CONVERGENCE OF ERROR-DRIVEN RANKING ALGORITHMS

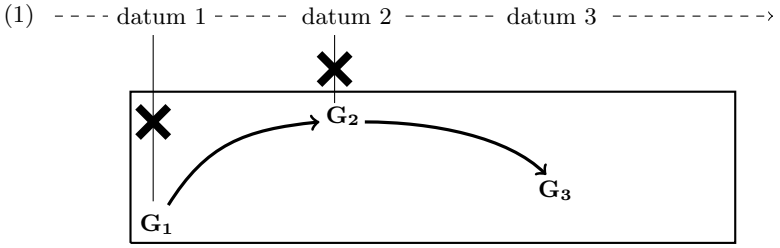
GIORGIO MAGRI

[This appeared with minor differences in *Phonology* 29.2: 213-269, 2012]

Abstract — According to the *OT error-driven ranking model* of language acquisition, the learner performs a sequence of slight re-rankings triggered by mistakes on the incoming stream of data, until it converges to a ranking that makes no more mistakes. This learning model is very popular in the OT acquisition literature, in particular because it predicts a sequence of rankings that models gradualness in child acquisition. Two implementations of this learning model have been developed in the OT computational literature, namely Tesar and Smolensky's (1998) *Error-Driven Constraint Demotion* (EDCD) and Boersma's (1997) *Gradual Learning Algorithm* (GLA). Yet, EDCD only performs constraint demotion, and it is thus shown to predict a ranking dynamics too simple from a modeling perspective. The GLA performs both constraint demotion and promotion, but has been shown not to converge. This paper thus develops a complete theory of convergence of error-driven ranking algorithms that perform both constraint demotion and promotion. In particular, it shows that convergent constraint promotion can be achieved (with an error-bound that compares well to that of EDCD) through a proper *calibration* of the amount by which constraints are promoted.

1. Introduction

1.1. **Error-driven ranking algorithms.** Assume that the learner is provided with the space of all possible grammars G_1, G_2, etcetera. Data come in a stream, one piece of data at the time. And the learner maintains a current grammar, which represents its current hypothesis on the target adult grammar. Suppose that at a certain time the current grammar, say G_1, fails at accounting for the current piece of data, say datum 1. Prompted by this error, the learner updates the current grammar G_1 to a slightly different grammar G_2 that sits nearby in the space of grammars. This process is repeated over and over again. Until the learner eventually stops making errors and converges to a final grammar consistent with the stream of input data, so that learning ceases.

(1) 

I wish to thank Adam Albright for lots of help and discussion. I also wish to thank Paul Boersma, Alan Prince, Jason Riggle, Paul Smolensky, Donca Steriade, and Bruce Tesar for useful conversations on the material presented in this paper. The reviewers and an associate editor of *Phonology* also provided me with very useful comments and suggestions, that greatly improved the paper. Earlier versions of this paper have been presented at NECPhon 3 (MIT, November 2009), at the 84th annual meeting of the LSA (Baltimore, January 2010), at the workshop *Computational Modeling of Sound Pattern Acquisition* (University of Alberta, February 2010), at the 33th annual meeting of the Cognitive Science Society (Boston, July 2011); at the 18th Machine Learning Summer School (Bordeaux, September 2011), and at the 48th annual meeting of the Chicago Linguistics Society (Chicago University, April 2012); I wish to thank the audiences at those venues for useful discussion. This work was supported in part by a 'Euryi' grant from the European Science Foundation ("Presupposition: A Formal Pragmatic Approach" to P. Schlenker) as well as by the LABEX-EFL grant.

This learning scheme is called *error-driven*, as the learning dynamics is driven by the errors performed on the incoming stream of data. This scheme has been thoroughly investigated in the Machine Learning literature (under the heading of *online learning*; for a review, see Kivinen 2003 and Cesa-Bianchi and Lugosi 2006, chapters 11, 12). Within the linguistic literature, error-driven learning dates back to at least Wexler and Culicover (1980).

This paper explores error-driven learning within the phonological framework of *Optimality Theory* (OT; Prince and Smolensky 2004, Kager 1999). Thus, the space of grammars is defined through the set of rankings over a given constraint set. And the slight shift from the current grammar to the updated grammar consists of a slight re-ranking of the constraints. Re-ranking can take different forms. For instance, the learner can *demote* those constraints that are causing the failure of the current ranking on the current piece of data. Or it can *promote* those constraints that would have prevented that failure. Or it can adopt a mixed re-ranking strategy, that combines both constraint demotion and promotion. These slight re-rankings continue until the constraints intersperse in a ranking consistent with the input stream of data, so that the learner makes no more mistakes. The model just sketched is called the OT *error-driven ranking algorithm* (henceforth: EDRA). Two EDRAs that have played an important role in the OT acquisition and computational literature are Tesar and Smolensky's (1998) *Error-Driven Constraint Demotion* (henceforth: EDCD) and Boersma's (1998) *Gradual learning algorithm* (henceforth: GLA). Building on this literature, Section 2 introduces EDRAs in full detail.

The intermediate rankings entertained by an EDRA can be interpreted as intermediate learning stages, thus modeling the observed gradual, stepwise child progression towards the target adult language. Furthermore, the model does not keep track of previously seen forms, and thus does not impose unrealistic memory requirements (contrary to so called *batch* models, that are instead allowed to glimpse at the entire set of data at once). For these reasons, most of the OT acquisition literature has endorsed EDRAs as a cognitively plausible model of child acquisition; see for instance Gnanadesikan (2004), Boersma and Levelt (2000), Bernhardt and Stemberger (1998), as well as Tesar (2004) and Tessier (2009) for critical discussion and alternative approaches.

Bridging cognitive plausibility with computational soundness, this paper looks at the most basic computational issue in the theory of EDRAs, namely *convergence*: the EDRA should eventually stop making errors, and thus settle on a final ranking consistent with the incoming stream of data. And convergence should be *efficient*: the number of mistakes made by the algorithm before converging should not only be finite, but should furthermore grow slowly with the complexity of the underlying OT typology, simply measured as the number of constraints. As clearly stated in Keller and Asudeh (2002), "convergence is a crucial property of a learning algorithm that should be investigated formally." This paper develops a complete theory of EDRAs' convergence, that provides both sufficient and necessary conditions for efficient convergence.

1.2. **Summary of the main results.** The first EDRA developed in the OT computational literature is Tesar and Smolensky's (1998) EDCD. Its signature property is that it demotes offending constraints to a lower position, but does not promote virtuous constraints. Lack of constraint promotion allows Tesar and Smolensky (but cf. also Boersma 2009) to prove that EDCD converges and that convergence is efficient, as it is achieved after a worst-case number of errors that grows only quadratically in the number of constraints. Tesar and Smolensky's result sets the standard for the theory of EDRAs. Furthermore, some of the tools used in their analysis turn out to extend beyond demotion-only to EDRAs that perform constraint promotion too. For these reasons, Section 3 offers a thorough review of Tesar and Smolensky's theory.

Although a virtue from a *computational* perspective, lack of constraint promotion turns out to be a liability from a *modeling* perspective. Section 4 argues for EDRAs that perform constraint promotion on top of demotion, by looking at one of the main modeling

applications of EDRAs, namely modeling the child early acquisition of phonotactics. In carefully controlled experimental conditions, nine-month-old infants already react differently to licit and illicit sound combinations (Jusczyk et al. 1993, among others). They thus display knowledge of phonotactics at an early stage, when other linguistic abilities are still not fully developed. In particular, *morphology* is still lagging behind at this early age, so that the child has still no access to phonological alternations (Hayes 2004). In order to model the fact that phonotactics is acquired before morphology kicks in and makes phonological alternations available, the error-driven model for the acquisition of phonotactics is trained on faithful mappings. This entails that the faithfulness constraints are never responsible for the failure of the current ranking vector. As EDCD only demotes the constraints that are responsible for the current failure (but never promotes those that could have prevented that failure), it thus never re-ranks the faithfulness constraints. And this cannot be right. If two languages in the typology require the opposite relative ranking of some faithfulness constraints, EDCD fails on at least one of them (Hayes 2004 and Prince and Tesar 2004 provide examples of such languages). Furthermore, EDCD is unable to model learning paths where the child's repair strategy for a certain marked structure changes over time (for instance McLeod et al. 2001 document leaning paths where complex onsets are simplified by different strategies over time, such as deletion and coalescence).

Although needed from a modeling perspective, convergent EDRAs that perform both constraint demotion and promotion are not easy to devise. Tesar and Smolensky (1998) explicitly warn against the danger of constraint promotion. Indeed, there is only one EDRA currently available in the literature that performs both constraint demotion and promotion, namely Boersma's (1998) GLA. Yet, Pater (2008) shows through a simple counterexample that the GLA does not converge in the general case. Section 5 reviews this literature on constraint promotion. In particular, it contributes the first explicit explanation of the GLA's failure on Pater's counterexample. In conclusion, computationally sound EDRAs that perform both constraint demotion and promotion are needed from the perspective of the OT acquisition literature but have so far eluded the efforts of the OT computational literature.

This paper solves this impasse, showing that efficient convergence can be achieved despite constraint promotion, through a proper calibration of the promotion component of the re-ranking rule. Let me illustrate the idea informally. Following Boersma (1997, 1998), I assume throughout this paper that EDRAs entertain a numerical representation of the current ranking, by assigning to each constraint a numerical *ranking value* whose relative size reflects the relative ranking of that constraint. Such a numerical representation of the current ranking allows for a numerical formulation of re-ranking rules: constraint demotion consists in a slight decrease of the current ranking value of offending constraints; and constraint promotion consists in a slight increase of the current ranking value of virtuous constraints. Assume that offending constraints are demoted by a small fixed *demotion amount*, say 1 for concreteness. And that virtuous constraints are promoted by a small *promotion amount*. Boersma's non-convergent GLA sets the promotion amount equal to the demotion amount, namely to 1. Yet, suppose there are few constraints that need to be demoted, say just one for concreteness; but a number of constraints that need to be promoted, say two. In this case, the GLA demotes *once* by 1 and promotes *twice* by 1. Overall, the GLA thus performs more constraint promotion than demotion. Pater's (2008) counterexample shows that this is not a good idea. Indeed, as demotion-only has been shown by Tesar and Smolensky (1998) to have a good convergent behavior, the promotion component of the re-ranking rule should not overwhelm the demotion component, so as not to disrupt too much its good convergent behavior. This requires a proper calibration of the promotion amount. For instance, if there are two constraints that are promoted and one that is demoted, then the promotion amount should intuitively be less than $1/2$. In fact, two promotions by less than $1/2$ lead to an overall promotion which is less than the overall demotion of 1, so that indeed the overall constraint promotion does not overwhelm

constraint demotion. These heuristic considerations suggest that in the general case, the promotion amount should be smaller than the number of constraints demoted divided by the number of constraints promoted, as stated in (2). A re-ranking rule that satisfies the strict inequality (2) is called *calibrated*.

$$\text{(2)}\quad \text{promotion amount} < \frac{\text{number of constraints demoted}}{\text{number of constraints promoted}}$$

Section 6 shows that a slight extension of Tesar and Smolensky's (1998) analysis of demotion-only EDRAs ensures convergence for any calibrated promotion/demotion EDRA, with the worst-case number of mistakes depending on the size of the promotion amount.

To consider a concrete case, define the promotion amount as in (3). For instance, if one constraint is demoted and two are promoted, then each of the latter two constraints will be promoted by 1/3 according to (3).

$$\text{(3)}\quad \text{promotion amount} = \frac{\text{number of constraints demoted}}{1 + \text{number of constraints promoted}}$$

This re-ranking rule is calibrated, as it obviously satisfies condition (2): the promotion amount is smaller than the ratio between the numbers of demoted and promoted constraints, although only slightly smaller. Section 6 shows that an EDRA with the calibrated promotion/demotion re-ranking rule (3) converges after a worst-case number of errors that grows only cubically in the number of constraints. This error bound compares well with Tesar and Smolensky's (1998) quadratic error bound for demotion-only EDRAs.

The ratio between the numbers of demoted and promoted constraints is called the *calibration threshold*. The calibration condition (2) requires the promotion amount to be *strictly* smaller than this calibration threshold. What happens if we increase the promotion amount up to the threshold, as in (4)?

$$\text{(4)}\quad \text{promotion amount} = \frac{\text{number of constraints demoted}}{\text{number of constraints promoted}}$$

Section 7 addresses this question, showing that convergence is retained but efficiency is lost: the worst-case number of errors is finite but grows exponentially in the number of constraints. Although Tesar and Smolensky's (1998) analysis of demotion-only can be extended to calibrated constraint promotion (2), it cannot be stretched further to apply also to the threshold case (4). A new line of analysis is thus needed in order to prove convergence in the latter case. The analysis developed will rest on a property of EDRAs that is interesting in its own right: they can never entertain again a ranking (vector) that had made a mistake at some earlier time. In other words, EDRAs explore the typology in a smart way: although they do not keep track of previously seen data and thus of the errors previously made, they implicitly manage to avoid repeating the same error twice. This surprising property will follow from a connection between the notion of OT-consistency and the geometric property of *conic independence*.

Although the results developed in this paper are computational in nature, I submit they have significant cognitive implications, discussed in Subsections 1.3-1.6. The paper is thus relevant also to a non-computational audience. In order to make it accessible, the discussion will be informal, with technical details relegated to a final Appendix.

1.3. **Implications for the OT vs. HG framework selection problem.** Children are able to solve the language learning problem efficiently, despite its complexity. Evolution must therefore have selected the actual child acquisition strategies because of their computational optimality. Provable computational soundness thus represents a necessary requirement that a learning model needs to satisfy in order to have a chance to qualify as a proper model of child language acquisition. In particular, efficient convergence represents the most basic requirement for computationally sound error-driven learning. It is

for this reason that the failure of the GLA on Pater's deadly counterexample has recently prompted various scholars to explore error-driven learning within frameworks alternative to OT, such as *Harmonic Grammar* (henceforth: HG; Legendre et al. 1990b,a), that is equipped with provably convergent error-driven learning algorithms. For recent applications of error-driven learning within HG, see for instance Coetzee and Pater (2008), Boersma and Pater (2007, to appear), Jesney and Tessier (2007, 2008), as well as Pater (2009) and Magri (2012b) for general discussion.

The results developed in this paper show that the GLA can be easily "fixed", in the sense that it is possible to develop variants of the GLA for standard OT that provably efficiently converge, through a proper calibration of the promotion amount, as in (2). These results thus show that the computational soundness of error-driven learning should not play any role in the recent debate concerning the OT vs HG framework selection problem, as both frameworks come with provably convergent error-driven algorithms.[1]

1.4. **Implications for modeling gradience.** OT grammars are usually parameterized by combinatorial objects, namely *rankings*. From this perspective, the OT framework looks very different from frameworks such as HG, that posits instead numerically weighted constraints. Yet, as originally noted in Boersma (1997, 1998), OT typologies can also be easily given a numerical re-parameterization, as rankings can be represented by assigning to each constraint a numerical ranking value, with the understanding that constraints with large ranking values are ranked above constraints with low ranking values. This technical observation immediately raises the following cognitive question: which one of these two parameterizations is cognitively more adequate? Does the learner actually assume that languages are parameterized by *combinatorial* objects such as rankings or by *continuous* objects such as ranking values?

This paper offers a new perspective on this issue. Demotion-only EDRAs can be implemented in terms of combinatorial rankings, with no need for a numerical re-parameterization in terms of ranking values. For example, Tesar and Smolensky's (1998) original formulation of EDCD uses rankings rather than ranking values. Sections 5-7 will show in detail that the situation is very different for re-ranking rules that perform constraint promotion on top of constraint demotion. In fact, as anticipated above, the promotion component of the re-ranking rule cannot overwhelm the demotion component. This requires a proper numerical calibration of the promotion amount, as in (2). Hence, constraint promotion is inherently numerical, and thus requires a numerical parameterization of OT typologies. The choice between demotion-only and demotion/promotion EDRAs thus bears on the issue of the proper parameterization of OT typologies. Section 4 will argue that demotion-only is insufficient from a modeling perspective, and that a certain amount of constraint promotion is needed. In conclusion, the computational investigation developed in this paper entails that the proper parameterization of OT grammars should be in terms of numerical ranking values rather than in terms of combinatorial rankings.

This conclusion in turn has theoretical implications for the framework selection problem between OT and alternative frameworks that adopt a weighted, numerical model of constraint interaction, such as HG. In particular, various authors have recently suggested that HG's numerical weights might have an advantage over OT's combinatorial rankings from the perspective of modeling gradience. For instance, Pater (2008) writes: "[I will] illustrate and extend existing arguments for the replacement of OT's ranked constraints with weighted ones: that the resulting theory can be adapted relatively straightforwardly

[1]Actually, error-bounds for OT error-driven learning are better than those for HG error-driven learning. In the case of OT, we get sharp error bounds polynomial in the number n of constraints, that do not depend on the training data. No such bounds are available for HG error-driven algorithms, such as the *Perceptron*. As Heinz and Riggle (2011, p. 71) note, in the case of HG, "it is possible to construct a sample sequence of arbitrary length in which each new data point causes an error that leads to an ever smaller change in the weighting. Thus, though learners that use strategies such as the Perceptron algorithm will eventually converge to a correct constraint weighting for any HG grammar [...], there is no general bound on the rate of convergence [...] that holds for all possible sets of training data."

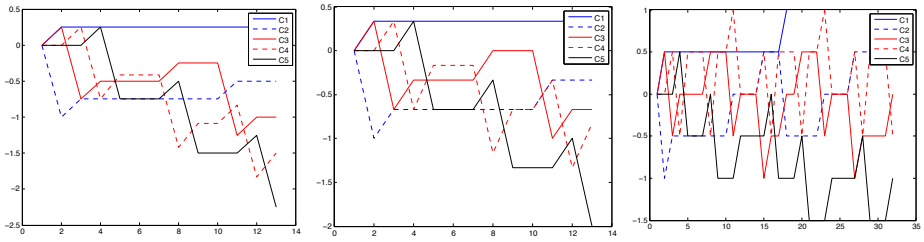
to deal with various types of non-categorical linguistic phenomena [...]. HG was in fact originally motivated as an account of gradient syntactic well-formedness [...]". Yet, if we can collect independent evidence in favor of a numerical representation of OT typologies, then the argument just mentioned in favor of an alleged superiority of HG over OT just evaporates.

1.5. **Implications for modeling the child acquisition of phonotactics.** One of the main applications of EDRAs considered in the literature concerns modeling the child acquisition of phonotactics. Section 4 offers a quick glimpse at this modeling application. I will argue that this important modeling application requires EDRAs to perform some constraint promotion, so as to motivate the computational developments of Sections 6 and 7. My argument will be twofold. First, I will argue that constraint promotion is needed in order for the sequences of rankings formally predicted by EDRAs to have a chance at *matching* child acquisition paths. Second, I will argue that constraint promotion is also needed in order for the final grammar entertained by EDRAs to have a chance of being *restrictive*, namely of correctly ruling out illicit forms, despite the fact that the algorithm is only trained on licit forms.

My discussion of this modeling application in Section 4 will be cursory, as I will only provide the *negative* part of the argument, namely I will only show that constraint promotion is *necessary* for restrictiveness and for matching child acquisition paths. I will not provide here the *positive* part of the argument, namely that constraint promotion is indeed also *sufficient* for restrictiveness and matching. Yet, this paper represents a first step of a larger research project that I am currently involved in, that tries to establish EDRAs as a proper model of the child acquisition of phonotactics, both from a computational and a modeling perspective. The next step in this project is Magri (2012d), that presents a detailed discussion of EDRAs restrictiveness, expanding substantially on the cursory remarks presented below in Subsection 4.4.2.

1.6. **Implications for modeling child acquisition paths.** As anticipated above, EDRAs assign to each constraint a current ranking value, whose relative size represents the current position of that constraint in the ranking. Throughout learning, these ranking values are slightly updated. The *ranking dynamics* can thus be plotted, with time on the horizontal axis and ranking values on the vertical axis. In (5), I plot the ranking dynamics for three runs on the same data (namely the test case considered in Pater 2008) with three slightly different choices of the promotion amount. In particular, (5b) corresponds to the case of the calibrated promotion amount in (3), whereby the denumerator of the promotion amount consists of the number of promotions increased by 1. And (5a) corresponds to a promotion amount only slightly smaller, whereby the number of promotions in the denominator is increased by 2 rather than by 1. These two ranking dynamics in (5a) and (5b) differ only minimally, and the number of updates performed in the two runs is identical. As one might naïvely expect, a small difference in the promotion amounts leads to only small differences in the ranking dynamics.

(5) a. $p = \dfrac{\#\text{ of demotions}}{\#\text{ of promotions} + 2}$ b. $p = \dfrac{\#\text{ of demotions}}{\#\text{ of promotions} + 1}$ c. $p = \dfrac{\#\text{ of demotions}}{\#\text{ of promotions}}$



That is not always the case, though. Figure (5c) plots the ranking dynamics on the same data when the promotion amount is set equal to the calibration threshold, so that the denominator now coincides with the number of promotions. Also this promotion amount in (5c) differs only very slightly from the one in (5b). And yet the ranking dynamics in (5c) looks completely different, and the number of updates is much larger.

The child acquisition of phonology is gradual, in the sense that the target adult phonology is approached through a stepwise progression of intermediate stages. As recalled in Subsection 1.1, EDRAs have been endorsed by the OT acquisition literature because they offer a way to model gradualness, as child acquisition paths can be matched against the sequence of grammars predicted by an EDRA's ranking dynamics. A naïve strategy to explore EDRAs' modeling predictions would fix the implementation details (such as the choice of the promotion amount) in some heuristic way, based on the assumption that only a few options for the implementation details need to be considered, as the modeling predictions will only vary little for small variations in the implementation details. But the example in (5) shows that this assumption is misleading: the behavior of the algorithm displays breakpoints, at which the modeling predictions of the algorithm change abruptly. In other words, (5) shows that the modeling implications of EDRAs cannot be explored naïvely, without a preliminary thorough computational understanding of the algorithm and its breakpoints. This paper thus contributes a new tool to the exploration of EDRAs' modeling prediction, by pinpointing the calibration threshold as a crucial breakpoint for the choice of the promotion amount.

2. Error-driven ranking algorithms

Subsection 2.1 introduces a basic formulation of EDRAs, after Tesar and Smolensky (1998). Subsection 2.2 restates them in terms of an alternative ERC representation of OT data, after Prince (2002). Finally, Subsection 2.3 restates them in terms of an alternative numerical parameterization of OT grammars, after Boersma (1997, 1998, 2009). Subsection 2.4 introduces the issue of EDRAs' convergence.

2.1. **Basic description.** The basic data unit in OT is a *data triplet* (6a), consisting of an underlying form /*x*/ and two surface forms [*y*] and [*z*], both drawn from the set *Gen*(/*x*/) of candidates for /*x*/. By convention, the first candidate [*y*] in the triplet is the intended *winner*, while the other candidate [*z*] is an intended *loser*. As a useful mnemonic, I adopt the convention of striking out the loser. To illustrate, the data triplet (6b) pairs up the underlying form /rad/ with the two candidates [rad] and [rat]. As the former is the intended winner, this triplet says that there is no final devoicing.

(6) a. winner / loser

(/*x*/, [*y*], ~~[*z*]~~)

b. winner / loser

(/rad/, [rad], ~~[rat]~~)

An OT-grammar is parameterized by a *ranking*, which is a linear order $\gg$ over the constraint set, as in (7). Constraint C_h is *$\gg$-ranked above* constraint C_k provided $C_h \gg C_k$. Without loss of generality, an arbitrary ranking $\gg$ can be assumed to be $C_1 \gg C_2 \gg \ldots \gg C_n$, as the numerical indices assigned to the constraints are arbitrary.

(7) top ranked … bottom ranked

$C_1 \gg C_2 \gg \ldots \gg C_n$

A ranking $\gg$ is called *(OT-)consistent* with an underlying/winner/loser form data triplet (/*x*/, [*y*], ~~[*z*]~~) provided condition (8) holds. Condition (8) says that the intended loser ~~[*z*]~~ violates the constraints "more severely" than the intended winner [*y*]. In the sense that,

among those constraints that distinguish between the winner [y] and the loser [$\not{z}$], the top $\gg$-ranked one, call it C_{top}, assigns more violations to the loser than to the winner.

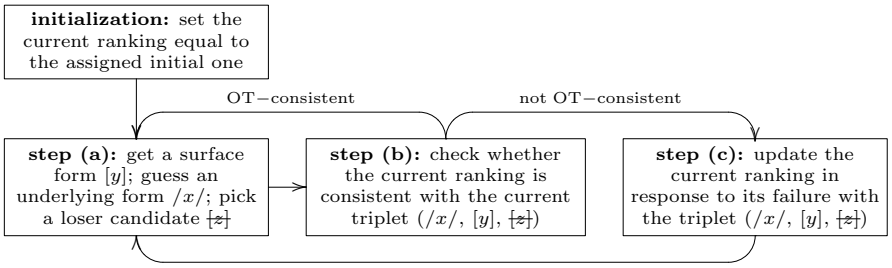
(8) $C_{\text{top}}(/x/, [\not{z}]) > C_{\text{top}}(/x/, [y])$ (violations of the loser [$\not{z}$] / violations of the winner [y])

where C_{top} = the top $\gg$-ranked constraint among those that assign a different number of violations to the loser [$\not{z}$] and to the winner [y]

A ranking $\gg$ is called *(OT-)consistent* with a set of data triplets provided it is consistent with every triplet in the set. And a set of data triplets is called *(OT-)consistent* provided it is consistent with at least one ranking.

With these preliminaries in place, an EDRA can be described as in (9), after Tesar and Smolensky (1998). The algorithm maintains a *current ranking*, which represents its current hypothesis on the target grammar. This current ranking is initialized to an *initial ranking*. And it is updated over time by looping through the three steps (9a)-(9c).

