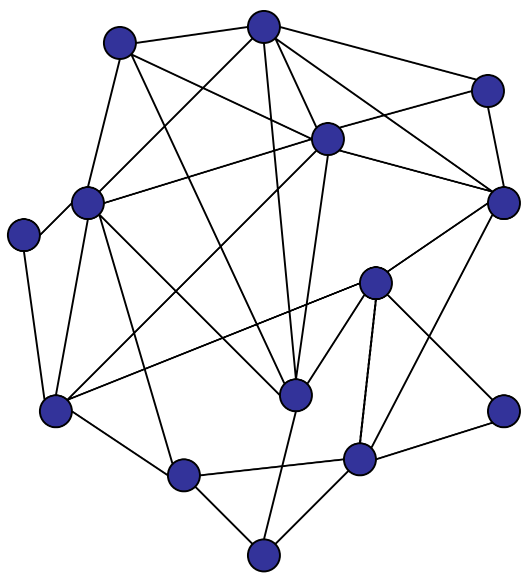
Heuristické řešení problémů

Seminář APS
Tomáš Müller
6. 7. 2002

Heuristické řešení problémů

- Popis několika základních metod
  - lokální prohledávání
  - branch and bound
  - simulated annealing, TABU
  - evoluční algoritmy
- Použití na problém obchodního cestujícího

Problém obchodního cestujícího (TSP)



- vrcholy – města
- hrany – cesty, ohodnocení
- nejkratší (nejlevnější) cesta
  - každé město navštíví právě jednou
- varianty
  - symetrie
  - trojúhelníková nerovnost
  - cesta mezi každými dvěma městy
- $|S| = {(n-1)!}/{2}$

Heuristické řešení problémů

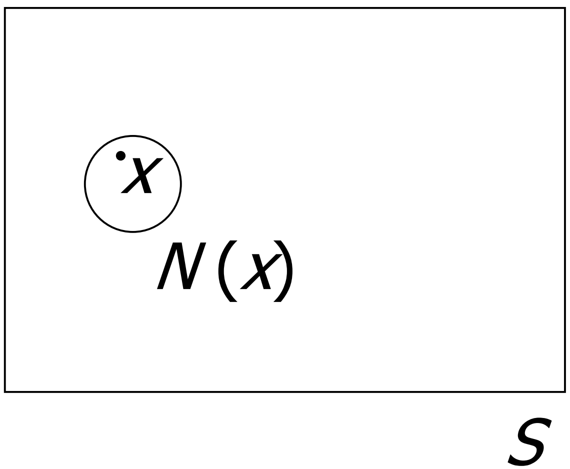
- reprezentace
  - pořadí navštívených měst – permutace
  - seznam hran
- cíl
  - nalezení cesty s minimální délkou
- ohodnocení
  - délka cesty

