The typological consequences of weighted constraints*

Max Bane and Jason Riggle
University of Chicago
January 25, 2010

A common 'typological criterion' on linguistic models is that they should predict (almost) all observed patterns while minimizing overgeneration. For optimization-based models, it has been argued that constraints should be ranked rather than weighted to minimize overgeneration. Recently, however, weighting has been shown to elegantly capture patterns that ranking misses. To evaluate the issue, we provide software that builds ranked/weighted-typologies. We find that some independently motivated restrictions eliminate much overgeneration but that, in general, weighting leads to numerous novel (and odd) constraint interactions.

1 Introduction

It is common to impose a typological criterion on linguistic models: that they should predict as many observed patterns and phenomena as possible while minimizing overgeneration—that is, the prediction of "unnatural" or implausible patterns whose lack of attestation is not deemed an accident. In constraint-based models, this desideratum has motivated proposed restrictions that can be divided into two kinds: (i) restrictions on the forms of constraints themselves, and (ii) restrictions on the ways they can interact. The former type of restriction (i) is the focus of work by Hayes et al. (2008), Steriade (2001), and others on characterizing the cognitive or phonetic "naturalness" of constraints, by McCarthy (2003) arguing against the use of gradiently evaluated constraints, and by McCarthy and Prince (1995) against templatic constraints. The latter concern (ii), the naturalness of constraint interactions, is also frequently invoked in assessing alternative models; for example, by Prince and Smolensky (1993/2004) in favor of ranked rather than weighted constraints, on the grounds of typologically strange weighted interactions.

In this paper, we present tools for taking a set of constraints and comparing the typological predictions that follow from ranking, as in Optimality Theory (OT; Prince and Smolensky 1993/2004), or weighting, as in Harmonic Grammar (HG; Legendre et al. 1990, Pater 2009). The typological criterion becomes especially

*Paper presented at CLS 45 as "Evaluating strict domination: The typological consequences of weighted constraints" (page #s here differ from CLS version). We would like to thank Joe Pater, Jeff Heinz, Morgan Sonderegger, and the participants of CLS 45 for useful comments and discussion.

relevant in HG because of the free range of interaction it allows—any constraints that can interact, will, under some weighting. Weighted optimization can give rise to what are called "gang" effects, where multiple violations of lower ranked constraints are worse than violation of a higher ranked constraint. This allows candidates that would be harmonically bounded in OT to be optimal in HG. We present here a second kind of gang effect that occurs across tableaux where individually viable OT candidates can be co-optimal in HG but not in OT.

(1)

/VC/	MAX (3)	ONS (4)	*COD (1)
a. VC		*	*
b. ∅	**		

/V/	MAX (3)	ONS (4)	*COD (1)
c. V		*	
d. ∅	*		

In OT, the optimality of candidate *a* entails the ranking Max $\gg$ {ONSET, *CODA} which, in turn, entails the optimality of candidate *c*. In HG, on the other hand, if optimization minimizes the weighted sum of the violations for the parenthesized weights, then the optima are *a* and *d*. We call these cross-tableaux gangs *cartels*.

The critical question is whether this cartel effect is a desirable prediction. We will return to this example in §4.1, but the short answer seems to be *no*. Under the weighting in (1), the set of syllables in surface forms is {CV, CVC, VC} but not V, which means that ONSET violations are permitted *only if* *CODA is also violated. This pattern is at odds with putative syllable-inventory universals (Jakobson 1962).

There may be other reasons to pursue grammars with weighted constraints (e.g., anti-bottleneck effects, Albright 2008), but, in terms of the typological criterion, their viability hinges on whether it is possible to choose constraints that do not overgenerate too much. If this additional restriction on constraints can be met without too much sacrifice of empirical coverage, then adopting a weighted model of constraint interaction adds to the set of tools that can be used to evaluate analyses. This is especially so if the "bad" constraints are problematic for independent reasons, as is the case with the alignment constraints discussed in §4.2. On the other hand, if massive overgeneration is unavoidable in weighted models, then we must either abandon models like HG or abandon (or modify) the typological criterion.

In §2 we review arguments for and against strict domination. In §3 we present a basic framework and tools for generating and evaluating typologies with and without strict domination. In §4 we present two case studies based on syllable structure and stress systems. Finally, in §5 we discuss our results.

2 The strict domination hypothesis

Harmonic Grammar was among several "weighted" phonological models (e.g., Goldsmith and Larson (1990), Goldsmith (1991, 1993), Larson (1992), Prince (1993)) being explored when Optimality Theory was formulated in the early 1990s. One core innovation of OT was the principle of *strict domination*, i.e., the criterion that obeying higher ranked constraints is strictly more important than obeying those ranked below them. In this section we discuss the arguments for strict domination.

2.1 Reductio ad monstrum

The adoption of strict domination was advocated on the grounds that weighted constraints make undesirable predictions. One such prediction involves *centering*. If the constraints ALIGN(**x**)-LEFT and ALIGN(**x**)-RIGHT have approximately equal weights then **x** will be centered in the form. Prince (1993:95) observes a variant of this prediction in the weighted models of Goldsmith and Larson (1990). Another prediction of weighting that has been more widely discussed is the possibility of *gangs*. In a one-versus-many battle between constrains, the many can overwhelm the one. For example, suppose that a language has stem-controlled nasal harmony generated by the constraints AGREE(NAS), IDENT(NAS)$_{stem}$, and IDENT(NAS).

(2)

/bã+dida/	AGREE(NAS)	IDENT(NAS)$_{st}$	IDENT(NAS)
a. bã.nĩ.nã			***
b. ba.di.da		*	*
c. bã.di.da	*		

Under strict domination, ranking AGREE(NAS)$\gg$ ID(NAS)$_{st}$ $\gg$ ID(NAS) generates stem-controlled harmony that applies across the board. In a weighted model, the stem-specific faithfulness constraint can eventually be overwhelmed by the general ID(NAS).[1] In (2), candidate *a* will be optimal *only if* the weight of the specific constraint is more than twice that of the general constraint.