(9)



At step (9a), the algorithm assembles an underlying/winner/loser form triplet. In certain applications (acquisition of phonotactics), the algorithm is only provided with the winner form, and needs to pick the corresponding underlying and loser forms. In some other applications (acquisition of alternations), the algorithm is provided with both underlying and winner forms, and only needs to pick a loser form. At step (9b), the algorithm checks the current ranking against the current underlying/winner/loser form data triplet. If the current ranking is consistent with the current data triplet, nothing happens: the algorithm goes back to step (9a) and waits for another piece of data. Otherwise, the algorithm modifies its current ranking at step (9c), and then goes back to step (9a).

2.2. **Restatement in comparative notation.** Given an underlying/winner/loser form data triplet (/x/, [y], [$\not{z}$]), the constraints can be sorted into *winner-preferring*, *loser-preferring* or *even* as in (10a), depending on whether they assign more (less or equal, respectively) violations to the loser [$\not{z}$] than to the winner [y].

(10) a. Constraint C_k is $\left\{ \begin{array}{c} \textit{winner-preferring} \\ \textit{loser-preferring} \\ \textit{even} \end{array} \right\}$ iff $\left\{ \begin{array}{l} C_k(/x/, [\not{z}]) > C_k(/x/, [y]) \\ C_k(/x/, [\not{z}]) < C_k(/x/, [y]) \\ C_k(/x/, [\not{z}]) = C_k(/x/, [y]) \end{array} \right\}$ (violations of the winner [y] / violations of the loser [$\not{z}$])

b. (/rad/, [rad], [$\not{\text{rat}}$]) (winner / loser) $\Longrightarrow$

F_{pos} = IDENT[VOICE]/ONSET:	even
F_{gen} = IDENT[VOICE]:	winner-preferring
M = *[VOICE]:	loser-preferring

To illustrate, consider again the data triplet (6b). The positional faithfulness constraint F_{pos} for voicing is even, as it does not distinguish between the two candidates [rad] and

~~[rat]~~. The general faithfulness constraint F_{gen} is winner-preferring, as the intended winner [rad] is fully faithful to the underlying form /rad/ contrary to the intended loser ~~[rat]~~. Finally, the markedness constraint M against voicing is loser-preferring, as it is violated by the intended winner [rad], and not by the intended loser ~~[rat]~~.

Usually, the relevant information concerning the data triplet (6b) is represented in the form of the OT-tableau (11a) (with the pointing finger marking the intended winner). This representation (11a) encodes the actual number of constraint violations, as the number of stars in a given cell.

(11) a. (/rad/, [rad], ~~[rat]~~) $\Longrightarrow$

/rad/	F_{pos}	F_{gen}	M
☞[rad]			⋆
[rat]		⋆	

b. (/rad/, [rad], ~~[rat]~~) $\Longrightarrow$ $\begin{bmatrix} F_{\text{pos}} & F_{\text{gen}} & M \\ \text{EVEN} & \text{WINNER-PREFERRER} & \text{LOSER-PREFERRER} \end{bmatrix} = \begin{bmatrix} F_{\text{pos}} & F_{\text{gen}} & M \\ e & \text{W} & \text{L} \end{bmatrix}$

Yet, the definition (8) of OT-consistency does not really care about the actual numbers of constraint violations. It only cares about whether each constraint is winner-preferring or loser-preferring or even. The relevant information concerning this data triplet can thus be represented as in (11b), specifying with a W, an L, or an e whether each constraint is winner-preferring or loser-preferring or even.

Following Prince (2002), the information provided by a data triplet that is useful for the sake of OT-consistency can thus be distilled as in (12). The data triplet is paired up with a tuple with n entries (one for every constraint), with the convention that the kth entry is equal to W, L or e depending on whether the kth constraint C_k is winner-preferring or loser-preferring or even. One such n-tuple of L's, e's and W's is called an *elementary ranking condition* (henceforth: ERC). I denote an ERC by $\mathbf{a}$ and its entries by $a_1, \ldots, a_n$.

(12) (/x/, [y], ~~[z]~~) $\Longrightarrow$ $\mathbf{a} = \begin{bmatrix} a_1 & \ldots & a_n \end{bmatrix}$ $\qquad a_k = \begin{cases} \text{W} & \text{if } C_k \text{ is winner-preferrer} \\ \text{L} & \text{if } C_k \text{ is loser-preferrer} \\ e & \text{if } C_k \text{ is even} \end{cases}$

A set of many, say m, data triplets, can be paired up with the corresponding *ERC matrix*, by organizing the ERCs corresponding to each triplet one underneath the other (the order does not matter), into a matrix with n columns (one for every constraint), m rows (one for every data triplet) and entries equal to W, L and e. I denote by $\mathbf{A}$ an arbitrary OT-comparative matrix; I often omit e's for readability.

(13) a. $\mathbf{A} = \underbrace{\begin{bmatrix} C_1 & \ldots & C_k & \ldots & C_n \\ \text{W} & \text{L} & \text{W} & \text{L} & e \\ \text{L} & \text{W} & \text{W} & e & e \\ e & \text{W} & \text{W} & \text{L} & \text{L} \end{bmatrix}}_{n \text{ columns}} \Big\} m \text{ rows}$

b. (/da/, [da], ~~[ta]~~), (/rad/, [rad], ~~[rat]~~) $\begin{bmatrix} F_{\text{pos}} & F_{\text{gen}} & M \\ \text{W} & \text{W} & \text{L} \\ e & \text{W} & \text{L} \end{bmatrix}$

An example is provided in (13b): this ERC matrix has three columns, because the constraint set in (10b) contains three constraints; it has two rows, because it corresponds to the two data triplets (/da/, [da], ~~[ta]~~) and (/rad/, [rad], ~~[rat]~~); its entries are W's, L's and e's according to the rule described in (12).

With this notation in place, condition (8) for OT-consistency between a ranking $\gg$ and a data triplet can be restated as condition (14) between that ranking and the corresponding ERC $\mathbf{a}$. In general, a ranking is called *(OT-)consistent* with an arbitrary ERC provided

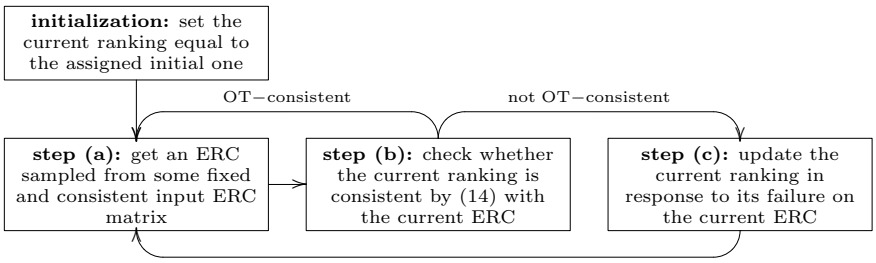
condition (14) holds. And it is called *(OT-)consistent* with an ERC matrix provided it is consistent with every ERC in the matrix. Finally, an ERC matrix is called *(OT-)consistent* provided it is consistent with at least one ranking.

(14) Once the n entries of the ERC $\mathbf{a}$ are reordered from left to right in decreasing order according to $\gg$, then the leftmost non-e entry is a W.

To illustrate the notion of OT-consistency in (14), note that the ERC matrix in (13b) is consistent with the ranking $F_{\text{pos}} \gg F_{\text{gen}} \gg M$, as its columns are ordered from left to right in $\gg$-decreasing order and the leftmost non-e symbol of every row is W.

In many cases, we are interested in studying the behavior of EDRAs irrespectively of the proper definition of its step (9a), namely irrespectively of the proper definition of the sub-routines that determine the underlying form and choose a loser form for the winner form the algorithm has been currently fed. In these cases, I can thus assume that the underlying and the loser forms are also provided to the algorithm, along with the winner form. In other words, I can assume that the algorithm is given as input at step (9a) an underlying/winner/loser form data triplet, or equivalently the corresponding ERC. It is thus useful to restate EDRAs in terms of ERCs, as in (15). According to (15), the algorithm is trained on a stream of ERCs and the current ranking is updated whenever inconsistent with the current ERC.

(15)



Throughout this paper, I assume that the ERCs fed to the algorithm at step (15a) are sampled from a fixed, given *input ERC matrix*. I will assume throughout the paper that the input ERC matrix is consistent. As we will see in Subsection 4.3, this assumption can be made without loss of generality in certain applications, as in the case of modeling the acquisition of phonotactics.

2.3. **Restatement in terms of ranking vectors.** So far, rankings have been represented as total orders on the constraint set. Boersma (1997, 1998, 2009) notes that a ranking over n constraints can equivalently be represented as an n-tuple of numbers, exploiting the natural ordering among numbers. To introduce the idea, let's pair up the three constraints in (10b) with three numbers as in (16). This triplet of numbers can be interpreted as follows: the positional faithfulness constraint F_{pos} is top ranked, because it corresponds to the largest number 100; the markedness constraint M is bottom ranked, because it corresponds to the smallest number 50; the general faithfulness constraint F_{gen} is ranked in between, because its corresponding number 70 lies in between the other two.

(16)
$$\begin{array}{c} \overset{F_{\text{pos}}}{100} \quad \overset{F_{\text{gen}}}{70} \quad \overset{M}{50} \end{array} \quad \Longrightarrow \quad F_{\text{pos}} \gg F_{\text{gen}} \gg M$$

In order for the correspondence in (16) to hold, it is crucial that the three numbers considered are all distinct. What if two of these numbers are identical? Consider for instance the case in (17), whereby the two constraints F_{gen} and M are assigned the same number. We can think of this triplet of numbers as representing two different rankings at the same time, depending on how the tie between M and F_{gen} is resolved. We can think of these two rankings as two different ways of "refining" the numerical tie.

(17)
$$\begin{array}{c} \begin{array}{ccc} {\scriptstyle F_{\text{pos}}} & {\scriptstyle F_{\text{gen}}} & {\scriptstyle M} \end{array} \\ \left(\begin{array}{ccc} 100 & 50 & 50 \end{array} \right) \end{array} \begin{array}{l} \nearrow \\ \searrow \end{array} \begin{array}{l} F_{\text{pos}} \gg M \gg F_{\text{gen}} \\ F_{\text{pos}} \gg F_{\text{gen}} \gg M \end{array}$$

These considerations allow OT grammars to be given a numerical parameterization, besides the usual combinatorial parameterization in terms of rankings. Here are the details.

A *ranking vector* is an n-tuple $\boldsymbol{\theta}$ of numbers $\theta_1, \ldots, \theta_n$ as in (18), one for each of the n constraints. The kth component θ_k is called the *ranking value* of constraint C_k.

(18)
$$\boldsymbol{\theta} = \begin{array}{c} \begin{array}{ccccccc} {\scriptstyle C_1} & & {\scriptstyle \ldots} & & {\scriptstyle C_k} & & {\scriptstyle C_n} \end{array} \\ \left(\begin{array}{ccccccc} \theta_1, & & \ldots & & \theta_k, & & \theta_n \end{array} \right) \end{array}$$

A ranking $\gg$ is a *refinement* of a ranking vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$ provided condition (19) holds for any pair of constraints C_h, C_k. If the two ranking values θ_h and θ_k tie, the antecedent of (19) fails, and different refinements can break the tie in either way. Otherwise, any refinement satisfies the ordering implicitly defined by the relative size of the two ranking values θ_h and θ_k.

(19) $\theta_h > \theta_k \implies C_h \gg C_k$.

Once ranking vectors are paired up with rankings through (19), notions that pertain to rankings can be extended to ranking vectors. In particular, we can extend the notion of OT-consistency from rankings to ranking vectors. If a ranking vector admits a unique refinement, then the extension is straightforward: we will say that the ranking vector is consistent with an ERC iff its unique refinement is consistent, according to the original notion of OT-consistency in (14). What if instead some of the ranking values tie and the ranking vector thus admits multiple refinements? One option would be to declare the ranking vector consistent with an ERC provided *at least one* of its refinements is consistent with it, in the spirit of Anttila (1997) and Anttila and Cho (1998). At convergence, the algorithm will thus return a ranking vector that has the property that at least one of its derived rankings is consistent with the input ERC matrix. But that is not very useful: how do we decide for a given refinement whether it is what we want or not? Thus, I will require consistency of *every* refinement in order to declare a ranking vector consistent with an ERC, as stated in (20).

(20) A ranking vector is *(OT-)consistent* with an ERC provided each of its refinements is consistent with that ERC, according to the original notion of consistency (14).[2,3]

---

[2] As stressed in Boersma (2009), the notion of OT-consistency (20) for ranking vectors with two or more identical components has nothing to do with the alternative notion of OT-consistency introduced by Tesar and Smolensky (2000), that allows for multiple constraints to be assigned to the same stratum with the corresponding tie resolved additively. Without getting into the details of this alternative definition of OT-consistency, let me illustrate the difference with an example. Consider the ERC (ia) together with the ranking vector (ib), with the two identical components $\theta_1 = \theta_2 = 2$.

(i) a. ERC = $\begin{array}{c} \begin{array}{ccc} {\scriptstyle C_1} & {\scriptstyle C_2} & {\scriptstyle C_3} \end{array} \\ \left[\begin{array}{ccc} \text{W} & \text{L} & \text{W} \end{array} \right] \end{array}$ b. $\boldsymbol{\theta} = \begin{array}{c} \begin{array}{ccc} {\scriptstyle C_1} & {\scriptstyle C_2} & {\scriptstyle C_3} \end{array} \\ \left(\begin{array}{ccc} 2 & 2 & 1 \end{array} \right) \end{array}$

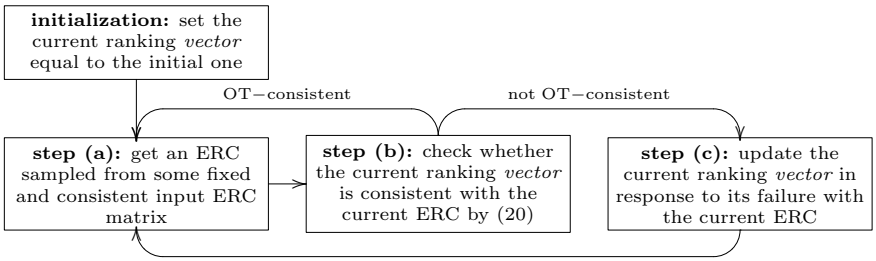
According to the alternative definition of OT-consistency introduced by Tesar and Smolensky (2000), the ranking vector (ib) is indeed consistent with the ERC (ia), because the L and the W of the two equally highest ranked constraints C_1 and C_2 "cancel out". But the ranking vector (ib) is *not* consistent with the ERC (ia) according to the definition (20), since the ranking vector (ib) admits the refinement $C_2 \gg C_1 \gg C_3$ that is not consistent with the ERC (ia). In the rest of the paper, I will stick to this classical notion of OT-consistency (20) and ignore the alternative notion of OT-consistency introduced by Tesar and Smolensky, that I have just alluded to. The entire discussion is thus framed squarely within standard OT. Contrary to what has been suggested by Tesar and Smolensky, there is no need to step outside of the standard framework for algorithmic purposes.

[3] A reviewer worries that this notion of OT consistency (20), that requires consistency to hold for *all* refinements, might not be efficiently computable. For instance, if all ranking values are identical, then don't we have to check consistency for all $n!$ refinements, causing a complexity explosion? That is *not* the case. Here is a way to see that. Let $W(\mathbf{a})$ and $L(\mathbf{a})$ be the sets of winner- and loser-preferring constraints relative to an ERC $\mathbf{a}$. It turns out that a ranking vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$ is consistent with

For instance, the ranking vector in (16) is consistent with the ERC in (11b), as the only refinement of the former is consistent with the latter. But the ranking vector in (17) is not consistent with that ERC, because of the inconsistent refinement $F_{\text{pos}} \gg M \gg F_{\text{gen}}$.

Boersma (1997, 1998, 2009) suggests restating the EDRA (15) in terms of ranking vectors as in (21), which is the final formulation of EDRAs considered in this paper. The current hypothesis on the target grammar is stored by the algorithm as a numerical ranking vector, rather than as a combinatorial ranking. The current ranking vector is updated whenever it is found to be inconsistent with the current ERC.

(21)



This restatement of EDRAs in terms of ranking vectors rather than rankings will prove to be crucial for the development of the theory of EDRAs' convergence.

2.4. **Convergent re-ranking rules.** Different EDRAs differ (mainly) because of the *re-ranking rule* used in step (21c) in order to update the current ranking vector in response to its failure on the current ERC. This re-ranking operation can take various different forms. One option is to demote by a certain amount (call it the *demotion amount*) certain loser-preferring constraints, as those are the constraints that are responsible for the failure of the current ranking vector on the current ERC. Another option is to promote by a certain amount (call it the *promotion amount*) certain winner-preferring constraints, as those are the constraints that would have helped to avoid the failure of the current ranking vector on the current ERC. These two options are schematized in (22), which thus provides a general scheme for re-ranking rules.

(22) a. Subtract a certain *demotion amount* from the current ranking value of some or all of the loser-preferring constraints;

b. add a certain *promotion amount* to the current ranking value of some or all of the winner-preferring constraints.

---

an ERC $\mathbf{a}$ according to condition (20) provided the following strict inequality (i) holds, that says that the largest ranking value over winner-preferrers is larger than the largest ranking value over loser-preferrers. Furthermore, the inequality (i) can be check in time linear in the number n of constraints. In the end, the consistency condition (20) can thus be efficiently computed.

(i) $\max_{k \in W(\mathbf{a})} \theta_k > \max_{h \in L(\mathbf{a})} \theta_h$

Let me explain why the consistency condition (20) is equivalent to the inequality (i). Suppose that the latter inequality (i) holds. Thus, every refinement of this ranking vector will rank the winner-preferrer that attains the maximum $\max_{k \in W(\mathbf{a})} \theta_k$ above the loser-preferrer that attains the maximum $\max_{h \in L(\mathbf{a})} \theta_h$. In other words, it will rank this winner-preferrer above every loser-preferrer. Every refinement is thus consistent with the ERC $\mathbf{a}$, and condition (20) holds. Vice versa, suppose that the inequality (i) does not hold. Namely that the largest ranking value over winner-preferrers is at most as large as the largest ranking value over loser-preferrers. Thus, the current ranking vector admits a refinement that ranks the loser-preferrer that attains the maximum $\max_{h \in L(\mathbf{a})} \theta_h$ above the winner-preferrer that attains the maximum $\max_{k \in W(\mathbf{a})} \theta_k$. In other words, it admits a refinement that ranks a loser-preferrer above every winner-preferrer. This refinement is thus not consistent with the ERC $\mathbf{a}$, and condition (20) fails. Finally, let me explain why the inequality (i) can be check in time linear in the number n of constraints. Start with $W = -\infty$ and $L = -\infty$. Scan through the current ranking values, for $k = 1, 2, \dots, n$. If θ_k is larger than W (larger than L) and C_k is winner-preferring (loser-preferring), then set $W = \theta_k$ (set $L = \theta_k$). After having scanned all ranking values, condition (i) holds iff $W > L$.

Re-ranking rules differ along three main dimensions. The first dimension is whether the re-ranking rule only performs *constraint demotion* or else performs *constraint promotion* too. The second dimension is whether the re-ranking rule *minimally* demotes only the loser-preferrers that need to be demoted or *maximally* demotes all of them. The third dimension is whether the re-ranking rule performs small, *gradual* updates or instead updates that are so "drastic" that one update suffices to make the current ranking vector consistent with the current ERC, so that no ERC can trigger two consecutive updates. The re-ranking rules considered in the literature are classified in (23) according to these three dimensions, together with the name of the corresponding EDRA.

(23) *re-ranking rules for EDRAs*

- *demotion-only*
  - *gradual*
    - *minimal* $\Longrightarrow$ $\mathrm{GLA}^{\mathrm{dem}}_{\mathrm{min}}$: convergent and efficient
    - *maximal* $\Longrightarrow$ $\mathrm{GLA}^{\mathrm{dem}}_{\mathrm{max}}$: convergent, but not efficient
  - *non-gradual* $\Longrightarrow$ EDCD: convergent and efficient
- *demotion-promotion*
  - *gradual*
    - *minimal* $\Longrightarrow$ $\mathrm{GLA}_{\mathrm{min}}$: not convergent
    - *maximal* $\Longrightarrow$ GLA: not convergent
  - *non-gradual* $\Longrightarrow$ —

Section 3 reviews what is currently known about re-ranking rules that perform constraint demotion only; Section 5 reviews what is currently known about re-ranking rules that perform both constraint demotion and promotion.

An EDRA (with a specific re-ranking rule and a specific initial ranking vector) is said to *converge* on a given input ERC matrix provided that the algorithm can perform only a finite number of updates when trained on any stream of ERCs sampled from that ERC matrix. In other words, the algorithm always eventually settles on a ranking vector consistent with each input ERC, so that the algorithm cannot make any more errors and learning ceases. An EDRA is called *(universally) convergent* provided that it converges no matter the input ERC matrix that it is trained on, as long as it is consistent. Of course, the restriction to consistent input ERCs makes good sense: if they are not consistent with any ranking, then of course the EDRA will not be able to find any consistent ranking. Furthermore, the restriction to consistent input ERCs can be enforced in some cases without loss of generality. For instance, in the case of the acquisition of phonotactics the input ERCs can be guaranteed to be consistent (under only mild assumptions on the constraint set), as we will see in Subsection 4.3. A convergent EDRA is called *efficient* provided that the number of errors made before convergence grows slowly (polynomially) with the number n of constraints, so that the algorithm also works in the case of very large constraint sets. For each of the main EDRAs considered in the literature, the synopsis (23) recalls what is currently known concerning (universal) convergence and efficiency. In particular, it reveals that the ambitious requirement of efficient convergence can be achieved. The next Section takes a close look at this outstanding result.

2.5. **Summary.** The acquisition of phonology is gradual, as the target adult language is approached through a path of intermediate learning stages. EDRAs model gradualness, as they define a sequence of rankings, that can be matched with child acquisition paths. The crucial ingredient in the development of an EDRA is the re-ranking rule used by the algorithm to move from the current to the updated ranking. In order to focus on this crucial implementation issue, I have restated EDRAs in ERC notation (Prince 2002). Furthermore, in order to be able to express re-ranking rules in the compact numerical form

(22), I have assumed that EDRAs entertain a numerical representation of their current ranking, in terms of ranking vectors (Boersma 1997, 1998, 2009).

3. Tesar and Smolensky's analysis of demotion-only EDRAs

This Section reviews the elegant analysis of demotion-only EDRAs developed in Tesar (1995, 1998) and Tesar and Smolensky (1996, 1998, 2000) (henceforth: T&S).[4] T&S's analysis is important because it shows that efficient convergence can indeed be attained. Furthermore, T&S's analysis provides the starting point for the developments in the rest of this paper. My presentation of T&S's analysis owes a lot to Prince's (2002) ERC notation. Furthermore, it underscores the benefit of framing the theory of EDRAs in terms of ranking vectors, rather than in terms of rankings, as they originally did.[5]

3.1. A minimal, gradual demotion-only re-ranking rule.

Consider the current ERC (24a) and the current ranking vector (24b). The two constraints C_4 and C_6 are both currently loser-preferrers. Yet, there is a crucial difference between them. The current ranking value of C_6 is 5, which is smaller than the current ranking value of the winner-preferrer C_1. The current ranking value of C_4 is instead 15, which is larger than the current ranking value of both winner-preferrers C_1 and C_2. This difference between the two loser-preferrers C_4 and C_6 is important enough to warrant a name. A loser-preferrer is called *undominated* in case there is no winner-preferrer with a strictly larger current ranking value. Thus, C_4 is currently undominated, while C_6 is not.

(24) a. $\mathbf{a} = \begin{bmatrix} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 \\ \text{W} & \text{W} & e & \text{L} & e & \text{L} \end{bmatrix}$ b. $\boldsymbol{\theta} = \begin{bmatrix} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 \\ 10 & 5 & 20 & 15 & 100 & 5 \end{bmatrix}$

We can now consider the re-ranking rule (25), that demotes all undominated loser-preferrers by a small fixed amount, here set equal to 1 for concreteness.

(25) a. Decrease by 1 the ranking value of each undominated loser-preferrer;
b. do nothing to the current ranking value of the other constraints.

For instance, if the current ERC is (24a) and the current ranking vector is (24b), then the updated ranking vector is (26): the ranking value of the currently undominated loser-preferrer C_4 is decreased by 1, from 15 to 14; all other ranking values are left unchanged.

(26) $\boldsymbol{\theta}_{\text{updated}} = \begin{bmatrix} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 \\ 10 & 5 & 20 & 14 & 100 & 5 \end{bmatrix}$

The re-ranking rule (25) is clearly *demotion-only*, as the current ranking values of winner-preferrers are never modified. Furthermore, it is *minimal*, because it only demotes the currently undominated loser-preferrers, rather than all of them. The intuition here is that, among the loser-preferrers, the undominated ones are those that really need to be demoted, as they are not currently ranked underneath any winner-preferrer. The other loser-preferrers instead do not really need to be taken care of, as they are already ranked underneath a winner-preferrer. We will see in Subsection 3.6 why it is indeed a good idea to only demote the currently undominated loser-preferrers, rather than all of them. Finally, the re-ranking rule (25) is *gradual*, in the sense that updates are performed by a small fixed amount, so that an ERC might have to trigger multiple consecutive updates before the current ranking vector becomes consistent with it. For instance, a single update

[4] T&S considered the re-ranking rule (38) below, which is the non-gradual counterpart of the re-ranking rule (25). Boersma (1998, p. 323-327) notes that T&S's analysis for the non-gradual re-ranking rule (38) straightforwardly extends to the gradual counterpart (25). Furthermore, Boersma (2009) fixes a small glitch in T&S's analysis.