$eval(x)$

$compare(x, y)$

- řešení
  - $x \in F \subseteq S; \forall y \in F \ eval(x) \leq eval(y)$

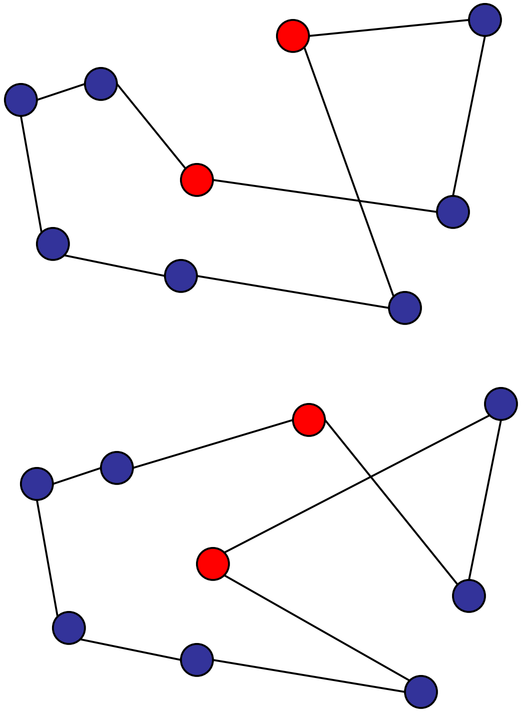
Lokální prohledávání



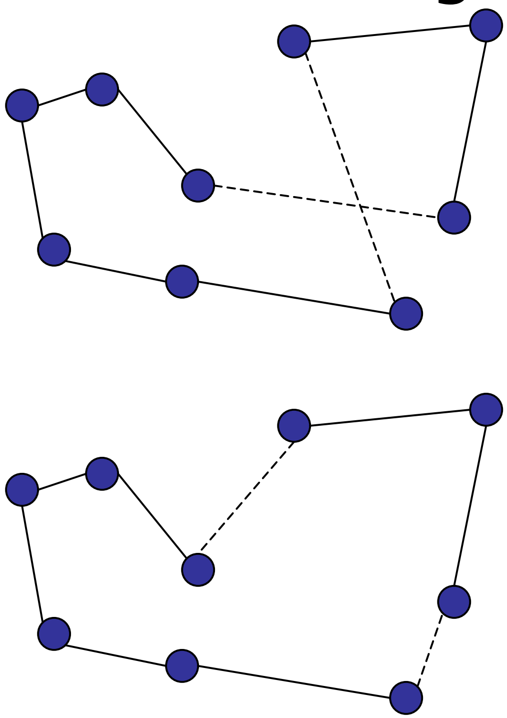
- množina sousedních řešení
  - operátor 2-swap
  - operátor 2-interchange
- best = **initial solution;**
  for i=1 step 1 until **MAX_TRIES** do begin
  s = **initial solution**;
  for j=1 step 1 until **MAX_ITERS** do begin
  s′ = **random neighbour**(s)
  if (eval(s′) **better** eval(s)) then s=s′
  end
  if (eval(s) **better** eval(best)) then best=s;
  end
  return best;

