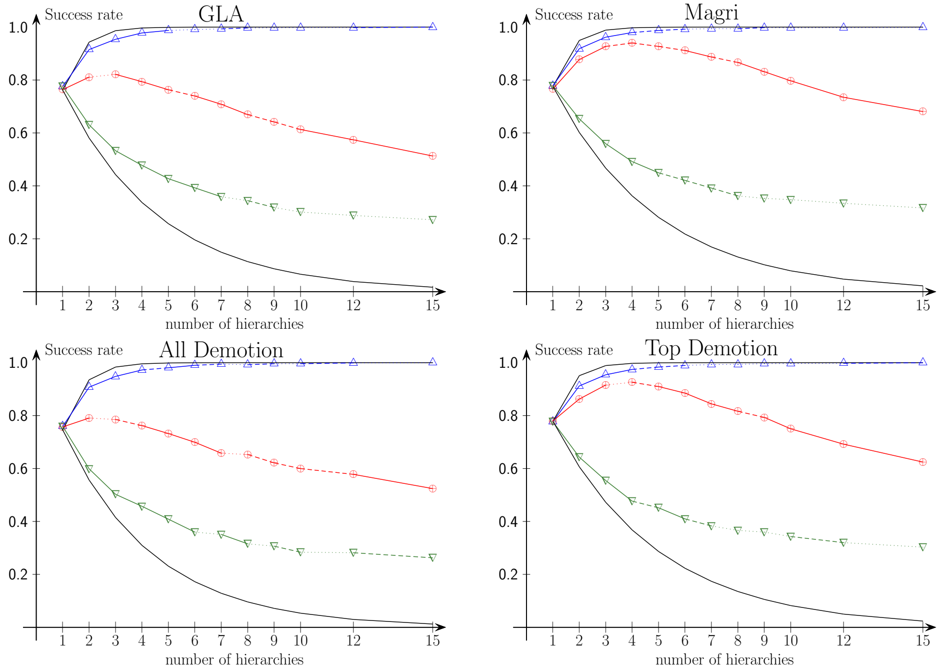

Figure 1: Success rates as a function of the number of hierarchies (r), for four update rules, under various success conditions: all hierarchies must succeed(∇), at least one must succeed ($\triangle$), as well as JRIP ($\oplus$; $\beta = \lambda = 0$). The upper and lower solid curves are $1 - (1 - p)^r$ and p^r respectively. Each data point corresponds to $N = 5000$ experiments. The significance of the difference of two data points is reflected by the style of the connecting line: solid for $p < 0.001$, dashed for $0.001 \leq p < 0.01$, dot-dashed for $0.01 \leq p < 0.05$, and dashed for $p \geq 0.05$.

for the selection of the winner-preferring constraints to be promoted, demotion-only algorithms are unaffected by the choice of β.

These two parameters influence the success rates in a complex, nontrivial way, which must be clarified by future research. A major mystery is the recurrent valley observable for $r = 5$ and $\beta = 0.6$. Nevertheless, we can conclude that a careful choice of the parameters largely improves the chance of success.

