When the Hothead Speaks

Simulated Annealing Optimality Theory for Dutch Fast Speech

Tamás Bíró

Humanities Computing, University of Groningen

Abstract

Simulated Annealing, a wide-spread technique for combinatorial optimisation, is employed to find the optimal candidate in a candidate set, as defined in *Optimality Theory* (OT). Being a heuristic techniques, simulated annealing does not guarantee to return the correct solution, and yet, some result is always returned within a constant time. Similarly to language production, this time framework can be diminished with the cost of diminishing correctness. We demonstrate how simulated annealing can model linguistic performance, built upon a competence theory, namely, OT. After having applied simulated annealing to OT, we attempt to reproduce empirical observations on metrical stress in Dutch fast speech. Simulated annealing necessitates defining a *topology* on the candidate set, as well as an exact formulation of the constraint OUTPUT-OUTPUT CORRESPONDENCE.

1 Introduction: OT and optimisation

Optimality Theory (OT; Prince and Smolensky (1003), aka Prince and Smolensky (2004)) has been an extremely popular model in linguistics in the last decade. The architecture of an OT grammar, as shown in Figure 1, is composed of two parts. Out of the input (the underlying representation *UR*), the GEN module generates a set of candidates (*GEN*(*UR*)), each of which is evaluated by the EVAL module, and the best element is returned as the output (the surface representation *SR*).

EVAL is usually seen as a pipeline, in which the *constraints* filter out the sub-harmonic candidates. Each constraint assigns violation marks to the candidates in its input, and candidates that have more marks than some other ones are out of the game. Alternatively, EVAL can also be seen as a function assigning a harmony value to the candidates, the most harmonic of which will surface in the language. This *Harmony function* has a remarkable property: being worse on a higher ranked constraint can never be compensated by a good behaviour on a lower ranked constraint. This phenomenon, referred to as *the categorical ranking of the constraints*, or as the *Strict Domination Hypothesis*, follows from the filtering approach: whoever is filtered out at an earlier stage never comes back.

The traditional way of representing the competing candidates is to use a *tableau*, such as the one in (1). The left column contains the elements of the candidate set, that is, *GEN*(UR). For a given candidate w_i, the number of violation marks $C_j(w_i)$—in most cases a non-negative integer—assigned by constraint C_j is given, and the exclamation mark brings the attention to the point where a given candidate meets its

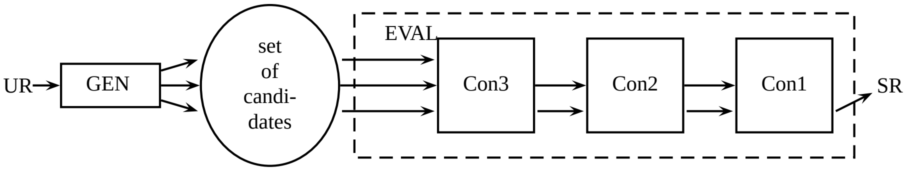

Figure 1: The basic architecture of an Optimality Theoretic grammar

Waterloo. The ☞ symbol points to the winning candidate.

(1)

/UR/	C_n	C_{n-1}	...	C_{k+1}	C_k	C_{k-1}	C_{k-2}	...
☞w_1	2	0		1	2	3	0	
w_2	2	0		1	3 !	1	2	
w_3	3 !	0		1	3	1	2	

If the constraints are functions mapping from the candidate set *GEN*(UR) to the set of non-negative integers ($\mathbb{N}_0$), then the EVAL module can be seen as an *Eval function* that assigns a vector—a *violation profile*, an (inverse) *Harmony value*, which is a shorthand for a row in a tableau—to each of the candidates:

(2) $$E(w) = \big(C_n(w), C_{n-1}(w), ..., C_1(w)\big) \in \mathbb{N}_0^n$$

together with an *optimisation algorithm*. The role of the optimisation algorithm is to find the optimal element of the candidate set, and to return it as the output (surface representation, SR, that is the grammatical form[1]):

(3) $$\mathit{SR}(\mathit{UR}) = \mathit{argmin}_{w \in \mathit{Gen}(\mathit{UR})} E(w)$$

Here, optimisation is with respect to *lexicographic ordering*, for this is the ordering realising the *categorical ranking* (strict hierarchy) of the constraints. *Lexicographic ordering* of vectors is the way words are sorted in a dictionary (e.g. *abacus*, *abolish*,..., *apple*,..., *zebra*): first compare the first element of the vectors, then, if they are the same, compare the second one, and so on. Formally speaking:

$E(w_1) > E(w_2)$, if there exists $k \in \{n, n-1, ..., 1\}$ such that

1. $C_k(w_1) > C_k(w_2)$, and
2. for all $j \in \{n, n-1, ..., 1\}$, if $j > k$ then $C_j(w_1) = C_j(w_2)$.

Constraint C_k, which determines the relative ordering of $E(w_1)$ and $E(w_2)$, will be called the *fatal constraint* (the highest ranked constraint with uncancelled marks).

[1] The form appearing in the language is not always the output of the OT grammar itself, but a trivial function F of it. For instance, parsing brackets may have to be removed. However, the inverse of the function F is not always functional, thus sometimes more outputs (parses) may describe the same observed phenomenon, posing a challenge to learning algorithms (Tesar and Smolensky 2000).