Lokální prohledávání – TSP

- 2-swap



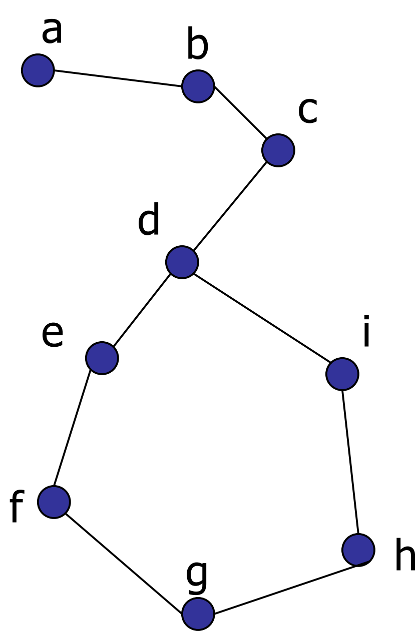
- 2-interchange



2-optimal řešení

Lin-Kernighan

- δ-path

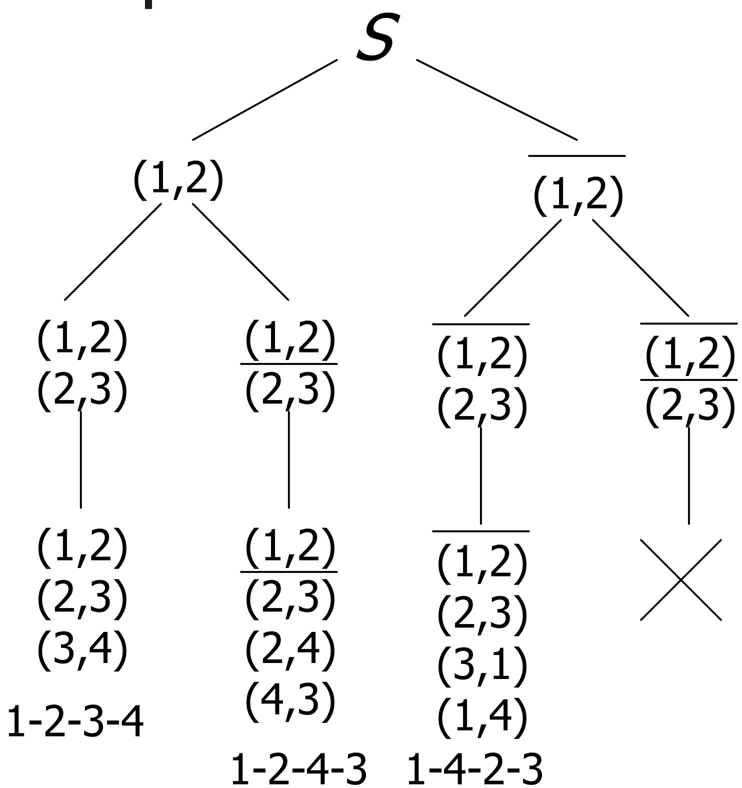


a-b-c-d-e-f-g-h-i-d

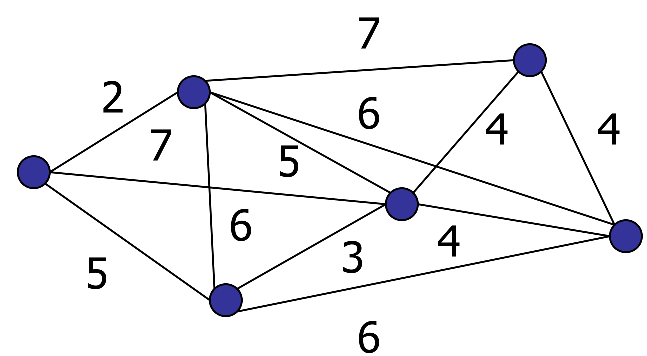
switch(d,e)

T= **random tour**; best_cost=**cost(T)**;
for **each node** k of T do
  for **each edge** (i,k) **of** T begin
C: if **there is** j!=k **such that** dist(i,j)<=dist(i,k)
      then p = **δ-path (T,i,k,j)** *// (i,k) -> (i,j)*
      else **goto** B;
A: T = **construct tour (p);**
    if (**eval(T)<=best_cost**)
      then **store** T; best_cost=**eval**(T);
    if there exists a switch of p resulting in
      δ-path with cost not greater then eval(T)
        then p = **that switch**; **goto** A;
B: if (**eval(T)<best_cost**)
      then **store** T; best_cost=**eval**(T);
    if **there remain untested node/edges comb.**
      then **goto** C;
  end

Branch and bound - TSP



- hranová reprezentace cesty
- odhad řešení
  - $$\left[\begin{matrix}(2+5)+(2+5)+(4+4)+\\(4+4)+(3+4)+(3+5)\end{matrix}\right]\Big/2$$



Stochastic hill-climer

- pravděpodobnost přijetí řešení

$$p = \frac{1}{1 + e^{\left(eval(s) - eval(s')\right)/T}}$$

T	$e^{-13/T}$	*P*
1	2e-7	1.00
5	0.0743	0.93
10	0.2725	0.78
20	0.52	0.66
50	0.77	0.56
10e10	0.9999	0.5…

- (maximalizace)

```
s = initial solution;
for i=1 step 1 until MAX_ITERS do begin
    s' = random neighbour(s)
    if (random(0,1)<1/(1+exp[(eval(s)-eval(s'))/T]) )
        then s=s';
end
return s;
```

Simulated annealing

- teplota T
- harmonogram chladnutí g(T,i)

$$p = e^{\left(eval(s') - eval(s)\right)/T}$$

s = **initial solution; i = 0;**
repeat
    repeat
        s’ = **random neighbour**(s)
        if (eval(s’) **>** eval(s)) or
            (**random[0,1)<exp[(eval(s’)-eval(s))/T]**)
            then s=s’;
    until (**terminal-condition**)
    T = g(T,i); i++;
until (**halting-condition**)
return s;

Tabu search

- paměť (pevné délky) – tabu list
  - n posledních ohodnocení
  - např. list (proměnná, hodnota)
- aspirační kritéria
- TSP: Knox alg.
  - 2-interchange
  - paměť – nové hrany v operátoru 2-interchange
  - tabu – pokud jsou obě v tabu listu
  - délky tabu-listu 3n, max. itrací kn^4

Evoluční algoritmy

- populace
  - cesty obchodního cestujícího
- nová populace
  - výběr a křížení několika jedinců (křížení jejich genů)
    - náhodný výběr
    - výběr dle „kvality“ jedinců (např. tournament selection)
  - mutace vyniklých jedinců – variation operator
    - náhodná změna genu
    - heuristika
    - optimalizace jedince
  - výběr cílové populace
    - pouze z nových jedinců, ze všech jedinců

