Ant Colony Optimization

I am lost! Where is the line?!
—A Bug's Life, Walt Disney, 1998

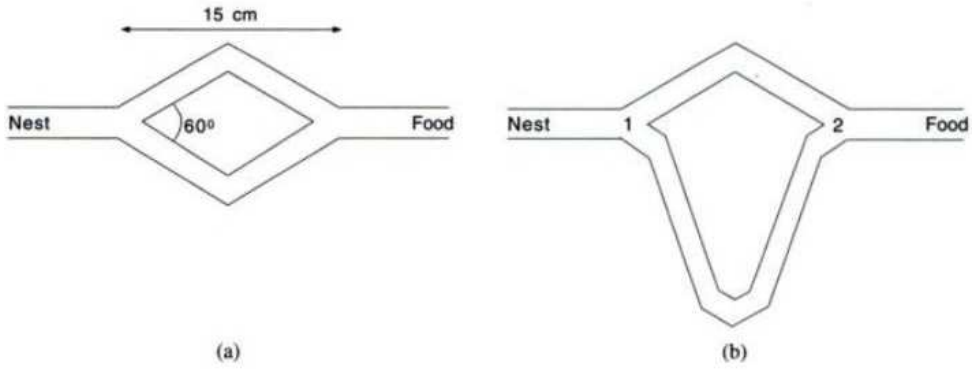
ACO je metaheuristika, shrnující poznatky ze studia společenstev různých druhů mravenců. Heuristické algoritmy postavené na ACO řeší úlohy způsobem podobným tomu, jakým mravenci v přírodě řeší své problémy. Mnoho aplikačních problémů patří do třídy $\mathcal{NP}$-těžkých a věří se, že optimální řešení těchto problémů nemůže být nalezeno algoritmem s polynomiální časovou složitostí. Proto je často k vyřešení velkých instancí $\mathcal{NP}$-těžkých problémů třeba použít aproximační metody, které dávají řešení blízké optimálnímu v relativně krátkém čase. ACO popisuje jeden dobrý přístup k řešení výpočetně, časově náročných problémů. Používá množství jednoduchých výpočetních agentů, kteří spolupracují při hledání globálního optimálního řešení. Časová složitost ACO-algoritmů je běžně kvadratická, může být i lineární.

Chování mravenců

Hledání potravy mnoha druhů mravenců je založeno na nepřímé komunikaci zprostředkované prostředím. Takový druh komunikace je nazýván *stigmergy*. Na cestě mezi mraveništěm a zdrojem potravy na zem mravenci kladou chemickou stopu - feromony. Tu cítí a pravděpodobnostně volí spíše cestu se silnou koncentrací feromonů. Po opuštění mraveniště mravenec prohledává okolí. Jakmile najde zdroj potravy, vrátí se zpět stejnou cestou, a poté jde (pravděpodobně) touto cestou znovu. Tím posílí feromonovou stopu a proto se stejnou cestou vydají i ostatní a tím ji opět posílí. Pokud stopa není udržována feromony vyprchají.

Double bridge experiment

Na obrázcích .a, .b je znázorněn double bridge experiment. Mravenci se časem pohybují pouze jednou cestou. Podle očekávání, v prvním případě k oboum stejným cestám skonvergují s přibližně stejnou pravděpodobností. Na druhém obrázku skonvergují ve většině případů ke kratší cestě. Nicméně pokusy ukázaly, že konečná hlavní cesta je příliš závislá na počátečních rozhodnutích. Tedy kolonie počítačových mravenců by si měla dát pozor na brzké nalezení suboptimálního řešení. Navíc nabídneme-li v .b nejprve pouze delší možnost a později umožníme u cestu kratší, mravenci zůstanou u delší. Vysvětlením je stálé posilování feromonové stopy na delší cestě v kombinaci s jejím pomalým vyprcháváním.