Furthermore, if $E(w_1) < E(w_2)$, than we shall say that candidate w_1 is *better* (*more harmonic*) than candidate w_2 ($w_1 \succ w_2$). A more detailed mathematical analysis is presented in Bíró (2005) and in Bíró (forthcoming)

The computational challenge posed by Optimality Theory is to realise the optimisation algorithm required by EVAL. Indeed, Eisner (2000) demonstrates that finding the optimal candidate (*generation* in OT) is OptP-complete. In addition, the candidate set is infinite in numerous linguistic models. Several solutions have been proposed, although each of them is built upon certain presuppositions, and they also require large computational resources. Finite state techniques (see references in Bíró (2003)) not only require GEN and constraints to be finite state, but work only with some further restrictions. The presuppositions of *Chart parsing* (*dynamic programming*, e.g. Tesar and Smolensky (2000), Kuhn (2000)) are more likely to be met by most linguistic models, yet it also makes use of a relatively large memory.

If our goal is, however, to find an optimisation technique which is cognitively adequate, we do not need an exact algorithm. Indeed, speech contains frequently performance errors.

The optimisation algorithm should, under normal conditions, find the "correct", i.e. the grammatical output—the optimal element of the candidate set—with high probability. Even more, the output is returned in constant time, since the partner in a conversation is not a computer user watching the sandglass. Further, human speakers sometimes speed up the computational algorithm, and the price paid is precision. We propose to see (some) fast speech phenomena (performance errors) as decreased precision (erroneous outputs) of the optimisation algorithm in EVAL, due to the increased speed.

This train of thought leads us straightforward to heuristic optimisation techniques, defined by Reeves (1995) as "*a technique which seeks good (i.e. near-optimal) solutions at a reasonable computational cost without being able to guarantee either feasibility or optimality, or even in many cases to state how close to optimality a particular feasible solution is.*" In the present paper, we implement Optimality Theory by using the simplest heuristic optimisation technique, *simulated annealing*.

We will see that simulated annealing meets all our criteria. Its computational requirements are minimal, compared to most other methods, and it returns a "good (i.e. near-optimal) solution" of even an NP-complete problem in limited time. This time interval can be reduced by paying on precision. In particular, by observing changes in the stress patterns in Dutch fast speech, we demonstrate how a proper competence grammar can produce correct outputs under normal conditions, but starts making human-like errors under time pressure. Thereby, we argue for the cognitive adequateness of the *Simulated Annealing Optimality Theory Algorithm* (SA-OT).

2 Simulated Annealing: a heuristic optimisation technique

Simulated annealing, also called as *Boltzmann Machines*, is a wide-spread stochastic technique for combinatorial optimisation (Kirkpatrick, Jr. and Vecchi 1983). It performs a random walk in the search space, and differs from *gradient descent* by allowing uphill moves—thereby escaping local minima—with a probability that decreases during the simulation. Only few have applied it in linguistics, for instance in parsing (Howells 1988, Kempen and Vosse 1989, Selman and Hirst 1994). Simulated annealing is also found in the pre-history of Optimality Theory (Smolensky 1986).

The idea originates in solid state physics. An interstitial defect in a crystal lattice corresponds to a local minimum in the energy E of the lattice. Although the perfect lattice would minimise the energy, the defect is stable, because any local change increases E. In order to reach the global minimum, one needs either to globally restructure the lattice within one step, or to be permitted to temporarily increase the energy of the lattice.

Heating the lattice corresponds to the second option. The lattice is allowed "to borrow" some energy, that is, to transform provisionally thermic energy into the binding energy of the lattice, thereby climbing the energy barrier separating the local minimum from the global minimum. At temperature T, the probability of a change that increases the lattice's energy by ΔE is $e^{\frac{-\Delta E}{kT}}$, where $k = 1.38 \times 10^{-23} JK^{-1}$ is Boltzmann's constant. The higher the temperature, the bigger energy jumps ΔE are allowed.

Annealing a metal means heating it to a high temperature, and then cooling it down slowly. The lower the temperature, the lower energy hills the system is able to climb; thus it gets stuck in some valley. At the end of the annealing, the system arrives at the bottom of the valley reached. With a slower cooling schedule, the likelihood of finding the valley including the global minimum is higher.

Now, the idea of *simulated annealing* is straightforward (cf. eg. Reeves 1995). We search for the state of a system minimising the quantity E (Energy or Evaluation) by performing a random walk in the search space. If the rule were to move always downhill (*gradient descent*), we would quickly get stuck in local minima. This is why we also allow moving upwards ("borrowing thermic energy") with some chance, which is higher in the beginning of the simulation, and which then diminishes.

For this purpose, a fictive "temperature" T is introduced. The random walk starts from an initial state w_0. At each step of the simulation, we randomly pick one of the neighbouring states (w') of the actual state w (cf. Fig. 2). Thus, a *topology* on the search space has to define the neighbours of a state (the *neighbourhood structure*), as well as the *a priori probability distribution* determining which neighbour to pick. Subsequently, we compare w' to w, and the random walker moves to w' with *transition probability* $P(w \rightarrow w' \mid T)$, where T is the temperature at that moment of the simulation. If $E(w)$ is the function to minimise, then:

(4) $$P(w \rightarrow w' \mid T) = \begin{cases} 1 & \text{if } E(w') \leq E(w) \\ e^{-\frac{E(w')-E(w)}{T}} & \text{if } E(w') > E(w) \end{cases}$$

