

Optimalizace pomocí mravenčích kolonií

Inspirace, definice, aplikace a perspektivy

Pavel Krömer
pavel.kromer@vsb.cz

22. února 2012

1 Úvod

Optimalizace pomocí mravenčích kolonií (Anto Colony Optimization), či méně formálně mravenčí optimalizace nebo mravenčí algoritmy, představují významný proud v oblasti počítačové vědy zvané rojová inteligence (Swarm Intelligence). Rojovou inteligenci můžeme zařadit mezi obory moderní výpočetní inteligence (Computational Intelligence). Výpočetní inteligence je termín, který zastřešuje velkou rodinu úspěšných algoritmů jež se zformovaly částečně z a částečně souběžně s metodami tradiční umělé inteligence, dnes označovaných také "stará dobrá umělá inteligence" (Good Old Artificial Intelligence). Mimo rojové inteligence a mravenčích algoritmů můžeme zařadit mezi algoritmy výpočetní inteligence také další přírodou inspirované výpočetní modely jako například evoluční algoritmy (např. [1], česky [2]), neuronové sítě, fuzzy systémy či umělé imunitní systémy.

Výpočetní inteligence se zabývá návrhem a aplikací algoritmů vycházejících ze studia adaptivních mechanismů, často inspirovaných fyzikálními či biologickými jevy a fenomény, za účelem dosažení inteligentního chování ve složitých a dynamických prostředích [3]. Algoritmy výpočetní inteligence jsou navrhovány tak, aby se dokázaly přizpůsobovat okolním podmínkám, učit se (nejen) ze svého okolí a pružně reagovat na změnu prostředí. Cílem algoritmů výpočetní inteligence je objevovat nová řešení komplexních problémů, abstrahovat a aplikovat nalezené vztahy a spojovat a zobecňovat znalosti odhalené v datech. Není neobvyklé, když algoritmus výpočetní inteligence překvapí nalezeným řešením své tvůrce a spíše než požadovanou odpověď naznačí svým výstupem nedostatek v položené otázce (např. nevhodný způsob hodnocení kvality potenciálních řešení).

Výpočetní inteligence je zpravidla klasifikována jako podmnožina [3] či odnož umělé inteligence [4]. Shrnuje nejnovější trendy ve výzkumu, vývoji a aplikaci inteligentních systémů. Na rozdíl od tradiční umělé inteligence, které staví na deterministických metodách, výpočetní inteligence sdružuje algoritmy, které vycházejí ze Soft Computingu, což je volné označení skupiny metod, které staví na pravděpodobnosti, fuzzy logice, fuzzy množinách atd. Velkou výhodou metod výpočetní inteligence je, že dokáží v přijatelném

čase přijít s přibližným řešením komplexních problémů o vysoké dimenzi, zatímco přesné metody, které vedou ke globálně optimálním řešením zkoumaných problémů, zpravidla vyžadují příliš dlouhý výpočet, což je činí nepoužitelnými pro převážnou většinu praktických aplikací.

Myšlenky simulace přírodních procesů v počítači, jež jsou základem bio-inspirovaných výpočtů, se objevují od prvopočátku výpočetních technologií [2, 1]. Průkopníci výpočetní techniky jako John von Neumann (1903-1957) a Alan Turing (1912-1954) se intenzivně zabývali problematikou umělé inteligence a umělého života ve smyslu návrhu počítačových programů schopných sebereplikace, adaptace, učení podobného lidskému a ovlivňování svého životního prostředí. Počítače od počátku sloužily mimo jiné k vědeckým pokusům o simulaci činnosti lidského mozku, zkoumání procesu učení a adaptivní evoluce. Z prvních dvou odvětví se vyvinuly obory neuronových sítí a strojového učení, z pokusů o simulaci evoluce v počítačových systémech se vyvinula problematika evolučních algoritmů [1].