Evoluční algoritmy - křížení

- 2 rodiče – operátor crossover
  - permutace

cut points

$$P_1 = [13|562|478] \qquad P_2 = [36215487] \longrightarrow O_1 = [56231487]$$

- 1 rodič
  - heuristic crossover
    - náhodně zvol hrany (i,j) (k,m), pokud

$$dist(i,j) + dist(k,m) > dist(i,m) + dist(k,j)$$

    - vyměn zvolené hrany za (i,m) a (k,j)

Evoluční algoritmy - křížení

- PMX (partially-mapped)

$$\begin{matrix} P_1 = [123|4567|89] \\ P_2 = [452|1876|93] \end{matrix} \longrightarrow \begin{matrix} O_1 = [xxx|1876|xx] \\ O_2 = [xxx|4567|xx] \end{matrix} \quad \begin{matrix} 1 \leftrightarrow 4 & 8 \leftrightarrow 5 \\ 7 \leftrightarrow 6 & 6 \leftrightarrow 7 \end{matrix}$$

$$\longrightarrow \begin{matrix} O_1 = [x23|1876|x9] \\ O_2 = [xx2|4567|93] \end{matrix} \longrightarrow \begin{matrix} O_1 = [423|1876|59] \\ O_2 = [182|4567|93] \end{matrix}$$

- OX (order)

$$\begin{matrix} P_1 = [123|4567|89] \\ P_2 = [452|1876|93] \end{matrix} \longrightarrow \begin{matrix} O_1 = [xxx|4567|xx] \\ O_2 = [xxx|1876|xx] \end{matrix} \longrightarrow \begin{matrix} O_1 = [218|4567|93] \\ O_2 = [345|1876|92] \end{matrix}$$

- CX (cycle)

$$\begin{matrix} P_1 = [123|4567|89] \\ P_2 = [412|8769|35] \end{matrix} \longrightarrow \begin{cases} O_1 = [1xxxxxxxx] = [1xx4xxxxx] \\ = [1xx4xxx8x] = [1234xxx8x] = [123476985] \end{cases}$$

Evoluční alg. + local search

```
P(0)=initial population; t=0;
apply local optimizer to P(0);
evaluate P(0);
while (not termination-condition) do
begin
   t++;
   select P(t) from P(t-1);
   alter P(t);
   apply local optimizer to P(t);
   evaluate P(t);
end;
```

- lokální prohledávání
  - Lin-Kernighan
  - 2-opt, 3-opt
- alter
  - rekombinace
  - mutace
- Padberg-Rinaldi
  - OX operátor

Iner-over operator

```
P=initial population;
while (not termination-condition) do begin
  for each tour T from P do
    T′<-T;
    select (randomly) a city c from T′;
    repeat
      if (rand()<=p)
        then select c′ from remaining cities in T′;
        else select (randomly) any tour from P;
             c′ = next city after c in selected tour;
      invert the selection from the next city c to the city c′ in T′;
      c = c′;
    if (eval(T′)<=eval(T)) T<-T′;
  end;
end;
```

- př.

c=3 c′=5

$$T = [2,3,9,4,1,5,8,6,7]$$

$$\downarrow$$

$$T' = [2,3,5,1,4,9,8,6,7]$$

TSP - benchmarks

- Náhodné rozmístění měst (Euklidovský prostor)
  - očekávaná délka nejkratší cesty $L^* = k\sqrt{n \cdot R}$
  - Held-Karp $n \geq 100$

$$k = 0.70805 + \frac{0.52229}{\sqrt{n}} + \frac{1.31572}{n} - \frac{3.07474}{n\sqrt{n}}$$

  - Bonomi-Lutton $k = 0.749$
- Kolekce veřejně dostupných testovacích úloh
  - [http://www.iwr.uni-heidelberg.de /iwr/comopt/soft/TSPLIB95/TSPLIB.html](http://www.iwr.uni-heidelberg.de/iwr/comopt/soft/TSPLIB95/TSPLIB.html)

Závěř

- Čerpáno z knihy
  - Z.Michalewicz, D.B.Fogel

  How to Solve It: Modern Heuristics

  Springer, 2000