[5] Although the idea of representing rankings in terms of numerical ranking vectors is actually implicitly already present in T&S's notion of the *offset* of a constraint w.r.t. a ranking, defined as the number of strata above that constraint in that ranking.

of the current ranking vector (24b) by the ERC (24a) is not enough to enforce consistency with that ERC, as the updated ranking vector (26) is still not OT consistent with it. As we will see in Subsection 3.7 , it is easy to construct the non-gradual counterpart of a gradual re-ranking rule. The EDRA (21) with this re-ranking rule (25) is Boersma's (1997) (non-stochastic) *minimal demotion-only Gradual Learning Algorithm*.

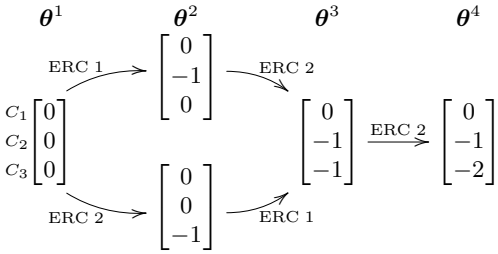
3.2. **The idea of the analysis.** To get some sense of the behavior of an EDRA with this re-ranking rule (25), I provide in (27b) the complete learning paths described by the algorithm on the input ERC matrix (27a) starting from null initial ranking values. At the first iteration, the algorithm can receive either ERC 1 or ERC 2. Suppose that it receives ERC 1. Since the initial ranking vector $\boldsymbol{\theta}^1$ is not consistent with that ERC, the algorithm updates $\boldsymbol{\theta}^1$ to $\boldsymbol{\theta}^2$ by demoting by 1 the ranking value of the currently undominated loser-preferrer C_2. At the next iteration, the algorithm can again receive either ERC 1 or ERC 2. Since the current ranking vector $\boldsymbol{\theta}^2$ is consistent with ERC 1, this ERC triggers no update and nothing happens until the algorithm is fed ERC 2. When that happens, the algorithm detects that the current ranking vector $\boldsymbol{\theta}^2$ is not consistent with ERC 2 and it consequently updates it to $\boldsymbol{\theta}^3$ by demoting by 1 the ranking value of the currently undominated loser-preferrer C_3. And so on.

(27)

a.

	C_1	C_2	C_3
ERC 1	W	L	W
ERC 2	*e*	W	L

b.



No matter the training sequence of ERCs, after three updates the algorithm entertains the ranking vector $\boldsymbol{\theta}^4$, whose only refinement is the ranking $C_1 \gg C_2 \gg C_3$. Since this ranking is consistent with both input ERCs, no further update will be triggered.

The input ERC matrix (27a) is only consistent with the ranking (28a), that assigns C_1 to the first stratum, C_2 to the second stratum and C_3 to the third stratum.

(28) a.

C_1 — stratum 1
|
C_2 — stratum 2
|
C_3 — stratum 3

b.

θ_1 never goes below 0

θ_2 never goes below -1

θ_3 never goes below -2

The ranking value of C_1 never goes below 0 in (27b), as stated in (28b). In fact, constraint C_1 is never loser-preferring in the input ERC matrix, and thus is never demoted from its initial null ranking value. Furthermore, the ranking value of C_2 never goes below -1 in (27b), as stated in (28b). In fact, C_2 can only be demoted through update by ERC 1; in order for that ERC to trigger an update, C_2 must be currently ranked above the winner-preferrer C_1; since C_1 is fixed at the ranking value $\theta_1 = 0$, this means that C_2 can only be demoted when its current ranking value is $\theta_2 = 0$; therefore C_2 cannot ever make it below $\theta_2 = -1$. Finally, the ranking value of C_3 never goes below -2 in (27b), as stated in (28b). In fact, C_3 can only be demoted through update by ERC 2; in order for that ERC to trigger an update, C_3 must be currently ranked above the winner-preferrer C_2; since C_2 can only drop down to $\theta_2 = -1$, this means that C_3 can only be demoted when its current ranking value is $\theta_3 = 0$ or $\theta_3 = -1$; therefore C_3 cannot ever make it below $\theta_3 = -2$.

Every time the EDRA makes an error on the current ERC, at least one constraint is demoted. Every time a constraint is demoted, its ranking value decreases by 1. Thus,

(28b) says that C_1 can never be demoted, C_2 can be demoted at most once and C_3 at most twice. This means in turn that the EDRA can perform at most $0+1+2=3$ errors. We have thus proved convergence on the simple input ERC matrix (27a).

The conclusion obtained in (28) can be summarized as follows: the ranking value θ_k of the constraint C_k assigned to the kth stratum (with the 1st stratum being the top one) can never make it below $-(k-1)$. This statement is called an *invariant*, as it is a property of the current ranking values that holds at any iteration throughout learning. T&S note that this invariant holds in full generality, as explained in Subsection 3.4. The only requirement is consistency of the input ERC matrix, spelled out in Subsection 3.3. Efficient convergence then follows straightforwardly from this invariant, as shown in Subsection 3.5.

3.3. **Characterization of OT-consistency.** T&S's analysis of demotion-only re-ranking rules (as well as the developments of that analysis presented in the rest of this paper) crucially relies on the assumption that the data fed to the EDRA make good sense, namely that the input ERC matrix is consistent. In order to distill useful computational consequences from this assumption, we need explicit characterizations of OT-consistency. I present here the characterization of OT-consistency developed by T&S, using the ERC notation of Prince (2002); in Appendix A.3, I will come back to the issue of the algorithmic implications of the notion of OT-consistency.

To introduce the idea, consider for instance the ERC matrix in (29a). It is consistent with the ranking $C_1 \gg C_2 \gg C_3 \gg C_4 \gg C_5$. Use this ranking as follows. Reorder the columns of the matrix from left-to-right according to this ranking, as in (29b). Then, place at the top the ERCs $\mathbf{a}_1$ and $\mathbf{a}_4$ that have a W corresponding to C_1, as in (29c). Place next the ERC $\mathbf{a}_3$ that has a W corresponding to C_2, as in (29d).

(29)

a.

	C_2	C_5	C_4	C_1	C_3
$\mathbf{a}_1$	W	L	W	W	L
$\mathbf{a}_2$		L			W
$\mathbf{a}_3$	W	L	W		
$\mathbf{a}_4$	L	L		W	
$\mathbf{a}_5$		W	L		W

b.

	C_1	C_2	C_3	C_4	C_5
$\mathbf{a}_1$	W	W	L	W	L
$\mathbf{a}_2$			W		L
$\mathbf{a}_3$		W		W	L
$\mathbf{a}_4$	W	L			L
$\mathbf{a}_5$			W	L	W

c.

	C_1	C_2	C_3	C_4	C_5
$\mathbf{a}_1$	W	W	L	W	L
$\mathbf{a}_4$	W	L			L
$\mathbf{a}_2$			W		L
$\mathbf{a}_3$		W		W	L
$\mathbf{a}_5$			W	L	W

d.

	C_1	C_2	C_3	C_4	C_5	
$\mathbf{a}_1$	W	W	L	W	L	} 1st block
$\mathbf{a}_4$	W	L			L	
$\mathbf{a}_3$		W		W	L	} 2nd block
$\mathbf{a}_2$			W		L	} 3rd block
$\mathbf{a}_5$			W	L	W	

The tableau obtained in (29d) can be described as follows. There is a top block of rows whose first entry is a W. Then there is a second block of rows (here, just one row), whose first entry is an e, followed by a W. Finally, comes a third block whose rows have the first two entries equal to e followed by a W. A straightforward generalization of the procedure illustrated in (29) yields the following characterization of consistent ERC matrices.

FACT 1. *An ERC matrix is consistent if and only if it can be brought into the shape (30), by properly reordering its rows and its columns and by relabeling the constraints.*

(30)

	C_1	C_2	...	...	...	...	C_n
1st block	W						
	|	...	...	...	...	...	...
	W						
2nd block	e	W					
	|	|	...	...	...	...	...
	e	W					
⋮			⋱	...	...	...	...
final block	e	e	—	e	W		
	|	|		|	|	...	...
	e	e	—	e	W		

Namely, it has a top block of rows whose first entry is W*; followed by a second block of rows whose first entry is* e *and whose second entry is* W*; and so on.* ■

3.4. **The crucial invariant.** Consider a run of the EDRA with the demotion-only re-ranking rule (25) and initial null ranking values. As the input ERC matrix is consistent, it can be brought into the form (30), by Fact 1. The ranking value θ_1 of constraint C_1 can never make it down to $\theta_1 = -1$. In fact, suppose by contradiction that it does. As C_1 starts out at $\theta_1 = 0$, this means that C_1 has been demoted at least once. But that is impossible, as the re-ranking rule (25) only demotes loser-preferrers and C_1 does not have a single L in (30). The ranking value θ_2 of constraint C_2 can never make it down to $\theta_2 = -2$. In fact, suppose by contradiction that it does. As the re-ranking rule (25) demotes by 1 only the undominated loser-preferrers, this means that at some point C_2 had a current ranking value of $\theta_2 = -1$ and was currently an undominated loser-preferrer. But that is impossible: in order for C_2 to be loser-preferring, the current ERC must belong to block 1, in which case C_2 is dominated by the winner-preferrer C_1, as the ranking value of C_2 is -1 by hypothesis while the ranking value of C_1 is always 0, as just shown. The cases $k > 2$ are dealt with analogously, by induction on k. We have thus proved the following crucial invariant.

FACT 2. *Assume that the input ERC matrix is consistent with a ranking* $\gg$*. Without loss of generality, assume that this ranking is* $C_1 \gg C_2 \gg \cdots \gg C_n$ *(otherwise, relabel the constraints). Then, the ranking vector* $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k, \ldots, \theta_n)$ *entertained at a generic time by the EDRA (21) run on those input ERCs with the re-ranking rule (25) starting from null initial ranking values satisfies condition (31) for every* $k = 1, \ldots, n$*.*

$$\theta_k \geq -(k-1) \tag{31}$$

Namely, the ranking value θ_k *of the constraint* C_k *assigned to the* k*th stratum (with the 1st stratum being the top one) never goes below* $-(k-1)$*.* ■

3.5. **Efficient convergence.** Suppose that we could construct an consistent input ERC matrix on which the EDRA run with the demotion-only re-ranking rule (25) does not converge but rather performs an infinite number of updates. This means in turn that some constraint gets demoted an infinite number of times. Its ranking value thus becomes arbitrarily small, violating the lower-bound (31). Thus, the algorithm needs to stop after a finite number T of updates. At each update, one or more demotions are performed. Thus, the total number of updates T must be smaller than the total number of demotions, as stated in (32).

$$T \leq \begin{matrix}\#\text{ of times} \\ C_1 \text{ is demoted}\end{matrix} + \begin{matrix}\#\text{ of times} \\ C_2 \text{ is demoted}\end{matrix} + \cdots + \begin{matrix}\#\text{ of times} \\ C_n \text{ is demoted}\end{matrix} \tag{32}$$

Without loss of generality, assume that the input ERC matrix is consistent with the ranking $C_1 \gg C_2 \gg \cdots \gg C_n$. Fact 2 says that C_1 is never demoted; C_2 is demoted at most once; C_3 is demoted at most twice; and so on. Thus, (32) becomes (33).

(33) $T \leq 0 + 1 + \ldots + (n-1)$

Using the well-known identity $0 + 1 + 2 + \ldots (n-1) = \frac{1}{2}n(n-1)$, we conclude with the following Theorem, which says that demotion-only converges efficiently, with a worst-case number of updates quadratic in the number n of constraints.

THEOREM 1. *The EDRA (21) with the demotion-only re-ranking rule (25) run on an arbitrary consistent input ERC matrix corresponding to n constraints starting from null initial ranking values can perform at most $\frac{1}{2}n(n-1)$ mistakes before converging to a ranking vector consistent with the input matrix.* ■

The bound $\frac{1}{2}n(n-1)$ on the worst-case number of updates provided by the Theorem is tight, and thus not improvable. In fact, suppose that there are $n = 4$ constraints and that the input ERC matrix is (34a), that Riggle (2009) calls *diagonal*. The learning path (34b) takes $6 = \frac{1}{2}4(4-1)$ updates to reach convergence.

(34) a. $\begin{matrix}\mathbf{a}_1\\ \mathbf{a}_2\\ \mathbf{a}_3\end{matrix}\begin{bmatrix} \text{W} & \text{L} & & \\ & \text{W} & \text{L} & \\ & & \text{W} & \text{L}\end{bmatrix}$ b. $\begin{bmatrix}0\\0\\0\\0\end{bmatrix} \overset{\mathbf{a}_1}{\to} \begin{bmatrix}0\\-1\\0\\0\end{bmatrix} \overset{\mathbf{a}_2}{\to} \begin{bmatrix}0\\-1\\-1\\0\end{bmatrix} \overset{\mathbf{a}_3}{\to} \begin{bmatrix}0\\-1\\-1\\-1\end{bmatrix} \overset{\mathbf{a}_2}{\to} \begin{bmatrix}0\\-1\\-2\\-1\end{bmatrix} \overset{\mathbf{a}_3}{\to} \begin{bmatrix}0\\-1\\-2\\-2\end{bmatrix} \overset{\mathbf{a}_3}{\to} \begin{bmatrix}0\\-1\\-2\\-3\end{bmatrix}$

So far, I have only considered the case of null initial ranking values. For the extension to arbitrary initial ranking values, see Appendix A.1.

3.6. **Why only demote the loser-preferrers that are undominated.** The re-ranking rule (25) only demotes the loser-preferrers that really need to be demoted, namely the currently undominated ones, that are not currently ranked underneath a winner-preferrer. What happens if we demote all loser-preferrers, namely both the dominated and the undominated ones? For the sake of explicitness, consider the variant in (35). The EDRA (21) with this re-ranking rule is Boersma's (1997) (non-stochastic) *demotion-only GLA*.

(35) a. Decrease by 1 the current ranking value of each loser-preferrer;
b. do nothing to the current ranking values of the winner-preferrers.

For instance, if the current ERC is again (36a) and the current ranking vector is again (36b), then the updated ranking vector is (36c): the ranking value of both loser-preferrers C_4 and C_6 is decreased by 1, despite the fact that only the former is currently undominated.

(36) a. $\mathbf{a} = \begin{matrix}C_1 & C_2 & C_3 & C_4 & C_5 & C_6\end{matrix}\begin{bmatrix} \text{W} & \text{W} & e & \text{L} & e & \text{L}\end{bmatrix}$ b. $\boldsymbol{\theta} = \begin{matrix}C_1 & C_2 & C_3 & C_4 & C_5 & C_6\end{matrix}\begin{bmatrix} 10 & 5 & 20 & 15 & 100 & 5\end{bmatrix}$

c. $\boldsymbol{\theta}_{\text{updated}} = \begin{matrix}C_1 & C_2 & C_3 & C_4 & C_5 & C_6\end{matrix}\begin{bmatrix} 10 & 5 & 20 & \textcircled{14} & 100 & \textcircled{4}\end{bmatrix}$

The invariant (31) says that constraints cannot be demoted too much, as long as we only demote the constraints that need to be demoted, namely the currently undominated loser-preferrers. This invariant does not hold anymore if all loser-preferrers are demoted, both the dominated and the undominated ones. As a counterexample, consider the run in (37b) with the re-ranking rule (35): the ranking value of the constraint C_2 drops down to -3, despite the fact that the input ERC matrix (37a) is consistent with the ranking $C_1 \gg C_2 \gg C_3 \gg C_4$ that assigns C_2 to the second stratum.

(37) a. $\begin{matrix}\text{ERC 1}\\ \text{ERC 2}\\ \text{ERC 3}\end{matrix}\begin{matrix}\begin{bmatrix} \text{W} & \text{L} & & \\ \text{W} & \text{L} & \text{L} & \\ \text{W} & \text{L} & \text{L} & \text{L}\end{bmatrix}\\ \begin{matrix}C_1 & C_2 & C_3 & C_4\end{matrix}\end{matrix}$ b. $\begin{matrix}C_1\\C_2\\C_3\\C_4\end{matrix}\begin{bmatrix}0\\0\\0\\0\end{bmatrix} \xrightarrow{\text{ERC 1}} \begin{bmatrix}0\\-1\\0\\0\end{bmatrix} \xrightarrow{\text{ERC 2}} \begin{bmatrix}0\\-2\\-1\\0\end{bmatrix} \xrightarrow{\text{ERC 3}} \begin{bmatrix}0\\-3\\-2\\-1\end{bmatrix}$

For the case of the re-ranking rule (25) that only demotes currently undominated loser-preferrers, the invariant (31) entails efficient convergence, with an error-bound quadratic in the number of constraints. For the case of the re-ranking rule (35) that demotes all loser-preferrers, convergence still holds but efficiency does not, as the worst-case number of errors can be shown to be exponential in the number of constraints; see Magri (2009). In conclusion, it is a good idea to only demote those loser-preferrers that need to be demoted, namely that are currently undominated.

3.7. **A non-gradual variant.** The re-ranking rule (25) is *gradual*, as it modifies the current ranking values only by a small fixed amount. Thus, multiple consecutive updates by a single ERC might be needed in order for the current ranking vector to become consistent with that ERC. For instance, the top path in the diagram (27b) shows that two consecutive updates by ERC 2 are needed before the current ranking vector becomes consistent with that ERC. We might have wanted to save time, replacing those two updates with a single jump. To get the jump, we should have demoted more the first time we encountered ERC 2, so that no further consecutive update by that ERC would have been necessary. A re-ranking rule is called *non-gradual* provided it performs updates large enough that the current ranking vector becomes consistent with the current ERC after a single update and thus no ERC can trigger two consecutive updates. To illustrate, consider the re-ranking rule (38) that demotes currently undominated loser-preferrers all the way underneath the currently top ranked winner-preferrer. The EDRA (21) with this re-ranking rule is T&S's *Error-Driven Constraint Demotion* (henceforth: EDCD).

(38) a. Decrease[6] the current ranking value of each undominated loser-preferrer to θ^*–1, where θ^* is the largest ranking value over winner-preferrers;
b. do nothing to the current ranking value of the other constraints.

For instance, if the current ERC is again (39a) and the current ranking vector is again (39b), then the updated ranking vector is (39c). Only the ranking value of C_4 is modified, as it is the only currently undominated loser-preferrer. And it is decreased from its original value 15 down to the updated value 9, namely the ranking value $\theta^* = 10$ of the currently top-ranked winner-preferrer C_1, further decreased by 1.

(39) a. $\mathbf{a} = \begin{bmatrix} \overset{C_1}{\text{W}} & \overset{C_2}{\text{W}} & \overset{C_3}{e} & \overset{C_4}{\text{L}} & \overset{C_5}{e} & \overset{C_6}{\text{L}} \end{bmatrix}$ b. $\boldsymbol{\theta} = \begin{bmatrix} \overset{C_1}{10} & \overset{C_2}{5} & \overset{C_3}{20} & \overset{C_4}{15} & \overset{C_5}{100} & \overset{C_6}{5} \end{bmatrix}$

c. $\boldsymbol{\theta}_{\text{updated}} = \begin{bmatrix} \overset{C_1}{10} & \overset{C_2}{5} & \overset{C_3}{20} & \overset{C_4}{\textcircled{9}} & \overset{C_5}{100} & \overset{C_6}{5} \end{bmatrix}$

The undominated loser-preferrer C_4 is demoted in (39) by an amount of 6. This amount is large enough that the updated ranking vector (39c) is consistent with the ERC after just one update. Indeed, the re-ranking rule (38) is *non-gradual* because no ERC can trigger two consecutive updates. The non-gradual re-ranking rule (38) is a "speeded up" version of the gradual re-ranking rule (25). Thus, it comes as no surprise that the analysis developed above for the gradual re-ranking rule (25) extends to the non-gradual variant (38), ensuring efficient convergence also in the latter case.

3.8. **Summary.** In this Section, we have seen that efficient convergence is possible, at least for the case of EDRAs that only perform constraint demotion. The core of the analysis can be summarized as follows: every time the algorithm performs an update, some ranking values are decreased; yet, ranking values cannot decrease too much, because of the invariant (31); thus, the algorithm cannot make too many updates. Furthermore, we have learnt some general lessons on how to devise good re-ranking rules. First, not

[6]The operation in (38a) indeed *decreases* the ranking value of the undominated loser-preferrers. In fact, a loser-preferrer is undominated provided that its current ranking value is larger than or at least equal to the ranking value θ^* of the currently top-ranked winner-preferrer.

all loser-preferrers should be demoted, but only those that really need to, namely the currently undominated ones. Second, it is easy to assemble a convergent non-gradual re-ranking rule from a gradual one, by collapsing multiple updates by the same ERC.

4. Some constraint promotion is needed too

In the preceding Section, we have looked at re-ranking rules from a purely computational perspective. In particular, we have focused on re-ranking rules that only perform constraint demotion but no constraint promotion. And we have seen that they have remarkable computational properties, as the corresponding EDRA efficiently convergences for any consistent input ERC matrix. In this section, we look at re-ranking rules from a modeling perspective: is demotion-only enough from a modeling perspective, or is some constraint promotion needed too? This issue has been addressed so far only marginally in the OT literature. Fikkert and De Hoop (2009) raise the question: "Does the (re)ranking of constraints involve the demotion of markedness constraints, the promotion of faithfulness constraints, or can it be achieved by both the demotion and the promotion of constraints?" (p. 311). But they quickly dismiss the issue, suggesting that "in practice the two approaches [constraint promotion and constraint demotion] are highly similar" (p. 319). Gnanadesikan (2004) endorses constraint promotion: "The process of acquisition is one of promoting the faithfulness constraints to approximate more and more closely the adult grammar, and produce more and more marked forms" (p. 73). But she does not provide arguments in support of constraint promotion, nor does she provide an explicit re-ranking rule. Bernhardt and Stemberger (1998), Stemberger and Bernhardt (1999, 2001), and Stemberger et al. (1999) defend constraint promotion too: "We are unsure as to how constraints are generally re-ranked. They may always be re-ranked higher. [...] We suggest that the typical way that children learn the ranking of constraints is to re-rank faithfulness constraints so that faithfulness increases" (1999). They discuss a few specific cases, but are not computationally explicit. Boersma (1997, 1998) and Boersma and Hayes (2001) provide the first computationally explicit argument in favor of constraint promotion, arguing that demotion-only is unable to model certain cases of language variation. Yet, this argument is framed within a stochastic variant of the traditional OT framework considered in this paper. In this section, I contribute to this debate a new explicit argument in favor of constraint promotion. I argue that demotion-only is insufficient for one of the most important modeling applications of EDRAs, namely modeling the early stage of the child acquisition of phonotactics.

4.1. Modeling the early stage of the acquisition of phonotactics.

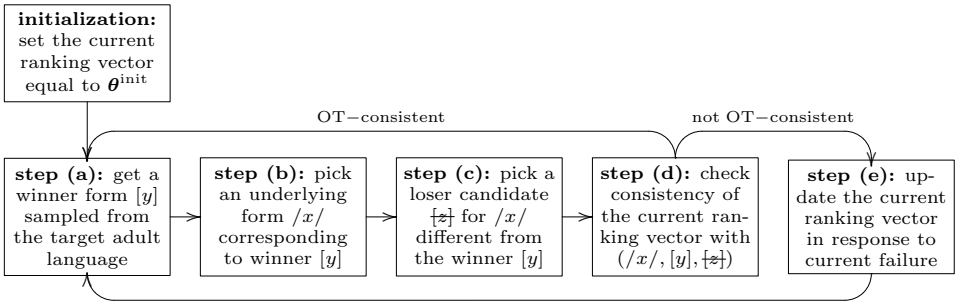
In carefully controlled experimental conditions, nine-month-old infants already react differently to licit and illicit sound combinations (Jusczyk et al. 1993, among others). They thus display knowledge of phonotactics at an early stage, when other linguistic abilities are still not fully developed. In particular, *morphology* is still lagging behind at this early age, so that the child still has no access to phonological alternations. Hayes (2004) thus concludes that "it seems a reasonable guess that in general, the learning of patterns of alternation [that only comes with knowledge of morphology] lags the learning of the contrast and phonotactic systems." In other words, there is a developmental stage, called the *early stage* of the acquisition of phonotactics, throughout which the child acquires phonotactics without the aid of phonological alternations.

Another crucial property of the acquisition of phonotactics is *gradualness*: the target adult grammar is approached through a path of intermediate stages (for classical examples, see Levelt et al. 2000, Gnanadesikan 2004, Pater and Barlow 2003 and Smit et al. 1990, among others; for a review, see Zamuner et al. 2005 and McLeod et al. 2001, among others). Assume that this gradualness reflects grammatical development, rather than just the slow development of performance factors that are orthogonal to linguistic competence (say the slow development of the articulators and of the corresponding motor programs);

see Smolensky (1996a), as well as Hale and Reiss (1998) for critical discussion. From this perspective, EDRAs provide an ideal model of the acquisition of phonotactics, as they describe a path within the space of possible phonotactics that can be matched with attested acquisition paths. What should a proper error-driven ranking model of the early stage of the acquisition of phonotactics look like?

To address this question, let's consider the full-blown description of EDRAs in (40). The algorithm maintains a current ranking vector; initializes it to an assigned *initial* ranking vector; and keeps updating it by looping through the five steps (40a)-(40e). At step (40a), the algorithm receives a form $[y]$ from the target adult language. At step (40b), the algorithm needs to apply some subroutine in order to figure out a corresponding underlying form $/x/$. And at step (40c), it needs to choose a corresponding loser form $[\not{z}]$ (so far, I had collapsed these three steps into a single one, as the focus was on the computational rather than the modeling side). At step (40d), the algorithm checks whether the current ranking vector is consistent with the underlying/winner/loser form triplet $(/x/, [y], [\not{z}])$ thus assembled. And if it isn't, then it takes action at step (40e).

(40)



Various implementation details now need to be specified, concerning the choice of the initial ranking vector; the subroutine that provides the underlying form x at step (40b); and the re-ranking rule to be used at step (40e).[7] I discuss these three issues in turn in Subsections 4.2-4.4. The conclusion of the argument will be that constraint promotion is crucially needed in order to turn the algorithmic kernel (40) into a proper model of the early stage of the child acquisition of phonotactics.