V padesátých letech dvacátého století započaly pokusy o využití simulované evoluce k řešení praktických optimalizačních úloh. Všechny přístupy měly společnou myšlenku hledání optimálního řešení daného problému prostřednictvím evoluční adaptace populace potenciálních řešení. Současně se objevilo více konkrétních implementací evolučních algoritmů. Rechenberg (1934) představil evoluční strategie, Fogel algoritmus evolučního programování. Patrně největšího úspěchu se ale dostalo metodám vycházejících z genetických algoritmů, které navrhl v padesátých letech John Holland (1929) a dále zkoumali a prohloubili jeho kolegové a studenti z Michiganské univerzity, zejména David Goldberg (1953). Holland na rozdíl od ostatních nehledal evoluční řešení pro specifický problém, ale formálně zkoumal samotný proces adaptace v přírodě a možnosti jeho simulace a využití v počítačových systémech. Jako první představil algoritmy vyvíjející populace potenciálních řešení pomocí simulované mutace a křížení. Navíc se jako jediný věnoval vytvoření a prohloubení formální teorie podporující evoluční výpočty [1].

Rojovou inteligenci je možno neformálně popsat jako vlastnost rozsáhlých systémů sestávajících z množství jednoduchých agentů, jejichž vzájemná lokální interakce (přímá nebo nepřímá) vede ke vzniku koherentních globálních funkčních vzorů [3]. Mravenčí optimalizace je oblastí rojové inteligence inspirovanou některými vzory chování mravenčích společenství. Jako jeden z prvních se zabýval systematickým studiem chování hmyzích společenství Eugène Marais (1872-1949), který studoval chování kolonie termitů a prováděl experimenty. Nepřímou komunikaci ve společenstvích termitů nazval Pierre-Paul Grassé (1895-1985) stigmergie (stigmergy). Popsal, že informace se společenstvím šíří prostřednictvím interakce mezi jednotlivými členy a prostřednictvím interakce členů s prostředím. Z jednoduchých komunikačních vzorů vyplývá komplexní chování kolonie. Studiem stigmergie založené na feromonech se zabýval Jean-Louis Deneubourg a kolegové [5, 6, 7] a vzory kolektivního chování kolonií argentinských mravenců zkoumal Goss a kolegové [8]. Jeden z prvních počítačových modelů chování mravenčí kolonie na předložila Maria Ebling a kolegové [9]. Základ optimalizacím pomocí mravenčích kolonií pak položil Marco Dorigo [10], Dorigo a kolegové [11, 12], Colorni a kolegové [13] a další, např. [14]. Dnes patří mravenčí optimalizace mezi nejrozšířenější bio-inspirované metaheuristické vyhledávací a optimalizační metody s řadou aplikací v mnoha různých

oborech.

2 Rojová inteligence

Rojová inteligence se zabývá návrhem multiagentních systémů inspirovaných inteligentním chováním společenských živočichů (např. hmyzu, ryb, ptáků a dalších). Zaměřuje se na analýzu a napodobování globálních vzorů chování komplexních přírodních systémů spíše než návrhem sofistikovaných kontrolních algoritmů, které by řídily danou aplikaci. Naopak, roj sestává z množství jednoduchých agentů, kteří spolu komunikují a jejichž vzájemná interakce přispívá k inteligentnímu chování celého společenstva [15].

2.1 Společenský hmyz: biologické inspirace mravenčích algoritmů

Hmyzí kolonie vykazují vysoký stupeň organizovanosti. Mravenci, včely, vosy či termity tvoří společenstva, ve kterých jednotlivci v různých rolích přispívají jednoduchými operacemi ke komplexnímu a překvapivě inteligentnímu chování celého roje [15, 16]. To vše bez jakéhokoliv centrálního mozku (kontroléru), který by řídil roj jako celek. V hmyzí kolonii se jednotliví jedinci zpravidla specializují na určité činnosti. Dělbá práce, při které jsou jednotlivé činnosti prováděny simultánně celými skupinami specializovaných agentů, přispívá k robustnosti chování celého roje a předpokládá se, že jsou prováděny efektivněji a lépe, než kdyby všechny druhy činností prováděl jeden typ agentů sekvenčním způsobem.

