A sz.ot.ag

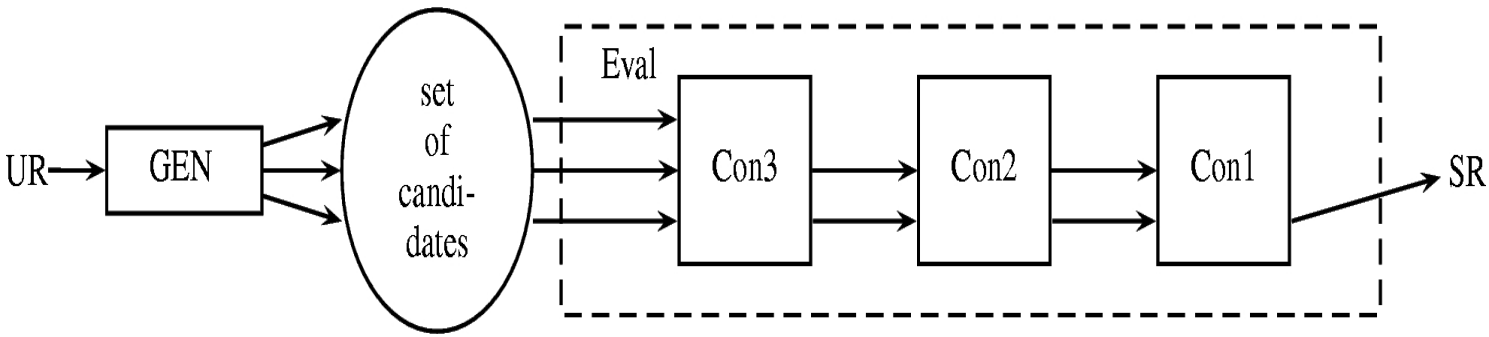
Optimalitáselmélet szimulált hőkezeléssel

Bíró Tamás
Humanities Computing, CLCG
University of Groningen, Hollandia
valamint
Eötvös Loránd Tudományegyetem, Budapest
birot@let.rug.nl, birot@nytud.hu

III. Magyar Számítógépes Nyelvészeti Konferencia
2005. december 8.

Optimalitáselmélet (Optimality Theory)

Prince & Smolensky, 1993 / 2004



OT tábla: a legjobb jelölt megkeresése, lexikografikus rendezés alapján

	C_N	C_{N-1}	...	C_{k+1}	C_k	C_{k-1}	C_{k-2}	...
w	2	0		1	2	3	0	
w'	2	0		1	3 !	1	2	
w''	3 !	0		1	3	1	2	

Áttekintés

- Az alapprobléma: az optimális jelölt megtalálása:

$$E(w) = \Big(C_N(w), C_{N-1}(w), ..., C_0(w)\Big) \in \mathbb{N}_0^{N+1}$$
$$SR(\,UR) = \text{argopt}_{w \in Gen(\,UR)} E(w)$$

- Szimulált hőkezelés (szimulált lehűtés, *simulated annealing*)
- OT + SA = SA-OT, ezért topológia bevezetése OT-ben
- A topológia szerepe: szótagolás, mint példa

Az alapprobléma: az optimális jelölt megtalálása

Meglehetősen nehéz feladat lehet (NP-teljes, Eisner 1997), különösen nagy vagy végtelen halmazon.

- Dinamikus programozás (*dynamic programing*, *chart parsing*; Tesar & Smolensky, 2000; Kuhn, 2000: LFG-OT)
- Véges állapotú automaták (FSA: Ellison, 1994; Karttunen, 1998; Frank & Satta, '98; Gerdemann & van Noord, 2000; Jäger, 2002; Bíró)
- Genetikus algoritmus (GA: Turkel, 1994)