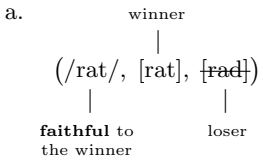
4.2. **Choice of the initial ranking vector.** There is wide agreement in the literature that markedness constraints should start out initially ranked above faithfulness constraints. For instance, Fikkert and De Hoop (2009, p. 325) write: "The recurrent pattern in child language data is that children's output is considerably less marked than the corresponding adult target forms. [...]. Hence, the starting hypothesis in much research on phonological acquisition is that children begin with markedness constraints outranking faithfulness constraints." See Smolensky (1996a,b) for theoretical arguments in favor of this hypothesis; Jusczyk et al. (2002) for empirical evidence; Davidson et al. (2004) for a review; and Hale and Reiss (1998) for critical discussion. I thus assume that the initial ranking value of markedness constraints is larger than that of faithfulness constraints. For instance, assume that the former is some large positive constant $\theta^{\text{init}} > 0$ while the latter is zero.

4.3. **Choice of the underlying form.** At step (40a), the EDRA is fed a licit surface form $[y]$ from the target adult language and needs to pick a proper corresponding underlying form $/x/$ at step (40b), as well as a loser form $[\not{z}]$ at step (40c). To see how to proceed, let's consider a concrete example. Suppose the learner is provided with the winner form $[y] = [\text{rat}]$. To keep things simple, suppose the only option for the loser is $[\not{z}] = [\not{\text{rad}}]$. And

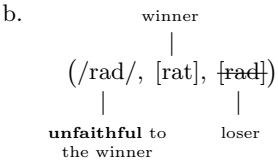
[7]Another implementation detail concerns the subroutine that provides the loser form $[\not{z}]$ at step (40c). Here, I ignore this issue, as it is immaterial to my argument. See Magri (2012c) for some discussion.

that there are two options for the underlying form: the learner could select the underlying form /x/ = /rat/ faithful to the winner, and thus construct the triplet (41a); or it could select the underlying form /x/ = /rad/ unfaithful to the winner, and thus construct the triplet (41b).

(41) a.



b.



The choice of the non-faithful underlying form in the data triplet (41b) is equivalent to the assumption that the target phonology enforces *final devoicing*. But this assumption might be dangerous. In fact, a crucial property of the early stage of the acquisition of phonotactics is that the learner is still blind to alternations, as recalled above. At this stage, the learner is thus not in a position to evaluate his assumption that the target phonology enforces final devoicing. The choice of a non-faithful underlying form might thus turn out to fool the learner into positing inconsistent data triplets at steps (40a)-(40c). I thus assume that the learner posits an underlying form faithful to the winner, as illustrated in the triplet (41a) and stated in (42). This assumption (42) models the fact that the child is blind to alternations throughout the early stage of the acquisition of phonotactics; see Prince and Tesar (2004) and Hayes (2004).

(42) At step (40b), the EDRA posits an underlying form /x/ identical to the given surface form [y], thus always building faithful triplets such as (41a).

Of course, this assumption (42) only makes sense provided that the set of underlying forms coincides with the set of surface forms, so that the same phonological structure can be construed both as an underlying and as a surface form. Assumption (42) is particularly well suited for segmental phonotactics. Now, assume that it is indeed the case that the set of underlying forms coincides with the set of surface forms. Is it the case that assumption (42) is computationally sound? In other words, can we guarantee that the underlying/winner/loser form triplets thus constructed at steps (40a)-(40c) are consistent? That turns out indeed to be the case under only mild assumptions on the constraint set, as recently shown by Tesar (2008). Tesar's result means in particular that, at least in the case of the acquisition of phonotactics, OT-consistency of the data fed to the algorithm can be assumed without loss of generality.

4.4. **Choice of the re-ranking rule.** We are now left with the crucial choice of the re-ranking rule that a proper error-driven ranking model of the early stage of the acquisition of phonotactics should use at step (40e). From a computational perspective, a crucial issue that bears on the choice of the re-ranking rule is that of *efficient convergence*: which choices ensure that the number of errors is finite and grows slowly with the number of constraints? From a modeling perspective, there are two more issues that bears on the choice of the re-ranking rule. Suppose that the EDRA indeed converges. Thus, its final grammar successfully rules in every *licit* form — otherwise the algorithm could still make mistakes and thus cannot have converged. Yet, phonotactics is the knowledge of the distinction between licit vs. illicit forms. If the final grammar, say, ranks all faithfulness constraints at the top, then it fails at ruling out any *illicit* form and the EDRA has effectively learned nothing. Thus, a second issue that bears on the proper choice of the re-ranking rule to be used at step (40e) is that of *restrictiveness*: which choices ensure that the final grammar entertained by the EDRA at convergence successfully manages to rule out *illicit* forms? Finally, suppose that the EDRA converges to a restrictive final grammar. On its way to that final grammar, the EDRA entertains a path of intermediate rankings, each corresponding to an intermediate OT phonotactics. Thus, a third issue that bears

on the proper choice of the re-ranking rule to be used at step (40e) is that of *matching*: which choices yield learning sequences that best match attested child acquisition paths?

The focus of this paper is squarely on the first of these three issues, namely efficient convergence. As reviewed in Section 3, demotion-only re-ranking rules fare well from the computational perspective of convergence. Yet, in this Subsection I will take a detour from my main focus on convergence, and look at the choice of the re-ranking rule from the modeling perspective of restrictiveness and matching between predicted learning sequences and child acquisition paths. I will argue that, from the latter perspective, demotion-only is insufficient and that some degree of constraint promotion is needed. This conclusion will motivate the rest of the paper, that develops provably convergent re-ranking rules that perform both constraint demotion and promotion.

4.4.1. *Choice of the re-ranking rule from the perspective of child acquisition paths.* Suppose that the EDRA uses at step (40e) a re-ranking rule that performs constraint demotion but no constraint promotion, as the re-ranking rules (25) or (38) studied in Section 3. By (42), the algorithm posits at step (40b) underlying forms that are fully faithful to the intended winners. Thus, the faithfulness constraints are never loser-preferring. As demotion-only re-ranking rules only re-rank loser-preferrers, the model will never re-rank the faithfulness constraints throughout learning. This means in turn that the model predicts learning sequences where the repair strategy for a given marked structure never changes over time, as the choice of the repair strategy is determined by the relative ranking of the faithfulness constraints. And this cannot be right, as child acquisition paths do show changes in the repair strategies over time. A certain amount of constraint promotion is thus needed in order for the model to have a chance at matching child acquisition paths. Let me make this point explicit with a couple of examples.

English learning children go through various intermediate learning stages where they reduce licit onset consonant clusters to a singleton consonant. Crucially, the strategies used to determine the singleton consonant vary over time. Based on a large review of the literature, McLeod et al. (2001) show that child cluster simplification typically starts out as deletion of one of the two consonants. And that at a subsequent stage, deletion is replaced by coalescence of the two target consonants into a singleton consonant, at least for certain targets. This learning dynamics crucially requires re-ranking of the two faithfulness constraints Max (that militates against consonant deletion) and Uniformity (that militates against consonant coalescence). By (42), the EDRA will be trained on faithful underlying/winner/loser form triplets such as (/kl/, [kl], [~~k~~]) or (/s_1m_2/, [s_1m_2], [~~$f_{1,2}$~~]). Thus, neither Max nor Uniformity will ever be loser-preferring constraints. Hence, neither of them will ever be re-ranked by demotion-only re-ranking rules, as these rules only re-rank the loser-preferrers. The demotion-only EDRA is thus unable to model even the very rough outline of the child acquisition path towards English consonant clusters.

The same point can be made by looking at different acquisition stages entertained by different children. Pater and Barlow (2003) review production data from two children, Gitanjali (age 2:3-2:9; Gnanadesikan 2004) and Julia (age 1:7-1:9; Compton and Streeter 1977). The two children display a different pattern of coalescence for targets consisting of a stop plus an approximant. Julia preserves the continuancy of the approximant (e.g. [fɪm] *cream*) while Gitanjali preserves the continuancy of the stop (e.g. [pait] *quite*). Pater and Barlow model the two different coalescence patterns assuming that Julia is currently ranking Ident[+Continuant] above Ident[−Continuant] while Gitanjali is entertaining the opposite ranking. Assume that the two children start from the same initial ranking vector, namely that UG has a built-in initialization step. By (42), the EDRA assumes underlying forms faithful to the winner. If it performs demotion-only, it will never re-rank the two faithfulness constraints Ident[+Continuant] and Ident[−Continuant] throughout learning, and will thus be unable to explain how the two children are currently entertaining two different relative rankings of those two faithfulness constraints.

4.4.2. *Choice of the re-ranking rule from the perspective of restrictiveness.* Assume that the EDRA adopts a demotion-only re-ranking rule, and thus only re-ranks loser-preferring constraints. Furthermore, assume by (42) that it posits underlying forms faithful to the winner, so that the faithfulness constraints are never loser-preferrers. Thus, the EDRA will never re-rank the faithfulness constraints. As just seen, this cannot be right from the perspective of modeling child acquisition paths. Let me now argue that it cannot be right also from the perspective of restrictiveness. In fact, if the faithfulness constraints are never re-ranked through learning, their final relative ranking is insensitive to the language the algorithm has been trained on. Suppose that restrictiveness requires the opposite relative ranking of some faithfulness constraints for a certain pair of anguages in the OT typology. A demotion-only EDRA that posits faithful underlying forms will not re-rank the faithfulness constraints and it will thus be unable to learn their different relative ranking needed for the two languages. Some constraint promotion is needed in order to re-rank the faithfulness constraints too, despite the fact that they are never loser-preferrers.

Let me illustrate these considerations with example (43), which is a simplified version of an example considered in Prince and Tesar (2004), based on Lombardi (1999). Both features [STOP-VOICING] and [FRICATIVE-VOICING] come with a specific faithfulness and markedness constraint, namely F_1, M_1 and F_2, M_2. The markedness constraint M_3, which requires adjacent obstruents to agree in voicing, lets the two features interact. The generating function only modifies voicing.

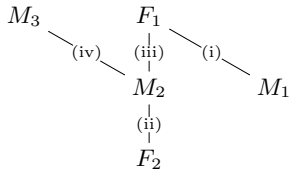
(43) a. $\{$ pa, ba, sa, za, apsa, apza, absa, abza, $\}$

b. $\left\{ \begin{array}{cc} F_1 = \text{IDENT[STOP-VOICING]} & M_1 = {}^*\text{[STOP-VOICING]} \\ F_2 = \text{IDENT[FRICATIVE-VOICING]} & M_2 = {}^*\text{[FRICATIVE-VOICING]} \\ & M_3 = \text{AGREE} \end{array} \right\}$

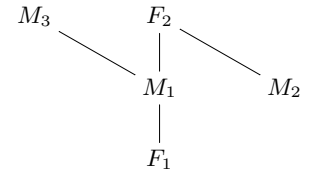
Among the languages in the OT typology (43), there are the two in (44). A ranking generates these two languages (44a) and 44b) provided it is a refinement of the partial orders in (45a) and (45b), respectively. Consider for instance the case of language (44a). In order for /ba/ not to be neutralized to [pa], the ranking (45a.i) is needed. In order for /za/ to be neutralized to [sa], the ranking (45a.ii) is needed. Given ranking (45a.ii), in order for /abza/ not to be neutralized to [apsa], the ranking (45a.iii) is needed. Furthermore, given the ranking (45a.ii), in order for /abza/ not to be neutralized to [absa], the ranking (45a.iv) is needed too.

(44) a. $\left\{ \begin{array}{cccc} \text{pa}, & \text{ba}, & \text{sa}, & \\ \text{apsa}, & & & \text{abza}, \end{array} \right\}$ b. $\left\{ \begin{array}{ccc} \text{pa}, & \text{sa}, & \text{za}, \\ \text{apsa}, & & \text{abza}, \end{array} \right\}$

(45) a.



b.



Because of the assumption (42) of underlying forms fully faithful to the intended winners, the two faithfulness constraints F_1 and F_2 are never loser-preferring. If the EDRA (40) is implemented with a demotion-only re-ranking rule that only demotes loser-preferrers, such as (25) or (38), then the two faithfulness constraints F_1 and F_2 will stay put at their initial ranking values. In other words, the final relative ranking of F_1 and F_2 will be the same, no matter which of the two languages (44) the algorithm is trained on. The algorithm will thus fail to acquire at least one of the two phonotactics, as they require the opposite relative ranking of F_1 and F_2, by (45). These considerations show that constraint promotion is a *necessary* condition in order for EDRAs to provide a proper model of the early stage

of the acquisition of phonotactics, as promotion allows the faithfulness constraints to be re-ranked too throughout learning, despite the fact that they are never loser-preferrers.[8]

4.4.3. *More on restrictiveness and constraint promotion.* What happens when an EDRA that performs both constraint demotion and promotion is trained on the two languages (44)? Is it indeed able to rank the faithfulness constraints correctly, as in the target rankings (45)? The input ERC matrices corresponding to the two languages (44a) and (44b) are provided in (46a) and (46b), respectively. Let me explain for instance how the matrix (46a) has been computed. The target language (44a) consists of the five surface forms [pa], [ba], [sa], [apsa], and [abza]. Of these five forms, only the two forms [ba] and [abza] are informative, as the other three forms are unmarked and therefore do not contribute to learning (their ERCs never trigger any updates). For the informative form [ba], we need to consider only one underlying/winner/loser triplet, namely the one that pairs up this winner surface form [ba] with the faithful underlying form /ba/ and the loser form ~~[pa]~~. For the other informative form [abza], we need to consider three triplets, that pair up this winner surface form [abza] with the faithful underlying form /abza/ and the three loser forms ~~[apsa]~~, ~~[apza]~~, and ~~[absa]~~. The ERC matrix (46a) is obtained by assembling, one on top of the other, the ERCs corresponding to these four triplets.

(46) a.

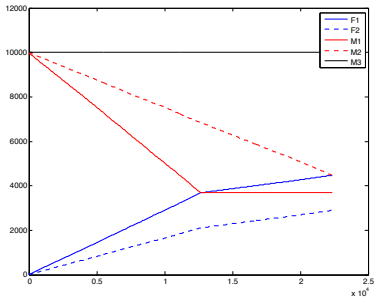
	F_1	F_2	M_1	M_2	M_3
(/ba/, [ba], ~~[pa]~~)	W		L		
(/abza/, [abza], ~~[apsa]~~)	W	W	L	L	
(/abza/, [abza], ~~[apza]~~)	W		L		W
(/abza/, [abza], ~~[absa]~~)		W		L	W

b.

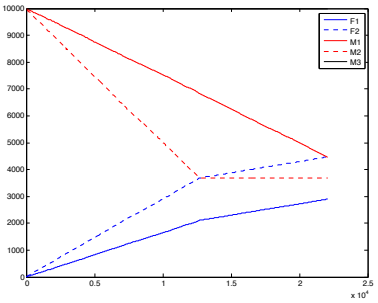
	F_1	F_2	M_1	M_2	M_3
(/za/, [za], ~~[sa]~~)		W		L	
(/abza/, [abza], ~~[apsa]~~)	W	W	L	L	
(/abza/, [abza], ~~[apza]~~)	W		L		W
(/abza/, [abza], ~~[absa]~~)		W		L	W

I provide in (47a) and (47b) the dynamics of the ranking values of the five constraints predicted by the EDRA with a promotion-demotion re-ranking rule trained on the two input matrices (46a) and (46b), respectively.[9] The initial ranking vector used in the simulations has been chosen according to Subsection 4.2: the faithfulness constraints are initially ranked at the bottom (with a null initial ranking value) and the markedness constraints at the top (with an initial ranking value of $\theta^{\text{init}} = 10,000$). The simulations use the (efficiently convergent) re-ranking rule (64) developed below in Section 6: the details of the re-ranking do not matter here, but for the fact that constraint promotion is performed too, besides demotion. Finally, the ERCs are sampled uniformly from the input ERC matrix.

(47) a.



b.



[8] As pointed out by an associate editor, Prince and Tesar's (2004) *Biased Constraint Demotion* and Hayes's (2004) *Low Faithfulness Constraint Demotion* succeed on the test case considered here. But this fact does not affect my point. In fact, the latter algorithms are *batch*: they are allowed to glimpse at the entire set of data at once. This paper focuses on the very different *error-driven* algorithmic scheme: the final ranking needs to arise as the result of a sequence of instantaneous choices based on a single piece of data at the time. My point here is just that constraint demotion is not sufficient for restrictiveness within *error-driven* learning.

[9] The diagrams (47) were drawn using the Python file `ERC-EDRA.py`, available on the author's website.

(48) a. $\begin{bmatrix} F_1 & F_2 & M_1 & M_2 & M_3 \\ 4473 & 2896 & 3684 & 4472 & 10000 \end{bmatrix}$ b. $\begin{bmatrix} F_1 & F_2 & M_1 & M_2 & M_3 \\ 2897 & 4473 & 4472 & 3684 & 10000 \end{bmatrix}$

The final ranking vectors that the promotion/demotion EDRA converges to in the two simulations (47a) and (47b) are provided in (48a) and (48b), respectively. Surprisingly, these two final ranking vectors (48a) and (48b) correctly represent the two target rankings (45a) and (45b). Consider for instance the final ranking vector (48a): the ranking value of F_1 is larger than both the ranking values of M_1 and M_2, thus representing the target rankings (45a.i) and (45a.iii); the ranking value of M_2 is larger than the ranking value of F_1, thus representing the target ranking (45a.ii); finally, the ranking value of M_3 is larger than the ranking value of M_2, thus representing the target ranking (45a.iv). In particular, the promotion component of the re-ranking rule has allowed the EDRA to learn the correct relative ranking of the two faithfulness constraints F_1 and F_2.

A quick look at the input ERC matrices (46) reveals what is behind this success. Consider for instance the case of the input matrix (46a). The three markedness constraints M_1, M_2 and M_3 start out with the same initial ranking value. After one update by the penultimate ERC, M_1 is demoted and M_3 is promoted. The current ranking vector thus enforces the ranking configuration $M_3 \gg M_1$, that ensures consistency with this ERC. Furthermore, this ranking configuration will never be disrupted in the rest of the learning process, as M_3 can never be demoted (because it is never loser-preferrer) and M_1 can never be promoted (because it is never winner-preferrer). Thus, the penultimate ERC can trigger at most one update, and is therefore pretty much irrelevant for the overall ranking dynamics. Analogous considerations hold for the last ERC. In the end, the ranking dynamics is thus completely determined by the first two ERCs. The first ERC only promotes F_1 while the second ERC promotes both F_1 and F_2. Thus, a single update by the first ERC is sufficient to ensure that F_1 will be ranked above F_2 from that time on. In other words, a single update by the first ERC is enough to guarantee that the EDRA will converge to the correct relative ranking of the two faithfulness constraints. As F_1 is thus ranked above F_2, the two markedness constraints M_1 and M_2 will intercept F_1 first in their free fall. As soon as they cross F_1, learning ceases, as the current ranking vector has become consistent with the entire ERC matrix. M_1 and M_2 thus find themselves squeezed in between F_1 and F_2, as desired. Although the input ERCs are sampled uniformly in the simulations reported above, the analysis just sketched reveals that the EDRA will converge to the correct final ranking no matter how the input ERCs are sampled, as long as the first ERC gets a chance to trigger a few updates. In conclusion, the OT typology in (43) has the following remarkable property: the only two languages (44) that require a faithfulness constraint to be ranked above another faithfulness constraint correspond to input ERC matrices that are able to train the EDRA to learn that relative ranking.

In Magri (2012d), I show that EDRAs are restrictive on any language that does not require any faithfulness constraint to be ranked above another faithfulness constraint. As expected, such languages are the vast majority: the relative ranking of the faithfulness constraints matters for the way illicit structures are repaired; only rarely it matters for the divide between licit and illicit structures. What about the remaining languages, that require a specific relative ranking of the faithfulness constraints? In Magri (2010, 2011b), I show that restrictiveness cannot be achieved in the general case by any algorithmic scheme. In other words, there is no learning algorithm from positive evidence that can ensure restrictiveness, unless we make assumptions on the underlying OT typologies. This result motivates the following conjecture: is it the case that phonologically plausible OT typologies happen to have the property just observed for the typology (43), namely that every language in the typology that requires a certain faithfulness constraint to be ranked above another faithfulness constraint correspond to an input ERC matrix that is able tot rain the EDRA to learn that relative ranking? If this conjecture turns out to be correct, it will provide formidable support for the hypothesis that error-driven learning

is a proper model of the child's acquisition of phonotactics. In Magri (2011a) and Magri (2012d), I report a first result in this direction. I consider all possible constraints of the type of M_3 in (43), that is responsible for the interaction between the two features that define the typology. And I show that promotion/demotion EDRAs are restrictive, but for phonologically implausible models of feature interaction.

4.5. **Summary.** Tesar and Smolensky (1998) develop EDCD, reviewed in Section 3. Its signature property is that it performs constraint demotion, but no constraint promotion. Lack of constraint promotion allows Tesar and Smolensky (1998) to prove that EDCD converges with an efficient error-bound (cf. also Boersma 2009). Although a virtue from a *computational* perspective, lack of constraint promotion turns out to be a liability from a *modeling* perspective, as argued in this Section. I have looked at one of the main modeling applications of EDRAs, namely modeling the early stage of the child acquisition of phonotactics. In this case, it makes sense to assume that the EDRA is provided with winner forms only, and that it assumes underlying forms faithful to the winners. Training on faithful mappings entails that the faithfulness constraints are never loser-preferrers. As EDCD only demotes loser-preferrers, it thus never re-ranks the faithfulness constraints. This cannot be right from the perspective of *restrictiveness*: if two languages in the typology require the opposite relative ranking of some faithfulness constraints, EDCD fails to learn that. Also, this cannot be right from the perspective of *matching* the predicted learning sequences with child acquisition paths: if an acquisition path shows a succession of different repair strategies for the same marked structure, EDCD fails to model that. In conclusion, EDCD is unable to implement the error-driven learning model of the acquisition of phonotactics. And some amount of constraint promotion is needed, in order to re-rank the faithfulness constraints too, despite the fact that they are never loser-preferrers. This conclusion motivates the research question addressed in the rest of this paper: is it possible to devise convergent re-ranking rules that perform constraint promotion in addition to demotion? To get started, Section 5 reviews what is currently known in the OT computational literature concerning constraint promotion.

5. The computational challenge raised by constraint promotion

Demotion-only re-ranking rules are easy to analyze by induction on the constraints, as in the proof of the crucial Fact 2 in Section 3. In fact, as they only re-rank loser-preferrers, there is always at least one constraint that is never re-ranked (because every consistent ERC matrix contains at least one constraint that is never loser-preferring). And this constraint can thus be used as the base of the induction. Furthermore, demotion-only re-ranking rules only decrease the current ranking values. They thus ensure a monotonic dynamics of the ranking values, that yields a simple inductive step. The situation is very different for re-ranking rules that perform constraint promotion in addition to demotion. In this Section, I illustrate the computational challenge raised by constraint promotion, with a review of the unsuccessful attempts at constraint promotion made so far in the literature.

5.1. **The credit problem.** Suppose that the current ERC fed to the EDRA has multiple W's, say it has two W's corresponding to the two constraints C_h and C_k, as in (49a). A promotion/demotion EDRA needs to decide which one of the two winner-preferrers C_h or C_k should be *credited* for OT-consistency with the current ERC. And the decision must be taken instantaneously, without looking at the rest of the ERC matrix. This is a very challenging task. In fact, the ERC matrix could contain another ERC like (49b), which says that only C_h should be credited for taking care of the ERC (49a). This is a specific instance of what Dresher (1999) called the *credit problem*.

(49) a.

		C_h	C_k		C_ℓ	
[	...	W	W	...	L	...]

b.

		C_h	C_k		C_ℓ	
[	...	e	L	...	W	...]

A demotion-only re-ranking rule gets around the credit problem by avoiding performing any promotion. But a promotion/demotion re-ranking rule might in principle get fooled by the credit problem. For this reason, Tesar and Smolensky (1998, pp. 244-245) explicitly warn against constraint promotion, in the passage quoted in (50).

(50) "At least one [winner-preferrer] must dominate all [loser-preferrers]. Demotion moves the [loser-preferrers]. [...] Once the highest-ranked [winner-preferrer] is identified, all of the [loser-preferrers] need to be dominated by it, so all [loser-preferrers] are demoted if not already so dominated. A hypothetical promotion operation would move the constraints corresponding to the [winner-preferrers] up in the hierarchy. But [...] it isn't clear which of the [winner-preferrers] should be promoted — perhaps all of them, or perhaps just one. Other data might require one of the [winner-preferrers] to be dominated by one of the [loser-preferrers]. [The current ERC] gives no basis for choosing."

5.2. **Gradualness and the credit problem.** Boersma (1997, 1998) suggests that *gradualness* might get around the credit problem. In fact, consider a re-ranking rule that makes only *small*, gradual adjustments at each iteration. Then, even if at a given iteration the algorithm incorrectly promotes a winner-preferrer that should in the end sit at the bottom of the ranking, nonetheless this will only be a small mistake. And hopefully small mistakes will in the end be overridden by subsequent better moves. Boersma and Hayes (2001, p. 52) explicitly state this conjecture as in (51).

(51) "[An ERC inconsistent with the current ranking] constitutes evidence for two things. First, it is likely that [the loser-preferring constraints] [...] are ranked too high. Second, it is likely that [the winner-preferring] constraints [...] are ranked too low. Neither of these conclusions can be taken as a certainty. However, this uncertainty is not crucial, since the ultimate shape of the grammar will be determined by the ranking values that the constraints will take on in the long term, with exposure to a full range of representative forms. The hypothesis [...] is that moderate adjustments of ranking values will ultimately achieve the right grammar."