Moving downhill is always possible, and moving uphill depends on the difference in E and on the actual temperature T. At the beginning of the simulation, T is assigned a high value, making any move very likely. The value of T is then decreased gradually, while even the smallest jump does not become highly improbable. When the temperature reaches its lowest value, the algorithm returns the state into which the random walker is "frozen" finally—this is a local minimum. Obviously, nothing guarantees

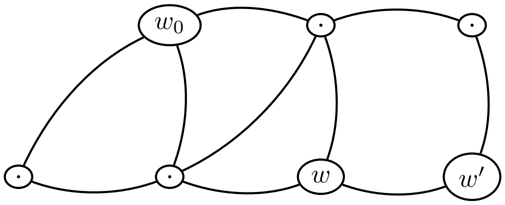

Figure 2: A schematic view of the search space—in SA-OT, the candidate set with a topology (neighbourhood structure)—in which simulated annealing realises a random walk.

finding the global minimum, but the slower the *cooling schedule* (the more iterations performed), the higher the probability of finding it.

3 Simulated Annealing for Optimality Theory

How to implement simulated annealing to Optimality Theory? The search space is the candidate set, defined by standard Optimality Theory. Yet, a *neighbourhood structure* (a *topology*) should be added to it (Fig. 2). which determines the picking of the next candidate w'. We propose to consider two candidates neighbours if they differ only minimally, that is, if a *basic operation* transforms one into the other. The algorithm gets stuck in local optima, candidates better than their neighbours. Thus, the definition of the topology influences crucially which candidates are returned besides the global optimum; these forms will be predicted to be the performance errors or the fast speech forms.

Enriching a model with further concepts—adding a topology to standard OT—could diminish the strength of a model. Yet, we have here a larger set of observations: not only the grammatical forms, but also speech errors and their frequencies. Standard OT predicts the grammatical form to be the globally optimal candidate, whereas the neighbourhood structure added to it accounts for performance errors. It is in a very non-trivial way that the interaction of the topology, the constraint hierarchy and the cooling schedule determines which local optimum is returned with what probability. Consequently, finding a simple, convincing—non *ad hoc*—topology reasonably accounting for the observed data is not a self-evident task.

If the topology determines the horizontal structure of the landscape in which the random walker roves, then the Harmony function to be optimised contributes its vertical structure. Here again, traditional Optimality Theory provides only the first part of the story. The transition probability $P(w \to w' \mid T) = 1$, if w' is better than w ($w' \succ w$, that is, $E(w') < E(w)$). But how to define the transition probability to a worse candidate, in function of the actual temperature T?

We begin by understanding the meaning of “temperature” in simulated annealing. According to equation (4), T defines the range of $E(w') - E(w)$ above which uphill moves are prohibited ($P(w \to w' \mid T) \approx 0$, if $E(w') - E(w) \gg T$), and below which they are allowed ($P(w \to w' \mid T) \approx 1$, if $E(w') - E(w) \ll T$).

In turn, our agenda is the following: first we define the difference of two violation profiles ($E(w') - E(w)$), then define temperature in an analogous way, and last adjust the definition (4).

The difference of two violation profiles seen as vectors (cf. (2)) is simply:

(5) $$E(w') - E(w) = (C_n(w') - C_n(w), ..., C_1(w') - C_1(w))$$

Yet, what interests us when comparing two candidates is only the fatal constraint (the highest ranked constraint with uncancelled violation marks). The general structure of Optimality Theory teaches us to neglect the lower ranked constraints.[2] Therefore, we define the *magnitude* of any vector $(a_n, ..., a_1)$ as

> $\|(a_n, ..., a_1)\| = \langle k, a_k \rangle$, where k is the lowest element of $\{n, ..., 1\}$ such that $\forall j \in \{n, ..., 1\}$: if $j > k$, then $a_j = 0$.
> Moreover, $\|(0, 0, ..., 0)\| = \langle 0, 0 \rangle$.

We shall use not the difference (5), but rather the magnitude of the difference of the violation profiles, $\|E(w') - E(w)\| = \langle k, C_k(w') - C_k(w) \rangle$ in simulated annealing. Take the following tableau to exemplify this idea:

	C_n	C_{n-1}	...	C_{k+1}	C_k	C_{k-1}	C_{k-2}	...
$E(w')$	2	0		1	2	3	0	
$E(w)$	2	0		1	3	1	2	
$E(w') - E(w)$	0	0		0	-1	2	-2	

Here, $\|E(w') - E(w)\| = \langle k, -1 \rangle$, since C_k is the fatal constraint, the highest constraint with uncancelled marks. We may ignore constraints ranked below C_k.

In short, the difference (5) of two violation profiles could be reduced from an n-tuple (n-dimensional vector) to a pair $\langle k, C_k(w') - C_k(w) \rangle$. The Strict Domination Hypothesis does not allow reducing it to a single real number, however (Bíró forthcoming).

In the next step, we introduce temperature. As explained, the role of temperature in simulated annealing is to gradually decrease the transition probability to a worse state. Initially, we want to allow all transitions; then prohibit transitions increasing the violation level of highly ranked constraints; then also prohibit the transitions that would increase the violation marks assigned by lower ranked constraints only, and so forth. Finally, the random walker can only move to neighbours that are not worse.

At each moment, uphill jumps much larger than T have a very low probability, and jumps much smaller than T are extremely likely. By equation (4), T *is equal to* the increase in E that has a likelihood of $1/e$. As the increase in E has been now defined as a pair, so will have to be the temperature T: a pair $\langle K_T, t \rangle \in \mathbb{Z} \times \mathbb{R}^+$.