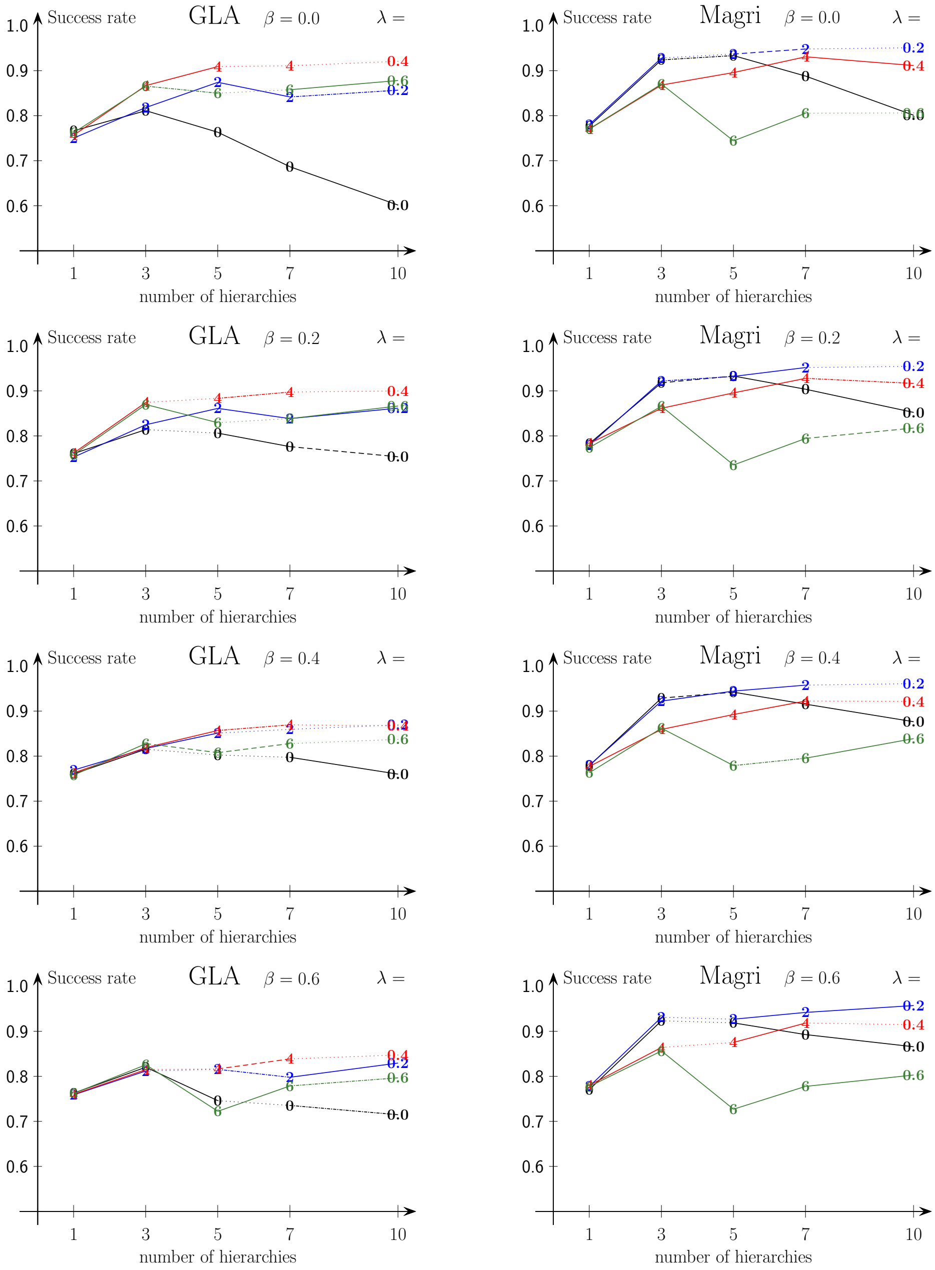

Figure 2: Success rate of JRIP as a function of the number of hierarchies (r), for the promotion-demotion update rules, and different values of parameters λ (per curve) and β (per panel). Each data point corresponds to $N = 6000$ experiments.