Komplexní aktivity společenského hmyzu jsou příkladem samoorganizace [16], kterou najdeme mimo biologii také například ve fyzice, chemie atd. Teorie samoorganizace ukazují, že komplexní kolektivní chování může vyplynout z četných interakcí mezi jednotlivými členy roje nebo hejna, kteří se řídí jednoduchými pravidly. Ukazuje se, že individuální složitost jednotlivce není podmínkou pro komplexní inteligentní kolektivní chování skupiny. Samoorganizace je podle posledních poznatků důležitou součástí celé řady kolektivních fenoménů v říši společenského hmyzu. Z pohledu řešení problémů můžeme na hmyzí společenství nahlížet jako na robustní decentralizovaný biologický stroj k řešení problémů složený z mnoha jednoduchých vzájemně interagujících komponent. Tyto dílčí komponenty (agenti a skupiny agentů) se specializují na různé úkoly. V přírodě řeší mravenčí stroj řadu různých problémů: hledání jídla, stavbu, údržbu a rozšiřování hnízda, přesuny společenstev, rozmnožování kolonie, péči o potomstvo, reakci na nebezpečí, šíření poplašného signálu a mnohé další.

Příkladem mravenčí samoorganizace mohou být postupy tropických mravenců (např. jihoamerických mravenců druhu *Eciton burchellii*) pro získávání potravy a materiálu. Jejich výboje zahrnují stovky tisíc jednotlivých mravenců a pokrývají tisíce čtverečních metrů během jediného dne. Největší roje tropických mravenců mohou sestávat i z více než 200 tisíc dělníků pohybujících se v husté koloně o šířce 15 i více metrů. Tyto nájezdy, přestože jsou prováděny slepými mravenčími dělníky, jsou příkladem zcela decentralizovaného složitého chování. Mravenčí falanga *E. burchellii* se skládá z čela, hustého koberce mravenců rozprostírajícího se přibližně jeden metra za čelem roje, a rozsáhlého systému výběžků, anastomózních stezek, po kterých se mravenci vydávají, aby z čela roje dopravili shromážděnou kořist a materiál do svého hnízda. Jejich stezky tvoří smyčky a oka,

malé poblíž čela pochodující armády a postupně se zvětšující se vzrůstající vzdáleností od čela falangy. Závěrečná mohutná smyčka vede k široké stezce, které permanentně spojuje mravenčí expediční sbor s jejich hnízdem. Toto chování a struktura mravenčích nájezdů je vysoce dynamická a adaptivní. Strategie různých druhů nájezdných mravenců přesto vždy vykazují výše představenou základní strukturu [16].

Mravenčí nájezdy, podobně jako další činnosti vykonávané hmyzími společenstvy, mohou být aplikovány na problémy v počítačových aplikacích, strojírenství, stavebnictví, plánování, ekonomice a podobně. Jejich velkou výhodou je, že dokáží řešit tyto často velmi složité problémy flexibilně a robustně [16].

Rojová inteligence je obsáhlá disciplína která se nezabývá pouze algoritmy inspirovanými chováním hmyzu. Optimalizace rojem částic (Particle Swarm Optimization) [17] se inspirovuje kolektivním chováním ptačích a rybích hejn či dokonce sociálním chováním v lidských společnostech. Je to metoda vhodná zejména k řešení problémů, u kterých je potenciální řešení možno snadno reprezentovat jako bod či křivku v n -rozměrném prostoru. Kandidátská řešení, interpretovaná jako prvky roje částic, jsou v tomto prostoru geometricky rozmístěna a je jim přiřazena počáteční rychlost a směr. Částice se prostorem pohybují a jimi reprezentovaná řešení jsou periodicky vyhodnocována. Podle jednoduchých pravidel následují částice v roji několik dobře se jevících řešení. Postupem času dochází k nasměřování celého hejna směrem, kterým se ubírají nejlepší nalezená řešení a kde se dá předpokládat optimální řešení problému. Jako u mnoha dalších multiagentních systémů přispívá kooperace celé populace jednoduchých částic k robustnímu postupu, který je odolný vůči uváznutí v lokálních extrémech a rychle konverguje k globálně dobrým řešením [17].

