

Optimalizační algoritmy inspirované chováním mravenců

- Motivace a biologická analogie
- ACO metaheuristic
- Ant system a jeho modifikace
- Specifikace problémů vhodných pro ACO
- Aplikace

Motivace

- **NP-hard problémy** – časová náročnost nalezení optimálního řešení roste exponenciálně s velikostí problému
 - Kombinatorické problémy
- Jednou z možností jsou **aproximační metody**, které naleznou řešení alespoň blízké optimálnímu v rozumném čase
- **Aproximační metody**
 - **Lokální prohledávání/optimalizace**
 - Iterativní zlepšování počátečního řešení až do uvážnutí v lok. ext.
 - **Konstruktivní algoritmy**
 - Budování řešení s využitím heuristické znalosti
- **Ant Colony Optimization (ACO)** – rozšíření tradičních konstruktivních heuristik o schopnost využití zkušeností získaných v průběhu řešení daného problému

Lokální prohledávání

Procedure *IterativeImprovement*($s \in S$)

$s' = \text{Improve}(s)$

while $s' \neq s$ **do**

$s = s'$

$s' = \text{Improve}(s)$

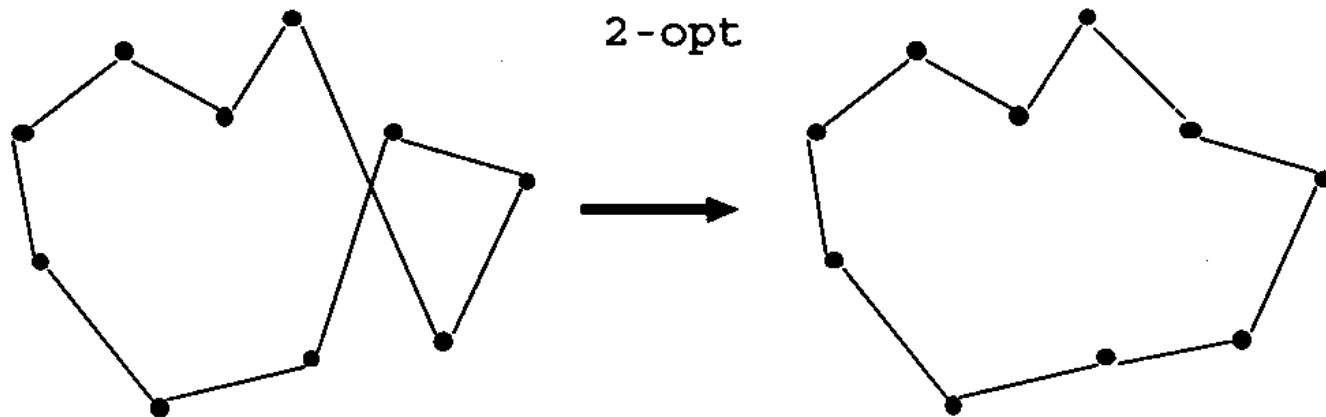
end

return s

end

Problémy

- často uvázne v lokálním extrému
- výsledek lokální optimalizace velice závisí na počátečním řešení



Konstruktivní algoritmy

- Pracují s částečně rozpracovaným řešením, snaží se jej rozšířit nejvýhodnějším způsobem – využívají hladový algoritmus