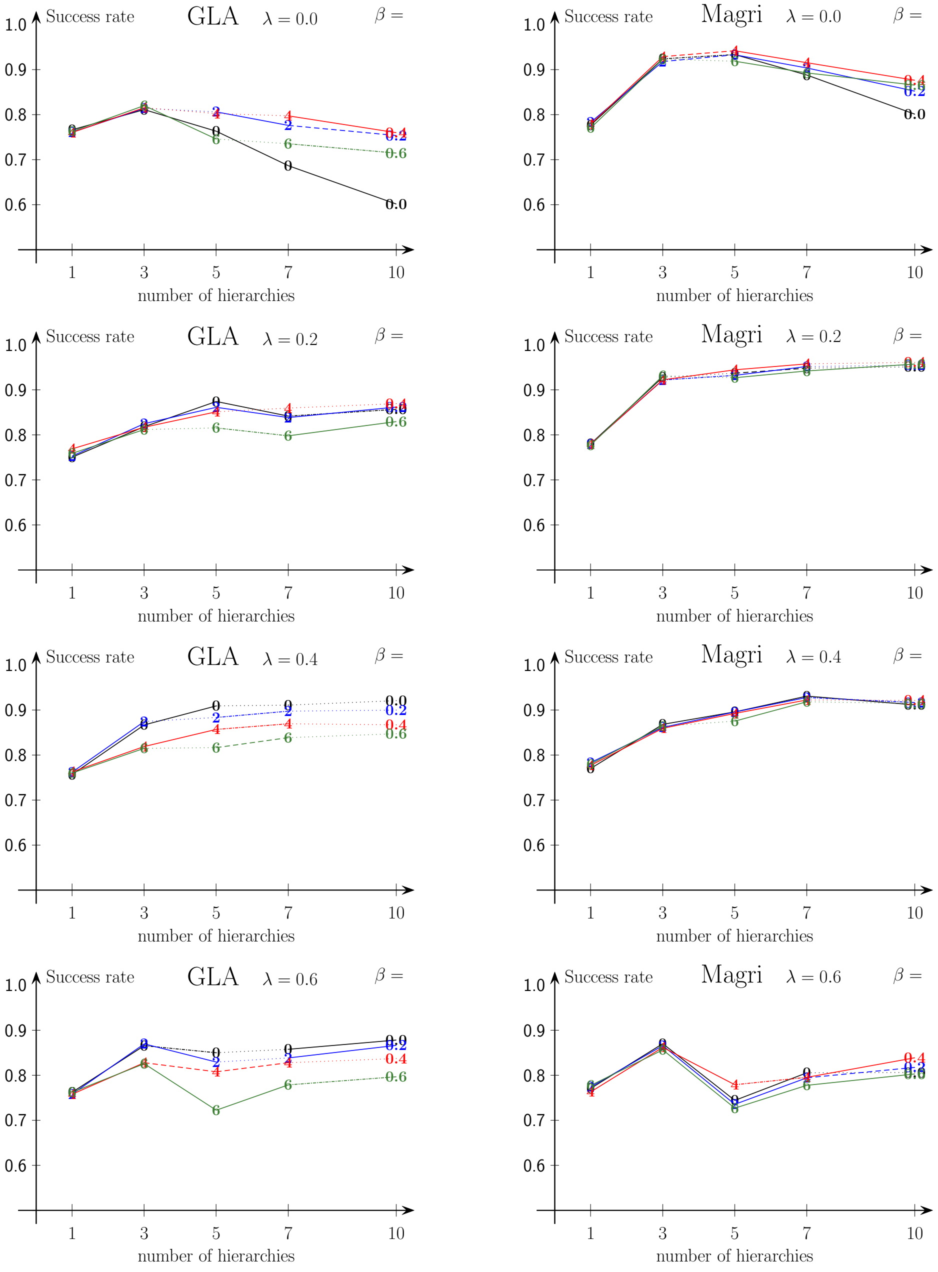

Figure 3: Success rate of JRIP as a function of the number of hierarchies (r), for the promotion-demotion update rules, and different values of parameters λ (per panel) and β (per curve). Each data point corresponds to $N = 6000$ experiments.