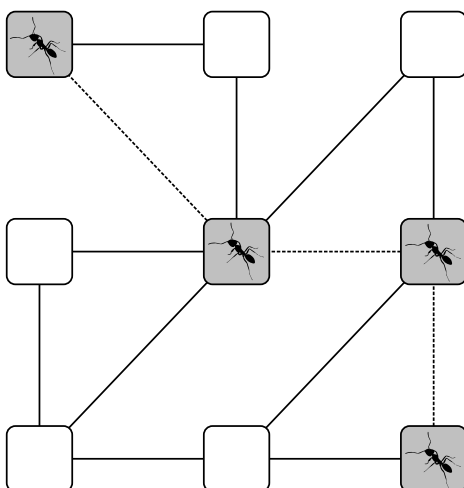
Jak jsme ukázali, společenstvo je v kontextu algoritmů rojové inteligence nutno chápat jako skupinu nezávislých agentů, kteří komunikují (přímo nebo nepřímo) aby vyřešili určitý problém. Robustní distribuované kolektivní strategie řešení problémů vyplývají z jejich vzájemné interakce. Využitím se zabývá výpočetní rojová inteligence, která navrhuje studuje algoritmické modely a aplikace [3].

3 Mravenčí algoritmy a jejich varianty

Základy mravenčích algoritmů navrhl Marco Dorigo [10, 11, 12] v roce 1992 jako metaheuristický algoritmus vycházející z pozorování některých vzorů chování mravenčích společenství při shromažďování potravy.

3.1 Ant systém

Základní formou mravenčích algoritmů je tzv. ant systém (v originále Ant System) [11, 12, 16], metaheuristický algoritmus, který přímo vychází z pozorování chování mravenčích sběračů. Ti vykazují schopnost najít velmi efektivní cesty mezi hnízdem a zdroji potravy. Hledání optimální cesty je u mravenců založeno na nepřímé komunikaci formou tzv. stigmergie. Stigmergie je komunikace mezi agenty pomocí modifikace prostředí, ve kterém se vyskytují. Změna prostředí (např. formou chemické značky, feromonu) způsobená jedním agentem ovlivňuje chování a akce ostatních agentů, kteří na značku narazí.



Obrázek 1: Mravenčí optimalizace pro hledání cesty v grafu. Volně podle [12].

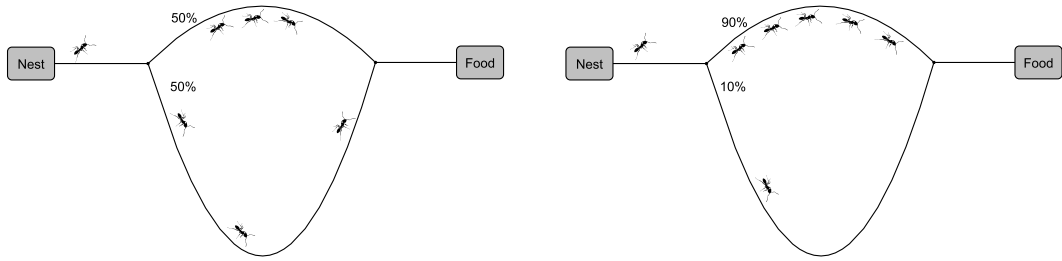
Mravenci pátrající po potravě se nahodile pohybují v okolí mraveniště. Jakmile naleznou potravu, vrací se stejnou cestou, jakou k potravě dospěli a cestu značí feromonem. Ostatní mravenčí pátrači, kteří narazí na feromonovou stopu, se po ní vydají spíše než aby pokračovali v průzkumu mraveniště. Čím více mravenců se pohybuje mezi zdrojem potravy a mraveništěm, tím silnější je feromonová stopa a tím větší je pravděpodobnost, že přitáhne další mravence (tím atraktivnější je pro ostatní mravence).