Weighted optimization with the constraints in (2) predicts stem control as long as the number of affixes is below a critical threshold. The problem with this prediction is that, for any value of *n*, there is an HG grammar that generates stem-controlled harmony so long as there are *n* or fewer disharmonic suffix vowels. This kind of dependence on a specific (and arbitrarily high) number of occurrences of a

[1] Legendre et al. (2006) discuss a similar pattern involving stress placement.

particular kind of segment is regarded as linguistically unrealistic, contradicting the adage that *grammars do not count* (e.g., McCarthy and Prince 1986).[2]

It turns out, however, that ranked constraints in OT give rise to a very similar prediction that Bakovic (2000) has dubbed *majority rule*. For instance, if the ranking of the two IDENT constraints in (2) is inverted, then the nasality of the surface form will depend on whether nasals or non-nasals are more prevalent in the underlying form. This leads to precisely the kind of counting that grammars are supposed not to do. Models of harmony have been proposed that avoid majority rule predictions, but the fact remains that optimization itself can introduce counting regardless of whether the constraints are weighted or ranked.

A similar story can be told regarding the centering predictions. Eisner (1997) points out that Generalized Alignment (McCarthy and Prince 1993) constraints are formally unusual in that the number of violations can mount quadratically with the length of the form (e.g., ALL-FT-LEFT assigns $[\sigma(\sigma\sigma)(\sigma\sigma)(\sigma\sigma)(\sigma\sigma)]_{wd}$ 1+3+5+7=16 violations). Eisner notes that, unlike most constraints proposed in the literature, these cannot be represented with finite-state automata[3] and, most critically, that they can do centering! For example, ALIGN-σ-L-H-L will align the left edge of a floating high tone with the left edge of *all* syllables and thus $[\sigma\sigma\acute{\sigma}\sigma\sigma]$ gets 6 violations while $[\sigma\sigma\sigma\acute{\sigma}\sigma]$ and $[\sigma\acute{\sigma}\sigma\sigma\sigma]$ get 7 violations. Again, the typologically odd predictions are not a consequence of weighting but are due to optimization and specific types of constraints. Even if alignment constraints are restricted to those that can be represented with finite-state means (see Ellison 1994) they make odd predictions due to their *gradient* nature that have led researchers to propose they be excluded from any universal constraint set (e.g., McCarthy 2003).

2.2 Motivations for weighting

A number of researchers have argued that gang-effects are, in fact, needed for the analysis of a range of cases such as cumulative markedness effects in Japanese OCP phenomena (Itô and Mester 1998), English genitive variation (Jäger and Rosenbach 2006), the processing of identical place avoidance in Dutch (Kager and Shatzman 2007), modeling acquisition (Jesney and Tessier 2008), cumulative faithfulness effects in Kikuyu nasal prefixation and Greek voicing assimilation (Farris-Trimble

[2] See (Pater et al. 2007:14-16) for a similar example presented in much greater detail.

[3] Potts and Pullum (2002) detail still more reasons that these constraints are formally unusual.

2008a,b, 2009/in press), and "anti-bottleneck" effects in gradient well-formedness judgment tasks (Albright 2007, 2008, Albright et al. 2008).[4]

These lines of research motivate a serious assessment of weighted constraint models, and how their typological predictions can be expected to diverge in general from those of ranked models. What is needed is a precise characterization of the kinds of patterns that can arise from weighting but are impossible under strict domination. Of these, we must then ask whether it is possible to formulate general principles that distinguish the typologically desirable gangs from the undesirable.

3 Contenders and typologies in OT and HG

The broad consequences of strict domination follow in a fairly obvious way from the restrictions it puts on constraint interaction. Under strict domination, typologies are finite (i.e., there are at most $k!$ languages if there are k constraints) and any degree of violation beyond a fatal violation in a candidate is irrelevant (i.e., violations cannot gang up). It is far more difficult, however, to foresee the specific consequences of strict domination for a particular set of constraints. To this end it is useful to employ algorithmic tools for generating cases where the two diverge.

In this section we describe a basic set of tools for OT and HG that make it possible to (i) select optima from a finite set of candidates, (ii) generate optima from a potentially infinite set of candidates, and (iii) generate the full typology realized by the input forms in any given lexicon.

3.1 Optima in OT and HG

The first and most basic task that we consider is the problem of taking a tableau containing a finite set of candidates and identifying those that can be optimal under some ranking or weighting of the constraints. Consider the candidates in (3):

[4] Some of these are not so easy to reanalyze by weighting the constraints. For example, loosely paraphrasing Itô and Mester (1998): 'suffixes may contain a falling tone only when the stem does not, so *HL^2 $\gg$ IDENT $\gg$ *HL'. The *conjoined* constraint *HL^2 (Smolensky 1995) is not obviated by weighting because the relevant candidates have one matched *HL violation (and thus no gangs). Potts et al. (2009) point out this contrast between weighting and conjunction and give more cases.

(3)

/VVC/	*CODA	ONSET	DEPC	DEPV	MAX	
a. V.VC	*	**				OT contenders
b. CV.CVC	*		**			OT contenders
c. ∅					***	OT contenders
d. VC	*	*			*	← OT-bounded by *a* & *c*
e. CVC	*		*		*	← OT-bounded by *b* & *c*
f. V.CVC	*	*	*			← HG-bounded by *a* & *b*
g. CVC.CVC	**		***			← simply-bounded by *b*

(Candidates *a*–*e*: HG contenders)

In order to be optimal, it is a necessary condition in both HG and OT that a candidate not have a constraint-wise superset of the violations of any other candidate.[5] In (3), candidate *g* cannot be optimal because *b* has fewer violations of *CODA and DEPC and equal violations elsewhere. Here we say that *g* is *simply-bounded* by *b*.