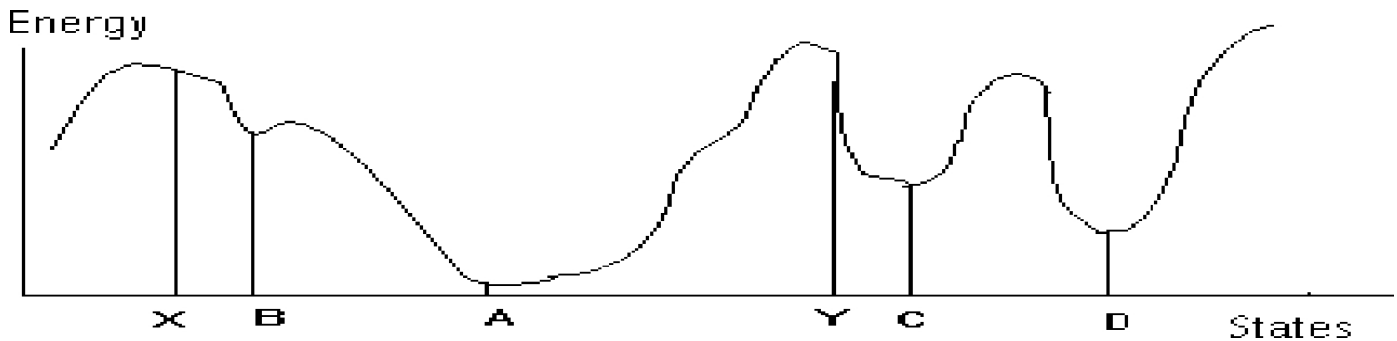
De vajon kell-e tökéletes megoldásra törekedni?

Szimulált hőkezelés

(Szimulált lehűtés, *simulated annealing*, SA)

Egy függvény *globális* minimumát keressük.

SA: elterjedt algoritmus, ötlet a statisztikus fizikából (termodinamikából)



$$P(w \to w') = \begin{cases} 1 & \text{ha } E(w') \leq E(w) \\ e^{-\frac{E(w')-E(w)}{kT}} & \text{ha } E(w') > E(w) \end{cases}$$

Gradient descent

```
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-fi


Until stopping condition = true

Return w          # w is an approximation to the optimal solution
```

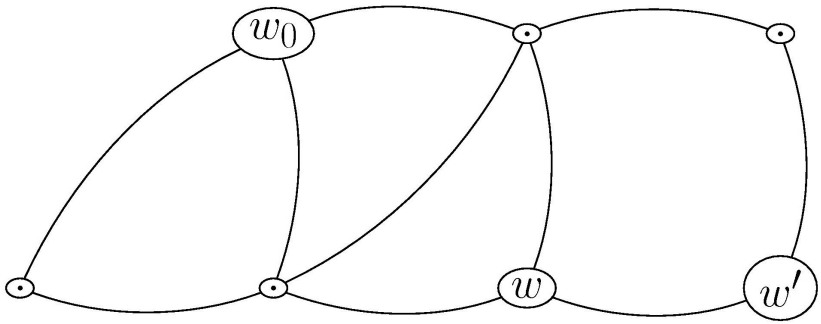
Szimulált hőkezelés

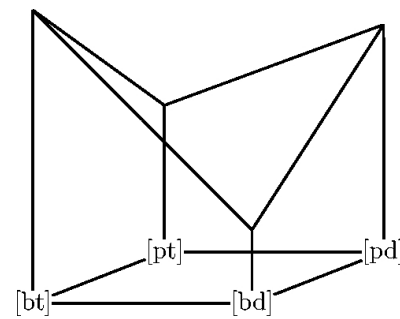
```
w := w_init ;       t := t_max  ;
Repeat
        Randomly select w' from the set Neighbours(w);
        Delta :=  E(w') - E(w) ;
        if   Delta < 0   then   w := w' ;
        else
               generate random r uniformly in range (0,1) ;
               if   r < exp(-Delta / t)   then   w := w' ;
        end-fi

   t := alpha(t)                              # decrease t
Until stopping condition = true

Return w          # w is an approximation to the optimal solution
```

OT + SA = SA-OT





- véletlen bolyongás az jelöltek halmazán
- `Neighbours(w)`: szomszédsági struktúra (topológia)
- lokális optimumok, amelyekbe beragadhat az algoritmus
- az algoritmus pontossága függ az algoritmus sebességétől.