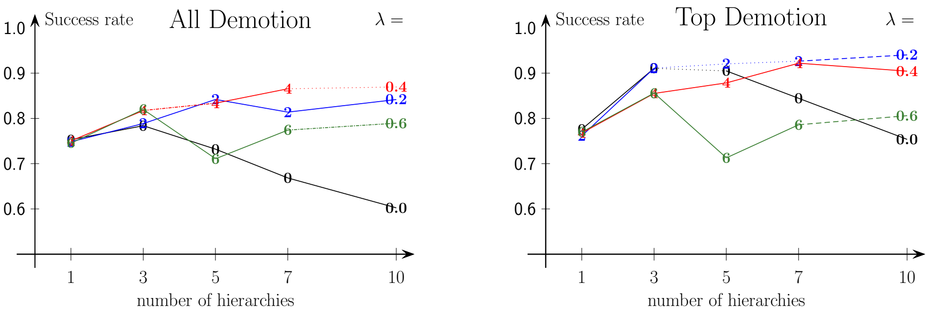

Figure 4: Success rate of JRIP as a function of the number of hierarchies (r), for the demotion-only update rules, and different values of the parameter λ (per curve). Parameter β does not play a role in these update rules. Each data point corresponds to $N = 6000$ experiments.

5. Concluding discussion

Using traditional Robust Interpretive Parsing (RIP, Tesar & Smolensky 2000), the probability of learning a random stress pattern is – given our OT grammar, a learner with a random initial hierarchy and the cyclic order of presenting the four words in the lexicon – $p = 76.54\% \pm 0.79\%$ in the case of Paul Boersma's GLA (Boersma 1998), and $p = 77.22\% \pm 0.78\%$ in the case of Giorgio Magri's algorithm (Magri 2011) (95% confidence intervals, based on 11000 experiments each). Learning metrical stress is a difficult task because the theory is based on unobservable feet, which the learners must surmise. A fifth of the learners fail, mainly because they mistakenly rely on their own hypotheses in interpreting the learning data. Joint Robust Interpretive Parsing (JRIP) improves on the success rate by calling for cooperation in a population of r hierarchies. The common wisdom of ten hierarchies, together with some conservative caution, helps us reach success rates as high as 92% for GLA ($\beta = 0.0$, $\lambda = 0.4$, $r = 10$), and 96% for Magri's update rule ($\beta = 0.4$, $\lambda = 0.2$, $r = 10$).

The problem of learning from data that hide crucial structural information from the learner has been around since the first discussions on the learnability of Optimality Theory. While the RIP algorithm of Tesar and Smolensky was long taken as the standard, a number of alternatives have recently been advanced. Jarosz (2013) introduced Resampling RIP and Expected Interpretive Parsing, while Biró (2013) advanced Generalised RIP. Their performance – the improvement relative to RIP – is comparable to the one of JRIP. Future research might wish to estimate which of RIP, RRIP, EIP, GRIP and JRIP (and using what parameter combination) is most successful by employing a standard dataset. I have declined to do so for a principled reason: I do not believe the merit or weakness of an algorithm depends on its performance on a specific arbitrary toy grammar. In practice, the specific task at hand will determine the best approach. I also concur with my reviewers: mathematical, analytical work on the algorithms in the future might shed more light on their virtue and vice. If, however, our goal is to understand language learning by the human mind, we should ponder other criteria, such as learning curves, error patterns, as well as the cognitive plausibility of the computational mechanisms. Probabilistic resampling and simulated annealing could be argued to be performed by the neural network in the brain, even if technical details of the proposed algorithms are quite abstract and mathematical. At the same time, JRIP's parallel grammars might show some affinity with parallel grammars in bilingual language development. Note, moreover, that RRIP and EIP crucially address the hidden information problem in stochastic OT and HG (Boersma 1997; Boersma & Hayes 2001; Boersma & Pater 2008) only, unlike GRIP and JRIP, which also work in the non-stochastic context.