The first element K_T of the pair is an integer, to be called the *domain* of the temperature. The second element t must be a positive real number. If C_K is an existing constraint, then $T = \langle K, t \rangle$ can be interpreted as if the violation level of

[2] For a more detailed analysis of this definition, see Bíró (forthcoming) and Bíró (2005).

constraint C_K were increased by t. Nonetheless, the domain of the temperature can be different from the indices of existing constraints.

Finally, we define the transition probability $P(w \rightarrow w' \mid T)$. As the rule is to assign a higher index to a higher ranked constraint, the first component of T places temperature somewhere relative to the constraint hierarchy. Lexicographic ordering compares adequately some $\|E(w') - E(w)\| = \langle k, C_k(w') - C_k(w)\rangle$ to $T = \langle K, t\rangle$. This is why the following definition reproduces equation (4):[3]

At temperature $T = \langle K_T, t\rangle$, if $\|E(w') - E(w)\| = \langle k, d\rangle$:

(6)
$$P(w \rightarrow w') = \begin{cases} 1 & \text{if } d \leq 0 \\ 1 & \text{if } d > 0 \text{ and } k < K_T \\ e^{-d/t} & \text{if } d > 0 \text{ and } k = K_T \\ 0 & \text{if } d > 0 \text{ and } k > K_T \end{cases}$$

This corresponds to the following *rules of transition*:

- If w' is better than w: move $w \rightarrow w'$!
- If w' loses due to fatal constraint $C_k > K_T$: don't move!
- If w' loses due to fatal constraint $C_k < K_T$: move!
- If w' loses due to the constraint $C_k = K_T$: move with probability $e^{-d/t}$.

In the beginning of the simulation, the domain K_T of the temperature will be higher than the index of the highest ranked constraint; similarly, at the end of the simulation, temperature will drop below the lowest ranked constraint. The most straightforward way to proceed is to use a double loop diminishing temperature.

The pseudo-code of *Optimality Theory Simulated Annealing* (OT-SA) can be presented finally (Fig. 3). The parameters of the algorithm are the initial candidate (w_0) from which the simulation is launched, as well as the parameters of the cooling schedule: K_{max}, K_{min}, K_{step}, t_{max}, t_{min}, t_{step}.

Typically, K_{max} is higher than the index of the highest ranked constraint, in order to introduce an initial phase to the simulation when the random walker may rove unhindered in the search space, and increase even the violation marks assigned by the highest ranked constraint. Similarly, the role of K_{min} is to define the length of the final phase of the simulation. By having K_{min} (much) below the domain (the index) of the lowest ranked constraint, the system is given enough time to "relax", to reach the closest local optimum, that is the bottom of the valley in which the system is stuck. Without such a final phase, the system will return any candidate, not only local optima, yielding an uninteresting model.

[3] As noted by an anonymous reviewer, a major difference between classical SA and SA-OT is that by equation (4), any increase in E has a small theoretical chance of being accepted in classical SA. Yet, SA-OT minimises not a real valued function, but a vector valued function for lexicographical order, due to the Strict Domination Hypothesis. Thus, the vague statement in classical OT that "if $\Delta E \gg T$ then $P \approx 0$" can and has to be formulated here in a more exact way as "if $k > K_T$ then $P = 0$". See Bíró (forthcoming) for further differences between classical SA and SA-OT.

```
ALGORITHM: Simulated Annealing for Optimality Theory
Paramters: w_0, K_max, K_min, K_step, t_max, t_min, t_step
w <-- w_0
   for K = K_max to K_min step K_step
        for t = t_max to t_min step t_step
             choose random w’ in neighbourhood(w)
             calculate < C , d >  = ||E(w’)-E(w)||
             if d <= 0 then w <-- w’
             else           w <-- w’ with probability
                         P(C,d) = 1         , if C < K
                                = exp(-d/t) , if C = K
                                = 0         , if C > K
        end-for
   end-for
return w
```

Figure 3: The algorithm of *Simulated Annealing Optimality Theory* (SA-OT).

Although other options are also possible, the way we shall proceed is placing our n constraints into the domains $K = 0$, $K = 1$,..., $K = n - 1$. That is, the highest ranked constraint receives index $n - 1$, and the lowest one is associated with index 0. Furthermore, $K_{max} = n$ and $K_{step} = 1$.

The parameters t_{max}, t_{min} and t_{step} drive the inner loop of the algorithm, that is, the decreasing of the second component t of temperature $T = \langle K, t \rangle$. This component plays a role only in the expression $e^{-d/t}$, used if the temperature is in the domain of the fatal constraint. Because the neighbouring candidates w and w' typically differ only minimally—a *basic operation* transforms w into w'—, their violation profiles are also similar, thus the difference d in violating the fatal constraint is expected to be low (usually $|d| = 1, 2$). Consequently, $e^{-d/t}$ vanishes if $t \gg 3$, and so the default values used will be $t_{max} = 3$ and $t_{min} = 0$.

The most interesting parameter is t_{step}, for it is inversely proportional to the number of iterations performed (if the other parameters are kept unchanged), and thereby it directly controls the speed of the simulation, that is, its precision. Therefore, we will tune this parameter. Other parameters also may change the number of iterations performed, but their effect is more complex, so tuning t_{step} is the most straightforward way to change the number of iterations. We also could introduce a new parameter for the number of repetitions within the core of the inner cycle.