Obrázek 1: Levý obrázek - stejné délky, pravý obrázek - rúzné délky

Řešení via ACO

Aplikační oblastí ACO jsou problémy hledání hodnot diskrétních proměnných takových, že hodnota dané objektivní funkce je optimální. Aby bylo možné problém s ACO řešit, je nutné ho přeformulovat na problém prohledávání grafu. Kolonie umělých mravenců pak problém řeší náhodnými procházením úplného grafu s uzly z množiny komponent (města, zdroje). Výběr jejich cesty je ovlivněn množstvím feromonů v části grafu, ve které se v daný okamžik nachází, a heuristickou informací. Procházka každého mravence je *kandidátním* řešením, procházka splňující předem stanovená omezení je *přípustné* řešení. Někdy je užitečné nechat mravence stavět řešení jenom kandidátní. Příkladem je požadavek, že vůz se zbožím má být na cestě omezenou dobu, ovšem někdy je lepší zaplatit řidiči přesčas než vyslat nový vůz (porušení omezení je pak promítnuto do hodnoty objektivní funkce). Po dokončení jsou procházky vydnoceny, aktualizovány feromonové stopy a případně i aktualizovány statistiky zahrnující třeba nejlepší dosud nalezenou cestu. Kolonie mravenců takto stálým opakováním hledání, vyhodnocování a aktualizace postupně vylepšuje svou představu globálně optimálního řešení. S pomocí matematického aparátu bylo ukázáno, že kolonie po nějakém čase vždy najde cestu, po které se pak většina mravnců pohybuje. Tedy ACO algoritmy vždy skonvergují k nějakému řešení.

Kostra ACO-algoritmů je na obrázku . Algoritmus nejprve nechá všechny mravence současně a asynchronně projít graf problému. Poté nechá vyprchat část feromonů a na prošlé cesty přidá nové. Následovat mohou centrálně řízené akce, které nemohou být vykonané jedním mravencem. Činnost se opakuje do doby, než jsou splněny nějaké podmínky, např. současné řešení je dostatečně dobré nebo už dlouho nebylo nalezeno lepší.

```
procedure ACO_MetaHeuristic
  while(not_termination)
     generateSolutions()
     pheromoneUpdate()
     daemonActions()
  end while
end prodcedure
```

Obrázek 2: Kostra ACO algoritmů

Aplikace ACO na TSP

Problém obchodního cestujícího je úloha nalezení nejkratší cesty z domovského města přes množinu stanovených měst zpět do domovského města. Zadání je možné v bodech zapsat takto:
Graf Úplný, s navštěvovanými městy jako uzly, každá cesta $a_{i,j}$ má přiřazenu váhu (např. vzdálenost z města i do j).
Omezení Všechna města musí být navštívena práve jednou.
Feromony Množství feromonů na cestě z města i do j $\tau_{i,j}$ vypovídá o vhodnosti navštívení města i po j (podle znalostí v daném čase).
Heuristická informace[1] $\mu_{i,j} = 1/d_{i,j}$.

Ant System - velmi jednoduchý ACO algoritmus

Ant System je jeden z prvních ACO algoritmů. Hledání optimálního řešení TSP probíhá následujícím způsobem. Na začátku proběhne inicializace hodnot feromonů, které jsou nastaveny tak, aby byly mírně vyšší, než očekáváné hodnoty položené v prvních běžích algoritmu. Je tak eliminováno přílišné ovlivnění konečného výsledku prvotním průzkumem. Následuje tělo cyklu. Mraveneci jsou náhodně rozmístěni po uzlech grafu a začnou procházet graf. Uzel, ke kterému se vydají, volí náhodně. Pravděpodonost $p^k_{i,j}$, že mravenec k, aktuálně v městě i, zvolí město j je popsána rovnicí

[1] V knize Ant Colony Optimization[1] je za heuritickou informaci považována předem známá vlastnost grafového zadání použitelná pro řešení

1.

$$p_{i,j}^k = \frac{[\tau_{i,j}]^\alpha [\mu_{i,j}]^\beta}{\sum_{l \in \mathcal{N}_i^k} [\tau_{i,l}]^\alpha [\mu_{i,l}]^\beta}, \tag{1}$$

kde $j \in \mathcal{N}_i^k, \mathcal{N}_i^k$ je množina přípustných uzlů (neobshuje již navštívené, každý mravenec si pamatuje cestu, kterou vykonal), α, β jsou parametry. Malé α posílí výběr bližších měst, malé β posílí význam feromonů. Po dokončení procházek všech mravenců je nechána část feromonů v grafu vyprchat příkazem

$$\tau_{i,j} \leftarrow (1-\rho)\tau_{i,j} \tag{2}$$

kde ρ je opět parametr. Posledním krokem je položení nových feromonů

$$\tau_{i,j} \leftarrow \tau_{i,j} + \sum_{k=1}^{m} \Delta_{i,j}^k, \tag{3}$$

$$\Delta_{i,j}^k = \begin{cases} 1/C^k, & \text{pokud hrana}(i,j) \in T^k \\ 0, & \text{jinak} \end{cases}, \tag{4}$$

m je počet mravenců, T^k je procházka mravence k a C^k je její délka. Nastavení parametrů je důležité. Experimentálně bylo zjištěno, že dobrá hodnota pro α je 1, β mezi 2 a 5 a 0.5 pro ρ. Pro velká ρ jsou všechny cesty mravenců zapomínány a ti stále jen prozkoumávají graf, pro velká ρ se příliš brzy shodnou na suboptimálním řešení.