Samek-Lodovici and Prince (1999) distinguish candidates like *g* from those like *d*, describing the latter as *collectively-harmonically-bounded* (i.e., *d* beats *a* if ONS $\gg$ MAX, but for *d* to beat *c* it must be the case that MAX $\gg$ ONS and thus *d* cannot be optimal under any ranking). On the other hand, there *are* weightings under which *d* is optimal (e.g.,*CODA(1), ONS(4), DEPC(4), MAX(3)). We refer to candidates like *d* as OT-bounded. A subset of these are also collectively bounded in HG; we call these HG-bounded. An example is candidate *f*, which is HG-bounded by candidates *a* and *b* (i.e., to beat the former the weight of ONS must be greater than that of DEPC but, to beat the latter, the opposite must hold). We refer to the candidates that can win under some ranking of the constraints as *OT-contenders* and those that can win under some weighting as *HG-contenders*. The latter are always a superset of the former, as we will establish below in §3.3.

Given a finite set of candidates, it is easy to weed-out those that are simply-bounded by comparing each pair of candidates. This is a good first step because detecting collective bounding can be costly. OT-bounded candidates can then be flagged for removal using a variant of recursive constraint demotion (RCD; Tesar and Smolensky 1996). This is a good second step because detecting collective bounding is less costly in OT than HG. Finally, if HG-contenders are desired, some of the OT-bounded candidates can be spared if they pass a feasibility check using linear programing (see Pater et al. 2009).

[5]This criterion corresponds to the notion of *Pareto efficiency* and the set of violation profiles that are not simply bounded is the *Pareto set*. There is no limit on the size of the Pareto set if $|Con| > 1$.

In carrying out the computations we will work with *difference vectors* that represent comparisons between candidates.[6]

(4) a. For candidates a and b with violation profiles $(a_1,\ldots,a_k),(b_1,\ldots,b_k)\in\mathbb{N}^k$, $\Delta(a,b)=(b_1-a_1,\ldots,b_k-a_k)$ is the difference vector for their violations.

b. We extend this to a candidate set S and $a\in S$ as $\Delta(a,S)=\{\Delta(a,s):s\in S\}$.

The difference vector $\Delta(a,b)$ is a concise representation of the conditions under which candidate a beats b. For example, in (3), the violation profiles for candidate a and candidate e are $(1,2,0,0,0)$ and $(1,0,1,0,1)$ and thus $\Delta(a,e)=(0,-2,1,0,1)$. For OT, the meaning of $\Delta(a,e)$ is that at least one of the constraints with a positive value must outrank all the constraints with negative values (i.e., for a to be more harmonic than e, MAX or DEPC must outrank ONS). For HG, $\Delta(a,e)$ encodes an inequality over the constraint weights under which candidate a beats e. That is, $\Delta(a,e)=(0,-2,1,0,1)\Rightarrow -2\,wt(ons)+wt(depV)+wt(max)>0$ or equivalently, and a bit more transparently, a beats e iff $wt(depV)+wt(max)>2\,wt(ons)$.

Selecting the OT and HG contenders from among a finite set of candidates S can be done by following the procedure in 5.

(5) CONTENDER SELECTION PROCEDURE (CSP)

1. Given a candidate set S, for each candidate $x\in S$: generate $\Delta(x,S)$
2. If $\Delta(x,S)$ contains a difference vector that has a negative value but has no positive values, remove x from S (x is simply bounded). Otherwise:
3. a. Let $D=\Delta(x,S)$:
   b. If any $(v_1,\ldots,v_k)\in D$ have a positive value $v_i>0$ where all other $v'\in V$ have non-negative values $v'_i\geq 0$, remove them from D and repeat step b.
   c. If D contains any vector with a negative value, flag x as OT-bounded.
4. To obtain OT-contenders, remove all flagged candidates. To obtain the HG-contenders, remove any flagged x for which there is no weighting that simultaneously satisfies all the inequalities implied by $\Delta(x,S)$.[7]

An implementation of CSP is available in PYPHON (`clml.uchicago.edu/pyphon`), a Python-based library of OT/HG tools. See PYPHON documentation for details.

[6]These are like Prince's (2002) Elementary Ranking Conditions—or the rows of comparative tableaux—but they preserve the difference magnitudes which are relevant in weighted optimization.

[7]There are several ways to do this but space does not permit a review here. See Potts et al. (2009) for an application of the simplex algorithm to this problem in HG.

3.2 Candidate generation

Using only insertion and deletion, it is possible to map any given underlying form to *every* conceivable surface form. Nonetheless, even though an input like /VVC/ admits an infinite range of candidates, just 17 of them simply-bound all the rest. Of these, 8 are OT-contenders, another 2 are HG-contenders, and the other 7 are collectively bounded in both OT and HG; tableau (3) is a subset of these candidates.

Though the CSP in (5) will identify the contenders among any *finite* set of violation profiles, the critical question is how to deal with infinite sets of candidates. This is actually a relatively minor variation on the issue that lies at the heart of most optimization problems and it can be overcome with techniques generally referred to as *dynamic programming*. The basic idea is to take a problem that admits an infinite (or unmanageably large) space of solutions and factor it into sub-problems whose solutions can then be combined to yield the solution to the whole problem. Optimization problems that can be factored in this way can then be solved directly.[8]

Early in the emergence of OT, Ellison (1994) and Tesar (1995) showed that optimization could be efficiently computed using dynamic programing strategies. Riggle (2004) proposed an algorithm for generating contenders based on Ellison's finite-state approach and, more recently, Riggle (2009) gives a different finite-state contender generation strategy whose complexity, modulo the number of contenders generated, is linear in the length of the underlying form with a multiplicative constant representing the size of the finite-state representation of the grammar.

In this work we extend Riggle's (2009) algorithm to generate HG contenders by applying CSP in (5) when iteratively selecting candidate fragments. This procedure is implemented in PyPhon along with a set of functions for constructing finite-state constraints from simple regular expressions describing marked structures and unfaithful mappings to be penalized (see PyPhon documentation for details).