Az SA-OT algoritmus

```
w := w_init ;
for K = K_max to K_min step K_step
      for t = t_max to t_min step t_step
            CHOOSE   random w' in Neighbours(w) ;
            COMPARE  w' to w:  C := fatal constraint
                               d := C(w') - C(w);
            if d <= 0 then w := w';
            else           w := w' with probability
                       P(C,d;K,t) = 1          , if C < K
                                  = exp(-d/t)  , if C = K
                                  = 0          , if C > K
      end-for
end-for
return w
```

Példa: szótagolás (1)

Szótag: (onset)+nukleusz+(kóda) ; Szó: szótag$^+$

Az OT modell (Prince & Smolensky 1993/2004):

- Bemeneti sztring (UR): pl. `anta`
- Szótagszerkezet, alulelemzés (törlés, underparsing), túlelemzés (epentézis, overparsing)
- Jelöltek halmaza (reguláris): pl. `N[a]D[n]O[t]N[a]`, `O[_]N[a]X[n]D[t]X[a]`, `O[_]N[a]X[n]D[t]X[a]N[_]`, `O[_]N[_]D[_]N[a]X[n]D[t]O[_]N[a]`.

Példa: szótagolás (2)

Constraint-ek:

- ONSET: a szótagkezdettel nem rendelkező szótagok száma.
- NOCODA: a kódával rendelkező szótagok száma
- PARSE: az alulelemzett szegmentumok száma.
- FILLNUCLEUS: a túlelemzett szótagmagok száma.
- FILLONSET: a túlelemzett szótagkezdetek száma.

Az SA-OT algoritmus (ismétlés)

```
w := w_init ;
for K = K_max to K_min step K_step
      for t = t_max to t_min step t_step
            CHOOSE   random w' in Neighbours(w) ;
            COMPARE  w' to w:  C := fatal constraint
                               d := C(w') - C(w);
            if d <= 0 then w := w';
            else           w := w' with probability
                       P(C,d;K,t) = 1          , if C < K
                                  = exp(-d/t)  , if C = K
                                  = 0          , if C > K
      end-for
end-for
return w
```

A topológia definíciója

```
CHOOSE   random w' in Neighbours(w) ;
```

w-ből w'-be pontosan egy *elemi lépés* vezet:

- $P_{reparse}$ valószínűséggel 1 szótaghatár elmozdul (onset-ből kóda vagy kódából onset)
- $1-P_{reparse}$ valószínűséggel epentézis vagy törlés $(50-50\%)$.

\+ $P_{postproc}$ valószínűséggel törlünk egy `O[_]N[_]`-t; vagy összevonunk egy `N[_]X[a]`-t vagy egy `X[a]N[_]`-t `N[a]`-vá, ill. egy `O[_]X[t]`-t vagy `X[t]O[_]`-t `O[t]`-vé.

Előzetes eredmények

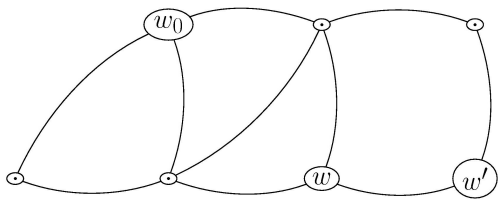
P_{rep}	%	P_{rep}	%
0.00	15	0.60	20
0.10	15	0.70	15
0.20	15	0.80	14
0.30	16	0.90	9
0.40	14	1.00	3
0.50	17		

P_{postp}	%	P_{postp}	%	P_{postp}	%
0.00	19	0.35	19	0.70	14
0.05	11	0.40	15	0.75	15
0.10	8	0.45	12	0.80	16
0.15	10	0.50	13	0.85	14
0.20	14	0.55	11	0.90	16
0.25	18	0.60	11	0.95	21
0.30	14	0.65	14	1.00	25

A helyes output (`O[_]N[a]D[t]O[t]N[a]`) gyakorisága (%), egy-egy paraméter függvényében (miközben a másik paraméterre átlagoltunk). Az input `anta` volt, a hierarchia: ONSET $\gg$ FILLNUCLEUS $\gg$ PARSE $\gg$ FILLONSET $\gg$ NOCODA. Minden $(P_{reparse}, P_{postproc})$ paraméterpár 10-szer futott.

Összefoglalás:

- Optimalitáselmélet kibővítése topológiával + paraméterekkel.



- A topológia paraméterei befolyásolják a kimenetet
- Lokális optimumok = performanciahibák
- Kipróbálható demo: `http://www.let.rug.nl/~birot/sa-ot/`

Köszönöm a figyelmet!

Bíró Tamás
birot@let.rug.nl, birot@nytud.hu