Ellitist Ant System - první vylepšení AS

Algoritmus přináší oproti Ant System změnu v pokládání feromonů. Tou je dodatečné položení feromonů na hrany, jež jsou součástí nejlepší do té doby známé procházky. Modifikovaná rovnice vypadá takto

$$\tau_{i,j} \leftarrow \tau_{i,j} + \sum_{k=1}^{m} \Delta_{i,j}^k + e\Delta_{i,j}^{bs}, \tag{5}$$

$$\Delta_{i,j}^{bs} = \begin{cases} 1/C^k, & \text{pokud hrana}(i,j) \in T^{bs} \\ 0, & jinak \end{cases}, \tag{6}$$

Vysledky experimentů ukazují, že tato změna vede k nalezení lepších řešení v kratším čase.

$\mathcal{MAX} - \mathcal{MIN}$ Ant System

$\mathcal{MM}$AS mění AS ve 4 hlavních bodech.

1. Klade velký důraz na nejlepší nalezenou cestu. Pouze buď nejlepší mravenec v dané iteraci (hlavního cyklu algoritmu) nebo nebo do té doby nejlepší mravenec položí feromony.

2. Tato strategie by mohla vést ke stagnaci. Celá kolonie by se mohla pohybovat pouze po jediné sice dobré ale suboptimální cestě vlivem velkého množství feromonů hromaděného na jedné trase. Proto $\mathcal{MM}$AS stanovuje horní a dolní limit na množství feromonů na jedné hraně grafu.

3. Feromonové stopy jsou inicializovány na horní limit, což společně s pomalejším vyprcháváním vede k intenzivnějšímu průzkumu v začátcích běhu algoritmu.

4. Feromonové stopy jsou reinicializovány projeví-li systém stagnaci nebo nebylo nalezeno nové řešení během daného počtu iterací.

Ant Colony System

ACS rozšiřuje AS ve 3 hlavních bodech.

1. Mravenci používají agresivnější pravidlo pro výběr příštího uzlu. S pravděpodobností q_0 (parametr) sledují nejlepší známou cestu z uzlu i. S opačnou pravděpodobností použijí pravidlo z AS. Volbou q_0 je možné podporovat průzkum grafu nebo koncentrovat hledání řešení okolo nejlepší známé cesty.

2. Pouze nejlepší mravenec pokládá svou feromonovou stopu a pouze jeho stopa vyprchává viz příkaz 7. Taková strategie snižuje složitost aktualizace feromonových stop z kvadratické na lineární.

3. Každý mravenec bezprostředně po každém svém kroku z i do j aktualizuje množství feromonů na hraně (i,j) příkazem 8, $\zeta \in (0,1)$ experimentálně zjištěná dobrá hodnota je 0.1, τ_0 je inicializační hodnota feromonů. Výsledkem aktualizace je snížení množství foromonů, což podpoří volbu ostatních uzlů a zabrání tak stagnaci.

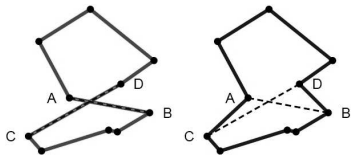
$$\tau_{i,j} \leftarrow (1-\rho)tau_{i,j} + \rho\Delta\tau_{i,j}^{bs} \tag{7}$$

$$\tau_{i,j} \leftarrow (1-\zeta)tau_{i,j} + \zeta\tau_0 \tag{8}$$

Záporem strategie je, že vylučuje paralelní implementaci.

Local search

Uvedené algoritmy lze navíc vylepšit variantami local search, např. 2-opt a 3-opt. Máme-li kandidátní řešení s, local search vytvoří všechna řešení s' taková, že s' vznikne z s změnou libovolných k hran libovolným možným způsobem. Mravenec použije nejlepší ze všech řešení s'.