One may ask then what the advantage is of using JRIP. Remember that the same number of hierarchies offered a much higher success rate when the hierarchies learned independently of each other until one of them was successful. Indeed, this kind of independent learning may be efficient from an engineer's perspective. Yet, it is psychologically implausible: while the number of metrical stress patterns in a language might be quite small, children cannot test their hypothetical grammars on the full vocabulary – let alone, scaling the grammar up, on the complex system of an entire language. Instead, they run their learning mechanism for some years, and after a certain "critical age" they stop learning. Imagine a child entertaining several, but independently developing grammars – a conjecture that would explain the large variability in child language (Yang 2002). Our simulations suggest that most probably some of the hierarchies will find the target language, but most probably not all of them. Thus, the child reaches the "critical age" with incompatible grammars. It is hard to imagine that she would then test these grammars for the entire target language – in the way our computer simulations did with a very restricted vocabulary – to find out which (if any) of the hierarchies is correct. Thus, such a model would predict that the adults display a variation of forms comparable to that of the children, alternating which grammar to use in production. Consequently, we need a learning method that makes it likely that **all** hierarchies acquire the target language. Independent learning with this stronger requirement proved to be much less successful than the joint learning algorithm JRIP.

Mapping the exact interplay between constraints, candidates, initial and target hierarchies, update rules and JRIP parameters is deferred to future work. A better understanding of the algorithms' behaviour will certainly contribute to a better understanding of OT learning algorithms in general. For instance, the stronger dependence of JRIP with Magri's update rule on λ than on β may be worth a longer discussion. It has also been surprising to see that rank-sensitiveness – differentiating between GLA and alldem, on the one hand, and Magri and topdem, on the other – is a major factor; whereas the distinction between promotion-demotion algorithms (GLA and Magri) versus demotion-only algorithms (alldem and topdem) only influences the importance of the choice of the β parameter.

The strong dependence of JRIP on its parameters (r, β and λ), hardly understood thus far, may be brought up as another point of criticism against JRIP. Some parameter combinations do it much better than traditional RIP, but others do much worse. Yet, it might be speculated that biological evolution could have optimised the parameter setting, and so our mind employs a (locally) optimal combination of the parameters.

Beyond the question whether JRIP is cognitively plausible or useful for language technology, the results presented here have more general consequences. Traditionally, generative linguistics accounts for non-existing types in language typology by postulating parameters or constraints such that no parameter setting or constraint hierarchy would predict this type. A language does not exist because the mind cannot encode it. When Tesar & Smolensky (2000) demonstrated that some metrical phonology grammars could not be learned due to the hidden structure problem, non-learnability was identified as a second reason for the lack of certain types (Boersma 2003). Two further reasons are the evolutionary instability of a type (Jäger 2003) and the too heavy computational load associated with it (Biró 2006, 215). Our results now remind us that the learnability of a type crucially depends on the learning algorithm. Hence, explaining the lack of a type by referring to its unlearnability is flawed unless we have independent arguments for the human mind using **this** learning algorithm. Or, reversing the train of thought: a learning algorithm can be argued for, and another learning algorithm can be argued against by comparing their learnability predictions to attested language types – provided, of course, that we strongly believe in the adequacy of the grammar architecture and its building blocks (principles, parameters, constraints, candidate sets, etc.) (cf. Boersma 2003, 443).

The experiments in this paper have focused on metrical stress within the framework of Optimality Theory. Some linguistic details of the grammar may be contested, but a different "dialect" of contemporary metrical phonology is not expected to display a very different computational behaviour. Moreover, as mentioned in the introduction, metrical stress is just one example for a far more general problem: crucial information is often covert in the learning data. I hope that JRIP offers a better solution to this universal challenge in linguistics, and that the results give ground for optimism and improvement regarding other linguistic phenomena (such as phrase brackets in syntax and semantic relations), as well. Finally, the solution proposed for OT may also inspire proponents of further theoretical frameworks struggling with similar learnability problems.

Acknowledgements

This research was supported by a *Veni Grant* (project number 275-89-004) offered by the Netherlands Organisation for Scientific Research (NWO), as well as by a *Marie Curie FP7 Integration Grant* (no. PCIG14-GA-2013-631599, "MeMoLI") within the *7th European Union Framework Programme*. I also thank Jan-Willem van Leussen and Paul Boersma, as well as a series of anonymous reviewers for comments and suggestions.

References

Biró, Tamás. 2003. Quadratic alignment constraints and finite state Optimality Theory. In Proceedings of the Workshop on Finite-State Methods in Natural Language Processing (FSMNLP), held within EACL-03, Budapest. 119–126. ROA-600.