More explicitly, Boersma (1997, 1998) and Boersma and Hayes (2001) consider the re-ranking rule (52): loser-preferrers (winner-preferrers) are demoted (promoted) by a small amount, say 1. This is the only example in the current literature of a re-ranking rule that performs both constraint demotion and promotion. The EDRA (21) with this re-ranking rule is Boersma's (1997) (non-stochastic) *Gradual Learning Algorithm* (henceforth: GLA).

(52) a. Decrease by 1 the ranking value of each loser-preferrer;
b. increase by 1 the ranking value of each winner-preferrer.

For instance, if the current ERC is (53a) and the current ranking vector is (53b), then the updated ranking vector is (53c). The ranking values of the two loser-preferrers C_4 and C_6 are decreased from 15 and 5 to 14 and 4, respectively. And the ranking values of the two winner-preferrers C_1 and C_2 are increased from 10 and 5 to 11 and 6, respectively.

(53) a. $\mathbf{a} = \begin{bmatrix} \overset{C_1}{\text{W}} & \overset{C_2}{\text{W}} & \overset{C_3}{e} & \overset{C_4}{\text{L}} & \overset{C_5}{e} & \overset{C_6}{\text{L}} \end{bmatrix}$ b. $\boldsymbol{\theta} = \begin{bmatrix} \overset{C_1}{10} & \overset{C_2}{5} & \overset{C_3}{20} & \overset{C_4}{15} & \overset{C_5}{100} & \overset{C_6}{5} \end{bmatrix}$

c. $\boldsymbol{\theta}_{\text{updated}} = \begin{bmatrix} \overset{C_1}{\textcircled{11}} & \overset{C_2}{\textcircled{6}} & \overset{C_3}{20} & \overset{C_4}{\textcircled{14}} & \overset{C_5}{100} & \overset{C_6}{\textcircled{4}} \end{bmatrix}$

The GLA converges in the case of input ERC matrices that have a unique winner-preferrer per ERC, and thus do not raise any credit problem.[10] But what about the case of input

[10] As far as I know, no proof of the convergence of the GLA for input ERC matrices with a unique W per row is currently available in the literature. Let me thus point out that convergence of the GLA in this special case follows from the theory developed in Section 7; see footnote 17 for details.

ERC matrices with multiple winner-preferrers per ERC? Does Boersma's conjecture (51), that gradualness gets around the credit problem, prove correct? As stressed in Keller and Asudeh (2002), this question has regrettably remained open for various years. Until Pater (2008) has recently shown that, indeed, gradual promotion/demotion re-ranking rules also get fooled by the credit problem, despite the fact that they make only small adjustments to the current ranking values. He considers the input ERC matrix (54), that exacerbates the credit problem by stacking multiple ERCs with two winner-preferrers. He reports that the GLA trained on this input ERC matrix (with the ERCs sampled uniformly and with identical initial ranking values for the five constraints) keeps increasing its current ranking values, without ever converging to an consistent vector.[11] Pater's counterexample thus settles the open issue of the convergence of the GLA.[12]

(54)
$$\begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & C_3 & C_4 & C_5 \end{array} \\ \left[\begin{array}{ccccc} \mathrm{W} & \mathrm{L} & \mathrm{W} & & \\ & \mathrm{W} & \mathrm{L} & \mathrm{W} & \\ & & \mathrm{W} & \mathrm{L} & \mathrm{W} \\ & & & \mathrm{W} & \mathrm{L} \end{array}\right] \end{array}$$

Pater does not explain why exactly the specific instance of the credit problem raised by his counterexample (54) fools the GLA. Furthermore, he leaves it open whether the GLA might converge provided the strong regularity of the ERC matrix (54) is broken by sampling its rows according to a skewed distribution, rather than the uniform distribution considered in his simulations. In subsection 5.3, I offer a detailed explanation of the failure of the GLA on Pater's counterexample: the algorithm fails because all ranking vectors consistent with the input ERC matrix lie outside of its search space! This explanation shows in particular that the GLA fails no matter how the ERCs in (54) are sampled and fed to the algorithm, thus slightly strengthening Pater's counterexample.

---

[11] A small caveat is in order here. Pater (2008) tests the GLA on his counterexample (54) using the Praat implementation of the GLA (with standard settings: uniform sampling of the input ERCs; constraints initially equally ranked with $\theta_1^{\text{init}} = \ldots = \theta_5^{\text{init}} = 100$; default values for the parameters, namely *noise variance* $\sigma = 2.0$ and *plasticity* $\eta = 0.1$); see Boersma (1999). The version of the GLA implemented in Praat differs from the one I am describing here in two respects. The *first* difference is that the GLA implemented in Praat perturbs the current ranking vector at each iteration with a small additive noise. Boersma suggests that this additive noise might be useful in order to model the acquisition of languages that display variation. Yet, I am ignoring the issue of variation in this paper. And I have thus gotten rid of the additive noise. The *second* difference is that the GLA implemented in Praat uses a slightly different notion of OT-consistency in order to decide at step (21b) whether the EDRA should update or not its current ranking vector. More precisely, it uses the notion of OT-consistency from Tesar and Smolensky (2000), that allows for multiple constraints to be assigned to the same stratum with the corresponding tie resolved additively. As noted in footnote 2, this notion of OT-consistency is different from the more standard notion of OT-consistency (20) adopted in this paper. Despite these two differences between the version of the GLA implemented in Praat and the version considered here, the Praat implementation can still be used to test the behavior of the variant of the GLA considered here, provided that the noise variance is set to zero (i.e. $\sigma = 0$), plasticity is an integer (e.g. $\eta = 1$), and the initial ranking values $\theta_1^{\text{init}}, \ldots, \theta_n^{\text{init}}$ are distinct and fractional. Since the initial ranking values are all distinct and fractional and since plasticity is an integer, then the current ranking values will stay all distinct throughout learning; thus the notion of OT-consistency built into the Praat implementation of the GLA coincides with the notion of OT-consistency (20) used in this paper. This way, Praat can be used to easily check that also the variant of the GLA considered in this paper fails on Pater's ERC matrix (54). This shows in particular that the failure of the GLA on Pater's counterexample has nothing to do with the internal randomization of the version of the algorithm considered by Boersma and tested by Pater.

[12] As discussed in Subsection 3.6, re-ranking rules that demote *all* loser-preferrers are worse than re-ranking rules that only demote the currently *undominated* loser-preferrers. Thus, it is natural to consider the variant of Boersma's re-ranking rule (52) that only demotes the *undominated* loser-preferrers rather than all of the loser-preferrers. The EDRA (21) with such a re-ranking rule is Boersma's (1997) (non-stochastic) *minimal Gradual Learning Algorithm* (GLA_{min}). Since Pater's counterexample (54) has a unique L per ERC, Boersma's original re-ranking rule and the variant just considered yield exactly the same ranking dynamics. Hence, Pater's ERC matrix also provides a counterexample against this variant of Boersma's re-ranking rule.

5.3. **A detailed explanation of Pater's counterexample.** Consider the beginning (55) of a run of the GLA on Pater's counterexample (54) starting from null initial ranking values. Suppose that at the first iteration, the GLA is fed ERC 1. Since the null ranking vector is not consistent with this ERC, update is performed. By (52), the ranking values of the winner-preferrers C_1 and C_3 are increased by 1 and the ranking value of the loser-preferrer C_2 is decreased by 1. Equivalently, the current ranking vector is updated by component-wise sum with the vector that has 1 in correspondence of the two winner-preferrers C_1 and C_3, has -1 in correspondence of the loser-preferrer C_2 and has 0's elsewhere. Suppose that at the second iteration, the GLA is fed ERC 2. Again the current ranking vector is not consistent with this ERC and update is thus performed. By (52), the ranking values of the winner-preferrers C_2 and C_4 are increased by 1 and the ranking value of the loser-preferrer C_3 is decreased by 1. Equivalently, the current ranking vector is updated by component-wise sum with the vector that has 1 in correspondence of the two winner-preferrers C_2 and C_4, has -1 in correspondence of the loser-preferrer C_3 and has 0's elsewhere. And so on.

(55)
$$\begin{array}{c} \\ \text{ERC 1} \\ \text{ERC 2} \\ \text{ERC 3} \\ \text{ERC 4} \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & C_3 & C_4 & C_5 \end{array} \\ \left[\begin{array}{ccccc} \text{W} & \text{L} & \text{W} & & \\ & \text{W} & \text{L} & \text{W} & \\ & & \text{W} & \text{L} & \text{W} \\ & & & \text{W} & \text{L} \end{array}\right] \end{array} \Longrightarrow \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \begin{array}{l} \xrightarrow{\text{ERC 1}} \begin{bmatrix} +1 \\ -1 \\ +1 \\ 0 \\ 0 \end{bmatrix} \\ \xrightarrow{\text{ERC 2}} \begin{bmatrix} +1 \\ -1 \\ +1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ +1 \\ -1 \\ +1 \\ 0 \end{bmatrix} \\ \xrightarrow{\text{ERC 3}} \begin{bmatrix} +1 \\ -1 \\ +1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ +1 \\ -1 \\ +1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ +1 \\ -1 \\ +1 \end{bmatrix} \\ \xrightarrow{\text{ERC 4}} \begin{bmatrix} +1 \\ -1 \\ +1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ +1 \\ -1 \\ +1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ +1 \\ -1 \\ +1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ +1 \\ -1 \end{bmatrix} \\ \longrightarrow \ \ldots \end{array}$$

The paper-and-pencil simulation in (55) shows that the ranking vector $\boldsymbol{\theta}$ entertained by the GLA at a generic iteration has the shape in (56): it is obtained by adding together the four column vectors that appear in (56), each multiplied by a nonnegative constant α_i. The ith column vector in (56) is the *update vector* corresponding to the ith ERC of Pater's counterexample (54). It is obtained by replacing each W, L, and e in the ERC with 1, -1 and 0, respectively. And it thus encodes the contribution of the ith ERC to the current ranking vector according to the GLA re-ranking rule (52). The corresponding coefficient α_i in (56) represents the number of updates triggered by the ith ERC of Pater's tableau in the run considered up to the time considered.

(56)
$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \end{bmatrix} = \alpha_1 \begin{bmatrix} 1 \\ -1 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \alpha_2 \begin{bmatrix} 0 \\ 1 \\ -1 \\ 1 \\ 0 \end{bmatrix} + \alpha_3 \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \\ 1 \end{bmatrix} + \alpha_4 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}$$

α_1: # of updates triggered by the 1st ERC of the matrix (54); α_4: # of updates triggered by the 4th ERC of the matrix (54); $\cdots$

$\begin{bmatrix} 1 \\ -1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$: update vector that corresponds to the 1st ERC of the matrix (54) according to rule (52); $\cdots$ $\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}$: update vector that corresponds to the 4th ERC of the matrix (54) according to rule (52)

Adding up the corresponding components in (56), we conclude that the search space of the GLA run on Pater's counterexample (54) starting from the null initial vector is a subset of the set of ranking vectors of the form (57), for nonnegative coefficients $\alpha_1, \alpha_2, \alpha_3, \alpha_4$. This conclusion makes good sense. Constraint C_1 starts out with a null initial ranking value and is promoted by 1 every time ERC 1 triggers an update. Thus, the current ranking value θ_1 of constraint C_1 must always be equal to the number α_1 of updates triggered by ERC 1, as stated in the first equation in (57). Constraint C_2 starts out with a null initial ranking value, is promoted by 1 every time ERC 2 triggers an update and is demoted by 1 every time ERC 1 triggers an update. Thus, the current ranking value θ_2 of constraint C_2 must always be equal to the number α_2 of updates triggered by ERC 2 minus the number α_1 of updates triggered by ERC 1, as stated in the second equation in (57). And so on.

(57)
$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_2 - \alpha_1 \\ \alpha_1 + \alpha_3 - \alpha_2 \\ \alpha_2 + \alpha_4 - \alpha_3 \\ \alpha_3 - \alpha_4 \end{bmatrix}, \quad \alpha_1,\, \alpha_2,\, \alpha_3,\, \alpha_4 \geq 0$$

There is only one ranking consistent with Pater's ERC matrix (54), namely $C_1 \gg C_2 \gg C_3 \gg C_4 \gg C_5$. Thus, a ranking vector $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)$ is consistent with Pater's ERC matrix only if it univocally represents this ranking, namely it satisfies the four strict inequalities $\theta_1 > \theta_2 > \theta_3 > \theta_4 > \theta_5$. By virtue of the characterization (57) of the ranking vectors entertained by the GLA, these four strict inequalities can be rewritten as the four strict inequalities (58), in terms of the coefficients $\alpha_1, \alpha_2, \alpha_3, \alpha_4$.

(58)
$$\begin{array}{lcl} \theta_1 > \theta_2 & \Rightarrow & \alpha_1 > \alpha_2 - \alpha_1 \\ \theta_2 > \theta_3 & \Rightarrow & \alpha_2 - \alpha_1 > \alpha_1 + \alpha_3 - \alpha_2 \\ \theta_3 > \theta_4 & \Rightarrow & \alpha_1 + \alpha_3 - \alpha_2 > \alpha_2 + \alpha_4 - \alpha_3 \\ \theta_4 > \theta_5 & \Rightarrow & \alpha_2 + \alpha_4 - \alpha_3 > \alpha_3 - \alpha_4. \end{array}$$

Crucially, the four strict inequalities in (58) are not feasible, namely there exist no coefficients $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ that satisfy all four of them at the same time.[13] In conclusion, the reason why the GLA fails on Pater's counterexample (54) is as follows: the search space

[13] To see that the inequalities (58) admit no solution, move everything on one side, as in (i).

(i)
$$\begin{array}{rrrrcl} +2\alpha_1 & -\alpha_2 & & & > & 0 \\ -2\alpha_1 & +2\alpha_2 & -\alpha_3 & & > & 0 \\ +\alpha_1 & -2\alpha_2 & +2\alpha_3 & -\alpha_4 & > & 0 \\ & +\alpha_2 & -2\alpha_3 & +2\alpha_4 & > & 0 \end{array}$$

If we sum all four inequalities (i) together, we obtain the inequality $\alpha_1 - \alpha_3 + \alpha_4 > 0$; if we sum together only the second and the third inequalities in (i), we obtain $-\alpha_1 + \alpha_3 - \alpha_4 > 0$. As the two inequalities thus derived are inconsistent, the four inequalities (i) admit no solution.

of the GLA is limited to ranking vectors of the form (57); but no such ranking vector is consistent with Pater's ERC matrix, namely satisfies the inequalities (58). In other words, the GLA fails because it struggles to reach a ranking vector that lies beyond its reach.[14] Pater detected the failure of the GLA on the counterexample (54) through simulations where the various ERCs were sampled uniformly. The explanation just provided slightly strengthens Pater's counterexample, showing that the GLA fails no matter how the input ERCs are sampled and fed to the algorithm.

5.4. **Summary.** On the one hand, Section 4 has argued in favor of constraint promotion from a *modeling perspective*: throughout the early stage of the acquisition of phonotactics, the learner can confidently only posit underlying forms faithful to the winners; the faithfulness constraints would therefore never be re-ranked by a demotion-only EDRA, as they are never loser-preferrers; and the predicted learning dynamics would thus be too simple to match the attested typological and acquisition complexity. On the other hand, this Section has shown that constraint promotion is hard to devise from a *computational perspective* because of the credit problem: only a subset of the current winner-preferrers should be credited for taking care of the current loser-preferrers; it is not possible to promote only these useful winner-preferrers, because we cannot select them based only on the current ERC; and it is not safe to promote all of them, because this strategy leads to repeated promotions of winner-preferrers that should be low ranked, as in Pater's counterexample (54). The rest of the paper presents a solution to this impasse, thus moving a step forward towards the integration of the modeling and the computational perspectives.

6. Calibration of the promotion amount ensures efficient convergence

Subsection 3.6 has shown that we get faster convergence by only demoting the loser-preferrers that need to be demoted, namely the currently undominated ones, rather than all of the loser-preferrers. Let the *demotion amount* by which the undominated loser-preferrers are demoted be a small, fixed amount, say 1 for concreteness. The demotion component of the re-ranking rule thus looks like (59a). Let the *promotion amount* be the small amount by which the winner-preferrers are promoted, that I will denote by p. Thus, the promotion component of the re-ranking rule looks like (59b).

(59) a. Decrease the ranking value of each undominated loser-preferrer by 1;
b. increase the ranking value of each winner-preferrer by p.

The demotion-only re-ranking rule (25) corresponds to the scheme (59) with the choice of a null $p = 0$ promotion amount. Lack of constraint promotion allowed T&S to prove convergence after a worst-case number of errors that grows only quadratically in the number of constraints, as reviewed in Section 3. Although a virtue from a computational perspective, lack of constraint promotion turns into a liability from a modeling perspective, as argued in Section 4. The only promotion-demotion re-ranking rule available in the literature is Boersma's (1997) GLA re-ranking rule (52). This corresponds to the scheme (59) with the choice of the promotion amount $p = 1$.[15] The addition of this promotion component disrupts the good convergent behavior of demotion-only, as shown by Pater's (2008) counterexample reviewed in Section 5.

[14]The discussion so far explains why the GLA fails to converge on Pater's counterexample (54), but it does not explain why in particular the current ranking values entertained by the GLA keep increasing. An explanation of this fact follows from two general properties of promotion/demotion re-ranking rules, that will be discussed in Sections 6 and 7. First, that the current ranking values cannot decrease below a certain threshold; see Fact **??**. Second, that the algorithm cannot entertain the same ranking vector twice within the same run; see Fact 5. The only way that learning can go on for ever, is thus that the ranking values keep increasing.

[15]I am ignoring here the fact that the GLA demotes *all* loser-preferrers, while the re-ranking rule (59) only demotes those loser-preferrers that need to be demoted, namely the currently *undominated* ones; see footnote 12.

The first three ERCs of Pater's ERC matrix (54) have two winner-preferrers and one loser-preferrer. The GLA demotes the loser-preferrer by 1 and promotes each of the two winner-preferrers by 1. Overall, the GLA thus demotes by 1 (as it demotes once) but promotes by 2 (as it promotes twice). In other words, the GLA performs overall more promotion than demotion. As the promotion component of the update overwhelms the demotion component, the good convergent behavior of demotion-only is disrupted. But what if the promotion-component (59b) of the re-ranking rule is properly calibrated, so that it never overwhelms the demotion component (59a)? Is the good convergence behavior of demotion-only retained in this case? And how should the promotion amount p be chosen?

The rest of this paper provides a complete answer to these questions. Let's say that a re-ranking rule of the form (59) is *calibrated* provided that the promotion amount p is chosen in such a way that the promotion component of the re-ranking rule never overwhelms the demotion component. Subsection 6.1 shows that calibration holds provided the promotion amount p is strictly smaller than the crucial threshold ℓ/w, namely the ratio between the number ℓ of currently undominated loser-preferrers and the number w of winner-preferrers. Subsections 6.2-6.4 then show that a slight extension of T&S's analysis shows that calibration is a *sufficient condition* for efficient convergence. For instance, set the promotion amount equal to $p = \ell/(w+1)$, which is calibrated, as it is slightly smaller than the calibration threshold ℓ/w. The corresponding EDRA converges after a worst-case number of errors that grows only cubically with the number of constraints. This bound compares well with the quadratic error-bound obtained by T&S for demotion-only. Section 7 will then show that calibration is also a *necessary condition* for efficient convergence. In fact, if the promotion amount is increased to coincide with the calibration threshold ℓ/w, then convergence still holds but efficiency fails, as the worst-case number of errors grows exponentially with the number of constraints.

6.1. **Calibration of the promotion amount.** We want the promotion component (59b) of the re-ranking rule not to overwhelm the demotion-component (59a), so that it will hopefully not disrupt too much the good convergent behavior of demotion-only. This requires a proper calibration of the promotion amount p. Let's work through various concrete cases. To get started, consider the special case where the current input ERC fed to the EDRA has a unique L and a unique W, as in (60).

(60) [... W ... L ...]

a. Decrease the ranking value of the loser-preferrer by 1;

b. increase the ranking value of the winner-preferrer by $p < 1$.

According to the scheme (59a), the unique loser-preferrer is demoted by 1, as stated in (60a). And the unique winner-preferrer is promoted by the promotion amount p. In order not to disrupt the convergence properties of demotion-only, we should promote less than we demote. Thus, we choose the promotion amount p smaller than 1, as in (60b).

Consider next the case where the current ERC fed to the EDRA again has a unique L, but now has multiple W's. For concreteness, suppose it has two W's, as in (61).

(61) [... W W ... L ...]

a. Decrease the ranking value of the loser-preferrer by 1;

b. increase the ranking value of both winner-preferrers by $p < \frac{1}{2}$.

Again, the loser-preferrer is demoted by 1 according to the scheme (59a). And both winner-preferrers are promoted by the promotion amount p. Thus, we overall promote by $2p$. In order not to disrupt the convergent behavior of demotion-only, we should promote overall less than we demote. This requires $2p$ to be smaller than 1. Thus, we choose the promotion amount p smaller than $1/2$, as in (61b).

The extension to the case where the current ERC fed to the EDRA has an arbitrary number w of winner-preferrers and again a unique loser-preferrer is straightforward.

(62) $\Big[\ldots \overbrace{\text{W} \;\text{—}\; \text{W}}^{w \text{ winner}-\text{preferrers}} \overset{\ldots}{\ldots} \text{L} \ldots \Big]$

a. Decrease the ranking value of the loser-preferrer by 1;

b. increase the ranking value of each of the w winner-preferrers by $p < \frac{1}{w}$.

Once more, the unique loser-preferrer is demoted by 1, according to (59a). And each one of the w winner-preferrers is promoted by the promotion amount p. Thus, we overall promote by wp. In order not to disrupt the convergent behavior of demotion-only, we should again promote overall less than we demote. This requires wp to be smaller than 1. Thus, we choose the promotion amount p smaller than $1/w$, as in (62b).

So far, I have only considered the case where the current ERC has a unique loser-preferrer. If such an ERC is not consistent with the current ranking vector, then its unique loser-preferrer must be currently undominated, namely it must be ranked above the currently top ranked winner-preferrer. If the current ERC has multiple loser-preferrers, then some of them might be currently undominated and some others might not be. Let ℓ be the total number of currently undominated loser-preferrers. If only one loser-preferrer is currently undominated, then we can of course use again the very same re-ranking rule (62). What if the number ℓ of currently undominated loser-preferrers is larger than one?

(63) $\Big[\ldots \overbrace{\text{W} \;\text{—}\; \text{W}}^{w \text{ winner}-\text{preferrers}} \overset{\ldots}{\ldots} \overbrace{\text{L} \;\text{—}\; \text{L}}^{\ell \text{ undom. loser}-\text{preferrers}} \ldots \Big]$

a. Decrease the ranking value of each of the ℓ undominated loser-preferrers by 1;

b. increase the ranking value of each of the w winner-preferrers by $p < \frac{\ell}{w}$.

Each one of the ℓ undominated loser-preferrers is demoted by 1, according to the scheme (59a). Thus, we overall demote by ℓ. Each one of the w winner-preferrers is promoted by the promotion amount p. Thus, we overall promote by wp. In order not to disrupt the convergent behavior of demotion-only, we should promote overall less than we demote. This requires wp to be smaller than ℓ. Thus, we choose the promotion amount p smaller than ℓ/w, as stated in (63b).

In conclusion, the crucial threshold for the calibration of the promotion amount is the ratio ℓ/w between the number ℓ of undominated loser-preferrers and the number w of winner-preferrers. Informally, this threshold can be justified in terms of the two following intuitions. As the number ℓ of undominated loser-preferrers increases, we demote more constraints, which buys us a larger promotion amount, without promotion overwhelming demotion. As the number w of winner-preferrers increases, we promote more constraints, and thus need to adopt a smaller promotion amount, in order for promotion not to overwhelm demotion. This ratio ℓ/w is called the *calibration threshold*. A re-ranking rule of the form (63), whose promotion amount is strictly smaller than the calibration threshold, is called *calibrated*.

For concreteness, let me consider a specific calibrated choice for the promotion amount. For instance, let's set the promotion amount p equal to $\frac{\ell}{w+1}$. This re-ranking rule (64) is calibrated, as the promotion amount is indeed slightly smaller than the calibration threshold $\frac{\ell}{w}$, so that the promotion component of the re-ranking rule never overwhelms the demotion component.

(64) a. Decrease the ranking value of each of the ℓ undominated loser-preferrers by 1;

b. increase the ranking value of each of the w winner-preferrers by $p = \frac{\ell}{w+1}$.

For instance, if the current ERC is (65a) and the current ranking vector is (65b), then the updated ranking vector is (65c). Of the two loser-preferrers C_4 and C_6, only the former

is currently undominated. Thus, its ranking value gets decreased by 1. And the ranking value of the two winner-preferrers C_1 and C_2 gets increased by 1/3, namely the number of undominated loser-preferrers (which is $\ell = 1$) divided by the number of winner-preferrers (which is $w = 2$) increased by 1.

(65) a. $\mathbf{a} = \begin{bmatrix} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 \\ \text{W} & \text{W} & e & \text{L} & e & \text{L} \end{bmatrix}$ b. $\boldsymbol{\theta} = \begin{bmatrix} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 \\ 10 & 5 & 20 & 15 & 100 & 5 \end{bmatrix}$

c. $\boldsymbol{\theta}_{\text{updated}} = \begin{bmatrix} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 \\ \textcircled{10.3} & \textcircled{5.3} & 20 & \textcircled{14} & 100 & 5 \end{bmatrix}$

The ranking dynamics in the case of the re-ranking rule (64) is rather complicated, as ranking values can oscillate over time up and down, contrary to the monotonically decreasing dynamics predicted by demotion-only re-ranking rules. Despite the complexity of the ranking dynamics, the current ranking values display invariant properties, spelled out in Subsections 6.2 and 6.3. These invariants lead to efficient convergence, as noted in Subsection 6.4. The reasoning turns out to be a small extension of T&S's analysis of demotion-only re-ranking rules.