Obrázek 3: 2-opt

Jiné $\mathcal{NP}$-těžké problémy

Směrování - Sequential Ordering - algoritmus HAS-SOP

Cílem SO je najít v grafu nejkratší cestu přes všechna uzly, přičemž některé uzly musí předcházet jiným. Graf je orientovaný s ohodnocenými hranami. SO může být formulováno jako asymetrické TSP. Pokud má uzel j předcházet uzlu i, hraně (i,j) je přiřazena váha -1. Na začátku jsou všichni mravenci umístěni do počátečního uzlu. Ve volbě dalšího pokračování své cesty neuvažují uzly, jejichž přidáním by porušili omezení. Hybrid AS-SOP, obohacený o specifický 3-opt local search je podle údaje z roku 2004 nejlepší algoritmus pro SOP.

Přiřazování - Quadratic Assignment Problem - algoritmus AS-QAP

QAP hledá přiřazení prvků z množiny *locations* prvkům z množiny *facilities*, s danými vzdálenostmi mezi místy v locations a toky mezi zařízeními ve facilities. Cílem je umístit zařízení tak, aby součet příslušných součinů toků a vzdáleností byl minimální.
Graf Graf je úplný a zahrnuje zařízení i místa.
Omezení Každému zařízení je přiřazeno právě jedno umístění a naopak.
Feromonová stopa $\tau_{i,j}$ odpovídá vhodnosti umístění zařízení i do j.
Heuristická informace $\mu_i = 1/d_i$, d_i je součet vzdáleností z místa i do ostatních míst.
Konstrukce řešení Zařízení jsou seřazena sestupně podle f_i, součtu toků ze zařízení i do ostatních zařízení. V každém svém kroku mravenci výběrem hrany vedoucí z i do j umístí jedno zařízení.
Aktualizace feromonvých stop Podobná jako u předchozích algoritmů. Pro určení kvality řešení se použije objektivní funkce $f = \sum_i^n \sum_j^n a_{i,j} b_{\pi_i,\pi_j}$, a je tok mezi i a j, b je vydálenost mezi místy i a j, π_i je umístění zařízení i.
Výsledky Algoritmus patří mezi nejlepší.

Strojové učení - Learning of Classification Rules - Ant-miner

Máme-li množinu atributů (každý atrubut má svou doménu), *class* atribut (jeden z množiny

atributů) a množinu training set obsahující data ke klasifikaci (mnoho přidělení hodnot atributům), pak úkolem je najít množinu pravidel tvaru IF<term1,term2,...>THEN<b>. THEN část obsahuje hodnotu class atributu a term je trojice (atribut, operátor, hodnota). Pro jednoduchost Ant-miner povoluje pouze operátor =. Př. atributy (Místo, Den, Rok, Počasí), class atribut Počasí, TS může obsahovat (Brno, 30.11, 2003, rainy),(Brno, 30.11, 2004, rainy),(Brno, 30.11, 2005, rainy),(Brno, 30.11, 2006, sunny), pak Ant-miner pravděpodobně najde pravidlo IF<Brno,30.11>THEN<rainy>.

ACO algoritmus je podobný AS, ale používá pouze jednoho mravence.

Graf Obsahuje uzel pro každý možný term a jeden startovní.

Omezení V jednom pravidle může být atribut použit nejvýše jednou a každé pravidlo obsahuje minimální počet atributů.

Feromonová stopa $\tau_{i,j}$ odpovídá vhodnosti přiřazení hodnoty j atributu i.

Heuristická informace $\mu_{i,j}$ podle rovnic 9, 10

Konstrukce řešení Mravenec začíná s prázdným pravidlem a postupně přidává termy podle rozhodovacího pravidla stejného jako v AS s pravděpodobností popsanou rovnicí 11. Se skládáním pravidla skončí použil-li všechny atributy nebo by jakýkoliv možný term snížil počet instancí v training set odpovídajících skládanému pravidlu pod stanovenou hodnotu.

Čištění pravidla Po dokončení je pravidlo zbaveno přebytečných termů snižujících jeho přesnost. Čištění je prováděno spekulativním odstraňováním termů a vyhodnocováním kvality pravidla bez nich, podle funkce 12, a odstraněním toho termu, jehož odstraněním se kvalita zvýší nejvíce. Proces se opakuje, dokud v pravidle nezůstane jediný term nebo by odstranění jakéhokoliv dalšího termu nezvýšilo kvalitu pravidla.

