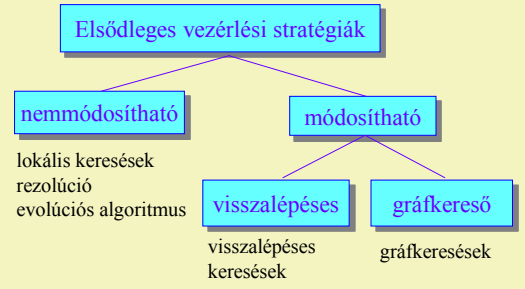
# III. Keresések



## *Elsődleges stratégiák osztályozása*






## 1. Lokális keresések

- A lokális keresés a reprezentációs gráf egyetlen csúcsát és annak környezetét látja (kicsi a globális munkaterület), az aktuális csúcsot minden lépésben annak környezetéből vett lehetőleg „jobb” (stratégia) csúccsal cseréli le (keresési szabály).
- A jobbság eldöntéséhez egy célfüggvényt (rátermettségi függvényt, heurisztikus függvényt) használunk, amely várhatóan annál jobb értéket ad egy csúcsra, minél közelebb esik az a célhoz.
- Alkalmazás:
  - Adott tulajdonságú elem keresése
  - Függvény optimumának keresése



## *Hegymászó algoritmus*

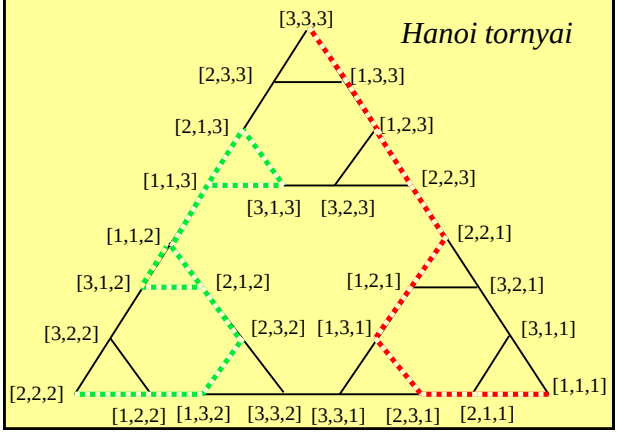
- Minden lépésben az aktuális ($n$) csúcs legjobb gyermekére lép, amelyik nem a szülője.

Procedure ***Hegymászó módszer*** (gráfreprezentációra)
1. $n :=$ *startcsúcs*
2. while *n nem célcsúcs* loop
3. $n := \arg\,\mathrm{opt}_{\gamma}( \Gamma(n) - \pi(n) )$ // üres halmazra kilép
4. endloop

end



## *Hanoi tornyai*




## *Megjegyzés*

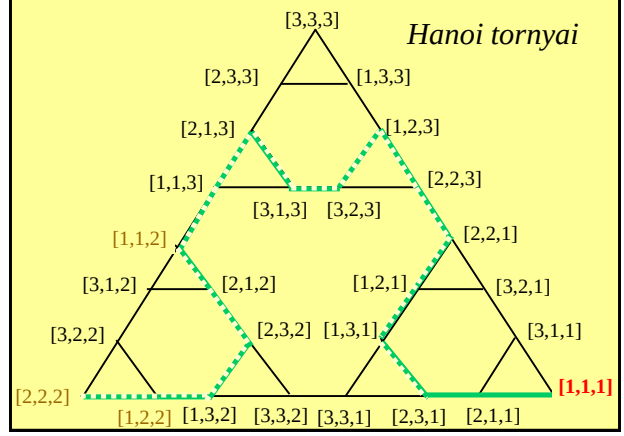
- Hátrányok:
  - nem végez körfigyelést, ezért lokális optimum hely körül, illetve ekvidisztans felületen eltévedhet
  - csak erős heurisztika esetén lesz sikeres
  - zsákutcába (szülőre nem léphet vissza) beragadhat
- A baj okai:
  - túl kicsi az algoritmus memóriája
  - túl erős az alkalmazott mohó stratégia
  - érzékeny a problématér alakjára

## *Tabu-keresés*

- Az aktuális csúcson ($n$) kívül nyilvántartja még
  - az eddig legjobbnak bizonyult csúcsot ($n^*$) és
  - az utolsó néhány érintett csúcsot; ez a tabu halmaz
- Minden lépésben
  - Az aktuális csúcs legjobb gyermekére lép, kivéve a tabu halmazban levőket
  - ha $n$ jobb, mint az $n^*$, akkor $n^*$-ot lecseréli
  - frissíti a tabu halmazt
- Terminálási feltételek:
  - ha a célfüggvény az $n^*$-ban optimális
  - ha az $n$ nem vagy az $n^*$ sokáig nem változik.