Procedure *GreedyConstructionHeur*

$s_p = \text{empty_solution}$

while not complete(s_p) **do**

$e = \text{GreedyComponent}(s_p)$

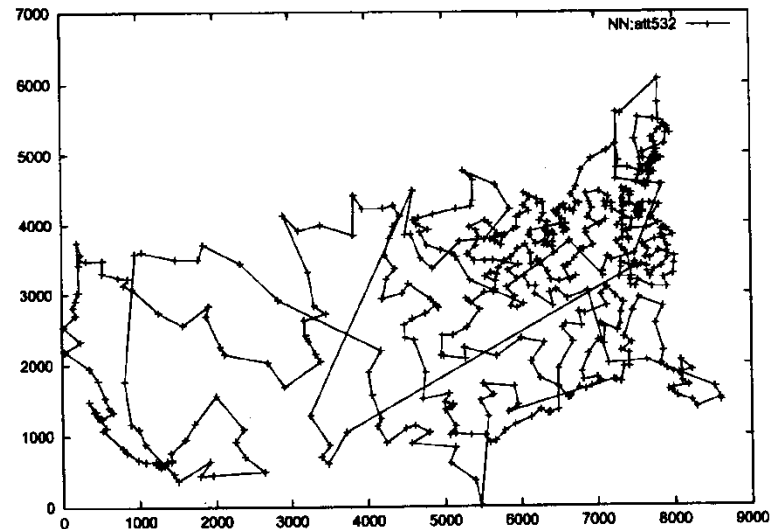
$s_p = s_p \otimes e$

end

return s_p

end

TSP: přidávání nejbližšího souseda



- **Výhody/nevýhody**

+ rychlé

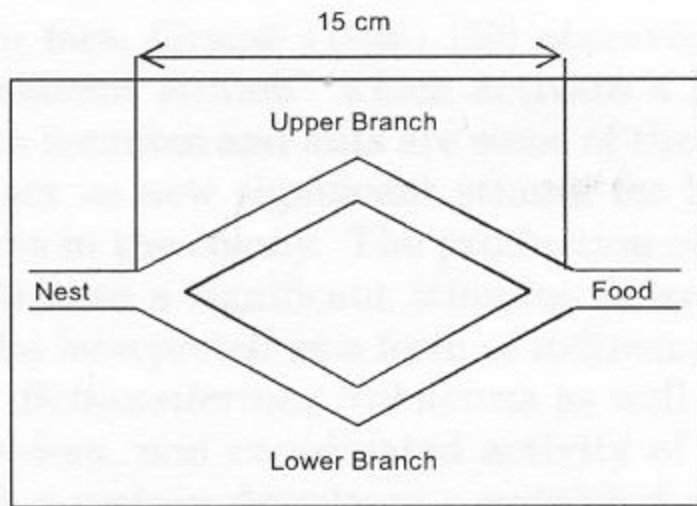
- mnohdy generují ne moc dobrá řešení (ikdyž o přijatelné kvalitě)
- generuje pouze velice omezený počet řešení
- rozhodnutí učiněná na začátku výpočtu značně omezují možnosti v pozdějších fázích

Mravenčí algoritmy: biologická inspirace

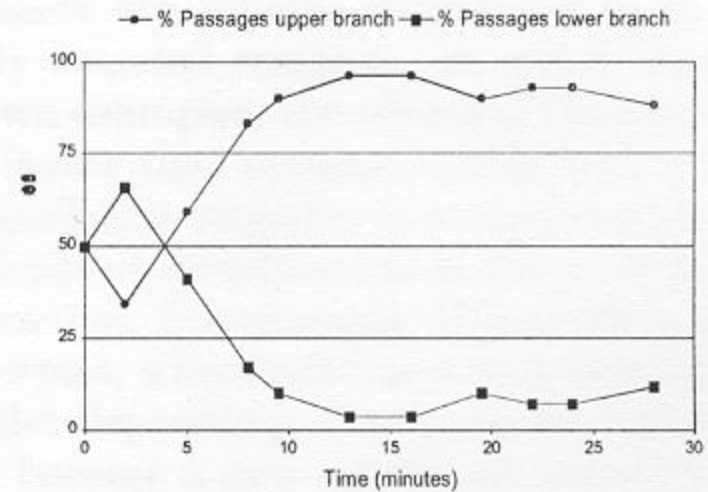
- **Inspirované chováním mravenčích kolonií**
 - Sociální hmyz – chování směřuje k zachování kolonie
 - Jednoduché chování jedinců × složité chování kolonie
- Schopnost nalézt nejkratší cestu od zdroje potravy k mraveništi pomocí **vzájemné komunikace prostřednictvím Feromonu**
 - **Zápis** - mravenci ukládají feromon po cestě za potravou
 - **Čtení** – mravenci umí detekovat feromony (a rozlišovat mezi různými koncentracemi) zanechané ostatními soukmenovci a vybírá si cestu s největší koncentrací feromonu
 - **Emergence** – toto chování aplikované celou kolonií mravenců může vést k emergenci nejkratší cesty

Experimenty se skutečnými mravenci

- Deneuborg et al. (mravenci *Linepithema humile*)
- Mraveniště oddělené od zdroje potravy dvojitým mostem
 - Obě cesty stejně dlouhé
 - Na začátku není žádný feromon na žádné z cest
 - Po čase jedna z cest díky náhodným fluktuacím převáží