Contender generation can be extended beyond finite-state constraints to those representable with context-free expressions using chart-parsing strategies such as the one presented by Goodman (1998) for generating the *n*-most-likely parses in probabilistic context-free grammars (see Riggle 2009:27).

[8]This is contrary to the oddly persistent misconception among some of OT's detractors that optimization entails exhaustive generation of every possible candidate (see, e.g., Calabrese 2005:2).

3.3 Recursive typology generation

HG contenders are always a superset of those in OT. This is so because, for k constraints, a tableau contains at most $k!$ contenders with distinct violation profiles. Thus, for any tableau T of OT contenders there is a finite number n that is the sum of the violation-counts in the distinct violation profiles in T. Given n, it is possible to devise a weighting that selects the same optima as any given ranking (see, e.g., Prince 2007). More generally, for any finite set of input forms U and any ranking R, there is a weighting that simulates R based on the highest n in any tableau for a form in U. Conversely, for *all* input forms, simulating a ranking is usually impossible.[9]

To generate a typology we assume that a function $\text{C}{\small\text{AND}}(i)$ provides a set of candidates for any underlying form. Ideally, $\text{C}{\small\text{AND}}(i)$ should return contenders (as in §3.2), but hand-crafted candidate sets can be used as well (*a la* OT-Soft; Hayes et al. 2003 or OT-Help; Becker et al. 2007). In constructing typologies, we will work with sets of difference vectors that (partially) define OT/HG grammars. If a set of vectors V can pass step 3 of the CSP in (5) we say that it is *consistent* and, if V can pass step 4, we say that it is *feasible*. We take a typology $\mathscr{T}$ to be a set of pairs (D,L) where D is a set of difference vectors and L is a language comprising a set of (i,k) pairs where i is an underlying form and $k=(v,S)$ is the candidate (violation-profile, surface-form-set) that is optimal given D. A typology can be constructed for a given set *Lex* via (6):

(6) RECURSIVE TYPOLOGY CONSTRUCTION (RTC)

a. The 'base' typology $\mathscr{T}$ is $\{(D_0,\emptyset)\}$.
b. For each underlying form i in the set *Lex*:
c. for each $(D,L)\in\mathscr{T}$: remove (D,L) from $\mathscr{T}$,
d. then, for each candidate $k\in K=\text{C}{\small\text{AND}}(i)$:
e. if $\hat{D}=\Delta(k,K)\cup D$ is consistent (step 3 of CSP)
or if generating for HG, if $\hat{D}$ is feasible (step 4 of CSP)
f. add $\hat{L}=L\cup\{(i,k)\}$ to $\mathscr{T}$: $\mathscr{T}=\mathscr{T}\cup\{(\hat{D},\hat{L})\}$.

RTC builds typologies using a kind of breadth-first search that is commonly used in constraint satisfaction problems. This is essentially the same as the approach of OT-Soft and OT-Help but RTC is more flexible in two respects. First, the argument

[9] In (2), for any weighting of $\text{I}{\small\text{D}}(\text{NAS})_{stem}$ and $\text{I}{\small\text{D}}(\text{NAS})$ there is an input form (possibly quite long) that allows the latter to overwhelm the former and thus ranking cannot be simulated.

D_0 can specify initial conditions (e.g., meta-rankings or preconditions on weights) and second, the CSP algorithm in step (e) makes it possible to generate both OT and HG typologies—simultaneously if need be. In generating HG typologies, CSP uses the simplex algorithm to check feasibility (as in Becker et al. 2007 and Potts et al. 2009), but it does so only when the consistency test fails. This is useful because consistency entails feasibility and checking the former is much less expensive.

The overall complexity of this procedure depends on the number of contenders for each input and the number of points in the typology which, in the case of OT, are both bounded at $k!$ for k constraints. The RTC algorithm is implemented in PyPhon (see the documentation for more details).

4 Case studies in OT/HG typology

In this section we present two case studies in applying the tools developed above to compute and compare the typological predictions of OT and HG. We first look at two versions of Prince and Smolensky's (1993/2004) model of syllabifying consonant-vowel (CV) sequences (section 4.1), and then consider Gordon's (2002) footless, alignment-based model of quantity-insensitive (QI) stress (section 4.2). In both cases, the typological "lexicon" of possible underlying forms is fixed, so that the set of HG languages must properly include the OT languages, the difference following from gang effects (both within and across tableaux) that are possible in HG but not in OT. HG overgenerates typologically in both cases, but moreso by several orders of magnitude in the QI stress model; we attribute this difference to the use of gradient alignment constraints in the latter model.

4.1 Syllable structure

Here we consider the basic model of syllable structure that Prince and Smolensky (1993/2004:Part II) present to account for Jakobson's (1962) typology of syllables. Jakobson observed that while some languages disallow onsetless and/or closed syllables, no language bans onsets or open syllables. Thus, languages are described by two independent choices: whether to require onsets and whether to forbid codas. Many logically possible languages are unobserved, such as, for instance, any that ban onsets and require codas (VC).

The basic version of the model makes the simplifying assumption that the segments to be syllabified comprise just two types: consonants (C), which must fill an onset or coda position, and vowels (V), which must be syllable nuclei. The model then consists of the five violable constraints summarized in Table 1, whose range of $5! = 120$ rankings are sufficient to capture Jakobson's generalization.