4 Dutch metrical stress

4.1 The empirical data

Schreuder and Gilbers (2004) analyse the influence of speech rate on stress assignment in Dutch, based on laboratory experiments forcing the participants to produce fast speech. For instance, in normal (slow, andante) speech, the compound word *fototoes-*

tel (‘photo camera’) is assigned a primary stress on its first syllable and a secondary stress on its third syllable (*fótotòestel*). However, in fast (allegro) speech, Schreuder and Gilbers observed a stress shift: the secondary stress moved in a number of cases from the third syllable to the fourth one.

The words used in their experiments belong to the following three groups (Types 1-3). No experiment has been performed with type 0 words. In the stress pattern of a word form or a candidate, s always refers to a syllable with a primary or secondary stress, and u refers to an unstressed syllable hereafter.

Type 0: andante: susu, allegro: suus (OO-correspondence to: su+su)

fo.to.toe.stel	’camera’

Type 1: andante: susuu, allegro: suusu (OO-correspondence to: su+suu)

stu.die.toe.la.ge	’study grant’
weg.werp.aan.ste.ker	’disposable lighter’
ka.mer.voor.zit.ter	’chairman of Parliament’

Type 2: andante: usus allegro: suus (OO-correspondence to: usu+s)

per.fec.tio.nist	’perfectionist’
a.me.ri.kaan	’American’
pi.ra.te.rij	’piracy’

Type 3: andante: ssus allegro: suus (OO-correspondence to: s+su+s)

uit.ge.ve.rij	’publisher’
zuid.a.fri.kaans	’South African’
schier.mon.nik.oog	name of in island

In slow (andante) speech, these words are pronounced in a way reflecting their inner structure. Types 0, 1 and 3 are compound words, and they keep the stress pattern of their components unchanged (e.g.: fóto+tòestel or stúdie+tòelage). Additionally, most of the examples in types 2 and 3 end in a suffix that must bear stress. Standard literature on OT phonology uses constraint Output-Output Correspondence to account for these morphologically based phenomena, as we shall explain it soon.

On the other hand, the fast speech (allegro) forms all display the suus pattern, (followed by an unstressed syllable in the five-syllable words of Type 1). This pattern matches best the markedness constraints, reflecting what the easiest is to pronounce. The markedness constraints used in the analysis advanced by Schreuder and Gilbers (2004) originate from the literature on metrical stress, supposing that parts of the syllables are parsed into *metrical feet*. These constraints are Foot Repulsion ($*\Sigma\Sigma$) punishing adjacent feet without an intervening unparsed syllable, as well as Parse-σ, punishing unparsed syllables.

Subsequently, Schreuder and Gilbers propose the re-ranking of the constraints Output-Output Correspondence and $*\Sigma\Sigma$ above a certain speech rate, after discarding the candidate *(fó)to(tòestel)*—a harmonic bound—from the candidate set. Careful speech is faithful to the morphological structure, as in (7), whereas fast speech optimises for pronunciation ease is (8).

(7) *Slow (andante) speech:*

fototoestel	OO-CORR.	$*\Sigma\Sigma$	PARSE-σ
☞(fóto)(tòestel)		*	
(fóto)toe(stèl)	*!		*

(8) *Fast (allegro) speech:*

fototoestel	$*\Sigma\Sigma$	OO-CORR.	PARSE-σ
(fóto)(tòestel)	*!		
☞(fóto)toe(stèl)		*	*

Yet, this proposal raises few questions. First, fast speech is usually seen rather as a performance phenomenon. If the competence (the knowledge of the language encoded in the brain) of the speaker is not altered, why would one model it with a new grammar? Second, if we still suppose a sudden change in the grammar at a certain speech rate, how can we explain that the fast speech form appears only in some percentage of cases? If the grammar is altered, then the new form should *always* appear, which is not the case. In fact, the difference between the two speech rates is rather a gradual shift in the frequency of two forms, both of which appear in both andante and allegro speech (Table 1 and Schreuder (2005)).

Stochastic Optimality Theory (Boersma and Hayes 2001) can model this phenomenon within one grammar. By adding a random *evaluation noise* to the ranking of the constraints, Stochastic OT allows for the re-ranking of the two constraints proposed by Schreuder and Gilbers (2004). If noise increases with speech rate, the probability of re-ranking the two constraints also grows, without a categorical switch within the grammar. It is unclear, however, why the evaluation noise should be higher in fast speech. Even worse, Stochastic OT cannot account for the different grammatical form / fast speech form-rates for different words, if they are to be explained by the reranking of the same constraints. The rank of the constraints and the evaluation noise may depend on the speech rate, but not on the specific input.

Third, this particular analysis is based on the re-ranking of *two* constraints, which cannot take place in more than 50% of the cases—leading to a false prediction of the model. In the case of two constraints, the probability of rerankig them converges to 0.5, as the evaluation noise (compared to the difference of their ranks) grows to infinity. However, a Stochastic Optimality Theoretic model with more constraints could correctly predict the fast speech form appearing in more than half of the cases. Which constraint should we then add to the model? An alternative would be to change the (unperturbed) ranks of the constraints, instead of increasing the evaluation noise in fast speech: why and how do the ranks of the different constraints change in function of the speech rate?

The advantage of the model to be presented using *Simulated Annealing Optimality Theory* will be manyfold. First, modelling fast speech by speeding up the algorithm is more convincing than postulating the increase in the evaluation noise or changing the underlying competence model (the OT grammar). More importantly, SA-OT correctly predicts which words are more likely to be pronounced erroneously. Last, the rate of the fast-speech form may exceed 50% in some cases without having to add new constraints.