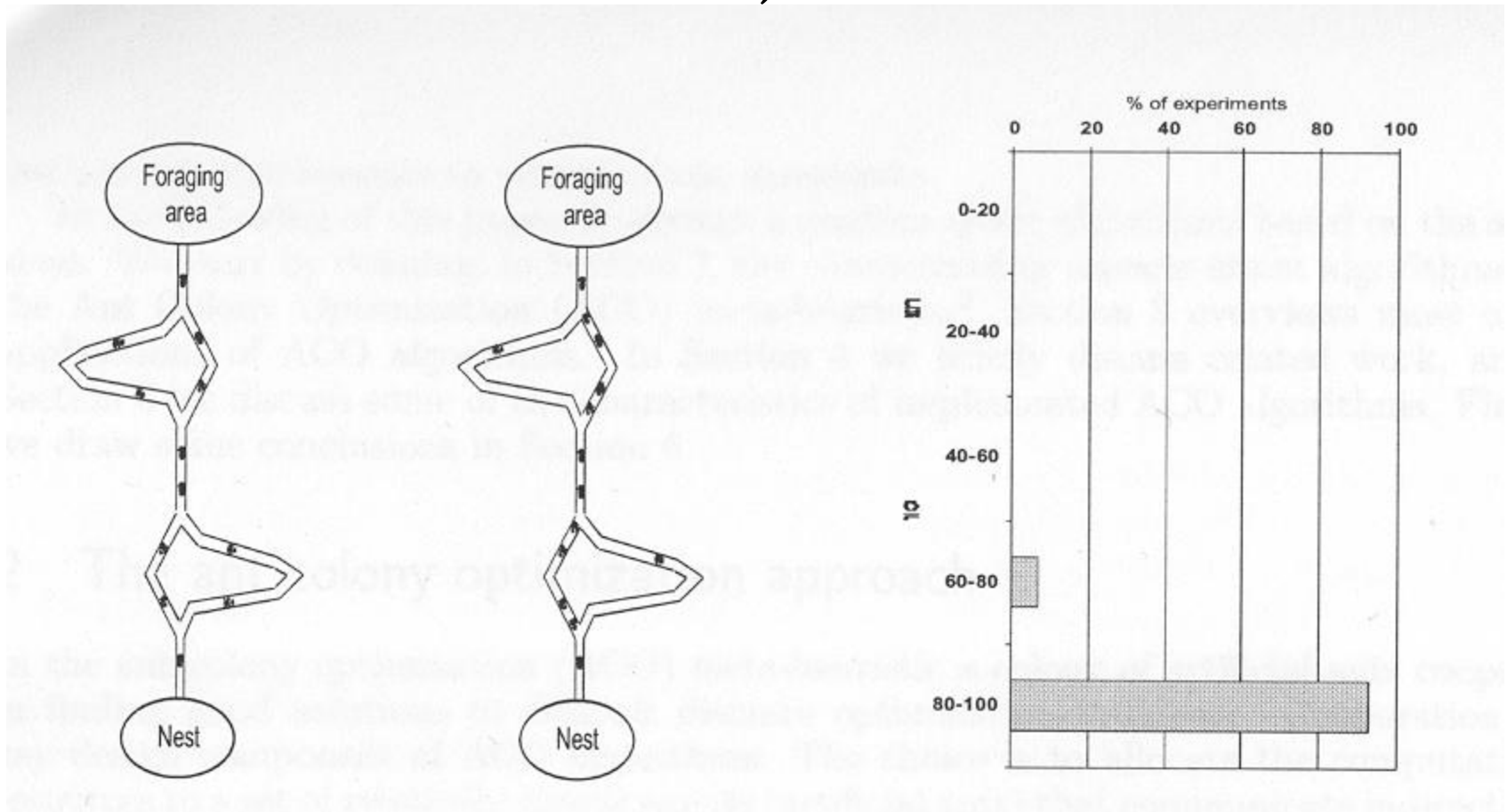
(a)



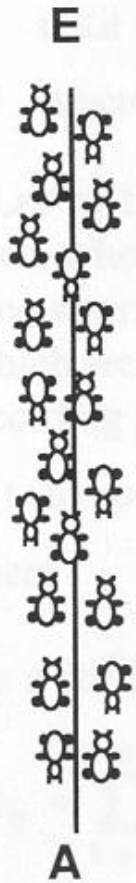
(b)

Mosty s různě dlouhými variantami

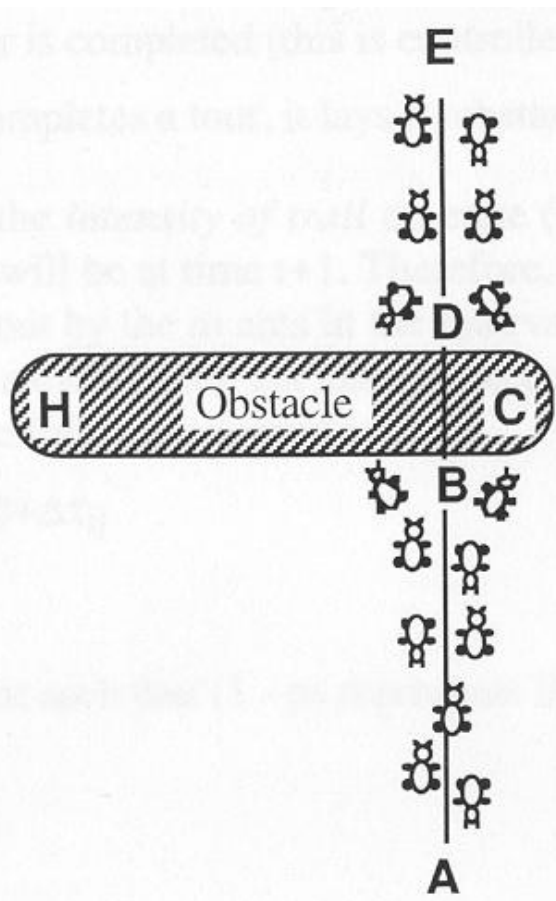
- Vliv náhodných fluktuací je značně redukován a většina mravenců nakonec volí tu nejkratší cestu



