Finding the Right Words
Implementing Optimality Theory with Simulated Annealing

Tamás Bíró

Work presented developed at:
Humanities Computing
CLCG
University of Groningen

Present affiliation:
Theoretical Linguistics
ACLC
University of Amsterdam

birot@nytud.hu

Amsterdam, ILLC, March 2, 2007
Overview

• Optimality Theory (OT) in a nutshell

• Simulated Annealing for Optimality Theory (SA-OT)

• Examples

• The \textit{dis}-harmonic mind?

• Conclusion
Optimality Theory in a nutshell

OT tableau: search the best candidate w.r.t lexicographic ordering
(cf. abacus, abolish, ..., apple, ..., zebra)

<table>
<thead>
<tr>
<th></th>
<th>$c_n$</th>
<th>$c_{n-1}$</th>
<th>...</th>
<th>$c_{k+1}$</th>
<th>$c_k$</th>
<th>$c_{k-1}$</th>
<th>$c_{k-2}$</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>(w)</td>
<td>2</td>
<td>0</td>
<td>...</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>(w')</td>
<td>2</td>
<td>0</td>
<td>...</td>
<td>1</td>
<td>3!</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>(w'')</td>
<td>3!</td>
<td>0</td>
<td></td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>
Optimality Theory in a nutshell

- Pipe-line *vs.* optimize the Eval-function

- Gen: \( UR \mapsto \{ w | w \text{ is a candidate corresponding to } UR \} \)

  E.g. assigning Dutch metrical foot structure & stress:
  \( \text{fototoestel} \mapsto \{ \text{fototoe(stél)}, (\text{fotó})(tòestel), (fó)to(toestèl), \ldots \} \)
Optimality Theory: an optimization problem

\[ UR \mapsto \{ w \mid w \text{ is a candidate corresponding to } UR \} \]

\[ E(w) = \left( C_N(w), C_{N-1}(w), \ldots, C_0(w) \right) \in \mathbb{N}_0^{N+1} \]

\[ SR(UR) = \arg\text{opt}_{w \in Gen(UR)} E(w) \]

Optimization: with respect to lexicographic ordering
OT is an optimization problem

The question is: How can the optimal candidate be found?

- Finite-State OT (Ellison, Eisner, Karttunen, Frank & Satta, Gerdemann & van Noord, Jäger...)

- chart parsing (dynamic programming) (Tesar & Smolensky; Kuhn)

These are perfect for language technology. But we would like a psychologically adequate model of linguistic performance (e.g. errors): Simulated Annealing.
How to find optimum: Gradient Descent 1

\[ w := w_{\text{init}} ; \]

Repeat

\[ w' := \text{best element of set Neighbours}(w) ; \]
\[ \text{Delta} := E(w') - E(w) ; \]
\[ \text{if} \quad \text{Delta} < 0 \quad \text{then} \quad w := w' ; \]
\[ \text{else} \]
\[ \quad \text{do nothing} \]
\[ \text{end-if} \]

Until stopping condition = true

Return \( w \) \# \( w \) is an approximation to the optimal solution
How to find optimum: Gradient Descent 2

w := w_init ;
Repeat
    Randomly select w’ from the set Neighbours(w);
    Delta := E(w’) - E(w) ;
    if Delta < 0 then w := w’ ;
    else
        do nothing
    end-if

Until stopping condition = true

Return w       # w is an approximation to the optimal solution
The Simulated Annealing Algorithm

\[ \text{w := w\textunderscore init} \; \quad \text{t := t\textunderscore max} \; \]
Repeat
\[
\text{Randomly select w'} \text{ from the set Neighbours(w);} \\
\text{Delta := E(w') - E(w);} \\
\text{if Delta < 0 then w := w';} \\
\text{else} \\
\quad \text{generate random r uniformly in range (0,1);} \\
\quad \text{if r < exp(-Delta / t) then w := w';} \\
\quad \text{end-if}
\]
\[ \text{t := alpha(t)} \quad \# \text{decrease t} \\
\]
Until stopping condition = true

Return w \quad \# \text{w is an approximation to the optimal solution}
Deterministic Gradient Descent for OT


- Based on a remark by Prince and Smolensky (1993/2004) on a “restraint of analysis” as opposed to “freedom of analysis”.

- Restricted Gen $\rightarrow$ Eval $\rightarrow$ Gen $\rightarrow$ Eval $\rightarrow$... ($n$ times).