4.2 Gen and the topology of the search space

Let us apply simulated annealing to stress assignment. The input is a word composed of a number of syllables. The set of candidates corresponding to this input is composed of all possible correct parses of this input. A parse is correct if: it contains the same number of syllable as the input; it contains at least one foot; feet do not overlap; a syllable not parsed into any foot is unstressed; finally, each foot contains one or two syllables, exactly one of which is stressed. Here, we ignore the difference between primary and secondary stress. For a four-syllable word input, possible parses include: u[s]uu, [su]uu, [us]u[s], [s][s][s][s], etc. Brackets represent foot borders; u and s refer to unstressed and stressed syllables, respectively.

Having defined the set of candidates, we now proceed to the topology of the search space. The *neighbours* of a candidate are the candidates reachable in one *basic step*, and a *basic step* is performing exactly one of the following actions:

- Insert a monosyllabic foot: turn an unparsed u into [s].
- Remove a monosyllabic foot: turn [s] into an unparsed u.
- Move one foot border: enlarge a foot by taking an unparsed syllable into a foot, or narrow a foot by taking an unstressed syllable out of a foot.
- Change the head syllable within a bisyllabic foot.

Defining the topology of the search space includes also determining the probability measure according to which one of the neighbours is picked at each step of the simulation. For the sake of ease, we assign equal probability to each neighbour.

The graph in Figure 4 presents the topology of the search space for a three-syllable input. (The candidate set of four-syllable words includes 43 candidates, and is too complex to reproduce here.) The arcs of the graph connect neighbours, and the arrow on an arc points towards the candidate which is more or equal harmonic, with respect to the toy ranking $*\Sigma\Sigma \gg$ PARSE-σ.

The arrows already bring us from the “horizontal” to the “vertical” structure of the landscape toured by the random walker. An arc on the graph with only one arrow points downhill, whereas an arc with two arrows represents a horizontal move. An arrow from candidate w to candidate w' means that the move from w to w' is possible with a transition probability of 100% during the entire simulation.

Eyeballing the graph, we can point to some phenomena. Candidate [s][s][s] represents a summit, a local maximum. It is also the global maximum, but this fact cannot be seen directly from the graph. The candidate [s]u[s] is a local minimum: the arrows from all its neighbours point towards it. We also find valleys of candidates of equal harmony, situated lower than their surroundings (for instance the one formed by u[su]

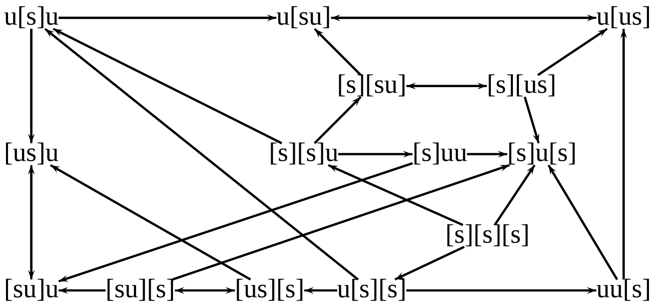

Figure 4: Search space (candidate set, neighbourhood structure) for a three-syllable word.

and u[us]). Comparing the local minima, [s]u[s], u[su], u[us], [us]u and [su]u, proves that all of them are global minima, as well. However, the graph itself would not help in determining which of them is a global minimum.

4.3 The vertical structure of the landscape: the constraints

Besides the constraints $^*\Sigma\Sigma$ and PARSE already mentioned, as well as besides OUTPUT-OUTPUT CORRESPONDENCE to which we are coming back in the next subsection, two further constraints will be used. From the large family of *alignment constraints*, we use the one requiring the left edge of the word matching the left edge of some foot (ALIGN(WORD, FOOT, LEFT), or in short, ALIGN-LEFT). Additionally, constraint TROCHAIC implements the tendency of Dutch to prefer trochaic feet. In sum, here are the constraints we are using:

- ALIGN-LEFT: assign one violation mark if left edge of word does not align with left edge of some foot.
- OUTPUT-OUTPUT CORRESPONDENCE: the stress pattern matches the expectations from the morphological structure.
- $^*\Sigma\Sigma$: one violation mark per adjacent feet borders.
- PARSE: one violation mark per unparsed syllable.
- TROCHAIC: one violation mark to each iambic foot ([us]).

Their ranking should make the grammatical form (the normal speech form) the optimal one, hence faithfulness to the morphological structure should dominate markedness—as it is the case in tableau (7). It is ranking $^*\Sigma\Sigma$ over PARSE which returns suus as the structure preferred by the markedness constraints. ALIGN-LEFT has to be ranked higher than OOC to help suus be a local optimum even for inputs, such as *perfectionist*, whose morphological structure would require usus. Finally, TROCHAIC is ranked low, and its only role is to distinguish between otherwise equal forms, such as [su]u[su] and [su]u[us] (in a word such as *studietoelage*, whose morphology requires susuu).

In short, without claiming that this is the only possible grammar describing the data, we used the following hierarchy:

(9) $\textsc{Align-Left} \gg \text{OOC} \gg^{*} \Sigma\Sigma \gg \textsc{Parse} \gg \textsc{Trochaic}$

We identify constraint ALIGN-LEFT with the domain (index) $K = 4$, constraint OOC with $K = 3$, ..., and finally constraint TROCHAIC with $K = 0$.