## *Hanoi tornyai*




## *Tabu kerersés algoritmusa*

Procedure ***Tabu keresés*** (gráfreprezentációra)

1. $n, n^*, Tabu := startcsúcs, startcsúcs, \emptyset$
2. while *not terminálási feltétel* ($n^*$ *nem célcsúcs*) loop
3. $n := \arg\operatorname{opt}_f(\Gamma(n) \backslash Tabu)$ // üres halmazra kilép
4. $Tabu :=$ Módosít($n$,*Tabu*)
5. if $f(n)$ jobb, mint $f(n^*)$ then $n^* := n$
6. endloop

end



## *Megjegyzés*

- Hátrányok:
  - A tabu méretét kísérletezéssel kell belőni
  - a tabuhalmaz miatt beragadhat (ha csak tabuhalmazbeli elemhez lehetne továbblépni)



## *Szimulált hűtés*

- A következő csúcs kiválasztása az aktuális ($n$) csúcs gyermekei közül véletlenszerű.
- Ha a kiválasztott $r$ csúcs célfüggvény-értéke nem rosszabb (itt $f(r) \le f(n)$), mint az $n$ csúcsé, akkor elfogadjuk, ha rosszabb ($f(r) > f(n)$), akkor az $r$ elfogadása is véletlenszerű: valószínűsége fordítottan arányos az $|f(n) - f(r)|$ különbséggel.

$$e^{\frac{f(n)-f(r)}{T}} > random[0,1]$$



## *Hűtési ütemterv*

- A $T$ (hőmérséklet) csökkentésével ugyanannak a rosszabb célfüggvényű új csúcs elfogadásának valószínűsége is kisebb lesz. *f(r)=120, f(n)=107*

| *T* | *exp(-13/T)* |
|---|---|
| $10^{10}$ | 0.9999… |
| 50 | 0.77 |
| 20 | 0.52 |
| 10 | 0.2725 |
| 5 | 0.0743 |
| 1 | 0.000002 |

- Adjunk ütemtervet a $T$ változására
  - Legyenek a $T$ értékei: $T_0, T_1, \ldots$
  - Ütemterv: $(T_k, L_k)$ $k= 1,2, \ldots$ ahol minden $T_k$ érték $L_k$ lépésen keresztül van érvényben.

## *Szimulált hűtés algoritmusa*

Procedure ***Szimulált hűtés*** (gráfreprezentációra)

1. $n :=$ *startcsúcs;* $k := 1$
2. while *not terminálási feltétel* (*n nem célcsúcs*) loop
3. for $i = 1 .. L_k$ loop
4. $r :=$ select( $\Gamma(n) \setminus \pi(n)$ )
5. if $f(r) \leq f(n)$ *or* $f(r) > f(n)$ *and* $e^{\frac{f(n)-f(r)}{T_k}} > rand[0,1]$
6. then $n := r$
7. endloop
8. endloop

end



## *Szimulált hűtés ereje*

- A szimulált hűtés algoritmusa (aszimptotikusan) egy optimális megoldáshoz konvergál
  - ha bármely elemből bármely elem véges lépésen belül elérhető
- Ahhoz azonban, hogy véges lépésen belül is egy elég jó megoldást találjunk, megfelelő hűtési ütemtervet kell találni.



## *Lokális kereséssel megoldható feladatok*

- A lokálisan hozott rossz döntés ne zárja ki a cél megtalálását
  - Ez biztos teljesül, ha reprezentációs-gráf erősen összefüggő
  - Jó heurisztikára épített célfüggvénnyel elkerülhetőek a zsákutcák vagy az olyan csúcsok, ahonnan már nem vezet út célcsúcsba.
  - Előnytelen, ha a reprezentációs-gráf egy irányított fa. (Ezért például az *n*-királynő probléma nem alkalmas lokális kereséssel való megoldáshoz, csak tökéletes heurisztikájú célfüggvény esetén)



## *Heurisztikák a Hanoi tornyai problémára*

- Darab: $C(v) = \sum_{\substack{i \\ v[i] \neq 1}} 1$
- Súlyozott darab: $WC(v) = \sum_{\substack{i \\ v[i] \neq 1}} i$
- Összeg: $S(v) = \sum_i v[i]$
- Súlyozott összeg: $WS(v) = \sum_i i * v[i]$
- Módosított összeg: $EWS(a) = WS(v) - \sum_{i=2,\dots} (v[i-1] > v[i]) + 2 * \sum_{i=2,\dots} X(v[i-1] = v[i+1] \wedge v[i] \neq v[i-1])$



## *Heurisztikák a Hanoi tornyai problémára*

- Tökéletes: $T(v) = f(n)_1$

  A korongokat megszámozzuk *1*-től *n*-ig, *1*-es lesz a legnagyobb.