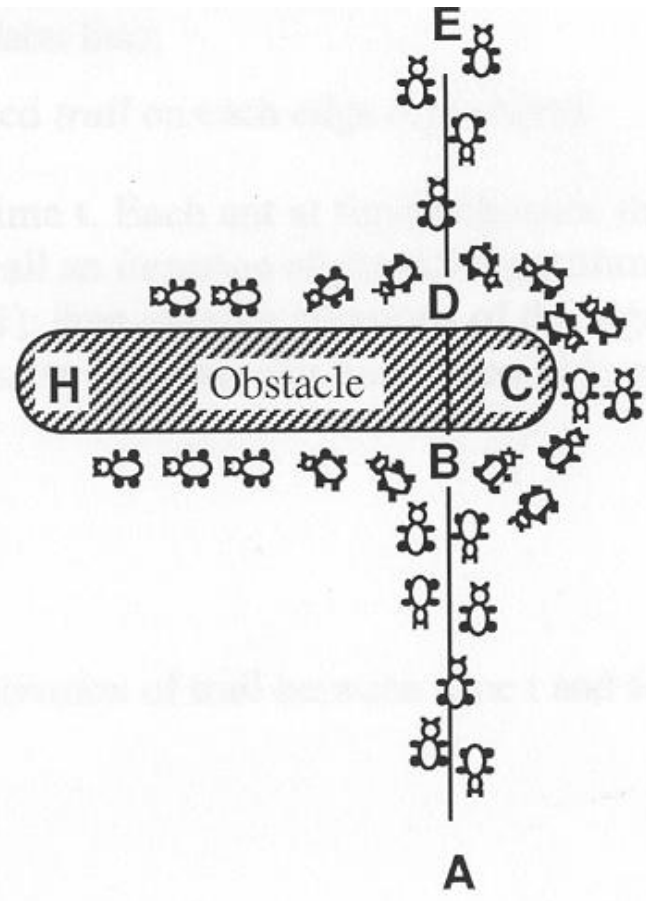
Příklad



a)



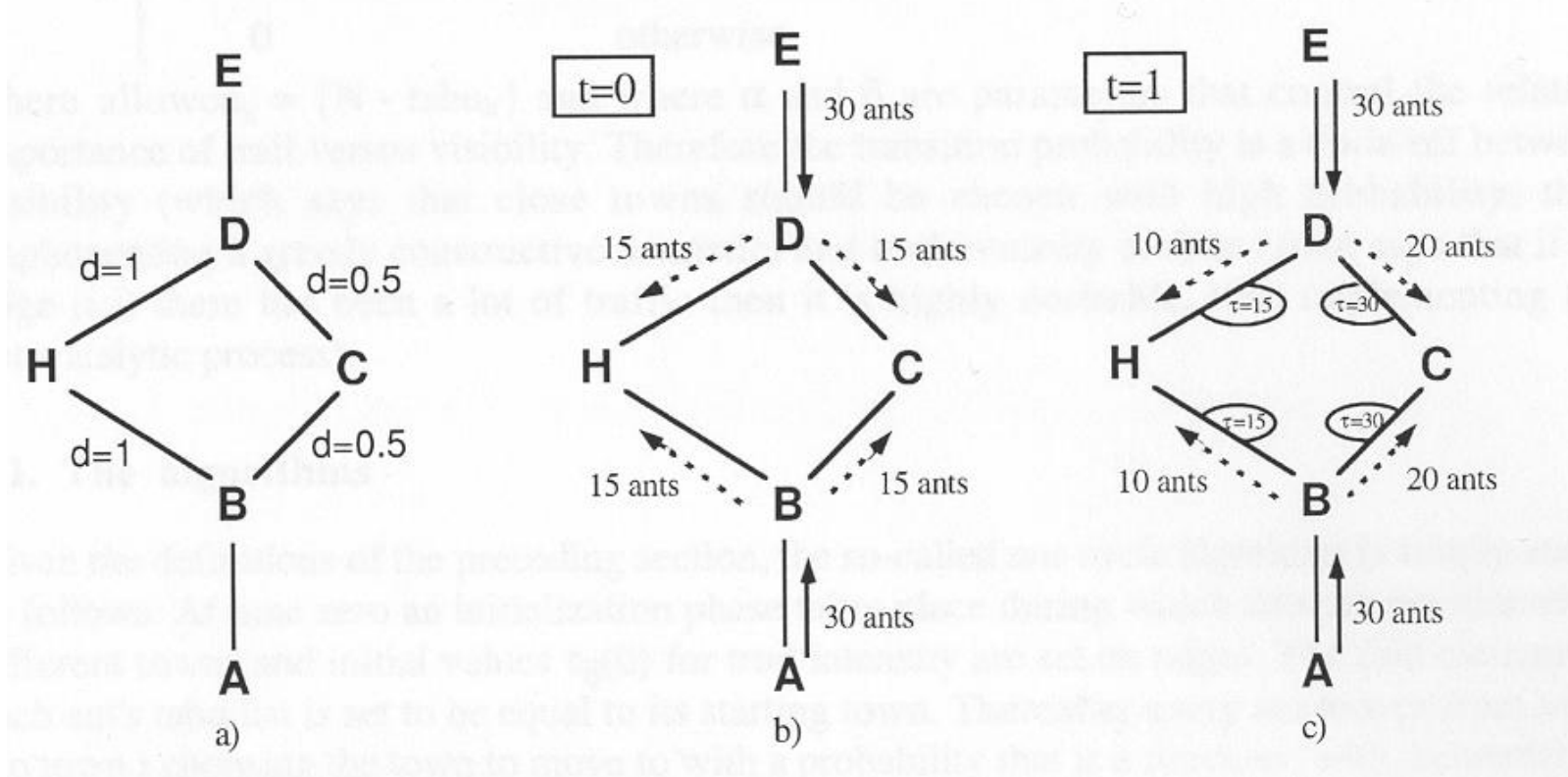
b)