Aktualizace feromonvých stop Množství pokládaných feromonů se odvozuje z hodnotící funkce f, viz rovnice 12.

$$h(B|a_i = v_{ij}) = -\sum_{b=1}^{l} P(b|a_i = v_{ij}) \cdot \log_2 P(B|a_i = v_{ij}) \tag{9}$$

l je počet hodnot class atributu, $P(b|a_i = v_{ij})$ je empirická pravděpodobnost výskytu hodnoty class atributu b, máli instance TS atribut a_i hodnotu v_{ij}

$$\mu_{ij} = \frac{\log_2 l - h(B|a_i = v_{ij})}{\sum_{j=1}^{f_i} \log_2 l - h(B|a_i = v_{ij})} \tag{10}$$

f_i je velikost domény atributu a_i

$$p_{ij} = \frac{\tau_{ij}\mu_{ij}}{\sum_{i=1}^{n}(x_i \sum_{j=1}^{f_i}(\tau_{ij}\mu_{ij}))} \tag{11}$$

x_i je 1 pokud atribut a_i mravenec ještě nepoužil, 0 jinak

$$f(rule) = \frac{TP}{TP + FN} \cdot \frac{TN}{FP + TN} \tag{12}$$

TP je počet instancí v training set, které mají hodnotu class atributu předpovězenou hodnoceným pravidlem a odpovídají IF části pravidla, FP je počet instancí v training set, které mají hodnotu class atributu jinou než předpovězenou hodnoceným pravidlem a odpovídají IF části, FN je počet instancí v training set, které mají hodnotu class atributu předpovězenou hodnoceným pravidlem, ale od pravidla se liší, TN je počet instancí v training set, které nemají hodnotu class atributu předpovězenou hodnoceným pravidlem, a od pravidla se liší.

AntNet

Mravenci mohou směrovat i datové sítě. AntNet je směrovací algoritmus, který se svými schopnostmi řadí mezi algoritmy jako OSPF, SPF, BF, Daemon (alespoň podle výsledků experimentů).

Každý směrovací uzel AntNetu náhodně, podle současného datového provozu, volí jeden síťový uzel. Vytvoří mravence a nechá ho najít si cestu do tohoto uzlu. Pravděpodobnost vytvoření *dopředného* mravence v uzlu s s cílem d je dána rovnicí 13. Každý směrovací uzel linkám ke svým sousedům

$$p_{sd} = \frac{f_{sd}}{\sum_{i=1}^{n} f_{si}}, \quad f_{sd} \text{ je datový tok v bitech z } s \text{ do} d \tag{13}$$

přiřazuje vlastní hodnoty feromonů. Ovšem rozlišuje nejen mezi tím, které lince k sousednímu uzlu feromony patří, ale i jaké cesty, určené cílovým uzlem, je linka součástí. Hodnoty feromonů τ_{ijd} mají tedy nyní 3 indexy. Každý směrovací uzel navíc vytváří a udržuje svůj lokální model sítě. Lokální model je trojice $(\nu_{id}, \sigma_{id}, \mathcal{W}_{id})$. ν_{id} je průměr délek trvání několika posledních cest a σ_{id} zaznamenává jeho kolísání. $\mathcal{W}_{id}$ uchovává nejlepší dobu cesty z i do d během posledních w aktualizací modelu. Model slouží k vyhodnocování kvality cesty. Vyhodnocování v síti je složitější, poněvadž k dispozici je pouze délka trvání cesty a tu neovlivňuje jen výběr uzlů ale i aktuální síťový provoz. Pokud provoz vzroste, jsou dobré časové hodnoty, které by dříve byly ohodnoceny velmi nízko. Lokální modely jsou udržovány podle rovnic 14 a 15. Váha k-té doby po j aktualizacích je potom $\varsigma(1-\varsigma)^{j-k}$, pro

$$\nu_{id} \leftarrow \nu_{id} + \varsigma(o_{i\to d} - \nu_{id}), \quad o_{id} \text{ je doba mravencovi cesty z } i \text{ do } d \tag{14}$$

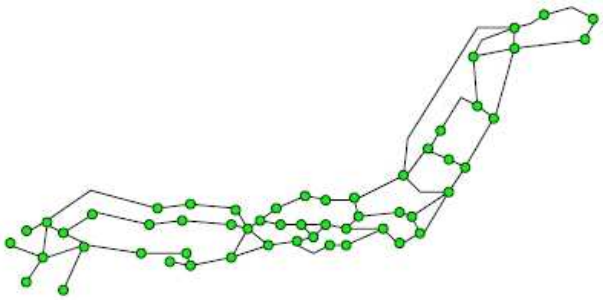
$\varsigma = 0.1$ výsledek ovlivní zhruba pouze 50 posledních údajů. V AntNetu jsou 2 prioritní skupiny pa-

$$\sigma_{id}^2 \leftarrow \sigma_{id}^2 + \varsigma((o_{i\to d} - \nu_{id})^2 - \sigma_{id}^2), \tag{15}$$