Popis tohoto chování, schematicky naznačeného na obrázku 2, vychází z laboratorního pokusu známého jako experiment s dvojitým mostem (Double Bridge Experiment) [12, 3, 18, 16]. Během pokusu s dvojitým mostem bylo hnízdo mravenců druhu *Linepithema humile* propojeno se zdrojem potravy dvěma trasami. Mravenci se časem naučili používat krátkou trasu a když jim byla nabídnuta nová, ještě kratší cesta, nedokázali ji využít. Mravenčí stroj uvázl v lokálním extrému řešeního problému. Feromonová stopa značící původně nejkratší trasu byla tak výrazná, že se novou cestou vydávalo příliš málo průzkumníků na to, aby ji označili jako nejvýhodnější. Další experimenty ukázaly, že různé druhy mravenců používají různé metody pro volbu trasy a například mravenci druhu *Lasius niger* vyvinuli postup, který jim umožňuje objevit novou výhodnější trasu i pokud se objeví až dodatečně [16].

Napodobování mravenčího chování slouží jako základ pravděpodobnostní výpočetní metaheuristické metody využitelné k řešení komplexních problémů, které můžeme vyjádřit jako hledání optimální cesty v grafu [12, 17]. Umělý mravenec k nacházející se ve vrcholu i grafu $G = (N, A)$ se přesune do sousedního vrcholu j s pravděpodobností p_{ij}^k :

$$p_{ij}^k = \frac{\tau_{ij}^\alpha \eta_{ij}^\beta}{\sum_{l \in N_i^k} (\tau_{il}^\alpha \eta_{il}^\beta)} \quad (1)$$

kde N_i^k představuje množinu všech vrcholů dostupných mravenci k (tzn. množinu vr-



(a) Počátek mravenčího algoritmu: mravenci si vybírají krátkou i dlouhou trasu se stejnou pravděpodobností. (b) Po mravenčí optimalizaci: feromonová stopa přitahuje mravence ke kratší trase.

Obrázek 2: Pravděpodobnost volby trasy před a po několika iteracích mravenčího algoritmu. Volně podle [12].

cholů, do kterých vede z i hrana), τ_{ij} odpovídá množství feromonu na hraně a_{ij} a η_{ij} reprezentuje heuristickou a-priori informaci jež vyjadřuje vhodnost přechodu hrany a_{ij} . Každý mravenec provede během jedné iterace výpočtu zadaný počet kroků vpřed. Trasu mezi jednotlivými vrcholy volí pomyslní mravenec podle pravidla (1). Jakmile mravenec dokončí své putování grafem, vrací se do hnízda s úlovkem. Konkrétní trasa k -tého mravence T^k , délka trasy C^k a množství potravy nalezené k -tým mravencem L_k vyjadřují kvalitu nalezeného řešení a jsou použity pro výpočet množství feromonu, který mravenec ukládá na každou hranu cesty, po které se v dané iteraci pohyboval:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{1}{C^k} & \text{pokud hrana } a_{ij} \text{ patří do } T^k \\ 0 & \text{jinak} \end{cases} \quad (2)$$

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (3)$$

Mimo ukládání feromonu dochází v mravenčích algoritmech také k vypařování feromonové stopy. V základní variantě optimalizace mravenčí kolonií se po každé iteraci zmenší množství feromonu uloženého na každé hraně podle vztahu:

$$\tau_{ij} = (1 - \rho)\tau_{ij} \quad (4)$$

Koeficienty α , β a ρ jsou obecnými parametry algoritmu. Algoritmus ant systému v pseudokódu je uveden ve výpisu 1.

Mravenčí optimalizace probíhá v několika po sobě jdoucích iteracích. V každé iteraci provede m mravenců své úkoly a vyhodnotí se cesty, které objevili. Proces optimalizace končí splněním ukončujících podmínek, ke kterým může patřit nalezení optimálního řešení, určitý počet iterací bez zlepšení nebo jednoduše dosažení určitého počtu iterací.