c)

Příklad

- V každém kroku jde 30 nových mravenců z A do B, 30 z E do D
 - mravenec jde rychlost 1 s^{-1}
 - za jednotku času položí 1 jednotku feromonu



Stigmergie

- Chování kolonie – **distribuovaný optimalizační algoritmus**, kde každý mravenec přispívá svou trochou do mlýna (umí zkonstruovat celou cestu).
- **Stigmergie** - nepřímá komunikace mravenců zprostředkovaná pomocí feromonu
 - **Fyzikální podstata informace** uvolněné komunikujícími mravenci – modifikace místa navštíveného mravencem
 - **Lokálnost zanechané informace** – je viditelná pouze mravenci, kteří navštíví místo s feromonem nebo jeho okolí
 - **Autokatalytický mechanismus** – čím silnější záznam, tím více se bude dále posilovat
- **Odpařování feromonu** – realizuje zapomínání, které zabraňuje předčasné konvergenci k suboptimálnímu řešení

Živí mravenci

- Téměř slepí
- Neschopní samostatně řešit složité úlohy
Neschopni určit nejkratší cestu s mraveniště ke zdroji potravy a zpět.
- Používají stigmergickou komunikaci prostřednictvím feromonu
S velkou pravděpodobností následují silnou feromonovou stopu.

Umělí mravenci

- **Podobnosti** se skutečnými mravenci:
 - Kolonie kooperujících mravenců
 - Feromonová stopa a stigmergie
 - Pravděpodobnostní rozhodování, lokálnost strategie
 - Apriorní informace daná problémovou specifikací
 - Lokální modifikace stavů, indukované předcházejícími mravenci
- **Rozdílnosti** oproti skutečným mravencům:
 - Diskrétní svět
 - Vnitřní stavy – osobní paměť zaznamenávající doposud vykonané akce
 - Nejsou zcela slepí
 - Množství zanechaného feromonu je funkcí kvality nalezeného řešení
 - Problémově závislé časování ukládání feromonu
 - Extras – lokální optimalizace, backtracking

Ant Colony Optimization Metaheuristic

- **ACO** může být použito pro řešení jakéhokoliv diskrétního optimalizačního problému, pro který lze použít nějakou konstruktivní heuristickou proceduru
- **Mravenci** použítí v ACO fungují jako stochastické konstruktivní procedury, které vytvářejí řešení iterativním přidáváním komponent do částečně rozpracovaného řešení, přičemž při výběru každé další komponenty uvažují
 - heuristickou informaci o řešeném problému, která směřuje výpočet ke slibným řešením a
 - zkušenosti získané všemi mravenci od začátku výpočtu, reprezentované feromonovými stopami, které se během výpočtu neustále adaptují
- **Stochastická složka** umožňuje vygenerování velkého počtu různých řešení

Ant System (AS) - definice

- **Problém:** Mějme n měst, cílem je nalézt nejkratší uzavřenou cestu, která prochází všemi městy právě jednou
 - uvažujeme **úplný graf**
 - d_{ij} je euklidovská vzdálenost z města i do města j
- **Definice**
 - m je celkový počet mravenců
 - $\tau_{ij}(t)$ je intenzita feromonu na spojnici (i, j) v čase t
 - η_{ij} je viditelnost vyjádřená hodnotou $1 / d_{ij}$
 - $(1-\rho)$ reprezentuje faktor vypařování, ρ je konstantní během výp.
 - $tabu_k$ je dynamicky rostoucí vektor, obsahuje seznam měst, kterými mravenec prošel v dané cestě
 - **iterace AS** je posun všech m mravenců o jeden krok
 - **cyklus AS** se skládá z n iterací během nichž mravenci dokončí cestu

AS – ukládání feromonu

- $\tau_{ij}(t+n) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij}$
- $\Delta\tau_{ij} = \sum_k \Delta\tau_{ij}^k$
- $\Delta\tau_{ij}^k = \begin{cases} Q/L_k, & \text{jestliže } k\text{-tý mravenec použil hranu } (i, j) \\ 0, & \text{když ji nepoužil.} \end{cases}$

kde

$\Delta\tau_{ij}^k$ je množství feromonu položeného na hranu (i, j) k -tým mravencem v intervalu $(t, t+n)$