Input alphabet:	Σ	=	$\{C, V\}$			
Lexicon:	L	=	$\{w \in \Sigma^* : 1 \leq \lvert w \rvert \leq 3\} = \{C,V\}^3$			
Output alphabet:	Ω	=	$\{C, V, .\}$			
Possible mappings:	GEN	=	$\{C \mapsto C, V \mapsto V, C \mapsto \varepsilon, V \mapsto \varepsilon,$ $\varepsilon \mapsto C, \varepsilon \mapsto V, \varepsilon \mapsto .\}$			
Output filter:	FILTER_5	=	$.((C\lvert\varepsilon)V(C\lvert\varepsilon).)^*$			
	FILTER_9	=	$.((C\lvert\varepsilon)(C\lvert\varepsilon)(V\lvert\varepsilon)V(C\lvert\varepsilon)(C\lvert\varepsilon).)^*$			
Violable constraints:	CON_5	=	{ONSET, *CODA, MAX, DEP-V, DEP-C}			
	CON_9	=	{ONSET, *CODA, MAX-V, MAX-C, DEP-V, DEP-C, *CXONSET, *CXCODA, *VV}			
Penalized structures:	ONSET	=	.V	MAX-V	=	$V \mapsto \varepsilon$
	*CODA	=	C.	MAX-C	=	$C \mapsto \varepsilon$
	MAX	=	$C \mapsto \varepsilon \lvert V \mapsto \varepsilon$	*CXONSET	=	.CC
	DEP-V	=	$\varepsilon \mapsto V$	*CXCODA	=	CC.
	DEP-C	=	$\varepsilon \mapsto C$	*VV	=	VV

Table 1: Parameters of the five- and ten-constraint syllable structure models.

The assumption that consonants may only appear in onset/coda position and that vowels may only be nuclei can be enforced via an additional set of constraints {NUC, *COMPLEX, *M/V, *P/C} held fixed at the top of the constraint hierarchy (Prince and Smolensky 1993/2004:96). Instead, we simplify things with a regular expression "filter" in (7) that requires every surface form to consist of zero or more syllables, delimited by '.' on either side with one nuclear V and a single optional coda or onset C. This is equivalent to having a single undominated (or infinitely weighted) constraint that penalizes structures matching the complement of FILTER_5.

(7) $\text{FILTER}_5 = .((C\lvert\varepsilon)V(C\lvert\varepsilon).)^*$ where ε is the empty string.

We assume that input forms are strings over the alphabet $\Sigma = \{C, V\}$. These are mapped, by the closure of the operations in GEN (in this case insertion and deletion), to an infinite set of strings over the output alphabet $\Omega = \{C, V, .\}$. Candidates that do not satisfy the filter are omitted, and among the OT/HG contenders optima are selected according to a ranking or weighting. For this study, we take all sequences of 1-to-3 consonants and vowels to be the set of input forms across languages. (By fixing the lexicon, we focus only on the way that the grammars shape typology.) Under strict domination, the 5-constraint model yields a typology of 12 languages summarized in Table 2.

		Onset required		Onset optional
		Onset Max Dep-C	Onset Dep-C Max	Max Dep-C Onset
Coda banned	*Coda Max Dep-V	**1.** CV ep. ep.	**3.** CV del. ep.	**5.** (C)V ep.
	*Coda Dep-V Max	**2.** CV ep. del.	**4.** CV del. del.	**6.** (C)V del.
Coda allowed	Max Dep-V *Coda	**7.** CV(C) ep. [*ep*]	**9.** CV(C) del. [*ep*]	**11.** (C)V(C) [*ep*]
	Dep-V Max *Coda	**8.** CV(C) ep. [*del*]	**10.** CV(C) del. [*del*]	**12.** (C)V(C) [*del*]

Table 2: The OT typology of the 5-constraint CV model. Each numbered cell represents the licit syllable types of one language, with parentheses indicating optionality. The notations “ep.” (epenthesis) and “del.” (deletion) in the lower left / right corners refer to the repair mechanisms for ONSET and *CODA violations respectively. Bracketed “[*ep*]” or “[*del*]” indicate the mechanisms that repair complex codas in languages that tolerate codas.

Jakobson’s four systems are predicted: syllables are restricted to [.CV.] in languages 1–4, to {[.V.], [.CV.]} in 5 & 6, to {[.CV.], [.CVC.]} in 7–10, and to {[.V.], [.CV.], [.VC], [.CVC.]} in languages 11 & 12. There are 12 languages, rather than 4, because the languages are mappings from inputs to outputs, so that two languages may have the same extension of possible outputs, while deriving those outputs from different choices of inputs by applying different “repair strategies” to obtain the licit outputs. For example, languages 1 and 2 both admit only [.CV.] syllables, but they differ in how they achieve this result for inputs containing post-vocalic consonants. Both avoid codas, but language 1 does so by epenthesizing vowels (e.g., /CVC/↦[.CV.CV.]), while 2 deletes consonants (/CVC/↦[.CV.]), the difference determined by the ranking of DEP-V and MAX.

By contrast, in HG, the same model (with the same lexicon, constraints etc.) yields a weighted typology of 23 languages. The HG typology necessarily includes the 12 languages of the OT typology (see §3.3), leaving 11 new languages made possible by weighted constraint interaction. We distinguish two non-exclusive avenues through which new HG-languages can enter the typology:

(8) Inputs for which HG-contenders are a proper superset of the OT-contenders (i.e., because the former include forms that are collectively bounded in OT);

(9) Sets of inputs for which weighting allows choices of co-optimal outputs that are not possible under any ranking (which may all be OT-contenders as in 1).

In (8), we have tableaux that contain candidates that are collectively bounded in OT but not HG; these are the source of differences between OT and HG that are most familiar (i.e., a gang effect within a single tableau allows an output not possible under strict domination). In (9) we have a set of *n* tableaux, in which it is possible to select a pattern of *n* winners via weighting that cannot be co-optimal under strict domination—we call this a *cartel* effect. Note that cartels can consist entirely of OT-contenders that are combined in a way that is possible only under weighting.

The mechanism behind (9) is similar to the collective bounding of (8), but instead of involving a single candidate that cannot be optimal under any ranking, it involves a pair (or more) of candidates in distinct tableaux that cannot be co-optimal under any ranking. In both cases, the set of optima—a singleton set in (8)—yields a set of difference vectors that is feasible but inconsistent.