Základní mravenčí algoritmus se dočkal velkého množství modifikací. V následujícím textu si představíme některé z nich.

```

1 Vygeneruj počáteční feromonovou matici  $\mathbf{P}$  aby odpovídala topologii dané úlohy;
2  $iterace = 0$ ;
3 while Ukončující podmínky nejsou splněny do
4   |   Rozmísti  $m$  mravenců náhodně na vrcholy grafu. Pro všechny mravence nechť
   |   množství potravy  $C^k$  je 0;
5   |   foreach  $k$  do
6   |     |   Jdi vpřed  $n$  kroků v grafu. Vybírej směr podle pravděpodobnostního
   |     |   pravidla (1);
7   |     |   Vyhodnot kvalitu řešení nalezeného mravencem;
8   |     |   Nech  $k$  uložit na jeho cestu v  $\mathbf{P}$  feromony podle (3);
9   |   end
10  |   Nech vypařit feromony v  $\mathbf{P}$  podle (4);
11  |    $iterace = iterace + 1$ ;
12 end

```

Algorithm 1: Algoritmus ant systému v pseudokódu.

3.2 Modifikace ant systému

Jedním z prvních fenoménů, které byly do algoritmů optimalizace mravenčích kolonií přidány, patří elitismus. Elitismus je dobře znám z jiných algoritmů výpočetní inteligence (např. evolučních algoritmů) a spočívá v posílení vlivu nejlepších řešení na průběh výpočtu. Toto chování zpravidla výrazně urychluje konvergenci algoritmu, ale zvyšuje pravděpodobnost uváznutí v lokálním optimu prostoru řešení daného problému.

Elitářský mravenčí systém (Elitist Ant System) byl prvním algoritmem vnášejícím elitismus do mravenčích optimalizací [12]. V rámci této metody dochází k posílení doposud nejlepšího řešení T^{bs} , nalezeného během uplynulého běhu mravenčí optimalizace. Feromonová matice se upravuje podle vztahů:

$$\Delta\tau_{ij}^{bs} = \begin{cases} \frac{1}{C^{bs}} & \text{pokud hrana } a_{ij} \text{ patří do } T^{bs} \\ 0 & \text{jinak} \end{cases} \quad (5)$$

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k + e\Delta\tau_{ij}^{bs} \quad (6)$$

kde $\Delta\tau_{ij}^k$ představuje přírůstek feromonu na hraně a_{ij} uložený k -tým mravencem vyjádřený podle (2), C^{bs} je délka nejlepší trasy a e je parametr. Potvrdilo se, že takto implementovaný elitismus, spolu s vhodně zvoleným e , vede k lepším výsledkům a způsobuje rychlejší konvergenci algoritmu [12].

Další variantou mravenčího algoritmu s elitismem je mravenčí systém založený na pořadí (Rank-based Ant System) [19, 12, 16]. V této metodě jsou mravenci po nalezení cesty seřazeni podle kvality nalezené cesty a množství uloženého feromonu se řídí pořadím mravence. Počet mravenců, kteří upravují feromonovou stopu, je omezen parametrem w . Jen $w - 1$ nejlépe umístěných mravenců získá právo uložit feromony. V každé

iteraci se zesilují feromony na dosud nejlepší nalezené trase. Feromony se upravují podle vztahu:

$$\tau_{ij} = \tau_{ij} + \sum_{r=1}^{w-1} (w-r)\Delta\tau_{ij}^r + w\Delta\tau_{ij}^{bs} \quad (7)$$

kde příspěvek r -tého mravence $\Delta\tau_{ij}^r$ vyjádříme podle (2) a $\Delta\tau_{ij}^{bs}$ podle (5).

Systém mravenčí kolonie (Ant Colony System) obohacuje původní mravenčí systém o elitismus a přináší také další modifikace [12]. Pro řízení pohybu mravenců po grafu používá pseudonáhodné proporční pravidlo. Každý mravenec se s pravděpodobností q_0 vydá směrem určeným feromonovou stopou a a-priori informací s pravděpodobností $(1 - q_0)$ se přesune na některou s dalších dostupných hran. Mravenec na vrcholu i se přesune na vrchol j zvolený podle:

$$j = \begin{cases} \operatorname{argmax}_{l \in N_i^k} \{\tau_{il}\eta_{il}^\beta\} & \text{pokud } q < q_0 \\ J & \text{jinak} \end{cases} \quad (8)$$

kde q_0 je parametr algoritmu, q je náhodná hodnota z intervalu $\langle 0, 1 \rangle$ a J je vrchol zvolený podle (1) při $\alpha = 0$.