Q je konstanta

L_k je délka cesty, kterou urazil k -tý mravenec

ρ musí být menší než 1, jinak by se neomezeně kumuloval feromon

$\tau_{ij}(0)$ je nastaveno na malé kladné hodnoty

AS - pravděpodobnost přechodu z i do j

- $p_{ij}^k(t) = \begin{cases} [\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta / \sum_{l \in \{N - \text{tabu}_k\}} [\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta, & \text{když } j \in \{N - \text{tabu}_k\} \\ 0, & \text{jinak.} \end{cases}$

kde

$$l \in \{N - \text{tabu}_k\}$$

α, β určují relativní důležitost feromonu a viditelnosti

- Pravděpodobnost je počítána jako kompromis mezi
 - **viditelností**, která říká, že bližší města by měla být preferována a
 - **intenzitou feromonu**, která říká, že často používaná hrana je zřejmě žádoucí

AS – algoritmus *Ant-cycle*

- *Ant-cycle*:

1. Inicializace

- času: $t=0$
- počtu cyklů: $NC=0$
- feromonu: $\tau_{ij}(t)=c$
- umístění m mravenců na n měst

2. Inicializace *tabu* seznamů

3. Činnost mravenců

- iterativní vybudování cesty
- výpočet délky cesty L_k pro všechna $k \in (1, \dots, m)$
- aktualizace nejkratší nalezené cesty
- výpočet $\Delta\tau_{ij}^k$ a aktualizace $\tau_{ij}(t+n)$