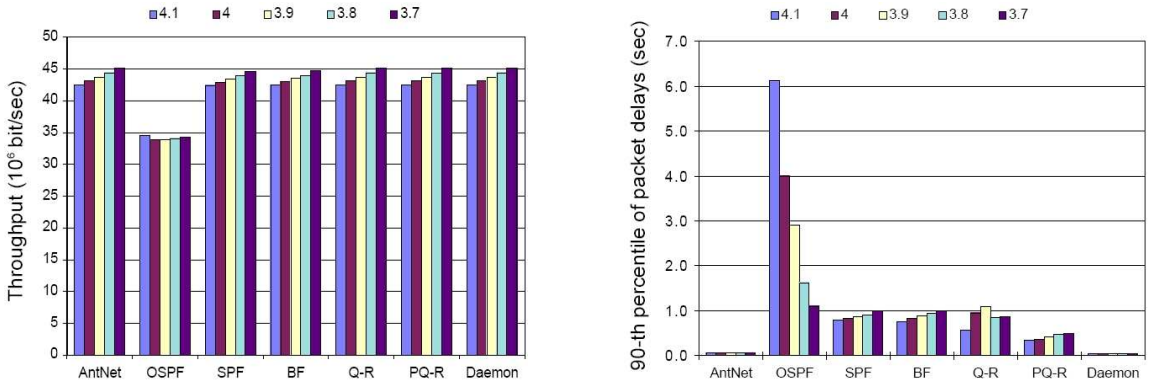
ketů: a) datové pakety a dopřední mravenci, b) zpětní mravenci, viz dále. Dopřední mravenci hledají co nejkratší cestu z uzlu, ve kterém byli vytvořeni, do cílového uzlu. Sítí cestují stejným způsobem jako data. Využívají přitom feromonovou stopu, heuristickou informaci a svojí paměť. Heuristická informace je dána rovnicí 16. Ná své cestě mravenec shromažďuje informace o hustotě provozu,

$$\mu_{ij} = \frac{q_{ij}}{\sum_{l=1}^{|\mathcal{N}_i|} q_{il}}, \quad q_{ij} \text{ je délka fronty na lince z } i \text{ do } j \tag{16}$$

ukládá identifikátory uzlů a pamatuje si jak dlouho je na cestě. Jakmile dorazí do cíle, přepne se do *zpětného* režimu, ve kterém má vyšší prioritu, a vrátí se stejnou cestou zpět do zdrojového uzlu (bez případných cyklů). Důvodem je rychlé rozšíření směrovací informace. V každém uzlu na zpáteční cestě aktualizuje feromony a lokální model a oznámí uzly, kterými prošel. Původní uzel mravence ničí.



Obrázek 4: Páteřní síť japonské společnosti NTT. Její model sloužil k testovíní AntNetu.



Obrázek 5: NTTnet: experimentální srovnání přenosové rychlosti a zpoždění AntNetu s ostatními směrovacími algoritmy.

Trendy v ACO

ACO algoritmy jsou při řešení mnoha akademických problémů, příkladem sequential ordering problem, vehicle routing problem with time window constraints, quadratic assignment, group shop sheduling, stejně dobré jako ty nejlepší v současnosti dostupné algoritmy, v některých případech lepší. A AntNet dává velmi dobré výsledky. Jsou ovšem používány i v komerčních aplikacích. Algoritmus Dyvoil řídí distribuci topných olejů rozvážených nehomogenním strojovým parkem společnosti Pina Petroli ve Švýcarsku. AntRoute plánuje cesty stovek vozidel Švýcarského potravinového řetězce Migras. Gravel, Price & Gagné použili ACO na plánování ve výrobě slévárny hliníku. Očekává se, že výzkum ACO se v budoucnu zaměří mimo jiné na paralelizaci, problémy s více obektivními funkcemi, problémy při jejihž řešení dává objektivní funkce hodnoty jen s určitou mírou přesnosti.

Reference

[1] Marco Dorigo Thomas Stützle. *Ant Colony Optimization.* The MIT Press, 2004.