4.4 Output-Output Correspondence

In the present subsection, we define the constraint OUTPUT-OUTPUT CORRESPONDENCE (OOC). Originally, Burzio (2002)'s proposal, based on an analogy from physics, required a sum over *all* elements of the lexicon. In practice, however, this constraint compares a candidate with its closest neighbours, that is, with the independent word forms of its morphological constituents. Used to account for phenomena related to morphology, it is usually defined only in a very vague way.

As SA-OT necessitates an exact definition, we propose to define OOC in the following way: candidate w is compared to a string σ of the same length, a stress pattern derived from the stress patterns of w's immediate morphological constituents. If w is the concatenation of a number of morphemes, σ is the concatenation of their stress patterns. Phonological arguments support that a candidate has to be compared to its immediate morphological components, and not to deeper levels in its morphological structure (e.g. Burzio (2002), Bíró (forthcoming)).

For instance, the stress pattern that parses of *individualist* are compared to is $\sigma =$ sususs: the stress pattern susus of *ìndivìduéel* followed by the pattern s of the stress attracting suffix *ist*. The pattern suus of *ìn.di.vi.dú* does not play a role.

After these preparations, we are ready to define the constraint OUTPUT-OUTPUT CORRESPONDENCE. The number of violation marks assigned to a candidate w is the number of mismatches with the corresponding string σ, after a pairwise comparison of the corresponding elements of the (equally long) strings:

(10) $$\text{OOC}_\sigma(w) = \sum_i \Delta(w_i, \sigma_i)$$

where w_i and σ_i represent the ith letter (now, the ith syllable's type) of the candidate w and the comparison string σ; and where $\Delta(w_i, \sigma_i) = \begin{cases} 1 & \text{if } w_i \neq \sigma_i \\ 0 & \text{if } w_i = \sigma_i \end{cases}$

The definition of OOC is thus complete, but not satisfactory. The result is maybe not exactly what we wish. Misplacing one stress should be a smaller difference than missing a stress entirely, or having extra stresses. If the target string is $\sigma =$ suus, then $w_1 =$ susu should be closer to it than $w_2 =$ suuu or $w_3 =$ suss. Yet, definition (10) will assign two violation marks to w_1, because there is a mismatch in both the third and the fourth syllable, whereas only one violation mark will be assigned to w_2 and to w_3. Candidate w_1 violates constraint OOC_σ on the same level as the "totally misconceived" candidate $w_4 =$ssss.

fo.to.toe.stel ‘camera’	*uit.ge.ve.rij* ‘publisher’	*stu.die.toe.la.ge* ‘study grant’	*per.fec.tio.nist* ‘perfectionist’
OOC to: susu	ssus	susuu	usus
fó.to.tòe.stel fast: *0.82* slow: *1.00*	*úit.gè.ve.rìj* fast: *0.65* / **0.67** slow: *0.97* / **0.96**	*stú.die.tòe.la.ge* fast: *0.55* / **0.38** slow: *0.96* / **0.81**	*per.féc.tio.nìst* fast: *0.49* / **0.13** slow: *0.91* / **0.20**
fó.to.toe.stèl fast: *0.18* slow: *0.00*	*úit.ge.ve.rìj* fast: *0.35* / **0.33** slow: *0.03* / **0.04**	*stú.die.toe.là.ge* fast: *0.45* / **0.62** slow: *0.04* / **0.19**	*pér.fec.tio.nìst* fast: *0.39* / **0.87** slow: *0.07* / **0.80**

Table 1: Simulated (in italics) and observed (in bold; Schreuder, 2005) frequencies. The simulation used $T_{step} = 3$ for fast speech and $T_{step} = 0.1$ for slow speech.

In turn, a modification of the constraint should assign additional violation marks to the difference in the stressed syllables. Let $\parallel \alpha \parallel$ denote the number of stresses (s) in the string α: $\parallel \alpha \parallel = \sum_i \Delta(\alpha_i, \mathrm{u})$. Then, OOC is re-defined as:

(11) $$\mathrm{OOC}_{z,\sigma}(w) = \sum_i \Delta(w_i, \sigma_i) + z \cdot \Big| \parallel w \parallel - \parallel \sigma \parallel \Big|$$

This definition introduces a new parameter z, which determines the relative weight of pointwise mismatch *vs.* difference in the global number of stresses.

4.5 Simulation results

After so much preparation, we can run the simulation. The algorithm of *Simulated Annealing Optimality Theory* has been given in Figure 3. The hierarchy (9) and further considerations mentioned earlier suggest using the following cooling schedule: $K_{max} = 5$ (one layer above the top constraint), $K_{step} = 1$, $t_{max} = 3$, $t_{min} = 0$. Parameter K_{min} was chosen in the function of t_{step}: $K_{min} = -2$ is low enough for $t_{step} = 0.1$ and $K_{min} = -100$ suffices for $t_{step} = 3$.

The simulation has been run with different t_{step} values, ranging between 0.03 and 3. For each parameter setting, we have run the simulation 600 times using each candidate as the initial point of the random walk. Hence, the simulation was run 25800 times for four-syllable inputs (43 candidates), and 71400 times for five-syllable inputs (119 candidates).

The results appear in Table 1, together with the outcome of Maartje Schreuder’s laboratory experiments (Schreuder 2005). Taking $T_{step} = 3$ as a fast speech model, and $T_{step} = 0.1$ as a slow speech model, the match between experiment and simulation is surprisingly good for the words belonging to the type of *uitgeverij*. The quantitative match is worse for other types of words, yet the simulation correctly predicts which types are more likely to be produced erroneously. Furthermore, the results—the 49% of *per.féc.tio.nìst* in fast speech—show that unlike Stochastic OT with the present underlying OT model, SA-OT can return the fast speech form with a frequency above 50% (the difference is significant).