Both (8) and (9) occur in the typology. Of the 11 new HG languages, all map at least one input to an output that is harmonically bounded in OT, and all exhibit combinations of OT-contenders that cannot not simultaneously be optimal in OT. In this case, cartel effects seem to contribute most to the novel points in the HG typology. In our lexicon $L = \{C,V\}^3$, only two input forms yield HG-contenders that are not also OT-contenders: /VVC/ and /CCC/. The new mappings and their incidence in the eleven novel HG languages is given in Table 3.

Most of the languages permit just one of these novel mappings, and four allow two of them. Though each of the 11 OT-impossible languages includes one of these mappings, they are not crucial to the HG typology, in the following sense: if the two inputs /VVC/ and /CCC/ are removed from the lexicon, and the HG typology is recomputed, the same 23 languages result, modulo the absence of those inputs.

Mapping(s)	Languages
/VVC/ $\mapsto$ [.CVC.]	2
/VVC/ $\mapsto$ [.VC.]	2
/CCC/ $\mapsto$ [.CVC.]	3
/VVC/ $\mapsto$ [.CVC.], /CCC/ $\mapsto$ [.CVC.]	2
/VVC/ $\mapsto$ [.VC.], /CCC/ $\mapsto$ [.CVC.]	2

Table 3: OT-impossible mappings in the HG typology.

Thus the typological divergence of HG from OT can be supported entirely by the action of new cross-tableaux patterns introduced by weighting.

All the cross-tableaux gangs involve asymmetric trade-offs between markedness and faithfulness constraints. One of the most relevant constraints is MAX, which plays a role in every gang effect. One way of seeing this is to remove MAX from the model and consider the resulting typologies. Removing MAX from CON also necessitates removing the two mappings that it penalizes ($C \mapsto \varepsilon, V \mapsto \varepsilon$) from GEN (otherwise [$\varepsilon$] would be optimal for every input). This gives us a model in which epenthesis is the only strategy available for repairing markedness violations. The OT typology for this model contains just four languages, one for each of the Jakobson types, with epenthesis as the only repair and the HG typology consists of exactly the same languages. This shows that all of the points of difference between OT and HG crucially depended on gangs involving MAX.

Removing DEP from the model by the same procedure reveals that MAX is not the only faithfulness constraint participating in the gangs. After removing DEP-V and DEP-C from CON (and $\varepsilon \mapsto C, \varepsilon \mapsto V$, from GEN), the OT typology again recapitulates the Jakobson typology with just four languages (this time using deletion as the repair), but in this case the HG typology yields just one additional language, for a total of five. The fact that most of the points of difference vanish in the absence of DEP implies that DEP and MAX work together in most gangs, but the fact that not all vanish shows that, in at least one of the novel HG languages, there is a gang in which MAX is the only faithfulness constraint.

Regarding overgeneration, we observe that though all 11 of the new languages introduced by HG in the 5-constraint model admit co-optimal forms that are not possible in OT, only four of them violate Jakobson's generalization; and furthermore, all four violate it in the same way, by allowing only {CV, CVC, VC} syllables (as in (1)). Thus, in terms of the Jakobson typology, we have just one unattested type predicted by weighting. The HG languages may also harbor unattested / undesirable patterns of repair strategies, but this is much harder to evaluate.

For the 5-constraint model, weighted optimization roughly doubles the number of languages (12 languages in OT, 23 in HG). To get a sense of how the typological differences grow with richer constraint sets, we consider the possibility of complex nuclei and margins using the 9-constraint model summarized in Table 1 as CON_9 and FILTER_9 (which permits 0–2 consonants in onsets/codas, and 1–2 nuclear vowels). In this case, the OT-typology contains 136 languages and the HG-typology contains 632 languages. The divergence can be illustrated with the two input forms in (10):

(10)

/VCVV/	*VV	Ons	DpC	MxV
a. CV.CVV	*		*	
b. CVV	*			*
c. V.CVV	*	*		

/VVCV/	*VV	Ons	DpC	MxV
x. CVV.CV	*		*	
y. CV				**
z. VV.CV	*	*		

In OT, there are rankings that make *a* & *x*, *b* & *y*, and *c* & *z* co-optimal. In HG, two additional languages result from weightings that make *b* & *x* and *b* & *z* co-optimal.

Whether this exercise implies that HG is hopelessly overgenerative is far from clear since one might reasonably expect HG to necessitate a different constraint set than OT (indeed, see §4.2 below). However, if we repeat our constraint-removal experiment, we find some cause for concern. If the MAX constraints are removed from the 9-constraint model the OT-typology contains 26 languages and HG 28. If DEP constraints are removed, the OT-typology contains 28 languages HG 56. Thus we again have the great majority of the difference between HG and OT being driven by MAX-gangs and (to a slightly lesser extent) DEP-gangs. Since these are rather fundamental constraints, and the markedness constraints with which they interact are so simple in this model, it is difficult to imagine how exactly one might formulate constraint sets that will restrain HG's prolific generative capacity.

4.2 Quantity-insensitive stress and infinite alignment gangs

This prolific capacity becomes most striking whenever trade-offs arise between a single violation of one constraint and arbitrarily many violations of another because sufficient numbers of violations of low-weight constraints can always overwhelm a high-weight constraint in HG. This allows 'one-vs-arbitrarily-many' gang effects to produce a range of bizarre typological predictions. For a brief illustration, we consider Gordon's (2002) OT model of quantity-insensitive (QI) stress systems.

Gordon's model uses 12 constraints stated in terms of a metrical grid, without reference to feet (i.e., over $\{\sigma, \grave{\sigma}, \acute{\sigma}\}$-sequences, where σ are syllables with no stress, $\acute{\sigma}$ primary stress, and $\grave{\sigma}$ secondary stress). The model is summarized in Table 4. Note that both *linear* and *quadratic* alignment constraints are employed:

(11) a ALIGN($\acute{\sigma}$, L/R): assigns one violation to each secondary stressed syllable that intervenes between primary stress and the left/right word edge.

b ALIGN($\{\grave{\sigma}, \acute{\sigma}\}$, L/R): assigns each stressed syllable violations equal to the number of syllables that intervene between it and the L/R word edge.