- Gradual progress toward (locally) max. harmony.

- Employed to simulate traditional derivations, opacity.
Simulated Annealing for OT – general idea

- Neighbourhood structure on the candidate set.
- Landscape’s vertical dimension = harmony; random walk.
- If neighbour more optimal: move.
- If less optimal: move in the beginning, don’t move later.
Simulated Annealing for OT – general idea

- Neighbourhood structure $\rightarrow$ local optima.
- System can get stuck in local optima: alternation forms.
- Precision of the algorithm depends on its speed (!!).
- Many different scenarios.
Sim. annealing with non-real valued target function

- Exponential weights if upper bound on $C_i(w)$ violation levels:

$$E(w) = C_N(w) \cdot q^N + C_{N-1}(w) \cdot q^{N-1} + ... + C_1(w) \cdot q + C_0(w)$$

- Polynomials:

$$E(w)[q] = C_N(w) \cdot q^N + C_{N-1}(w) \cdot q^{N-1} + ... + C_1(w) \cdot q + C_0(w)$$

- Ordinal weights:

$$E(w) = \omega^N C_N(w) + ... + \omega C_1(w) + C_0(w)$$
Sim. annealing with non-real valued target function

Transition probability if $w'$ worse than $w$: what is $e^{\frac{E(w') - E(w)}{t}}$?

- Polynomials:

  $$ T[q] = \langle K_T, t \rangle [q] = t \cdot q^{K_T} $$

  $$ P\left( w \rightarrow w' \mid T[q] \right) = \lim_{q \rightarrow +\infty} e^{-\frac{E(w')[q] - E(w)[q]}{T[q]}} $$

- Ordinals: move iff the generated $r \in [0, 1]$ is s.t.

  $$ \forall \alpha \in \mathbb{N}^+: r^{-\alpha} > 2^q \left( \Delta \left( E(w'), E(w) \right) \cdot \alpha, T \right) $$
Domains for temperature and constraints

- Temperature: $T = \langle K_T, t \rangle \in \mathbb{Z} \times \mathbb{R}^+$ (or "$\mathbb{Z} \times \mathbb{R}^+$").

- Constraints associated with domains of $K_T$:

<table>
<thead>
<tr>
<th></th>
<th>$C_0$</th>
<th>$C_1$</th>
<th>$C_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K = -1$</td>
<td>$K = 0$</td>
<td>$K = 1$</td>
<td>$K = 2$</td>
</tr>
<tr>
<td>... 0.5 1.0 1.5 2.0 2.5</td>
<td>... 0.5 1.0 1.5 2.0 2.5</td>
<td>... 0.5 1.0 1.5 2.0 2.5</td>
<td>... 0.5 1.0 1.5 2.0 2.5</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Rules of moving

**Rules of moving from** $w$ to $w'$

at temperature $T = \langle K_T, t \rangle$:

If $w'$ is better than $w$: move! $P(w \rightarrow w'|T) = 1$

If $w'$ loses due to fatal constraint $C_k$:

- If $k > K_T$: don’t move! $P(w \rightarrow w'|T) = 0$
- If $k < K_T$: move! $P(w \rightarrow w'|T) = 1$
- If $k = K_T$: move with probability

$$P = e^{-(C_k(w')-C_k(w))/t}$$
The SA-OT algorithm

\[ w := w_{\text{init}} ; \]
for \( K = K_{\text{max}} \) to \( K_{\text{min}} \) step \( K_{\text{step}} \)
\quad for \( t = t_{\text{max}} \) to \( t_{\text{min}} \) step \( t_{\text{step}} \)
\quad \quad \text{CHOOSE random } w' \text{ in neighbourhood}(w) ;
\quad \quad \text{COMPARE } w' \text{ to } w: \ C := \text{fatal constraint} \]
\quad \quad \quad d := C(w') - C(w); \]
\quad if \( d \leq 0 \) then \( w := w' ; \)
\quad else \quad w := w' \text{ with probability} \]
\quad \quad \quad P(C,d;K,t) = 1 \quad \text{, if } C < K \]
\quad \quad \quad = \exp(-d/t) , \quad \text{if } C = K \]
\quad \quad \quad = 0 \quad \quad \quad \quad \text{, if } C > K \]
\quad \quad \quad \text{end-for} \]
\quad \text{end-for} \]
\quad return w \]
SA-OT as a model of linguistic performance