4. Inkrementace diskrétního času

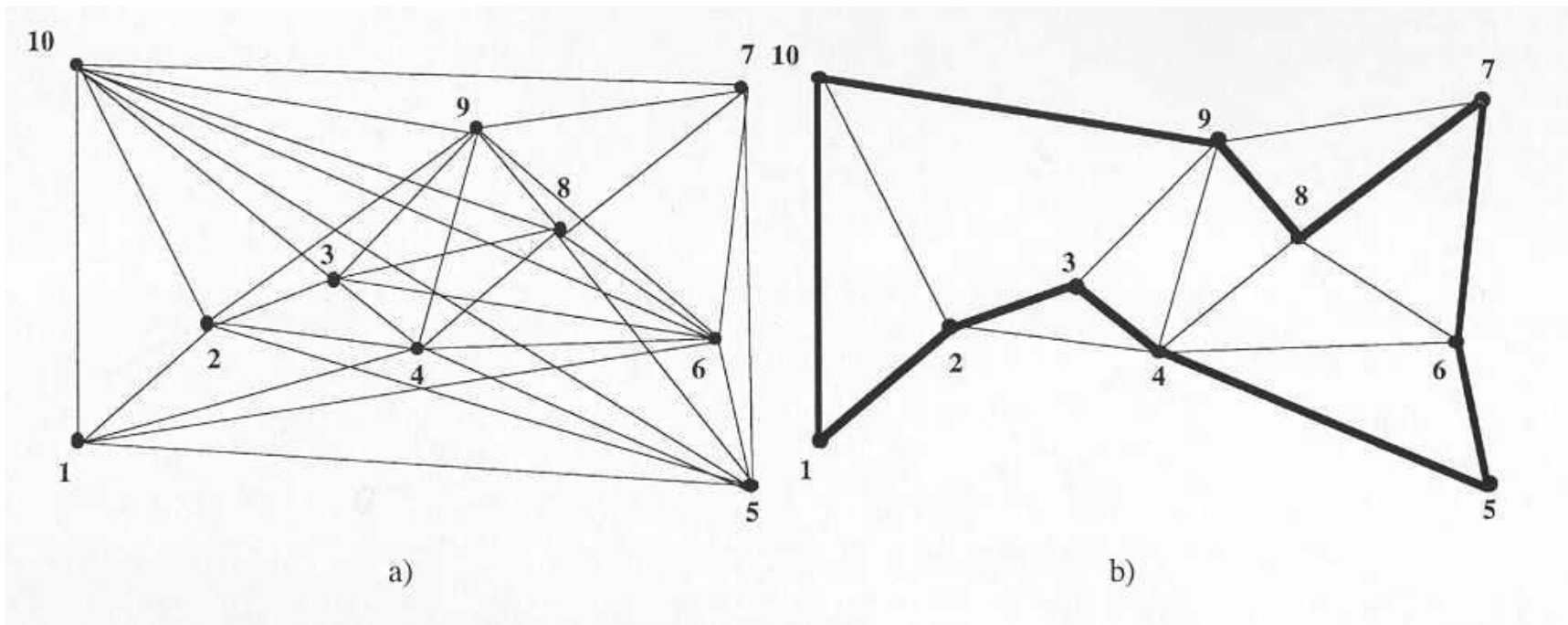
- $t = t+n, NC = NC+1$

5. If($NC < NC_{max}$) then goto step 2

else stop

AS – nastavení ρ

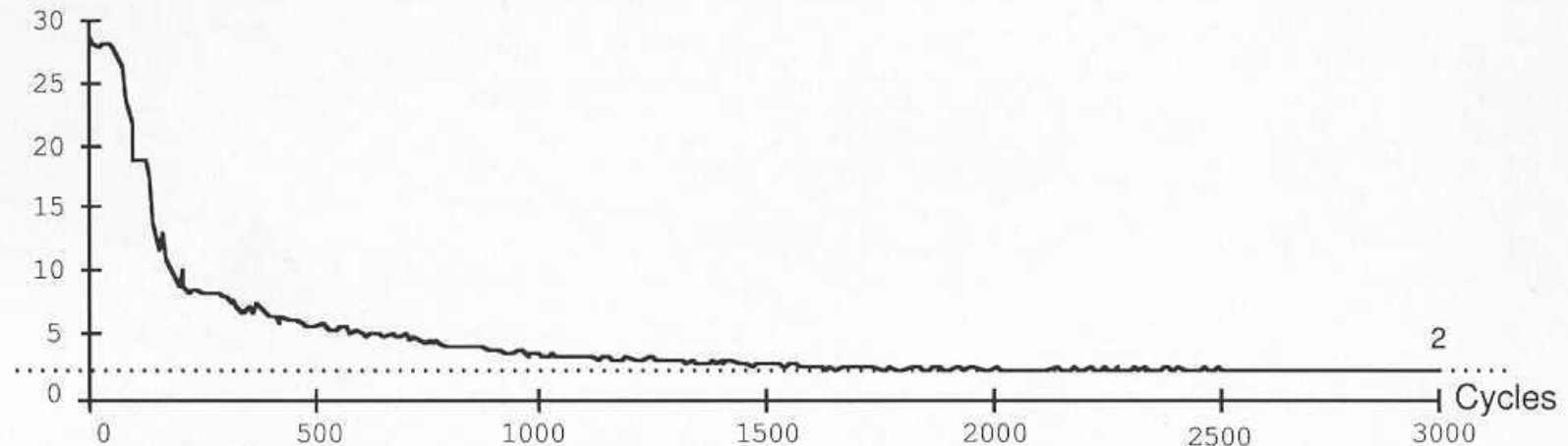
- Optimální hodnota ρ je 0.5
 - po úvodním hladovém prohledávání je třeba dát prostor adaptaci globální informace uložené v $\tau_{ij}(t)$
 - jinými slovy, je třeba zapomenout



AS – význam α , β

- význam α , β (optimální $\alpha=1$, $\beta=2$)
 - Vysoká hodnota α znamená, že intenzita feromonu je prioritní, takže mravenci volí cestu, kterou šli jejich předchůdci
 - Nízká nebo nulová α mění metodu na stochastický restartovaný hladový algoritmus

Average node branching



- **Stagnace** – větvící faktor 2, všichni mravenci jdou stejnou cestou

AS – elitismus

- Intenzita feromonu je posílena na hranách, které leží na nejkratší cestě
 - Množství přidaného feromonu: $e \cdot Q/L^*$,
kde e je počet „elitních“ mravenců a L^* je nejkratší cesta
 - **Ale pozor na předčasnou konvergenci**

Obecná ACO metaheuristika

procedure *ACO metaheuristics*

ScheduleActivities

 ManageAntActivity()

 EvaporatePheromone() // zapomínání

 DaemonActions() {optional} // centralizované akce

end ScheduleActivities

end *ACO metaheuristics*

- Nalezení vhodné grafové reprezentace
- Definování pozitivní zpětné vazby
- Výběr konstruktivní heuristiky
- Model práce s omezeními – *tabu* seznamy u TSP

Shrnutí: Chování umělého mravence

- **Mravenec funguje jako** konstruktivní procedura, která vytváří řešení procházením konstrukčního grafu $G=(C, L)$, kde komponenty C jsou vrcholy grafu a L reprezentuje hrany (úplně propojený graf).
- **Komponenty $c_i \in C$ a spoje $l_{ij} \in L$ mohou mít přiřazenu feromonovou stopu τ , která reprezentuje dlouhodobou paměť** (zkušenost mravenců) upravovanou mravenci a heuristickou hodnotu η , která reprezentuje apriorní informaci o řešení problému (odhad ceny spojené s rozšířením daného stavu). Tyto hodnoty používá mravenec při rozhodování o následujícím tahu.
- **Každý mravenec má následující vlastnosti:**
 - Prohledává graf G s cílem najít optimální proveditelné řešení
 - Má paměť M^k , která uchovává doposud provedené tahy. Využívá ji pro
 - vytváření proveditelných řešení (implementace omezení)
 - ohodnocování nalezeného řešení (rekonstrukce celé cesty TSP)
 - zpětné trasování nalezené cesty během něhož ukládá feromon
 - Vybírá tahy pomocí pravděpodobnostního pravidla, které je funkcí
 - lokálně dostupného feromonu, paměti M^k a omezení Ω
 - Feromon může být upraven při každém přidání komponenty do řešení (online step-by-step pheromone update) nebo najednou až po zkompletování celého řešení (delayed pheromone update)