Constraints like (11a) are only linearly gradient because each word has exactly one primary stress (as ensured by the FILTER in Table 4). The constraints in (11b), however, assess quadratically many violations as a function of input length.

Input alphabet:	$\Sigma = \{\sigma\}$
Lexicon:	$L = \{w \in \Sigma^* : 1 \leq \|w\| \leq 8\}$
Output alphabet:	$\Omega = \{\sigma, \acute{\sigma}, \grave{\sigma}\}$
Possible mappings:	GEN = $\{\sigma \mapsto \sigma, \sigma \mapsto \acute{\sigma}, \sigma \mapsto \grave{\sigma}\}$
Inviolable filter:	FILTER = $(\sigma\|\grave{\sigma})^* \acute{\sigma} (\sigma\|\grave{\sigma})^*$
Violable constraints:	CON = {ALIGN($\acute{\sigma}$, L/R) ALIGN($\{\grave{\sigma}, \acute{\sigma}\}$, L/R), *CLASH, ALIGNEDGE, *LAPSE, *LAPSELEFT/RIGHT, *EXTLAPSE, *EXTLAPSERIGHT, NONFINALITY}

Table 4: Parameters of Gordon's QI stress model, see Gordon (2002) for definitions.

Gordon's model implements a "meta-constraint" on rankings whereby one of the primary alignment constraints is always lowest ranked, so either ALIGN($\acute{\sigma}$, L) or ALIGN($\acute{\sigma}$, R) is "active," but never both. This ensures that primary stress always occurs to one side of the secondary stresses and does not vacillate at different word lengths. We model this with two constraint sets: CON_L with ALIGN($\acute{\sigma}$, R) excluded, and CON_R with ALIGN($\acute{\sigma}$, L) excluded. The typology is the union of the typologies for by CON_L and CON_R. Gordon's model is thus capable of representing $2 \cdot 11! = 79{,}833{,}600$ QI stress grammars in OT.

Using an extensive database of QI stress systems, Gordon developed the model so as to predict as many attested QI systems as possible but as few unattested ones. Bane and Riggle (2008) compare Gordon's model against the database of 306 languages provided by Heinz (2007), which contains Gordon's database as a subset (accessible at `http://phonology.cogsci.udel.edu/dbs/stress/`). In OT, the model predicts a factorial typology of 152 distinct possible stress systems, which includes all but two of the 26 distinct attested systems in the database plus 128 unattested patterns. Some overgeneration is to be expected of any model, since the majority of distinct attested systems are extremely rare; thus failure to observe a pattern in a limited sample is not strong evidence that the pattern is impossible. Gordon's OT model of QI stress offers a comparatively "tight" typological fit.

In HG, by contrast, when the same constraints are weighted rather the ranked, the predicted typology explodes into 36,846 distinct possible stress systems (for words up to 8 syllables; for arbitrary lengths, the HG typology is infinite). Most of the difference is due to alignment-gang languages that are quite bizarre: some permit otherwise ordinary-looking stress patterns to "peter out" at a critical length, others have stress that vacillates between edges at different lengths, and still others center stress in the word. Some researchers (e.g., Eisner 1997, McCarthy 2003) have argued against these kinds of alignment constraints on unrelated grounds. Thus, one might deem it favorable that HG requires a restriction on constraints that converges with independent restrictions from other constraint-based analyses.

5 Discussion

There seem to be empirical phenomena that require something beyond optimization with strict domination (§2.2). Among options such as positing new constraints for each case, locally-conjoining constraints (Smolensky 1995), and using weighted constraints, HG stands out as particularly elegant by accounting for the phenomena via the model's basic operation: weighted optimization. Moreover, this mechanism can be evaluated in terms of its behavior elsewhere. This paper describes tools that allow us to begin this evaluation. Our preliminary findings are mixed: on the one hand, many infinite gangs are easily fixed by ditching alignment constraints (and this converges with other work); on the other hand, even simple constraint sets expand into huge HG-typologies suggesting something is needed to rein them in.

Perhaps it is possible to choose the 'right' HG constraints. The tools we describe can undoubtedly help in this strategy. We note, however, that the enterprise may be doomed by the ways that even very basic constraints yield novel interactions under weighting. Another tack would be to identify properties unique to 'bad' gangs. The infinite gangs have vanishingly small volume in the weighting-space and might be eliminated by conditions on the 'margin' of weightings as linear classifiers. Another strategy might be to restrict the action of faith-gangs by segregating markedness and faithfulness (e.g., strictly ranked strata with weighted violations within each stratum plus a restriction against M and F constraints in a stratum). These options and many others should be explored, *but* it is crucial to keep in mind that the elegance-argument for HG diminishes with the complexity of the restrictions.

References

Albright, A., 2007. Gradient phonological acceptability as a grammatical effect. Ms, MIT.

Albright, A. 2008. From clusters to words: grammatical models of nonce word acceptability. Handout from 82nd Annual Meeting of the LSA, Chicago.

Albright, A., G. Magri, and J. Michaels. 2008. Modeling doubly marked lags with a split additive model. In *BUCLD 32: Proceedings of the 32nd annual Boston University Conference on Language Development*, ed. by H. Chan, H. Jacob, and E. Kapia.

Bakovic, E., 2000. *Harmony, Dominance, and Control*. New Brunswick, NJ: Rutgers University dissertation.

Bane, M., and J. Riggle. 2008. Three correlates of the typological frequency of quantity-insensitive stress systems. In *Proceedings of the Tenth Workshop of the Association for Computational Linguistics' Special Interest Group in Morphology and Phonology*. ROA-966.

Becker, M., J. Pater, and C. Potts. 2007. Ot-help 1.2 software package, university of massachusetts, amherst.