Parametr $q_0 \in \langle 0, 1 \rangle$ představuje poměr využití informací uložených ve feromonové matici (exploitaci) k průzkumu nových tras (exploraci), což je činnost který může vést k objevení nových slibných cest. Elitismus je v systému mravenčí kolonie implementován pomocí restrikcí pro ukládání feromonů. Jen nejlepší mravenec smí po každé iteraci mravenčí optimalizace uložit feromony. Vypařováním se upravují pouze feromony na nejlepší nalezené trase. Novým prvkem systému mravenčí kolonie je lokální úprava feromonů. Pokaždé, když nějaký mravenec přejde hranu a_{ij} , množství feromonu na ní uložené je bezprostředně upraveno podle vztahu:

$$\tau_{ij} = (1 - \xi)\tau_{ij} + \xi\tau_0 \quad (9)$$

kde $\xi \in (0, 1)$ a τ_0 jsou parametry. Pro τ_0 se zpravidla volí malá konstantní hodnota [3].

Max-min mravenčí systém (Max-Min Ant System) [12] je další modifikací, která se snaží zlepšit výkonnost mravenčí optimalizace. V této variantě dochází k celé řadě změn oproti základnímu mravenčímu systému. Max-min v názvu algoritmu odkazuje k omezení hodnoty feromonů na hranách intervalem $\langle \tau_{min}, \tau_{max} \rangle$. Tím se zamezuje příliš brzké dominanci jedné cesty v grafu. Lepšímu využití informace zakódované ve feromonové matici napomáhá inicializace hodnot feromonů na hranách grafu na τ_{max} a nízká hodnota ρ . Elitismus je v max-min mravenčím systému implementován tak, že ukládání feromonů je povoleno jen na nejlepší dosud nalezenou trasu či nejlepší trasu v dané iteraci. Opatřením proti uváznutí v lokálním extrému je reinicializace feromonové matice pokaždé, když během řešení nastane stagnace, tzn. že nedojde ke zlepšení nalezeného řešení během daného počtu po sobě jdoucích iterací.

Algoritmus ant-Q [20, 12, 3] je variantou systému mravenčí kolonie u kterého byly vztahy pro modifikaci množství feromonu inspirovány Q-učením a pomyslné feromony

jsou nahrazeny tzv. AQ-hodnotami. Cílem algoritmu je najít takovou kombinaci AQ-hodnot, která povede k nalezení optimální cesty. Ant-Q se oproti systému mravenčí kolonie liší zejména volbou τ_0 :

$$\tau_0 = \gamma \max_{j \in N_i^k} \{\tau_{ij}\} \quad (10)$$

kde γ představuje parametr a maximum je spočteno ze všech feromonových stop na hranách vycházejících z vrcholu i , které mravenec nenavštívil. V algoritmu Ant-Q dochází při každém přesunu mravence k okamžité úpravě množství uložených AQ-hodnot. Přitažlivost hrany a_{ij} se tedy sníží s každým mravencem, který po ní přijde a zároveň zvýší o hodnotu proporčně vycházející z množství feromonu na nejlépe hodnocené hraně vycházející z vrcholu i . Ukázalo se, že jednodušší systém mravenčí kolonie dosahuje zpravidla stejných výsledků jako Ant-Q [3].