Biró, Tamás. 2006. Finding the right words: Implementing Optimality Theory with simulated annealing. Doctoral dissertation. University of Groningen. ROA-896.

Biró, Tamás, 2010. OTKit: Tools for Optimality Theory. A software package. http://www.birot.hu/OTKit/

Biró, Tamás. 2013. Towards a robuster interpretive parsing: Learning from overt forms in Optimality Theory. Journal of Logic, Language and Information 22. 139–172.

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

Boersma, Paul. 1998. Functional Phonology: Formalizing the interactions between articulatory and perceptual drives. Doctoral dissertation. University of Amsterdam. (Published by Holland Academic Graphics, The Hague.)

Boersma, Paul. 2003. Review of B. Tesar & P. Smolensky (2000): *Learnability in OT*. Phonology 20. 436–446.

Boersma, Paul. 2009. Some correct error-driven versions of the Constraint Demotion algorithm. Linguistic Inquiry 40. 667–686.

Boersma, Paul and Bruce Hayes. 2001. Empirical tests of the Gradual Learning Algorithm. Linguistic Inquiry 32. 45–86.

Boersma, Paul and Joe Pater, 2008. Convergence properties of a gradual learning algorithm for Harmonic Grammar. ROA-970.

Eisner, Jason. 1997. Efficient generation in Primitive Optimality Theory. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-1997) and 8th EACL. Madrid, 313–320. Also: ROA-206.

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

Jäger, Gerhard. 2003. Simulating language change with functional OT. In S. Kirby (ed.) Language evolution and computation. Proceedings of the Workshop at ESSLLI, Vienna. 52–61.

Jarosz, Gaja. 2013. Learning with hidden structure in Optimality Theory and Harmonic Grammar: Beyond Robust Interpretive Parsing. Phonology 30. 27–71.

Magri, Giorgio. 2011. An online model of the acquisition of phonotactics within Optimality Theory. In L. Carlson, C. Hölscher and T. F. Shipley (eds.) Expanding the space of cognitive science: Proceedings of the 33rd Annual Conference of the Cognitive Science Society. Austin, TX: Cognitive Science Society, 2012–2017.

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

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

Niyogi, Partha. 2006. The computational nature of language learning and evolution. Cambridge, MA: MIT Press.

Pater, Joe. 2008. Gradual learning and convergence. Linguistic Inquiry 39. 334–345.

Prince, Alan and Paul Smolensky. 1993. Optimality theory: Constraint interaction in generative grammar. Technical Report TR-2, Center for Cognitive Science, Rutgers University, New Brunswick, N.J. and Technical Report CU-CS-697-93, Department of Computer Science, University of Colorado, Boulder.

Pulleyblank, Douglas and William J. Turkel. 2000. Learning phonology: Genetic algorithms and Yoruba tongue-root harmony. In J. Dekkers, F. van der Leeuw and J. van De Weijer (eds.) Optimality Theory: Phonology, syntax, and acquisition. Oxford: Oxford University Press. 554–591.

Reeves, Colin R. (ed.). 1995. Modern heuristic techniques for combinatorial problems. London: McGraw-Hill.

Riggle, Jason. 2004. Contenders and learning. In B. Schmeiser, V. Chand, A. Kelleher and A. Rodriguez (eds.) Proceedings of the 23rd West Coast Conference on Formal Linguistics (WCCFL 23). Somerville, MA: Cascadilla Press.

Riggle, Jason, 2009. Generating contenders. Technical report, ROA-1044.

Samek-Lodovici, Vieri and Alan Prince, 1999. Optima. ROA-363.

Tesar, Bruce and Paul Smolensky. 1998. Learnability in Optimality Theory. Linguistic Inquiry 29. 229–268.

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

Turkel, Bill, 1994. The acquisition of Optimality Theoretic systems. ROA-11.

Yang, Charles D. 2002. Knowledge and learning in natural language. Oxford: Oxford University Press.