  $f:[0..n] \to \mathbb{N} \times \mathbb{N}$

  $f(i)_1$ : lépések száma, ha csak az első *i* darab (legnagyobb) koronggal számolunk.

  $f(i)_2$ : hol kellene lennie az *i+1.* korongnak ahhoz, hogy a nálánál nagyobb korongok mozgatását azonnal elkezdhessük

  $f(0) = (0, 1)$ -- nincs korong

$$f(i) = \begin{cases} (2*f(i-1)_1 \quad , f(i-1)_2 \;) & \text{ha } v[i]=f(i-1)_2 \\ (2*f(i-1)_1+1,\ 6-(f(i-1)_2+ v[i])\;) & \text{ha } v[i]\neq f(i-1)_2 \end{cases}$$



## *Heurisztikák a 8-as (15-ös) tologató játékra*

- Rossz helyen levő cellák száma:

  $W(a) = \sum_{\substack{i,j \\ a_{i,j} \neq cél_{i,j}}} 1$
- Cellák célbeli helyüktől vett minimális távolságainak összege (Manhattan):

  $P(a) = \sum_{i,j} |i - célbelisor(a_{i,j})| + |j - célbelioszlop(a_{i,j})|$

  $célbelisor(a_{i,j})$ ~ az $a_{i,j}$ célállapotbeli helyének sora

  $célbelioszlop(a_{i,j})$ ~ az $a_{i,j}$ célállapotbeli helyének oszlopa
- „Széleken levő cellák legyenek jók” (frame):
  - Hány olyan cella van a szélen, amelyiket nem a célbeli szomszédja követ az óra járásával megegyező irányban
  - Hány sarokban nincs még a cél szerinti cella

## *Heurisztikák a Fekete-fehér kirakóra*

- Inverziószám:
  *I(v)= minimálisan hány csere kell ahhoz, hogy minden fehér minden feketét megelőzzön*
- Duplázott inverziószám:
  *D(v)= 2*I(v)*
- Módosított inverziószám:
  $M(v)= 2*I(v) - \sum_{i=1,\dots} X(v[i..i+2] =$ „_BW” ∨ „BW_”)



## *Gráfszínezés*

- Feladat:
  - Adott egy véges egyszerű gráf, amelynek a csúcsait a lehető legkevesebb szín felhasználásával úgy kell kiszínezni, hogy a szomszédos csúcsok eltérő színűek legyenek.
- Cél:
  - A gráf csúcsainak olyan minimális osztályozását (egy osztályba tartozó csúcsok azonos színűek) keressük, ahol egy osztályhoz tartozó csúcsok között nem vezet él.



## *Gráfszínezés állapottér-reprezentációja*

- *Állapot*: a csúcsoknak a célhoz képest gyengített osztályozása, ahol
  - egy osztályhoz tartozó csúcsok között lehetnek élek
  - a gráf maximális fokszámánál több osztály van, de lehet egy osztály üres is
- *Művelet*: Egy osztályból egy csúcsot egy másik osztályba helyezünk.
- *Kezdő állapot*: tetszőleges
- *Célállapot*: a legjobb osztályozás
- *Állapot-gráf*: Exponenciális méretű az eredeti gráf csúcsszámához mérve.



## *Gráfszínezés célfüggvénye*

- Annál jobb egy $(O_1, \dots, O_k)$ osztályozás,
  - minél több csúcs van az első néhány osztályában (ezáltal minél kevesebb nem üres osztálya van), és
  - minél kevesebb egy osztályon belül vezető élek száma.
- $f(n)=\Sigma_j w_j( \lambda |A(O_j)| - |O_j| )$
  - ahol $A(O_j)$ az $O_j$ osztálybeli élek halmaza
  - a $w_j>0$ számok szigorúan növő sorozatot alkotnak.
- Könnyű a „szomszédos” osztályozás célfüggvény-értékét kiszámolni.



## *Feladatok*

- Korábban már említett problémák állapottér-reprezentációjának ismeretében ítéljük meg
  - mennyire alkalmas a problématerük a lokális kereséshez?
  - adhatunk-e jó heurisztikát a célfüggvényhez?
- Problémák:
  - SAT probléma
  - Misszionárius-kannibál
  - Kancsók problémája
  - Huszár bejárás