Optimality Theory: a model of competence

Simulated Annealing: a model of performance
Proposal: three levels

<table>
<thead>
<tr>
<th>Level</th>
<th>its product</th>
<th>its model</th>
<th>the product in the model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Competence in narrow sense: static knowledge</td>
<td>grammatical form</td>
<td>standard OT grammar</td>
<td>globally optimal candidate</td>
</tr>
<tr>
<td>of the language</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dynamic language production process</td>
<td>acceptable or attested forms</td>
<td>SA-OT algorithm</td>
<td>local optima</td>
</tr>
<tr>
<td>Performance in its outmost sense</td>
<td>acoustic signal, etc.</td>
<td>(phonetics, pragmatics)</td>
<td>??</td>
</tr>
</tbody>
</table>
The art of using Simulated Annealing Optimality Theory

- Take a traditional OT model
- Add *convincing* neighbourhood structure to candidate set
- Local (non-global) optima = alternation forms
- Run simulation (e.g., [http://www.let.rug.nl/~birot/sa-ot](http://www.let.rug.nl/~birot/sa-ot)):
  - Slowly: likely to return only the grammatical form
  - Quickly: likely to return local (non-global) optima
Parameters of the algorithm

- $t_{step}$ (and $t_{max}$, $t_{min}$)
- $K_{max}$ (and $K_{min}$)
- $K_{step}$
- $w_0$ (initial candidate)
- Topology (neighbourhood structure)
- Constraint hierarchy
How to make the topology convincing?

A connected (weighted) “graph”; universal;...

- Observation-driven strategies:
  - Many phenomena in many languages
    or even better: cross-linguistic typologies
  - Based on existing theories based on cross-linguistic
    observations (cf. Hayes’s *metrical stress theory*)

- Theory-driven strategies:
  - Principles (e.g. minimal set of basic transformations)
  - Psycholinguistically relevant notions of similarity, etc.
**Example: Fast speech: Dutch metrical stress**

<table>
<thead>
<tr>
<th></th>
<th><code>camer</code>a</th>
<th><code>publisher</code></th>
<th><code>study grant</code></th>
<th><code>perfectionist</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>susu</td>
<td>ssus</td>
<td>susuu</td>
<td>usus</td>
<td></td>
</tr>
<tr>
<td><strong>fó.to.tòe.stel</strong></td>
<td><code>ùit.gè.ve.říj</code></td>
<td><code>stú.dìe.tòe.là.ge</code></td>
<td><code>per.fèc.tìo.níst</code></td>
<td></td>
</tr>
<tr>
<td>fast: 0.82</td>
<td>fast: 0.65 / 0.67</td>
<td>fast: 0.55 / 0.38</td>
<td>fast: 0.49 / 0.13</td>
<td></td>
</tr>
<tr>
<td>slow: 1.00</td>
<td>slow: 0.97 / 0.96</td>
<td>slow: 0.96 / 0.81</td>
<td>slow: 0.91 / 0.20</td>
<td></td>
</tr>
<tr>
<td><strong>fó.to.toe.stèl</strong></td>
<td><code>ùit.ge.ve.říj</code></td>
<td><code>stú.dìe.toe.là.ge</code></td>
<td><code>pèr.fèc.tìo.níst</code></td>
<td></td>
</tr>
<tr>
<td>fast: 0.18</td>
<td>fast: 0.35 / 0.33</td>
<td>fast: 0.45 / 0.62</td>
<td>fast: 0.39 / 0.87</td>
<td></td>
</tr>
<tr>
<td>slow: 0.00</td>
<td>slow: 0.03 / 0.04</td>
<td>slow: 0.04 / 0.19</td>
<td>slow: 0.07 / 0.80</td>
<td></td>
</tr>
</tbody>
</table>

*Simulated / observed* (Schreuder) frequencies.

In the simulations, $T_{step} = 3$ used for fast speech and $T_{step} = 0.1$ for slow speech.
Example: Irregularities

- Local optimum that is not avoidable.
Example: string-grammar

- **Candidates:** $\{0, 1, ..., P - 1\}^L$
  E.g. ($L = P = 4$): 0000, 0001, 0120, 0123, ..., 3333.

- **Neighbourhood structure:** $w$ and $w'$ neighbours iff one basic step transforms $w$ to $w'$.