In order to appreciate the results, one has to realise that not only did the model reproduce the grammatical forms, but it also correctly predicted which among the 43 or 119 candidates is the alternative fast speech form. In fact, in the case of *perfectionist*, a third form has also been returned (2% in slow speech, 12% in fast speech), namely [s][su]u (*pérfèctionist*)—by using $z = 1$ in the definition (11) of OOC. Different values for z returned the non-attested [s][su]u form even more frequently. This difficulty underlines the non-triviality of the present results.

5 Summary

The present paper has implemented a heuristic technique, simulated annealing, to Optimality Theory. The standard algorithm had to be slightly modified in order to use it to find the optimal candidate of the candidate set. Simulated annealing does not guarantee maximal precision, and this “drawback” could model the lack of precision in human speech: faster production yields more performance errors. Despite quantitative mismatches so-far, the approach seems to be promising.

Simulated annealing required the introduction of some new concepts in Optimality Theory: a *topology* (a *neighbourhood structure*) on the candidate set, the *difference* of two violation profiles, temperature, as well as a more precise definition of OUTPUT-OUTPUT CORRESPONDENCE.

We propose to see *Simulated Annealing Optimality Theory* (SA-OT) as a model for (part of) the linguistic performance. If traditional Optimality Theory represents linguistic competence (that is, the static knowledge of the language encoded in one’s brain), then simulated annealing models the dynamic computations involved in producing utterances. The arguments for why simulated annealing can be an adequate model of (part of) the performance included the fact that it does not require complex computing capacities even in the case of NP-complete problems; that it returns a “nearly good” solution in limited time; and that this time interval can be reduced (just like speech can be speeded up) by paying in precision. The last fact was demonstrated on the case of Dutch stress assignment in fast speech.

The reader is welcome to try out the demo of SA-OT and the implementation of the model introduced for Dutch stress at http://www.let.rug.nl/ birot/sa-ot/.

Acknowlegments

I acknowledge the support of the *High Performance Computing* program of the University of Groningen, the Netherlands. The author is thankful to *Gosse Bouma*, *Dicky Gilbers*, *Gertjan van Noord* and *Maartje Schreuder* for data and for valuable discussions, as well as to an anonymous reviewer for useful remarks.

References

Bíró, T.(2003), Quadratic alignment constraints and finite state optimality theory, *Proc. FSMNLP, within EACL-03, Budapest*, also: ROA-600[4], pp. 119–126.

[4] ROA stands for *Rutgers Optimality Archive* at `http://roa.rutgers.edu/`.

Bíró, T.(2005), How to define Simulated Annealing for Optimality Theory?, *Proc. Formal Grammar 10 and MOL 9*, Edinburgh.

Bíró, T.(forthcoming), *Finding the Right Words: Implementing Optimality Theory*, PhD thesis, Rijksuniversiteit Groningen, Groningen, Netherlands.

Boersma, P. and Hayes, B.(2001), Empirical tests of the gradual learning algorithm, *Linguistic Inquiry* **32**, 45–86.

Burzio, L.(2002), Missing players, *Lingua* **112**, 157–199.

Eisner, J.(2000), Easy and hard constraint ranking in OT, *Finite-State Phonology: Proc. SIGPHON-5*, Luxembourg, pp. 57–67.

Howells, T.(1988), Vital: a connectionist parser, *Proceedings of 10th Annual Meeting of the Cognitive Science Society*, pp. 18–25.

Kempen, G. and Vosse, T.(1989), Incremental syntactic tree formation in human sentence processing, *Connection Science* **1**, 273–290.

Kirkpatrick, S., Jr., C. D. G. and Vecchi, M. P.(1983), Optimization by simulated annealing, *Science* **220**(4598), 671–680.

Kuhn, J.(2000), Processing optimality-theoretic syntax by interleaved chart parsing and generation, *Proc.ACL-38, Hongkong*, pp. 360–367.

Prince, A. and Smolensky, P.(2004), *Optimality Theory: Constraint Interaction in Generative Grammar*, Blackwell, Originally: RuCCS-TR-2, 1993.

Reeves, C. R. (ed.)(1995), *Modern Heuristic Techniques for Combinatorial Problems*, McGraw-Hill, London, etc.

Schreuder, M.(2005), *Prosodic Processes in Language and Music*, PhD thesis, Rijksuniversiteit Groningen, Groningen, Netherlands.

Schreuder, M. and Gilbers, D.(2004), The influence of speech rate on rhythm patterns, *in* D. Gilbers, M. Schreuder and N. Knevel (eds), *On the Bounderies of Phonology and Phonetics*, University of Groningen, pp. 183–201.

Selman, B. and Hirst, G.(1994), Parsing as an energy minimization problem, *in* G. Adriaens and U. Hahn (eds), *Parallel Natural Language Processing*, Ablex, Norwood, NJ, pp. 238–254.

Smolensky, P.(1986), Information processing in dynamical systems: Foundations of harmony theory, *Rumelhart et al.: Parallel Distributed Processing*, Vol. 1, Bradford, MIT Press, Cambridge, London, pp. 194–281.

Tesar, B. and Smolensky, P.(2000), *Learnability in Optimality Theory*, The MIT Press, Cambridge, MA - London, England.