Calabrese, A. 2005. *Markedness and economy in a derivational model of phonology (Studies in Generative Grammar 80)*. Mouton de Gruyter.

Eisner, J. 1997. Efficient generation in primitive optimality theory. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL)*, 313–320, Madrid.

Ellison, T. M. 1994. Phonological derivation in optimality theory. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING)*, 1007–1013, Kyoto.

Farris-Trimble, A. W., 2008a. *Cumulative faithfulness effects in phonology*. Indiana University, Bloomington dissertation.

Farris-Trimble, A. W. 2008b. Cumulative faithfulness effects: Opaque or transparent? In *IUWPL6: Phonological Opacity Effects in Optimality Theory*, ed. by A. W. Farris-Trimble and D. A. Dinnsen, 119–145, Bloomington, IN. IULC Publications.

Farris-Trimble, A. W. 2009/in press. Nothing is better than being unfaithful in multiple ways. In *Proceedings of the 44th annual meeting of the Chicago Linguistic Society*, ed. by M. Bane, J. Bueno, T. Grano, A. Grotberg, and Y. McNabb, Chicago. Chicago Linguistic Society.

Goldsmith, J. 1991. *A dynamic computational theory of accent systems*, chapter 1, 1–28. Stanford: CSLI.

Goldsmith, J. 1993. *Harmonic phonology*, 221–269. Chicago: University of Chicago Press.

Goldsmith, J., and G. Larson. 1990. *Local modeling and syllabification*, 129–142. Chicago: Chicago Linguistics Society.

Goodman, J., 1998. *Parsing Inside-Out*. Harvard: Harvard University dissertation.

Gordon, M. 2002. A factorial typology of quantity-insensitive stress. *Natural Language and Linguistic Theory* 20.491–552.

Hayes, B., B. Tesar, and K. Zuraw. 2003. OTSoft 2.3 software package: /www.linguistics.ucla.edu/people/hayes/otsoft/.

Hayes, B., K. Zuraw, P. Siptár, and Z. Londa, 2008. Natural and unnatural constraints in hungarian vowel harmony. Manuscript, submitted.

Heinz, J., 2007. *Inductive Learning of Phonotactic Patterns*. UCLA dissertation.

Itô, J., and A. Mester, 1998. Markedness and word structure: OCP effects in Japanese. Ms., ROA 255-0498.

Jäger, G., and A. Rosenbach. 2006. The winner takes it all—almost: Cumulativity in grammatical variation. *Linguistics* 44.937–971.

Jakobson, R. 1962. *Selected writings, vol 1*. The Hague: Mouton.

Jesney, K., and A.-M. Tessier. 2008. Gradual learning and faithfulness: Consequences of ranked vs. weighted constraints. In *Proceedings of the 38th Meeting of the North East Linguistics Society (NELS 38)*, ed. by M. Abdurrahman, A. Schardl, and M. Walkow, Amherst, MA. GLSA.

Kager, R., and K. Shatzman. 2007. Phonological constraints in speech processing. *Linguistics in the Netherlands* 24.99–111.

Larson, G., 1992. *Dynamic computational models and the representation of phonological information*. University of Chicago dissertation.

Legendre, G., Y. Miyata, and P. Smolensky. 1990. 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. Lawrence Erlbaum Associates.

Legendre, G., A. Sorace, and P. Smolensky. 2006. The Optimality Theory-Harmonic Grammar connection. In *The Harmonic Mind: From Neural Computation to Optimality-Theoretic Grammar*, ed. by P. Smolensky and G. Legendre, 339–402. The MIT Press.

McCarthy, J., and A. Prince, 1986. Prosodic morphology. Ms., University of Massachusetts and Brandeis.

McCarthy, J., and A. Prince. 1995. *Faithfulness and Reduplicative Identity*, 249–384. Amherst, Mass.: GLSA.

McCarthy, J. J. 2003. OT constraints are categorical. *Phonology* 20.75–138.

McCarthy, J. J., and A. Prince. 1993. Generalized alignment. In *Yearbook of Morphology*, 79–153. Springer.

Pater, J. 2009. Weighted constraints in generative linguistics. *Cognitive Science* . To appear, ROA-982.

Pater, J., R. Bhatt, and C. Potts. 2007. Linguistic optimization. Ms. UMASS Amherst. ROA 924.

Potts, C., J. Pater, K. Jensey, R. Bhatt, and M. Becker. 2009. Harmonic grammar with linear programming: From linear systems to linguistic typology. *Phonology* . ROA-984.

Potts, C., and G. Pullum. 2002. Model theory and the content of OT constraints. *Phonology* 19.361–393.

Prince, A. 1993. In defense of the number *i*: Anatomy of a linear dynamical model of linguistic generalizations. Technical Report TR-1, RuCCS.

Prince, A., 2002. Entailed ranking arguments. Ms.

Prince, A. 2007. Let the decimal system do it for you: A very simple utility function for OT. ROA 934 1207.

Prince, A., and P. Smolensky. 1993/2004. *Optimality Theory: Constraint Interaction in Generative Grammar*. MIT Press.

Riggle, J., 2004. *Generation, Recognition, and Learning in Finite State Optimality Theory*. University of California, Los Angeles dissertation.

Riggle, J. 2009. Generating contenders. ROA 1044-0809.

Samek-Lodovici, V., and A. Prince, 1999. Optima. Ms.

Smolensky, P. 1995. On the internal structure of the constraint component Con of UG. ROA 86-0000. Handout of talk presented at UCLA, ROA 86-0000.

Steriade, D., 2001. The phonology of perceptibility effects: the p-map and its consequences for constraint organization. Ms. University of California, Los Angeles and Massachusetts Institute of Technology.

Tesar, B., 1995. *Computational Optimality Theory*. University of Colorado dissertation.

Tesar, B., and P. Smolensky. 1996. Learnability in optimality theory (long version). Technical report, Deparment of Cognitive Science, Johns Hopkins University.