- **Basic step:** change exactly one character $\pm 1$, $\text{mod } P$ (cyclicity).

- Each neighbour with equal probability.
Example: string-grammar

Markedness Constraints \( w = w_0w_1...w_{L-1}, \ 0 \leq n < P \): 

- **No-\( n \):** \( *n(w) := \sum_{i=0}^{L-1} (w_i = n) \)

- **No-initial-\( n \):** \( *\text{INITIAL}n(w) := (w_0 = n) \)

- **No-final-\( n \):** \( *\text{FINAL}n(w) := (w_{L-1} = n) \)

- **Assimilation** \( \text{ASSIM}(w) := \sum_{i=0}^{L-2} (w_i \neq w_{i+1}) \)

- **Dissimilation** \( \text{DISSIM}(w) := \sum_{i=0}^{L-2} (w_i = w_{i+1}) \)
Example: string-grammar

- Faithfulness to UR $\sigma$:

$$\text{FAITH}_\sigma(w) = \sum_{i=0}^{L-1} d(\sigma_i, w_i)$$

where $d(a, b) = \min(|a - b|, |b - a|)$

(binary square, feature-combination?)
Example: string-grammar

\[ L = P = 4, \quad T_{\text{max}} = 3, \quad T_{\text{min}} = 0, \quad K_{\text{step}} = 1. \]

Each of the 256 candidates used 4 times as \( w_0 \).

Grammar:

*0 \gg Assim \gg \text{Faithf}_{\sigma=0000} \gg *\text{Init}1 \gg *\text{Init}0 \gg *\text{Init}2 \gg *\text{Init}3 \gg *\text{Fin}0 \gg *\text{Fin}1 \gg *\text{Fin}2 \gg *\text{Fin}3 \gg *3 \gg *2 \gg *1 \gg \text{Dissim} \]

Globally optimal form: 3333

Many other local optima, e.g.: 1111, 2222, 3311, 1333, etc.
Example: string-grammar

Output frequencies for different $T_{step}$ values:

<table>
<thead>
<tr>
<th>output</th>
<th>0.0003</th>
<th>0.001</th>
<th>0.003</th>
<th>0.01</th>
<th>0.03</th>
<th>0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>0.40</td>
<td>0.40</td>
<td>0.36</td>
<td>0.35</td>
<td>0.32</td>
<td>0.24</td>
</tr>
<tr>
<td>3333</td>
<td>0.39</td>
<td>0.39</td>
<td>0.41</td>
<td>0.36</td>
<td>0.34</td>
<td>0.21</td>
</tr>
<tr>
<td>2222</td>
<td>0.14</td>
<td>0.14</td>
<td>0.15</td>
<td>0.18</td>
<td>0.19</td>
<td>0.17</td>
</tr>
<tr>
<td>3311</td>
<td>0.04</td>
<td>0.04</td>
<td>0.04</td>
<td>0.05</td>
<td>0.06</td>
<td>0.05</td>
</tr>
<tr>
<td>1133</td>
<td>0.03</td>
<td>0.04</td>
<td>0.04</td>
<td>0.04</td>
<td>0.05</td>
<td>0.04</td>
</tr>
<tr>
<td>others</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>0.04</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.29</td>
</tr>
</tbody>
</table>
What does SA-OT offer to standard OT?

- A new approach to account for variation:
  - Non-optimal candidates also produced (cf. Coetzee);
  - As opposed to: more candidates with same violation profile; more hierarchies in a grammar.

- A topology (neighbourhood structure) on the candidate set.


- Arguments for including losers (never winning candidates).
The *dis*-harmonic mind?

ICS (Integrated Connectionist/Symbolic Cognitive Architecture):

“[T]here is no symbolic algorithm whose internal structure can predict the time and the accuracy of processing; this can only be done with connectionist algorithms” (Smolensky and Legendre (2006): *The Harmonic Mind*, vol. 1, p. 91).

SA-OT:

- symbolic computation only
- predicts time and accuracy of processing
Summary of SA-OT

- Implementing OT: lang. technology? cognitively plausible?
- A model of variation / performance phenomena.
- *Errare humanum est*: heuristics in cognitive science.
- Time and accuracy with a symbolic-only architecture.
- Much work needed: learnability, linguistic examples, etc.
Thank you for your attention!

Tamás Bíró

birot@nytud.hu