Applications of ACO algorithms

- **Static problems**
 - Traveling salesman
 - Quadratic assignment
 - Job-shop scheduling
 - Vehicle routing
 - Graph colouring
 - Shortest common supersequence
- **Dynamic problems**
 - Network routing

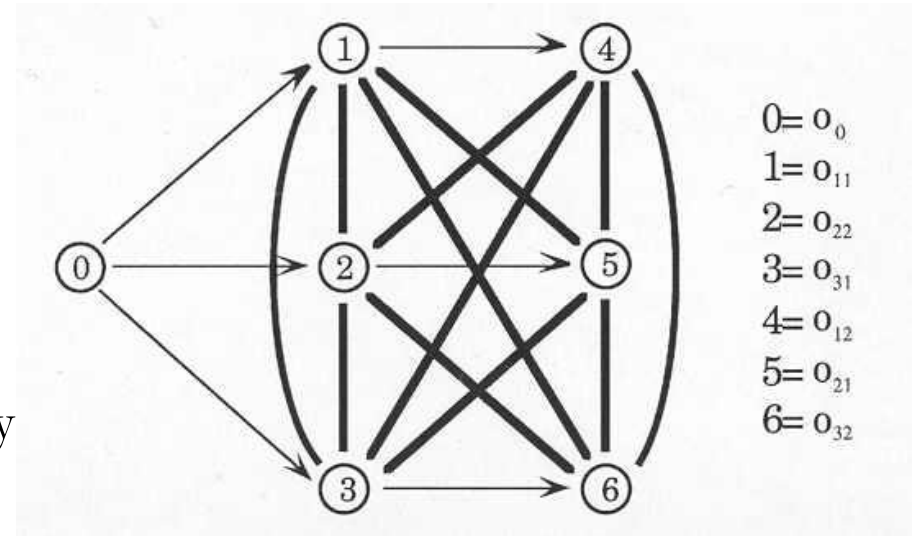
Aplikace ACO na JSP

- **Definice problému:**
 - M strojů a J úkolů
 - j -tý úkol se skládá z uspořádané posloupnosti operací z množiny $O = \{\dots, o_{jm}, \dots\}$, celkový počet operací je $N = |O|$
 - každá operace má danou délku trvání
 - cílem je nalézt nejkratší rozvrh prací, bez konfliktů, tj. žádné dva úkoly nebudou naplánovány současně na stejném stroji

Aplikace ACO na JSP

- **Implementace**

- Orientovaný graf $Q=(O', A)$,
kde $O'=O \cup \{o_0\}$ a A je množina hran spojující o_0 s první operací každého úkolu a realizující úplné propojení operací z O , vyjma operací stejného úkolu.
- Ke každé hraně jsou přiřazeny váhy $\{\tau_{ij}, \eta_{ij}\}$
 - τ_{ij} intenzita feromonu
 - η_{ij} viditelnost, odvozená pomocí *Longest Processing Time* nebo *Shortest Completion Time* heuristik



- **Příklad**

- 3 úkoly, 2 stroje
- 1. stroj: o_1, o_3, o_5
- 2. stroj: o_2, o_4, o_6

Aplikace ACO na JSP

- **Průběh výpočtu:**

1. Na začátku jsou všichni mravenci v uzlu o_0
2. V každém kroku mravenec vybírá další tah z
 - množiny G_k , která obsahuje všechny uzly, které mají být navštíveny a
 - množiny S_k , která obsahuje uzly, které mohou být navštíveny v daném kroku

Př.: na začátku je $G_k = \{1, 2, 3, 4, 5, 6\}$ a $S_k = \{1, 2, 3\}$

3. Když je vybrán další uzel u , tak je
 - přidán do tabu seznamu a
 - vymazán z G_k a S_k
4. Pokud u nebyl poslední uzel daného úkolu, tak přidej do S_k bezprostředního následníka u
5. Opakuj kroky 2.-4. dokud $G_k = \emptyset$

Zdroje

[Dorigo et al., 1996]

Dorigo M., V. Maniezzo & A. Colorni (1996). The Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 26(1):29-41

[Dorigo & Gambardella, 1997]

Dorigo M. & L.M. Gambardella (1997). Ant Colonies for the Traveling Salesman Problem. *BioSystems*, 43:73-81.

[Dorigo et al., 1999]

Dorigo M., G. Di Caro & L. M. Gambardella (1999). Ant Algorithms for Discrete Optimization. *Artificial Life*, 5(2):137-172.

[Dorigo & Stützle, 2002]

M. Dorigo and T. Stützle, 2002. The ant colony optimization metaheuristic: Algorithms, applications and advances. In F. Glover and G. Kochenberger editors, *Handbook of Metaheuristics*, volume 57 of International Series in Operations Research & Management Science, pages 251-285. Kluwer Academic Publishers, Norwell, MA.

<http://iridia.ulb.ac.be/~mdorigo/ACO/ACO.html>