6.2. **T&S's invariant extends beyond demotion-only.** As recalled in Section 3, T&S's analysis of demotion-only re-ranking rules rests on the core observation that the current ranking values cannot drop too low over time, as stated in Fact 2. A careful look at the proof of this invariant reveals that it does not actually hinge in any way on the demotion-only assumption. Rather it follows from the fact that not all loser-preferrers are demoted, but only those that really need to be demoted, namely only the currently *undominated* loser-preferrers. In other words, if we only demote the constraints that need to be demoted, we cannot demote too much. T&S's Fact 2 thus extends to an arbitrary promotion/demotion re-ranking rule, as long as it only demotes the loser-preferrers that are undominated, as made explicit in the following generalized formulation of Fact 2

FACT 2 (GENERALIZED). *Assume that the input ERC matrix is consistent with a ranking* $\gg$*. Without loss of generality, assume that this ranking is* $C_1 \gg C_2 \gg \cdots \gg C_n$*. Let* $\theta_1, \ldots, \theta_n$ *be the current ranking values entertained by an EDRA in a generic run on those input ERCs, up to a generic time, starting from null initial ranking values. Assume that the re-ranking rule use by the EDRA has the shape (59), namely it demotes by 1 only the currently* undominated *loser-preferrers. The current ranking values thus satisfy condition (66) for every* $k = 1, \ldots, n$*.*

(66) $\theta_k \geq -(k-1)$

Namely, the ranking value θ_k *of the constraint* C_k *assigned to the* k*th stratum (with the 1st stratum being the top one) never goes below* $-(k-1)$*.* ■

6.3. **An invariant for calibrated re-ranking rules.** The sum of the current ranking values in (65b) is $10+5+20+15+100+5 = 155$. The updated ranking values in (65c) add up to $10.3+5.3+20+14+100+5 = 154.6$. The fact that the sum of the updated ranking values is smaller than the sum of the current ranking values is not a coincidence but rather a general property of calibrated re-ranking rules. Let me illustrate why that is the case, focusing on the calibrated re-ranking rule (64). Suppose that the ERC triggering the current update has ℓ undominated loser-preferrers and w winner-preferrers. The demotion component (64a) of the re-ranking rule subtracts the demotion amount 1 for ℓ times, so that the sum of the current ranking values gets overall decreased by $\ell = 1 \times \ell$. And the promotion component (64a) adds the promotion amount $\frac{\ell}{w+1}$ for w times, so that the sum of the current ranking values gets overall increased by $\frac{\ell}{w+1} \times w$. As the amount that gets overall added to the sum of the current ranking values (i.e., $\frac{w\ell}{w+1}$) is always smaller than the amount that gets subtracted (i.e., ℓ), the sum of the current ranking values decreases at each update. And the amount by which it decreases can be computed as in (67).

$$\underbrace{\frac{w\ell}{w+1}}_{\substack{\text{amount added to the sum of the current ranking values}}} \underbrace{-\ell}_{\substack{\text{amount subtracted from the sum of the current ranking values}}} = \underbrace{-\frac{\ell}{w+1}}_{\substack{\text{amount by which the sum of the ranking values decreases at each update}}} \tag{67}$$

To trigger an update, the current ERC must have at least one undominated loser-preferrer, i.e., ℓ must be at least 1. As the number of constraints is n and at least one of them is loser-preferring, there can be at most $n-1$ winner-preferrers, i.e., w can be at most $n-1$. Replacing ℓ with 1 and w with $n-1$ in the right hand side of (67), I conclude that at each update the sum of the current ranking values decreases by at least $1/n$.

Fact 3. *Each update according to the calibrated re-ranking rule (64) decreases the sum of the current ranking values by at least $1/n$, where n is the number of constraints.* ■

6.4. **Calibration ensures efficient convergence.** The proof of efficient convergence now follows straightforwardly. Consider a run of the EDRA on a consistent ERC matrix using the new calibrated re-ranking rule (64). Suppose for concreteness that the initial ranking values are all null. As the input ERC matrix is consistent by hypothesis, the generalized Fact 2 ensures that the current ranking values $\theta_1, \dots, \theta_n$ cannot become too small, as in (66). Because of the identity $0+1+2+\ldots+(n-1) = \frac{1}{2}n(n-1)$, inequalities (66) entail that the sum of the current ranking values never gets smaller than the constant $-\frac{1}{2}n(n-1)$ (that only depends on the number n of constraints), as stated in (68).

$$\sum_{k=1}^{n} \theta_k \geq -\frac{1}{2}n(n-1) \tag{68}$$

By Fact 3, the sum of the current ranking values decreases by at least $1/n$ with every update. After t updates, it thus has decreased by at least t/n. As we start from initial null ranking values, the sum of the current ranking values after t updates is thus smaller than or at most equal to $-t/n$, as stated in (69).

$$\sum_{k=1}^{n} \theta_k \leq -\frac{t}{n} \tag{69}$$

By combining together the two inequalities (68) and (69), we get the upper bound (70) on the number t of updates. In other words, as the sum of the current ranking values decreases by at least $1/n$ with every update but cannot get smaller than $-\frac{1}{2}n(n-1)$, then the algorithm cannot perform too many updates, namely no more than $\frac{1}{2}n^2(n-1)$.

$$\frac{t}{n} \leq \frac{1}{2}n(n-1) \tag{70}$$

We have thus proved Theorem 2, that guarantees efficient convergence for the calibrated promotion/demotion re-ranking rule (64). The reasoning just outlined easily extends to an arbitrary calibrated re-ranking rule, with the quality of the error-bound depending on how much the promotion amount is smaller than the calibration threshold ℓ/w; see Appendix A.2 for details. The extension from null initial ranking values to arbitrary ones is straightforward, as discussed in Appendix A.1.

Theorem 2. *The EDRA (21) with the calibrated promotion/demotion re-ranking rule (64) run on a consistent input ERC matrix corresponding to n constraints starting from null initial ranking values can perform at most $\frac{1}{2}n^2(n-1)$ mistakes before converging to a ranking vector consistent with all input ERC.* ■

This Section started from the intuition that the good convergent behavior of demotion-only EDRAs should be retained or only slightly affected provided that the promotion component of the re-ranking rule does not overwhelm the demotion-component. This requires a proper calibration of the promotion amount, as in the new calibrated re-ranking rule (64). Theorem 2 confirms that this initial intuition is correct. In fact, T&S's Theorem 1 ensures an error-bound for demotion-only that grows only quadratically with the number of constraints. And Theorem 2 provides an error-bound for the new calibrated promotion/demotion re-ranking rule (64) that is only slightly worse, namely grows cubically rather than quadratically in the number of constraints.

6.5. **Summary.** As recalled in Section 5, constraint promotion is delicate, because of the credit problem. Tesar and Smolensky (1998) are drastic: they suggest that we do not promote at all. It turns out that we can be a bit less drastic: we can promote, as long as the promotion amount is properly calibrated, so that the promotion component of the re-ranking rule does not overwhelm the demotion-component. Furthermore, the proof of convergence for the demotion-only case and the calibrated promotion/demotion case are analogous. T&S's original proof of convergence for demotion-only reviewed in Section 3 was based on two crucial properties of the sum of the current ranking values. The first property is that this sum can never become smaller than a certain constant that depends on the number of constraints, as stated in (71a). The second property is that, as long as we perform demotion only, the current ranking values can only decrease over time, and thus in turn their sum can only decrease over time, as stated in (71b). Convergence immediately follows: as the sum of the current ranking values keeps decreasing by (71b) but cannot decrease too much by (71a), then learning must stop at a certain point.

(71) The sum of the current ranking values:

a. can never become smaller than a certain constant;

b. starts null and decreases a little bit with every update.

Property (71a) has nothing to do with the promotion issue. It rather follows from the fact that constraints cannot drop too much as long as we only demote the constraints that really need to be demoted, namely the currently undominated ones. This property thus extends to any re-ranking rule that only demotes undominated loser-preferrers (generalized Fact 2). The case of property (71b) is more tricky. For instance, it does not hold for Boersma's GLA re-ranking rule (52): if there are two winner-preferrers that are each promoted by 1 and one loser-preferrer that is demoted by 1, then the sum of the ranking values increases by 1, instead of decreasing. Yet, property (71b) is not altogether incompatible with constraint promotion. In fact, calibration of the promotion amount was designed in Subsection 6.1 in order to ensure that the sum of the current ranking values decreases with each update, despite promotion (Fact 3). Convergence for calibrated promotion/demotion re-ranking rules thus follows again from the two properties (71).

7. Calibration is a necessary condition for efficient convergence

Section 6 has shown that efficient convergence holds for re-ranking rules that perform both constraint demotion and promotion, as long as the promotion component of the update never overwhelms the demotion component. This requires the promotion amount to be calibrated: it must always be *strictly smaller* than the calibration threshold ℓ/w, which is the ratio between the number ℓ of currently undominated loser-preferrers and the number w of winner-preferrers. To complete the computational theory of re-ranking rules that perform both constraint demotion and promotion, I now want to investigate the limiting case where the promotion amount is set *equal* to the calibration threshold ℓ/w, as described in Subsection 7.1. The proof of convergence for calibrated re-ranking rules developed in Section 6 does not extend to this limiting case, as explained in Subsection

7.2. A different line of analysis is thus needed, developed in Subsection 7.4. This analysis exploits a property of EDRAs that is interesting in its own right: they can never entertain again a ranking (vector) that has made a mistake at some earlier time, as explained in Subsection 7.3. In other words, they explore the typology in a smart way: although EDRAs do not keep track of previously seen data and thus of the errors previously made, they implicitly manage to avoid repeating the same error twice. This alternative proof of convergence does not provide a bound on the number of errors made before converging. Subsection 7.5 takes on the issue of the number of errors, showing how to construct cases where even the best-case number of errors grows exponentially in the number of constraints. In conclusion, efficiency breaks down at the calibration threshold ℓ/w. And proper calibration is thus a necessary condition for efficient convergence.

7.1. **Smallest non-calibrated promotion.** In this Section, I study the re-ranking rule (72). Its crucial property is that the promotion amount is set equal to the calibration threshold ℓ/w, namely the ratio between the number ℓ of undominated loser-preferrers and the number w of winner-preferrers. In other words, this is the update rule with the smallest possible promotion amount, among those that are not calibrated, and thus do not fall under the theory developed in Section 6.

(72) a. Decrease the ranking value of each of the ℓ undominated loser-preferrers by 1;
b. increase the ranking value of each of the w winner-preferrers by $\frac{\ell}{w}$.

For instance, if the current ERC is (73a) and the current ranking vector is (73b), then the updated ranking vector is (73c). Of the two loser-preferrers C_4 and C_6, only the former is currently undominated. Thus, its ranking value gets decreased by 1. And the ranking value of the two winner-preferrers C_1 and C_2 gets increased by 1/2, namely the number of undominated loser-preferrers (which is $\ell = 1$) divided by the number of winner-preferrers (which is $w = 2$).

(73) a. $\mathbf{a} = \begin{bmatrix} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 \\ \text{W} & \text{W} & e & \text{L} & e & \text{L} \end{bmatrix}$ b. $\boldsymbol{\theta} = \begin{bmatrix} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 \\ 10 & 5 & 20 & 15 & 100 & 5 \end{bmatrix}$

c. $\boldsymbol{\theta}_{\text{updated}} = \begin{bmatrix} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 \\ \textcircled{10.5} & \textcircled{5.5} & 20 & \textcircled{14} & 100 & 5 \end{bmatrix}$

7.2. **An invariant for smallest non-calibrated promotion.** The sum of the current ranking values in (73b) is $10+5+20+15+100+5 = 155$. The updated ranking values in (73c) add up to the same number, as $10.5 + 5.5 + 20 + 14 + 100 + 4 = 155$. The fact that the sum of the current ranking values remains constant is not a coincidence. In fact, the promotion component (72b) of the re-ranking rule adds the promotion amount ℓ/w for w times, so that the sum of the current ranking values gets overall increased by $\frac{\ell}{w} \times w = \ell$. Furthermore, the demotion component (72a) subtracts the demotion amount 1 for ℓ times, so that the sum of the current ranking values gets overall decreased by $1 \times \ell = \ell$. As the same quantity gets added to and subtracted from the sum of the current ranking values, the promotion and demotion components of the re-ranking rule balance each other and the sum of the current ranking values remains constant over time.

Fact 4. *The sum of the current ranking values entertained by the EDRA with the smallest non-calibrated promotion amount (72) never changes throughout learning, and is thus always equal to the sum of the initial ranking values.* ■

Unfortunately, Fact 4 entails that the strategy used in Section 6 to prove convergence for the case of calibrated promotion amounts does not extend to the case of the smallest non-calibrated promotion amount (72). T&S's lower bound on the sum of the current ranking values holds for any re-ranking rule (as long as only undomintaed loser-preferrers are demoted), as stated in the generalized Fact 2. If the sum of the current ranking values

decreases with each update (as in the case of demotion-only or calibrated promotion), then T&S's lower bound on the sum of the ranking values straightforwardly translates into a bound on the number of possible updates. But if the sum of the current ranking values does not decrease with each update but rather stays constant, then T&S's lower bound does not straightforwardly have anything to say about the number of updates. A more indirect reasoning is needed, illustrated in Subsection 7.4. This alternative analysis rests on another property of EDRAs discussed in Subsection 7.3, namely that they cannot loop.

7.3. **EDRAs cannot loop.** The sequence of ranking vectors entertained by a demotion-only EDRA cannot contain a subsequence such as (74), whereby the same ranking vector $\boldsymbol{\theta}$ is entertained at two different times but with some other ranking vector $\boldsymbol{\theta'} \neq \boldsymbol{\theta}$ entertained at some time in between. The reason is straightforward. When moving from the ranking vector $\boldsymbol{\theta} = (\theta_1, \ldots \theta_n)$ to a different ranking vector $\boldsymbol{\theta'} = (\theta'_1, \ldots, \theta'_n)$, at least one ranking value must have decreased from θ_k to some smaller value θ'_k. In order for the algorithm to go back from $\boldsymbol{\theta'}$ to $\boldsymbol{\theta}$, that ranking value would need to increase back from θ'_k to its original value θ_k. And that is impossible with a demotion-only re-ranking rule, as ranking values can only decrease but not increase over time.

(74) $\ldots \longrightarrow \boldsymbol{\theta} \longrightarrow \ldots \longrightarrow \boldsymbol{\theta'} \longrightarrow \ldots \longrightarrow \boldsymbol{\theta} \longrightarrow \ldots$

Thus, demotion-only EDRAs cannot loop: once a ranking vector is deemed unsuitable and thus updated, the algorithm can never consider it again in that same learning path. In other words, demotion-only EDRAs explore the search space in a smart way, as they avoid making the same mistake twice.

This special property does not carry over from demotion-only to promotion/demotion EDRAs. A trivial counterexample is provided in (75). Start from a null initial ranking vector. Update in response to the ERC 1, according to the promotion/demotion re-ranking rule (72). Then update again in response to the ERC 2. The algorithm has looped back to the initial null ranking vector.

(75) a.

	C_1	C_2
ERC 1	W	L
ERC 2	L	W

b.

$$\begin{matrix} C_1 \\ C_2 \end{matrix}\begin{bmatrix} 0 \\ 0 \end{bmatrix} \xrightarrow{\text{ERC 1}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \xrightarrow{\text{ERC 2}} \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\boldsymbol{\theta} \qquad \boldsymbol{\theta'} \qquad \boldsymbol{\theta}$$

The details of the re-ranking rule are irrelevant here. The point is that, as soon as we perform some constraint promotion too, ranking values can oscillate up and down and in particular can fall back to values entertained earlier on.

Yet, the input ERC matrix (75a) used to get the EDRA to loop is *in*consistent. What if we restrict ourselves to consistent ERC matrices? Fact 5 ensures that EDRAs cannot loop in this case, even if the re-ranking rule performs constraint promotion besides demotion. Again, the details of the re-ranking rule do not matter, and the statement holds for any re-ranking rule of the form (59).[16]

FACT 5. *If the input ERC matrix is consistent, the EDRA (21) with any promotion/demotion re-ranking rule of the form (59) can never loop back to a current ranking vector that it had previously made a mistake in that same run.* ■

In the demotion-only case, the impossibility of loops follows trivially from the monotonicity of the ranking dynamics, and is thus independent of the properties of the input ERC

[16]The proof of Fact 5 presented in Appendix A.3 only really uses the fact that the re-ranking rule (59) promotes *all* winner-preferrers. The proof would not work if only some of the winner-preferrers were promoted. On the other hand, the exact promotion amount does not matter, nor does it matter whether the winner-preferrers are all promoted by the same amount or each by a different amount.

matrix. In the promotion/demotion case instead, the impossibility of loops follows from a property of the input ERC matrix, namely its consistency. Understanding why promotion/demotion EDRAs cannot loop thus brings out an interesting and non-trivial property of the notion of OT-consistency. In particular, non-looping follows from a connection between OT-consistency and the geometric property of *conic independence*. The details of this reasoning are somewhat technical, and thus relegated to Appendix A.3.

7.4. **Convergence.** I am now ready to prove convergence for the re-ranking rule (72), with the smallest non-calibrated promotion amount. To start, recall the generalized Fact 2 from Subsection 6.2, that guarantees that T&S's lower bound on the current ranking values holds for any re-ranking rule that demotes only the loser-preferrers that need to be demoted, namely the undominated ones. Thus, the current ranking values in any run of the EDRA with the re-ranking rule (72) cannot become arbitrarily *small*, provided the input ERC matrix is consistent.

Can the current ranking values get arbitrarily *large*? As seen is Subsection 5.2, that is precisely what happens when Boersma's GLA (52) is run on Pater's counterexample (54): the ranking values increase indefinitely. But this cannot happen in the case of the re-ranking rule (72). In fact, the ranking values start out all null. As their sum must remain constant over time by Fact 4, the current ranking values must always sum up to zero. As they cannot become too small and must add up to zero, the current ranking values cannot become too large either.

As the current ranking values cannot become too large nor too small, the current ranking vector must live in a bounded region. Each time a ranking value is updated, it is increased by 1 or increased by at least $\frac{1}{n}$. This means in turn that the ranking values entertained by the EDRA live on a grid of points, separated by at least $1/n$ one from the other. As the search space of the algorithm consists of a bounded grid, it only contains a finite number of ranking vectors. Since the algorithm cannot loop by Fact 5, finiteness of the search space entails finite time convergence. We have thus proved the following convergence Theorem 3.[17] For a very different proof, see Magri (2012b).

THEOREM 3. *The EDRA (21) with the re-ranking rule (72), with the smallest non calibrated promotion amount, converges, namely it always makes a finite number of updates on any consistent input ERC matrix.* ■

7.5. **Number of errors.** T&S's convergence Theorem 1 for demotion-only EDRAs provides a (tight) bound on the worst-case number of errors. Theorem 2 extends T&S's analysis to calibrated promotion/demotion EDRAs, with a comparable bound on the worst-case number of errors. Unfortunately, that analysis does not extend further to the case of the non-calibrated re-ranking rule (72) considered here. An alternative analysis thus had to be developed in order to prove the convergence Theorem 3. Unfortunately, this alternative analysis does not provide a bound on the worst-case number of errors. This final Subsection takes on this issue, showing that the worst-case number of errors in the case of the non-calibrated re-ranking rule (72) grows exponentially in the number of constraints, so that efficiency is lost.

Following Riggle (2009), let the *diagonal matrix* for n constraints be the ERC matrix with n columns and $n-1$ rows, whose kth row has all entries equal to e but for the kth entry which is a W and the following entry which is an L. To illustrate, I give in (76) the diagonal ERC matrices for $n = 4, 5, 6$ constraints.

[17] For input ERC matrices with a unique winner-preferrer and a unique loser-preferrer per ERC, Boersma's re-ranking rule (52) coincides with the re-ranking rule (72) considered in this Section. Theorem 3 thus ensures that Boersma's re-ranking rule converges on these special input ERC matrices.

(76)
$$\begin{bmatrix} \mathrm{W} & \mathrm{L} & & \\ & \mathrm{W} & \mathrm{L} & \\ & & \mathrm{W} & \mathrm{L} \end{bmatrix}, \begin{bmatrix} \mathrm{W} & \mathrm{L} & & & \\ & \mathrm{W} & \mathrm{L} & & \\ & & \mathrm{W} & \mathrm{L} & \\ & & & \mathrm{W} & \mathrm{L} \end{bmatrix}, \begin{bmatrix} \mathrm{W} & \mathrm{L} & & & & \\ & \mathrm{W} & \mathrm{L} & & & \\ & & \mathrm{W} & \mathrm{L} & & \\ & & & \mathrm{W} & \mathrm{L} & \\ & & & & \mathrm{W} & \mathrm{L} \end{bmatrix}$$

For diagonal matrices, a different line of analysis developed in Magri (2012b) guarantees that the worst-case number of errors made by the EDRA with the non-calibrated re-ranking rule (72) is feasible (smaller than $n(n^2-1)/6$ where n is the number of constraints).

Pater (2008) considers ERC matrices obtained from the diagonal one by adding a W to the right of every L. More precisely, let *Pater's matrix* for n constraints be the ERC matrix with n columns and $n-1$ rows obtained from the diagonal matrix by "adding" a W at the right of every L (but in the last row). The case with $n = 5$ was already considered in (54). To illustrate, I give in (77) Pater's matrices for $n = 4, 5, 6$ constraints.

(77)
$$\begin{bmatrix} \mathrm{W} & \mathrm{L} & \mathrm{W} & \\ & \mathrm{W} & \mathrm{L} & \mathrm{W} \\ & & \mathrm{W} & \mathrm{L} \end{bmatrix}, \begin{bmatrix} \mathrm{W} & \mathrm{L} & \mathrm{W} & & \\ & \mathrm{W} & \mathrm{L} & \mathrm{W} & \\ & & \mathrm{W} & \mathrm{L} & \mathrm{W} \\ & & & \mathrm{W} & \mathrm{L} \end{bmatrix}, \begin{bmatrix} \mathrm{W} & \mathrm{L} & \mathrm{W} & & & \\ & \mathrm{W} & \mathrm{L} & \mathrm{W} & & \\ & & \mathrm{W} & \mathrm{L} & \mathrm{W} & \\ & & & \mathrm{W} & \mathrm{L} & \mathrm{W} \\ & & & & \mathrm{W} & \mathrm{L} \end{bmatrix}$$

The extra W's added at the right of the diagonal in Pater's matrices do not contribute anything to consistency, in the sense that they do not enlarge the set of consistent rankings. They only serve the purpose of confounding re-ranking rules that perform constraint promotion and are thus sensitive to these W's. Indeed, Pater's extra layer of W's is able to fool the GLA, as recalled in Subsection 5.3. But it is not able to fool the non-calibrated re-ranking rule (72): the analysis developed in Magri (2012b) guarantees again a feasible worst-case number of errors (smaller than n^5, where n is the number of constraints).

Let's thus try to further aggravate the learning challenge by adding yet another layer of useless W's. Thus, let the *aggravated Pater's matrix* for n constraints be the ERC matrix with n columns and $n-1$ rows obtained from the diagonal matrix by "adding" *two* (rather than just *one*) W's at the right of every L (but for the penultimate row, where only one W is added; and for the last row, where no W's are added). To illustrate, I give in (78) the aggravated Pater's matrices for $n = 4, 5, 6$ constraints.

(78)
$$\begin{bmatrix} \mathrm{W} & \mathrm{L} & \mathrm{W} & \mathrm{W} \\ & \mathrm{W} & \mathrm{L} & \mathrm{W} \\ & & \mathrm{W} & \mathrm{L} \end{bmatrix}, \begin{bmatrix} \mathrm{W} & \mathrm{L} & \mathrm{W} & \mathrm{W} & \\ & \mathrm{W} & \mathrm{L} & \mathrm{W} & \mathrm{W} \\ & & \mathrm{W} & \mathrm{L} & \mathrm{W} \\ & & & \mathrm{W} & \mathrm{L} \end{bmatrix}, \begin{bmatrix} \mathrm{W} & \mathrm{L} & \mathrm{W} & \mathrm{W} & & \\ & \mathrm{W} & \mathrm{L} & \mathrm{W} & \mathrm{W} & \\ & & \mathrm{W} & \mathrm{L} & \mathrm{W} & \mathrm{W} \\ & & & \mathrm{W} & \mathrm{L} & \mathrm{W} \\ & & & & \mathrm{W} & \mathrm{L} \end{bmatrix}$$

Theorem 3 guarantees that the EDRA with the non-calibrated re-ranking rule (72) converges on aggravated Pater's matrices. Yet, it turns out that convergence is not efficient: even in the best, shortest run, the algorithm makes way too many errors before it finally converges on a consistent ranking vector. In Appendix A.4, I show how to compute the best-case number of mistakes made by the EDRA with the re-ranking rule (72) on aggravated Pater's matrices. The computation is analogous to the explanation provided in Subsection 5.3 for Pater's counterexample against the GLA's convergence. The the best-case number of errors is provided in (79a) for various choices of the number $n = 5, 7, 9, 11, 13, 15$ of constraints. For instance, the EDRA will always perform nothing less than 107,920 updates before converging to a ranking vector consistent with the aggravated Pater's matrix corresponding to $n = 15$ constraints. In

(79)

	$n=5$	$n=7$	$n=9$	$n=11$	$n=13$	$n=15$
a.	12	86	532	3,159	18,495	107,920
b.	32	128	512	2,048	8,192	32,768

In (79b), I report the values of 2^n for the corresponding values of n. The table thus shows that the best-case number of mistakes made by the EDRA with the re-ranking rule (72) on aggravated Pater's matrices grows exponentially in the number of constraints. In other words, the number of mistakes performed before reaching convergence grows so quickly with n that it is unfeasible even for a small number n of constraints.

7.6. **Summary.** This Section has studied the re-ranking rule (72), that increases the promotion amount up to the calibration threshold ℓ/w, namely the ratio between the number ℓ of undominated loser-preferrers and the number w of winner-preferrers. It has shown that convergence is retained in this case, but efficiency is lost: an exponential number of errors might be required before reaching convergence. This result is somewhat surprising. In fact, one might expect that, in order to bring down the number of errors from exponential to something feasible, the promotion amount would need to be exponentially reduced, so as to become practically useless from the modeling perspective of Section 4. This intuitively plausible conjecture turns out to be false. In fact, as seen in Section 6, it is sufficient to decrease the promotion amount just slightly, for example from $\frac{\ell}{w}$ down to $\frac{\ell}{w+1}$, in order to obtain efficient convergence. This shows the importance of a thorough computational understanding of the algorithm when setting the implementation parameters of the model.

8. Conclusion

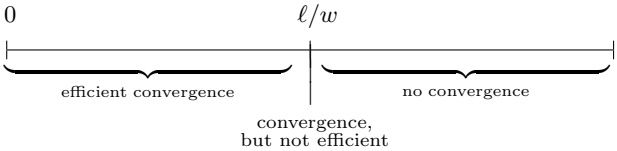
Section 2 has reviewed from the literature an important learning scheme in computational OT, namely EDRAs. The crucial property of EDRAs is that they are trained on a stream of data and predict a sequence of ranking vectors. The switch from the current ranking vector to the updated one is prompted by an error on the current piece of data. The crucial implementation detail for EDRAs is the *re-ranking rule* used by the algorithm to switch from the current to the updated ranking vector. In this paper, I have focused on re-ranking rules of the form (59), repeated in (80). Only loser-preferrers that need to be demoted are indeed demoted, namely only the currently *undominated* ones. And they are demoted by a small fixed *demotion amount*, say 1 for concreteness. All winner-preferrers are promoted by a certain *promotion amount* p, which can be null or positive.

(80) a. Decrease the ranking value of each undominated loser-preferrer by 1;
b. increase the ranking value of each winner-preferrer by p.

A proper EDRA should eventually stop making errors on the training sequence of data, and thus settle on a ranking vector consistent with the input data (*convergence*). Furthermore, the number of errors made before reaching convergence should not grow exponentially with the number of constraints (*efficiency*). Within the update scheme (80), the crucial issue is then how to choose the promotion amount p to ensure efficient convergence.

Building on previous work (in particular, Tesar and Smolensky 1998, Boersma 1997, 1998, and Pater 2008), this paper has developed a complete answer to this question, summarized in (81). The pivot of the theory is the *calibration threshold*, namely the ratio ℓ/w between the number ℓ of undominated loser-preferrers that get demoted and the number w of winner-preferrers that get promoted.

(81)



As reviewed in Section 3, T&S showed that efficient convergence holds for *demotion-only* re-ranking rules, namely rules of the form (80) with a null promotion amount (i.e. $p = 0$). Building on their analysis, Section 6 has shown that efficient convergence extends

(with comparable error-bounds) from demotion-only re-ranking rules to *calibrated* promotion/demotion re-ranking rules, namely rules of the form (80) with a promotion amount strictly smaller than the calibration threshold (i.e. $p < \ell/w$). This result cannot be extended any further. In fact, Section 7 has looked at re-ranking rules of the form (80) with a promotion amount that coincides with the calibration threshold (i.e. $p = \ell/w$). And has shown that, although convergence holds, efficiency breaks down, as the number of errors can grow exponentially in the number of constraints. The calibration threshold is thus the tipping point for the theory of EDRAs' convergence. Above that threshold, convergence is lost too. Indeed, in Section 5 I have discussed in detail Pater's (2008) counterexample, that shows that convergence also breaks down once we pass the calibration threshold, namely for re-ranking rules of the form (80) with a promotion amount that exceeds the calibration threshold ($p > \ell/w$), such as Boersma's re-ranking rule with $p = 1$.

The focus of the paper has been mainly computational, looking at the crucial learnability requirement of EDRAs' efficient convergence. Yet, Section 4 has provided an initial glimpse into the modeling implications of these computational results. I have looked at an application of EDRAs that has figured prominently in the literature, namely modeling the child early acquisition of phonotactics. It is common in this literature to assume that the child posits underlying forms faithful to the adult winner forms. As Pater and Barlow (2003, p. 490) write "the [underlying form] in [...] child phonology [is] taken to correspond to the child's stored lexical representation" under the assumption that it "is likely [that] children do perceive and store [forms] accurately". This assumption makes good sense computationally, as it never leads to inconsistent ERCs (under mild assumptions on the constraint set), as shown by Tesar (2008). In this paper, I have pointed out that the modeling implications of this assumption that EDRAs be trained on faithful mappings. In fact, it predicts that the faithfulness constraints are never loser-preferrers. Hence they would never be re-ranked by a demotion-only re-ranking rule. This cannot be right from the perspective of the *restrictiveness* of the final grammar, namely how well it rules out illicit forms. Also, it cannot be right from the perspective of the *matching* between predicted learning sequences and child acquisition paths. The current literature thus displays a gap between what the OT acquisition literature needs (namely sound EDRAs that perform both constraint promotion and demotion) and what the OT computational literature has been able to deliver (namely EDRAs that are sound but don't perform constraint promotion, such as EDCD; or EDRAs that perform constraint promotion but are not sound, such as the GLA). This paper fills this gap: it develops sound EDRAs that perform constraint promotion too. These results thus contribute to a general line of research that tries to establish EDRAs as proper models of the child acquisition of phonotactics, both from a computational and a modeling perspective.

Child phonology displays a characteristic degree of variation. Boersma (1997, 1998) suggests to model variation through a stochastic variant of EDRAs. The idea of this variant is that the mapping from the current ranking vector to its refinements used in order to check consistency with the current piece of data is not deterministic but rather stochastic. This is achieved by looking not at the refinements of the current ranking vector but rather at the refinements of a corrupted version thereof, obtained by adding to the current ranking values a small additive noise. The relative size of current ranking values close to each other can switch due to the additive noise, thus modeling variation. As shown in Magri (2012a), all convergence results and error-bounds presented in this paper for deterministic EDRAs trivially extend to this stochastic implementation. The extension is straightforward if the additive noise is bounded (say a gaussian or a uniform distribution truncated to zero outside of the some interval $[-\Delta, +\Delta]$). The extension consists of a simple probabilistic argument, if the additive noise is unbounded but concentrated around zero (say, a gaussian with null mean, as originally suggested by Boersma).

Appendix

A.1. **Extension to arbitrary initial ranking vectors.** Throughout the paper, I have investigated convergence in the case where the initial ranking values $\theta_1^{\text{init}}, \dots, \theta_n^{\text{init}}$ were all identical, say all equal to zero (the actual value does not really matter). In this Appendix, I show how to obtain error bounds for demotion-only and calibrated re-ranking rules in the case of an arbitrary initial ranking vector $\boldsymbol{\theta}^{\text{init}} = (\theta_1^{\text{init}}, \dots, \theta_n^{\text{init}})$. The extension is straightforward, but has never appeared in the literature. It turns out that the properties of the initial ranking vector that are relevant for the error bounds can be extracted through the quantity $\Delta(\boldsymbol{\theta}^{\text{init}})$ defined in (82), namely the sum of the difference between each initial ranking value θ_k^{init} and the smallest ranking value $\min_{h=1} \theta_h^{\text{init}}$. Intuitively, this quantity measures how *scattered* the initial ranking values are. In fact, $\Delta(\boldsymbol{\theta}^{\text{init}})$ is null for the case of identical initial ranking values, small for ranking values close to each other and large if there are some ranking values that are very small and some other ranking values that are very large.

$$\Delta(\boldsymbol{\theta}^{\text{init}}) = \sum_{k=1}^{n} \left(\theta_k^{\text{init}} - \min_{h=1} \theta_h^{\text{init}} \right) \tag{82}$$

Consider the general re-ranking rule (59) from Section 6, repeated in (83). It demotes each undominated loser-preferrer by 1 and it promotes each winner preferrer by a promotion amount $p \in [0, 1]$.

(83) a. Decrease the ranking value of each undominated loser-preferrer by 1;
b. increase the ranking value of each winner-preferrer by p.

A.1.1. *The crucial invariant.* In the case of null initial ranking values, T&S's (generalized) Fact 2 from Subsection 6.2 provided a crucial invariant for the current ranking values entertained by the EDRA: they can never get much smaller than zero. The reasoning trivially extends to arbitrary initial ranking values, yielding the following further generalization of Fact 2: the current ranking values can never get much smaller than the smallest initial ranking value.

Fact 2 (further generalized). *Assume that the input ERC matrix is consistent with a ranking $\gg$. Without loss of generality, assume that this ranking is $C_1 \gg C_2 \gg \cdots \gg C_n$. Let $\theta_1, \dots, \theta_n$ be the current ranking values entertained by an EDRA in a generic run on those input ERCs, up to a generic time, starting from arbitrary initial ranking values $\theta_1^{\text{init}}, \dots, \theta_n^{\text{init}}$. Assume that the re-ranking rule used by the EDRA has the shape (83), namely it demotes by 1 only the currently* undominated *loser-preferrers. The current ranking values thus satisfy condition (84) for every $k = 1, \dots, n$.*

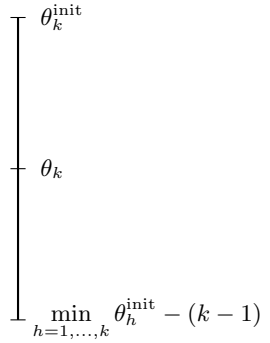
$$\theta_k \geq \min_{h=1,\dots,n} \theta_h^{\text{init}} - (k-1) \tag{84}$$

Namely, the ranking value θ_k of the constraint C_k assigned to the kth stratum (with the 1st stratum being the top one) never drops by more than $(k-1)$ underneath the smallest initial ranking value $\min_{h=1,\dots,n} \theta_h^{\text{init}}$. ■

A.1.2. *Demotion-only re-ranking rules.* Consider the demotion-only re-ranking rule (25) studied in Section 3, which is a special case of (83) with $p = 0$. At least one constraint is demoted at each update. Hence, the total number T of updates is at most the sum of the number of times C_1 has been demoted, and the number of times C_2 has been demoted, etcetera, as stated in (85a). Each time constraint C_k is demoted, it is demoted by 1. And it is never promoted. Hence, the number of times that constraint C_k has been demoted up to the time considered is equal to the distance $\theta_k^{\text{init}} - \theta_k|$ between its initial ranking value θ_k^{init} and its current ranking value θ_k, as stated in (85b). The inequality (84) says that θ_k sits in between θ_k^{init} and $\min_{h=1,\dots,n} \theta_h^{\text{init}} - (k-1)$, as depicted in (86). Thus,

the distance between the latter two points upper bounds the distance $\theta_k^{\text{init}} - \theta_k$ between θ_k and θ_k^{init}, as stated in (85c). Finally, step (85e) follows from the definition (82) of the constant $\Delta(\boldsymbol{\theta}^{\text{init}})$ and from the identity $\sum_{k=1}^{n}(k-1) = \frac{1}{2}n(n-1)$.

$$
\begin{aligned}
(85) \quad T &\overset{(a)}{\leq} \sum_{k=1}^{n} \big(\#\text{ of demotions of } C_k\big) \\
&\overset{(b)}{=} \sum_{k=1}^{n} \big(\theta_k^{\text{init}} - \theta_k\big) \\
&\overset{(c)}{\leq} \sum_{k=1}^{n} \Big(\theta_k^{\text{init}} - \big(\min_{h=1\ldots n} \theta_h^{\text{init}} - (k-1)\big)\Big) \\
&\overset{(d)}{=} \sum_{k=1}^{n} \Big(\theta_k^{\text{init}} - \min_{h=1\ldots n} \theta_h^{\text{init}}\Big) + \sum_{k=1}^{n}(k-1) \\
&\overset{(e)}{=} \Delta(\boldsymbol{\theta}^{\text{init}}) + \tfrac{1}{2}n(n-1)
\end{aligned}
\qquad (86)
$$



We have thus proven the following extension of Theorem 1 from null to arbitrary initial ranking values. Recall that, if the initial ranking values are all identical (say, all null), then $\Delta(\boldsymbol{\theta}^{\text{init}}) = 0$, and thus we obtain back the error bound $\frac{1}{2}n(n-1)$ already obtained in Section 3.

THEOREM 1 (EXTENDED). *The EDRA (21) with the demotion-only re-ranking rule (25) run on a consistent input ERC matrix corresponding to n constraints starting from an arbitrary initial ranking vector $\boldsymbol{\theta}^{\text{init}} = (\theta_1^{\text{init}}, \ldots, \theta_n^{\text{init}})$ can perform at most $\Delta(\boldsymbol{\theta}^{\text{init}}) + \frac{1}{2}n(n-1)$ errors before converging.* ■

The bound $\Delta(\boldsymbol{\theta}^{\text{init}}) + \frac{1}{2}n(n-1)$ on the worst-case number of errors is tight, as shown by the same example in (34) with the ERCs fed in the fixed order $\mathbf{a}_1 \to \mathbf{a}_2 \to \mathbf{a}_3$ and with the initial ranking vector $\boldsymbol{\theta}^{\text{init}} = (4, 3, 2, 1)$.

A.1.3. *Calibrated re-ranking rules.* Consider next the calibrated demotion/promotion re-ranking rule (64) introduced in Section 6, which is a special case of (83) with $p = \frac{\ell}{w+1}$, where ℓ is the number of currently undominated loser-preferrers and w is the total number of winner-preferrers. The invariant (84) ensures that the sum of the current ranking values can be lower bounded as in (87).

$$
(87) \quad \sum_{k=1}^{n} \theta_k \geq \sum_{k=1}^{n} \left(\min_{h=1,\ldots,n} \theta_h^{\text{init}} - (k-1)\right) = n \min_{h=1,\ldots,n} \theta_h^{\text{init}} - \frac{1}{2}n(n-1)
$$

As seen in Subsection 6.3, the sum of the current ranking values is decreased by at least $1/n$ with every update. After T updates, it has thus decreased by at least T/n from the sum of the initial ranking values, as stated in (88).

$$
(88) \quad \sum_{k=1}^{n} \theta_k \leq \sum_{k=1}^{n} \theta_k^{\text{init}} - \frac{T}{n}
$$

Combining the two inequalities (87) and (88), I conclude that the number T of updates must be smaller than $n\Delta(\theta^{\text{init}}) + \frac{1}{2}n^2(n-1)$. We have thus proven the following extension of Theorem 2 from null to arbitrary initial ranking values.

THEOREM 2 (EXTENDED). *The EDRA (21) with the calibrated promotion/demotion re-ranking rule (64) run on a consistent input ERC matrix corresponding to n constraints starting from an arbitrary initial ranking vector $\boldsymbol{\theta}^{\text{init}} = (\theta_1^{\text{init}}, \ldots, \theta_n^{\text{init}})$ can perform at most $n\Delta(\theta^{\text{init}}) + \frac{1}{2}n^2(n-1)$ errors before converging.* ■

Also in the case of an arbitrary initial ranking vector, the error bound for the calibrated case is worse by a factor of n than the error-bound for the demotion-only case.

A.2. **Convergence of a generic calibrated re-ranking rule.** In Section 6, I have looked for concreteness at a specific calibrated re-ranking rule, namely the one in (64), that demotes each of the l undominated loser-preferrers by 1 and promotes each of the w winner-preferrers by $\frac{l}{w+1}$. In this Appendix, I look at a generic calibrated re-ranking rule (89).

(89) a. Decrease the ranking value of each of the l loser-preferrers by 1;
b. increase the ranking value of each of the w winner-preferrers by $p = \frac{l}{w+\delta}$.

This re-ranking rule is calibrated as long as $\delta > 0$. Indeed, the distance of the promotion amount p from the calibration threshold l/w is controlled by the constant δ: the larger δ, the smaller the promotion amount p is w.r.t. the calibration threshold. In particular, the case $\delta = 1$ corresponds to the re-ranking rule (64) already considered in Section 6. And the case where δ goes to infinity corresponds to the demotion-only case $p = 0$ considered in Section 3.

The reasoning for the case $\delta = 1$ presented in Section 6 trivially extends to an arbitrary $\delta > 0$, yielding the following generalization of Theorem 2 of Section 6.

THEOREM 2 (GENERALIZED). *An EDRA with the general calibrated re-ranking rule (89) run on a consistent input ERC matrix corresponding to n constraints starting from null initial ranking values can perform at most*

$$\frac{1}{2}\frac{W+\delta}{\delta}n(n-1) \tag{90}$$

mistakes before converging, where W is the largest number of winner-preferrers over all input ERCs.

Proof. With every update, the sum of the current ranking values is decreased by l, as each of the l undominated loser preferrers is demoted by 1. And it is furthermore increased by $\frac{wl}{w+\delta}$, as each of the w winner-preferrers is promoted by $\frac{l}{w+\delta}$. In the end, the sum of the current ranking values is thus decreased by $l - \frac{wl}{w+\delta} = \frac{\delta l}{w+\delta}$. As the number l of undominated loser-preferrers is at least 1 and the number w of winner-preferrers is at most W, I conclude that the sum of the current ranking values is decreased by at least $T\frac{\delta}{W+\delta}$ after T updates. On the other hand, the sum of the current ranking values starts at zero and can never get smaller than $-\frac{1}{2}n(n-1)$, by the generalized Fact 2 stated in Subsection 6.2. In conclusion, the number of updates T in the case of the re-ranking rule (89) must satisfy the inequality $T\frac{\delta}{W+\delta} \leq \frac{1}{2}n(n-1)$, which yields the error-bound in (90). □

As there are a total of n constraints and each ERC must have at least a loser-preferrer (ERCs that have no loser-preferrers cannot ever trigger any update and can therefore be ignored), then the largest number W of winner-preferrers is upper bound by $n-1$ and the bound (90) becomes (91).

$$\frac{1}{2}\frac{n-1+\delta}{\delta}n(n-1) \tag{91}$$

The bound (91) for $\delta = 1$ gives back the bound $\frac{1}{2}n^2(n-1)$ of the original Theorem 2 of Section 6. As δ increases and the promotion amount $p = \frac{l}{w+\delta}$ thus gets further away from the calibration threshold $\frac{l}{w}$, the bound (91) on the number of mistakes decreases, ensuring faster convergence. In the limit of δ going to infinity, the coefficient $\frac{n-1+\delta}{\delta}$ goes to 1, and the bound (91) thus becomes the bound $\frac{1}{2}n(n-1)$ already obtained in Theorem 1 of Section 3 for the case with null $p = 0$ promotion amount.

Note that the *cubic* rather than *quadratic* growth in n of the bound (91) comes from the fact that I have upper bounded the largest number W of winner-preferrers in a generic input ERC with $n-1$. But in most applications, W is much smaller than $n-1$, as the winner and the loser forms that correspond to an ERC differ only under a few respects

and thus most of the constraints are even. Furthermore, if the loser forms are properly chosen so that the input ERCs have as few winner-preferrers as possible, then W might be forced into a constant in certain applications. In that case, the error-bound (90) for calibrated promotion grows only quadratically in the number of constraints n, just as the bound $\frac{1}{2}n(n-1)$ for the demotion-only case.

A.3. **Why EDRAs cannot loop.** Consider the general re-ranking rule (59) from Section 6, repeated once more in (92). Throughout this Subsection, I assume that the promotion amount p is never null.

(92) a. Decrease the ranking value of each undominated loser-preferrer by 1;
b. increase the ranking value of each winner-preferrer by p.

This Subsection shows that the EDRA with the re-ranking rule (92) cannot loop on consistent input ERC matrices, as stated in Fact 5 repeated below. The proof is based on a connection between OT-consistency and *conic independence*.

FACT 5. *If the input ERC matrix is consistent, the EDRA (21) with the re-ranking rule (92) can never loop back to a current ranking vector that it had previously dismissed.* ■

Let m be the total number of input ERCs. To simplify the presentation, let me start by assuming that the input ERC matrix has a unique L per ERC. The contribution of the ith ERC $\mathbf{a}_i$ to the current ranking vector according to this re-ranking rule can thus be summarized with the corresponding *update vector* $\overline{\mathbf{a}}_i$ as in (93): the entry corresponding to the loser-preferrer is equal to -1; the entries corresponding to winner-preferrers are set equal to the corresponding promotion amount $p > 0$; all other entries are 0.

$$\text{(93)}\quad \mathbf{a}_i = [a_1, \ldots, a_n] \longrightarrow \overline{\mathbf{a}}_i = \begin{bmatrix} \overline{a}_1 \\ \vdots \\ \overline{a}_n \end{bmatrix} \text{ where } \overline{a}_k = \begin{cases} p & \text{if } a_k = \text{W} \\ -1 & \text{if } a_k = \text{L} \\ 0 & \text{otherwise} \end{cases}$$

Suppose that the initial ranking values are all null. The current ranking vector $\boldsymbol{\theta}^t$ entertained by the EDRA with the re-ranking rule (92) can be described as in (94), namely as a combination of the update vectors, each multiplied by the number of updates α_i^t triggered by the corresponding ith ERC in the run considered up to time t. Equation (56) obtained in the discussion of Pater's counterexample in Subsection 5.3, is a special case of this general equation (94). Of course, the coefficients α_i^t are by definition all non-negative. Thus, the identity (94) can be summarized by saying that the current ranking vector is a *conic combination* of the m update vectors.

$$\text{(94)}\quad \boldsymbol{\theta}^t = \underset{\substack{\text{number of updates triggered by} \\ \text{the } i\text{th ERC up to time } t \\ \downarrow}}{\alpha_1^t} \overline{\mathbf{a}}_1 + \ldots + \underset{\downarrow}{\alpha_i^t} \overline{\mathbf{a}}_i + \cdots + \alpha_m^t \overline{\mathbf{a}}_m$$
$$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \downarrow$$
$$\qquad\qquad\qquad\qquad\qquad\qquad \text{update vector corresponding to the } i\text{th ERC}$$

As the current ranking vector is a conic combination of the update vectors, it is interesting to study the *conic geometry* of these vectors, namely the formal properties of their *conic combinations*. Here is a particularly important conic property. The update vectors are called *conically independent* provided that there are no coefficients $\alpha_1, \ldots, \alpha_m$ that satisfy the three conditions (95); see Bertsekas et al. (2003). These conditions say that it is not possible to synthesize the null vector as a conic combination of the update vectors, unless of course the coefficients are all set equal to zero

(95) a. $\alpha_1 \overline{\mathbf{a}}_1 + \ldots \alpha_m \overline{\mathbf{a}}_m = \mathbf{0}$;
b. $\alpha_i \geq 0$ for all $i = 1, \ldots, m$;

c. $\alpha_i \neq 0$ for some $i = 1, \dots, m$.

Fact 6 below says that OT-consistency of the input ERC matrix (with a unique L per row) entails conic independence of the corresponding update vectors. And Fact 7 says that conic independence of the update vectors in turn entails that the EDRA cannot loop. Fact 5 thus follows from these two auxiliary Facts 6 and 7. The assumption that the input ERCs have a unique L per row can be easily dropped, as discussed at the end of this Subsection.

FACT 6. *Consider an input ERC matric that has a unique* L *per row. If it is consistent, then the corresponding update vectors (93) are conically independent.* ■

Proof. Recall from Fact 1 that any consistent ERC matrix has the shape in (30) repeated in (96), modulo re-ordering of its rows and its columns and relabeling of the constraints. The tableau (96) has a top block of rows whose first entry is W; followed by a second block of rows whose first entry is e and whose second entry is W; and so on.

(96)

	C_1	C_2	...		C_d		C_n
1st block	W						
	|	...	...	...	...	...	...
	W						
2nd block	e	W					
	|	|	...	...	...	...	...
	e	W					
⋮			⋱	...	...	...	...
final block	e	e	—	e	W		
	|	|		|	|	...	...
	e	e	—	e	W		

For consistency with standard notation from Linear Algebra, in (93) I have paired up a *row* of the input ERC matrix with a corresponding *column* update vector. To get around this rows/columns mismatch, let me turn (96) upside down (i.e. transpose) so that rows become columns, as in (97).

(97)

	1st block			2nd block				final block		
1st column	W	——	W	e	——	e		e	——	e
2nd column		...		W	——	W		e	——	e
⋮							⋱			
dth column		...			...			W	——	W
⋮		⋮			⋮		⋮		⋮	

The update vectors can now be read straightforwardly out of (97): the ith update vector is obtained by looking at the ith column of (97). Recall that the mapping (93) from ERCs into update vectors replaces a e with a 0 and a W with the positive quantity $p > 0$. The collection of update vectors can thus be made a bit more explicit as in (98).[18]

(98)

$$\underbrace{\begin{bmatrix} p \\ \hline \\ \hline \\ \hline \\ \hline \end{bmatrix}, \dots, \begin{bmatrix} p \\ \hline \\ \hline \\ \hline \\ \hline \end{bmatrix}}_{\text{1st block}}, \underbrace{\begin{bmatrix} 0 \\ \hline p \\ \hline \\ \hline \\ \hline \end{bmatrix}, \dots, \begin{bmatrix} 0 \\ \hline p \\ \hline \\ \hline \\ \hline \end{bmatrix}}_{\text{2nd block}}, \dots, \underbrace{\begin{bmatrix} 0 \\ \hline \vert \\ \hline 0 \\ \hline p \\ \hline \end{bmatrix}, \dots, \begin{bmatrix} 0 \\ \hline \vert \\ \hline 0 \\ \hline p \\ \hline \end{bmatrix}}_{\text{final block}}$$

[18]Each update vector can have a different value for the promotion amount p. This fact does not play any role in the reasoning, so I do not encode it explicitly in the notation, and use the same p for all update vectors.

Suppose that a conic combination of these update vectors (98) with some nonnegative coefficients yields the null vector, namely that conditions (95a) and (95b) hold. Let's focus on the first component of this conic combination, as in (99). The first component of the update vectors in the 1st block is always positive (recall that $p > 0$ by hypothesis). Suppose there are k vectors in the first block. The first component of the remaining $m-k$ update vectors is always null. In order for the first component of this conic combination to be zero, the nonnegative coefficients that multiply the update vectors in the 1st block must be all null, namely $\alpha_1 = \ldots = \alpha_k$.

$$(99)\quad \underbrace{\alpha_1 \begin{bmatrix} p \\ \\ \end{bmatrix} + \ldots + \alpha_k \begin{bmatrix} p \\ \\ \end{bmatrix}}_{\text{1st block}} + \underbrace{\alpha_{k+1} \begin{bmatrix} 0 \\ \\ \end{bmatrix} + \ldots + \alpha_m \begin{bmatrix} 0 \\ \\ \end{bmatrix}}_{\text{remaining blocks}} = \begin{bmatrix} 0 \\ \\ \end{bmatrix}$$

As their coefficients are null, the update vectors in the 1st block can be ignored in the conic combination. By looking at the second component and reasoning analogously, I conclude that also the coefficients that multiply the update vectors in the 2nd block are null. By repeating the reasoning, I conclude that these multiplicative coefficients are all null, contradicting condition (95c) in the definition of conic independence. □

FACT 7. *If the update vectors are conically independent, then the EDRA cannot loop back to a current ranking vector it had previously updated.* ■

Proof. Suppose by contradiction that the EDRA can indeed loop back to a ranking vector that it had dismissed at a previous time. This means that it is possible for the algorithm to walk through a learning path with the properties (100)

(100) a. The EDRA entertains the same ranking vector at two times t and t';
b. assume for concreteness that time t precedes time t';
c. the EDRA entertains a different ranking vector at a time in between t and t'.

Assumption (100a) that the ranking vectors $\boldsymbol{\theta}^t$ and $\boldsymbol{\theta}^{t'}$ entertained at times t and t' coincide, can be expressed as the identity (101a), using the description (94) of the current ranking vector in terms of update vectors. Here, α_1^t and $\alpha_1^{t'}$ are the number of updates triggered by ERC 1 up to time t and t', respectively; an analogous interpretation holds for the other coefficients. As the number of updates grows with time, assumption (100b) that time t' follows time t thus entails that the coefficient $\alpha_i^{t'}$ at time t' is larger than or equal to the corresponding coefficient α_i^t at time t, as stated in (101b). Furthermore, assumption (100c) entails that some update has happened at some time in between t and t', so that at least one of the coefficients has increased by at least 1 from time t to time t', as stated in (101c).

(101) a. $\alpha_1^t \overline{\mathbf{a}}_1 + \ldots + \alpha_m^t \overline{\mathbf{a}}_m = \alpha_1^{t'} \overline{\mathbf{a}}_1 + \ldots + \alpha_m^{t'} \overline{\mathbf{a}}_m$
b. $\alpha_i^{t'} \geq \alpha_i^t$ for all $i = 1, \ldots, m$
c. $\alpha_i^{t'} \neq \alpha_i^t$ for some $i = 1, \ldots, m$

By moving everything to the right hand side, (101a) can of course be restated as in (102a), where I have introduced the coefficients $\alpha_i = \alpha_i^{t'} - \alpha_i^t$ for all $i = 1, \ldots, m$. The property (101b) that $\alpha_i^{t'}$ is larger than or equal to α_i^t because time t' follows time t, can then be restated as the property (102b) that all coefficients α_i are non-negative. And the property (101c) that some coefficient $\alpha_i^{t'}$ is different from the corresponding coefficient α_i^t because some update has happened in between times t and t', can be restated as the property (102c) that at least one of the coefficients α_i is non-null.

(102) a. $\alpha_1 \overline{\mathbf{a}}_1 + \ldots + \alpha_m \overline{\mathbf{a}}_m = \mathbf{0}$

b. $\alpha_i \geq 0$ for all $i = 1, \ldots, m$

c. $\alpha_i \neq 0$ for some $i = 1, \ldots, m$

Conditions (102) say that the null vector can be synthesized as a conic combination of the update vectors, without the coefficients $\alpha_1, \ldots, \alpha_m$ being all null. This contradicts the hypothesis that the update vectors are conically independent. □

To conclude the proof of Fact 5, I need to consider the case where the input ERC matrix contains rows with multiple L's. The additional difficulty in this case is that the contribution of the ith ERC to the current ranking vector depends on the number of currently undominated loser-preferrers, namely it can be different at different times, and thus cannot be distilled into a unique update vector $\overline{\mathbf{a}}_i$ as in (93). But this difficulty can be straightforwardly overcome, at the expense of a slightly more cumbersome notation. Let m be the total number of ERCs. Suppose that the ith ERC has ℓ_i loser-preferrers $C_{k_1}, \ldots, C_{k_{\ell_i}}$. Let $\mathcal{C}^i_1, \ldots, \mathcal{C}^i_{2^{\ell_i}-1}$ be all $2^{\ell_i}-1$ non-empty subsets of the set $\{C_{k_1}, \ldots, C_{k_{\ell_i}}\}$ of loser-preferrers. For every such subset $\mathcal{C}^i_j$, let $\overline{\mathbf{a}}_{i,j}$ be the *update vector* defined as follows: the components corresponding to the loser-preferrers in the subset $\mathcal{C}^i_j$ are equal to -1; the components corresponding to winner-preferrers are equal to p; the remaining components are equal to 0. Furthermore, let $\alpha^t_{i,j}$ be the number of updates triggered by this ith ERC up to time t because all and only the loser-preferrers in the set $\mathcal{C}^i_j$ were currently undominated. The current ranking vector can then be expressed as a conic combination of these update vectors through these non-negative coefficients, namely $\boldsymbol{\theta}^t = \sum_{i=1}^m \sum_{j=1}^{2^{\ell_i}-1} \alpha^t_{i,j} \overline{\mathbf{a}}_{i,j}$. Again, these update vectors $\overline{\mathbf{a}}_{i,j}$ are conically independent. And I can thus trivially extend the preceding reasoning.

A.4. **On the number of updates for smallest non-calibrated promotion.** Recall from Section 7.5 that the *aggravated Pater's ERC matrix* for n constraints is obtained from the corresponding diagonal tableaux by adding two W's to the right of each L. To illustrate, I give in (103) the matrix corresponding to $n = 7$ constraints. It has $6 = n - 1$ ERCs; it has a W on every diagonal entry, followed by an L followed in turn by two more W's (but for the last two rows, whose L's are followed by one and zero W's, respectively).

(103)

	C_1	C_2	C_3	C_4	C_5	C_6	C_7
ERC 1	W	L	W	W			
ERC 2		W	L	W	W		
ERC 3			W	L	W	W	
ERC 4				W	L	W	W
ERC 5					W	L	W
ERC 6						W	L

Consider a run of the EDRA on the aggravated Pater's input ERC matrix for n constraints. Suppose the algorithm starts from null initial ranking values. And that it uses the re-ranking rule (72) with the smallest non-calibrated promotion amount, repeated in (104) for the case of input ERCs with a single loser-preferrer, as in the case of aggravated Pater's ERC matrix.

(104) a. Decrease the ranking value of the loser-preferrer by 1;

b. increase the ranking value of each of the w winner-preferrers by $1/w$.

The convergence Theorem 3 ensures that after a finite number of errors the EDRA will converge to a final ranking vector consistent with the input ERC matrix, and learning will cease. Yet, the Theorem does not provide any estimate of the number of errors made before convergence. This Subsection shows that this number grows exponentially with the number n of constraints, as anticipated in Subsection 7.5.

For concreteness, suppose the input matrix is the aggravated Pater's matrix (103) corresponding to $n = 7$ constraints. Let $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_7)$ be the final ranking vector the EDRA has converged on. Let $\alpha_1, \ldots, \alpha_6$ be the total number of updates triggered by each of the six input ERCs in the run considered. The final ranking values $\theta_1, \ldots, \theta_7$ can be expressed in terms of the coefficients $\alpha_1, \ldots, \alpha_6$ as in (105).

$$
\begin{aligned}
(105)\quad \theta_1 &= \tfrac{1}{3}\alpha_1 \\
\theta_2 &= -\alpha_1 + \tfrac{1}{3}\alpha_2 \\
\theta_3 &= \tfrac{1}{3}\alpha_1 - \alpha_2 + \tfrac{1}{3}\alpha_3 \\
\theta_4 &= \tfrac{1}{3}\alpha_1 + \tfrac{1}{3}\alpha_2 - \alpha_3 + \tfrac{1}{3}\alpha_4 \\
\theta_5 &= \tfrac{1}{3}\alpha_2 + \tfrac{1}{3}\alpha_3 - \alpha_4 + \tfrac{1}{2}\alpha_5 \\
\theta_6 &= \tfrac{1}{3}\alpha_3 + \tfrac{1}{3}\alpha_4 - \alpha_5 + \alpha_6 \\
\theta_7 &= \tfrac{1}{3}\alpha_4 + \tfrac{1}{2}\alpha_5 - \alpha_6
\end{aligned}
$$

Here is how these equations (105) are obtained. The ranking value θ_1 of constraint C_1 starts out null. It is only modified when ERC 1 triggers an update. In which case, it is increased by 1/3, as ERC 1 contains $w = 3$ winner-preferrers. In other words, the final ranking value θ_1 of constraint C_1 is 1/3 times the total number α_1 of updates triggered by ERC 1, as stated by the first equation in (105). Analogously, the ranking value θ_2 of constraint C_2 starts out null, is decreased by 1 every time ERC 1 triggers an update, and it is increased by 1/3 every time ERC 2 triggers an update, whereby we get the second equation in (105). The remaining equations in (105) are obtained analogously.

The comparative tableau (103) is only consistent with the ranking $C_1 \gg C_2 \gg \ldots \gg C_7$. As the final ranking vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_7)$ entertained by the EDRA at convergence is consistent with that tableau, it must thus satisfy the six strict inequalities $\theta_1 > \theta_2$, …, $\theta_6 > \theta_7$. Let me consider for instance the first of these six inequalities, repeated in (106a). Using the first two equations (105), this inequality (106a) can be rewritten as in (106b), in terms of the numbers of updates α_1 and α_2 triggered by ERC1 and ERC 2, respectively. If both sides of inequality (106b) are multiplied by the constant 3, we get the equivalent inequality (106c). As the variables α_1, α_2 as well as the coefficients are integers, the strict inequality (106c) is equivalent to the loose inequality (106d), where I have added a +1 to the right hand side.

$$
\begin{aligned}
(106)\quad &\text{a. } \theta_1 > \theta_2 \\
&\text{b. } \tfrac{1}{3}\alpha_1 > -\alpha_1 + \tfrac{1}{3}\alpha_2 \\
&\text{c. } \alpha_1 > -3\alpha_1 + \alpha_2 \\
&\text{d. } \alpha_1 \geq -3\alpha_1 + \alpha_2 + 1
\end{aligned}
$$

By reasoning this way, I conclude that the six strict inequalities $\theta_1 > \theta_2$, …, $\theta_6 > \theta_7$ are equivalent to the six inequalities (107) in terms of the number of updates $\alpha_1, \ldots, \alpha_6$ triggered by ERC1 through ERC 6, respectively.

$$
\begin{aligned}
(107)\quad \theta_1 > \theta_2 &\iff \alpha_1 \geq 1 - 3\alpha_1 + \alpha_2 \\
\theta_2 > \theta_3 &\iff -3\alpha_1 + \alpha_2 \geq 1 + \alpha_1 - 3\alpha_2 + \alpha_3 \\
\theta_3 > \theta_4 &\iff \alpha_1 - 3\alpha_2 + \alpha_3 \geq 1 + \alpha_1 + \alpha_2 - 3\alpha_3 + \alpha_4 \\
\theta_4 > \theta_5 &\iff 2\alpha_1 + 2\alpha_2 - 6\alpha_3 + 2\alpha_4 \geq 1 + 2\alpha_2 + 2\alpha_3 - 6\alpha_4 + 3\alpha_5 \\
\theta_5 > \theta_6 &\iff 2\alpha_2 + 2\alpha_3 - 6\alpha_4 + 3\alpha_5 \geq 1 + 2\alpha_3 + 2\alpha_4 - 6\alpha_5 + 6\alpha_6 \\
\theta_6 > \theta_7 &\iff 2\alpha_3 + 2\alpha_4 - 6\alpha_5 + 6\alpha_6 \geq 1 + 2\alpha_4 + 3\alpha_5 - 6\alpha_6
\end{aligned}
$$

The total number of updates performed by the EDRA in the run considered coincides with the sum $\alpha_1 + \ldots + \alpha_6$ of the number α_1 of updates triggered by ERC 1 plus the number α_2 of updates triggered by ERC 2 and so on down to the number α_6 of updates triggered by ERC 6. Furthermore, these nonnegative numbers $\alpha_1, \ldots, \alpha_6$ must satisfy the inequalities (107). Thus, the number of updates performed by the EDRA to reach

convergence cannot be smaller than the solution of the optimization problem (108). In other words, the solution of this optimization problem provides a bound on the best-case number of updates performed by the EDRA on the input matrix (103). As (108) is a linear program, it can be easily solved with standard linear programming techniques.

(108) minimize: $\alpha_1 + \ldots + \alpha_6$
subject to: $\alpha_1, \ldots, \alpha_6$ satisfy the inequalities (107)
$\alpha_1, \ldots, \alpha_6 \geq 0$

The reasoning just developed in the concrete case of the aggravated Pater's matrix (103) corresponding to $n = 7$ constraints extends to the case of an arbitrary number n of constraints. I can always construct an optimization problem akin to (108) that provides a bound on the best-case number of updates performed by the EDRA on that aggravated Pater's matrix. The solution of the optimization problems thus obtained for the aggravated Pater's matrices corresponding to various choices of the number n of constraints have been reported in (79a).[19]

Let me close by pointing out the close parallelism between the reasoning presented in this Subsection and the explanation for Pater's (2008) counterexample against the GLA's convergence provided in Subsection 5.3. The equations (105) are analogous to those in (57) in Subsection 5.3, and both are a special case of the vector equation (94) considered in Appendix A.3. The inequalities (107) are analogous to those in (58) in Subsection 5.3. Finally, showing that there are no coefficients α's that solve the inequalities (107) and add up to a small number corresponds to the final step of the explanation of Pater's counterexample, that showed that there are no coefficients α's that solve inequalities (58).

References

Anttila, Arto. 1997. Variation in finnish phonology and morphology. Doctoral Dissertation, Stanford University.

Anttila, Arto, and Young-mee Yu Cho. 1998. Variation and change in optimality theory. *Lingua* 104:31–56.

Bernhardt, Barbara Handford, and Joseph Paul Stemberger. 1998. *Handbook of phonological development from the perspective of constraint-based nonlinear phonology*. Academic Press.

Bertsekas, Dimitri P., Angelia Nedic, and Asuman E. Ozdaglar. 2003. *Convex Analysis and Optimization*. Athena Scientific.

Boersma, Paul. 1997. How We Learn Variation, Optionality and Probability. In *IFA Proceedings 21*, 43–58. University of Amsterdam: Institute for Phonetic Sciences.

Boersma, Paul. 1998. Functional Phonology. Doctoral Dissertation, University of Amsterdam. The Hague: Holland Academic Graphics.

Boersma, Paul. 1999. Optimality Theoretic Learning in the `Praat` Program. In *IFA Proceedings 23*, 17–35. University of Amsterdam: Institute for Phonetic Sciences. .

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 for the Gradual Learning Algorithm. *Linguistic Inquiry* 32:45–86.

[19] These values were computed using the Matlab file `MinimumRunningTime.m`, available on the author's website. It takes as input the aggravated Pater's ERC matrix corresponding to n constraints, for any n. It constructs the corresponding optimization problem akin to (108), generalizing the reasoning just presented here in the special case of the aggravated Pater's comparative tableaux corresponding to $n = 7$ constraints. And it solves this optimization problem using Matlab built-in subroutines for linear programming. Aggravated Pater's comparative matrices for $n = 5, 7, 9, 11, 13, 15$ constraints are provided in the file `AggravatedPaterMatrices.txt`, available on the author's website too. They can be copied and pasted directly into the Matlab Command Window.

Boersma, Paul, and Clara Levelt. 2000. Gradual Constraint-Ranking Learning Algorithm Predicts Acquisition Order. In *Proceedings of the 30th Child Language Research Forum*, 229–237. Stanford University: CSLI. Corrected version (ROA 361, 1999/08/28).

Boersma, Paul, and Joe Pater. 2007. Convergence Properties of a Gradual Learner for Harmonic Grammar. In *Proceedings of NELS 38*. .

Boersma, Paul, and Joe Pater. to appear. Convergence properties of a gradual learning algorithm for harmonic grammar. In *Harmonic grammar and harmonic serialism*, ed. John McCarthy and Joe Pater. London: Equinox Press.

Cesa-Bianchi, Nicolò, and Gábor Lugosi. 2006. *Prediction, Learning, and Games*. Cambridge University Press.

Coetzee, Andries W., and Joe Pater. 2008. Weighted constraints and gradient restrictions on place co-occurrence in muna and arabic. *Natural Language and Linguistic Theory* 26:289–337.

Compton, A. J., and M. Streeter. 1977. Child phonology: data collection and preliminary analyses. *Papers and Reports on Child Language Development* 7:99–109.

Davidson, Lisa, Peter W. Jusczyk, and Paul Smolensky. 2004. The initial and final states: Theoretical implications and experimental explorations of richness of the base. In *Constraints in Phonological Acquisition*, ed. R. Kager, J. Pater, and W. Zonneveld, 158–203. Cambridge University Press.

Dresher, E. 1999. Charting the Learning Path: Cues to Parameter Setting. *Linguistic Inquiry* 30:27–67.

Fikkert, Paula, and Helen De Hoop. 2009. Language acquisition in optimality theory. *Linguistics* 47.2:311–357.

Gnanadesikan, Amalia E. 2004. Markedness and Faithfulness Constraints in Child Phonology. In *Constraints in phonological acquisition*, ed. René Kager, Joe Pater, and Wim Zonneveld, 73–108. Cambridge: Cambridge University Press. Circulated since 1995.

Hale, Mark, and Charles Reiss. 1998. Formal and Empirical Arguments Concerning Phonological Acquisition. *Linguistic Inquiry* 29.4:656–683.

Hayes, Bruce. 2004. Phonological Acquisition in Optimality Theory: The Early Stages. In *Constraints in Phonological Acquisition*, ed. R. Kager, J. Pater, and W. Zonneveld, 158–203. Cambridge University Press.

Heinz, Jeffrey, and Jason Riggle. 2011. Learnability. In *Blackwell Companion to Phonology*, ed. Marc van Oostendorp, Colin Ewen, Beth Hume, and Keren Rice. Wiley-Blackwell.

Jesney, Karen, and Anne-Michelle Tessier. 2007. Re-evaluating learning biases in Harmonic Grammar. In *University of massachusetts occasional papers 36: Papers in theoretical and computational phonology*, ed. Michael Becker.

Jesney, Karen, and Anne-Michelle Tessier. 2008. Gradual learning and faithfulness: consequences of ranked vs. weighted constraints. In *Proceedings of NELS38*, –.

Jusczyk, P. W., A. D. Friederici, J. M. I. Wessels, V. Y. Svenkerud, and A. Jusczyk. 1993. Infants' sensitivity to the sound patterns of native language words. *Journal of Memory and Language* 32:402–420.

Jusczyk, Peter, Paul Smolensky, and Theresa Allocco. 2002. How English-learning infants respond to Markedness and Faithfulness constraints. *Language Acquisition* 10:31–73.

Kager, René. 1999. *Optimality Theory*. Cambridge University Press.

Keller, Frank, and Ash Asudeh. 2002. Probabilistic Learning Algorithms and Optimality Theory. *Linguistic Inquiry* 33.2:225–244.

Kivinen, Jyrki. 2003. Online learning of linear classifiers. In *Advanced lectures on machine learning (lnai 2600)*, ed. S. Mendelson and A.J. Smola, 235–257. Berlin Heidelberg: Springer-Verla.

Legendre, Géraldine, Yoshiro Miyata, and Paul Smolensky. 1990a. Harmonic Grammar: A formal multi-level connectionist theory of linguistic well-formedness: An application. In *Proceedings of the twelfth annual conference of the Cognitive Science Society*, 884–891.

Cambridge, MA: Lawrence Erlbaum.

Legendre, Géraldine, Yoshiro Miyata, and Paul Smolensky. 1990b. Harmonic Grammar: A formal multi-level connectionist theory of linguistic well-formedness: Theoretical foundations. In *Proceedings of the twelfth annual conference of the Cognitive Science Society*, 388–395. Cambridge, MA: Lawrence Erlbaum.

Levelt, Clara C., Niels O. Schiller, and Willem J. Levelt. 2000. "The Acquisition of Syllable Types". *Language Acquisition* 8(3):237–264.

Lombardi, Linda. 1999. Positional faithfulness and voicing assimilation in Optimality Theory. *Natural Language and Linguistic Theory* 17:267–302.

Magri, Giorgio. 2009. The acquisition of Dutch syllable types: from linear OT to standard OT. Poster presented at the LSA Annual Meeting, San Francisco.

Magri, Giorgio. 2010. Complexity of the Acquisition of Phonotactics in Optimality Theory. In *Proceedings of SIGMORPHON 11: the 11th biannual meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, ed. Jeffrey Heinz, Lynne Cahill, and Richard Wicentowski, 19–27. Uppsala, Sweden: Association for Computational Linguistics.

Magri, Giorgio. 2011a. An online model of the acquisition of phonotactics within Optimality Theory. In *Proceedings of CogSci 33: the 33rd annual conference of the Cognitive Science Society*, ed. L. Carlson, C. Hölscher, and T. Shipley. Austin, TX:: Cognitive Science Society.

Magri, Giorgio. 2011b. Complexity of the acquisition of Phonotactics in Optimality Theory. ENS manuscript. Available as ROA-1138. Accepted conditional on revisions by *Linguistic Inquiry*.

Magri, Giorgio. 2012a. Convergence of error-driven ranking algorithms: extension to the stochastic case. Manuscript.

Magri, Giorgio. 2012b. HG has no computational advantages over OT: towards a new toolkit for Computational OT. Accepted with revisions by *Linguistic Inquiry*.

Magri, Giorgio. 2012c. A note on the gla's choice of the current loser from the perspective of factorizability. Manuscript submitted to the *Journal of Logic, Language, and Information*.

Magri, Giorgio. 2012d. Restrictiveness of error-driven ranking algorithms: an initial assessment. Manuscript in progress.

McLeod, S., J. van Doorn, and V. Reed. 2001. Normal acquisition of consonant clusters. *American Journal of Speech-Language Pathology* 10:99–110.

Pater, Joe. 2008. Gradual Learning and Convergence. *Linguistic Inquiry* 39.2:334–345.

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

Pater, Joe, and Jessica A. Barlow. 2003. Constraint conflict in cluster reduction. *Journal of Child Language* 30:487–526.

Prince, Alan. 2002. Entailed Ranking Arguments. ROA 500.

Prince, Alan, and Paul Smolensky. 2004. *Optimality Theory: Constraint Interaction in Generative Grammar*. Blackwell. As Technical Report CU-CS-696-93, Department of Computer Science, University of Colorado at Boulder, and Technical Report TR-2, Rutgers Center for Cognitive Science, Rutgers University, New Brunswick, NJ, April 1993. Rutgers Optimality Archive 537 version, 2002.

Prince, Alan, and Bruce Tesar. 2004. Learning Phonotactic Distributions. In *Constraints in Phonological Acquisition*, ed. R. Kager, J. Pater, and W. Zonneveld, 245–291. Cambridge University Press.

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

Smit, A., L. Hand, J. Freilinger, and A. Bernthal, J. Bird. 1990. The Iowa Articulation Norms Project and its Nebraska replication. *Journal of Speech and Hearing Disorders* 55:779–798.

Smolensky, Paul. 1996a. On the Comprehension/Production Dilemma in Child Language. *Linguistic Inquiry* 27.4:720–731.

Smolensky, Paul. 1996b. The Initial State and Richness of the Base in Optimality Theory. John Hopkins Technical Report.

Stemberger, Joseph Paul, and Barbara Handford Bernhardt. 1999. The Emergence of Faithfulness. In *The Emergence of Language*, ed. B. MacWhinney, 417–446. Mahweh, NJ: Erlbaum.

Stemberger, Joseph Paul, and Barbara Handford Bernhardt. 2001. U-shaped learning in language acquisition, and restrictions on error correction. Poster presented at the 2001 Biennial Meeting of the Society for Research in Child Development.

Stemberger, Joseph Paul, Barbara Handford Bernhardt, and Carolyn E. Johnson. 1999. U-shaped learning in the acquisition of prosodic structure. Poster presented at the sixth International Child Language Congress.

Tesar, Bruce. 1995. "Computational Optimality Theory". Doctoral Dissertation, University of Colorado, Boulder. ROA 90.

Tesar, Bruce. 1998. Error-Driven Learning in Optimality Theory via the Efficient Computation of Optimal Forms. In *Is the Best Good Enough? Optimality and Competition in Syntax*, ed. Pilar Barbosa, Danny Fox, Paul Hagstrom, Martha McGinnis, and David Pesetsky, 421–435. Cambridge, MA: MIT Press.

Tesar, Bruce. 2004. Using inconsistency detection to overcome structural ambiguity. *Linguistic Inquiry* 35.2:219–253.

Tesar, Bruce. 2008. Output-Driven Maps. Ms., Rutgers University; ROA-956.

Tesar, Bruce, and Paul Smolensky. 1996. Learnability in Optimality Theory (long version). Technical Report 96-3, Department of Cognitive Science, Johns Hopkins University, Baltimore. Available as Rutgers Optimality Archive 156, http://ruccs.rutgers.edu/roa.html.

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: The MIT Press.

Tessier, Anne-Michelle. 2009. Frequency of Violation and Constraint-based Phonological Learning. *Lingua* 119.1:6–38.

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

Zamuner, Tania S., LouAnn Gerken, and Michael Hammond. 2005. The acquisition of phonology based on input: a closer look at the relation of cross-linguistic and child language data. *Lingua* 115(10):1329–1474.