Antabu [3] doplňuje mravenčí algoritmy o lokální vyhledávání s prvky tabu vyhledávání. Tabu vyhledávání (Tabu Search) je stručně řečeno postup, při kterém je nějaký prostor prohledáván tak, že žádná bod není navštíven dvakrát. Jinými slovy, každý navštívený bod je uložen do tzv. tabu seznamu a dále jsou procházeny pouze body, které v tabu seznamu nejsou. navíc algoritmus antabu upravuje vztah pro ukládání feromonů na:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \left(\frac{\rho}{C^k}\right) \left(\frac{C^{ws} - C^k}{C^{bs}}\right) \quad (11)$$

kde C^{ws} představuje délku nejhorší nalezené cesty, C^{bs} cenu nejlepší nalezené cesty a C^k délku cesty nalezené k -tým mravencem.

Algoritmus ANTS [3] patří k dalším rozšířením mravenčího systému. Název ANTS (v překladu mravenci) je vlastně zkratkou pro přibližné nedeterministické prohledávání stromu (Approximated Non-deterministic Tree Search). Od mravenčího systému algoritmus ANTS odlišuje jiný výpočet přírůstku feromonu:

$$\Delta\tau_{ij}^k = \tau_0 \left(1 - \frac{C^k - \epsilon}{\bar{C}_n - \epsilon}\right) \quad (12)$$

kde C^k představuje cenu cesty, kterou v dané iteraci absolvoval k -tý mravenec a \bar{C}_n je klouzavý průměr ceny posledních n_t nejlepších řešení nalezených algoritmem. \bar{C}_n je dáno vztahem:

$$\bar{C}_n = \frac{\sum_{t'=t-n_t}^t C_{t'}^k}{n_t} \quad (13)$$

ve kterém $C_{t'}^k$ představuje nejlepší řešení známé v iteraci t' a ϵ je dolní mez ceny optimálního řešení. Pokud je k dispozici méně než n_t předchozích řešení, počítá se klouzavý průměr pouze z existujících řešení. Jako v předchozích případech je i zde změna výpočtu přírůstku feromonů motivována snahou o snížení pravděpodobnosti uvíznutí v lokálním optimu.

Rychlý mravenčí systém (Fast Ant System) [3] byl vyvinut pro řešení jedné konkrétní úlohy, kvadratického přiřazovacího problému (Quadratic Assignment Problem). Rychlý mravenčí systém na rozdíl od dalších zmíněných variant mravenčích algoritmů používá pouze jediného mravence, odlišný vztah pro ukládání feromonů a zcela opomíjí vypařování feromonů a využití a-priori informace. Vypařování feromonů je v rychlém mravenčím dáno vztahem:

$$\tau_{ij} = \tau_{ij} + w_1 \Delta \tilde{\tau}_{ij} + w_2 \Delta \hat{\tau}_{ij} \quad (14)$$

Parametry w_1 a w_2 určují, jak k přírůstku feromonů přispěje dosud nejlepší řešení a aktuální řešení. Množství uložených feromonů je dáno:

$$\Delta \tilde{\tau}_{ij} = \begin{cases} 1 & \text{pokud hrana } a_{ij} \text{ patří do cesty nalezené v této iteraci} \\ 0 & \text{jinak} \end{cases} \quad (15)$$

$$\Delta \hat{\tau}_{ij} = \begin{cases} 1 & \text{pokud hrana } a_{ij} \text{ patří do nejlepší dosud nalezené cesty } T^{bs} \\ 0 & \text{jinak} \end{cases} \quad (16)$$

Feromony na všech hranách grafu se v rychlém mravenčím systému inicializují na hodnotu 1 na počátku prohledávání a také pokaždé, když je nalezena nová nejlepší trasa. Pokud v nějaké iteraci dojde k nalezení stejné trasy jako je dosud nejlepší nalezená, zvětší se hodnota w_1 o jednu. To vede k postupnému snižování vlivu nejlepší dosud nalezené trasy na přírůstek feromonů a zmenšuje riziko uváznutí v lokálním optimu odpovídajícím právě nejlepší dosud nalezené trase.

Existuje celá řada dalších modifikací a variant mravenčích algoritmů. Jsou úspěšné ve vyhledávání téměř optimálních či optimálních cest v komplexních grafech. Výborných výsledků dosahují mravenčí algoritmy zejména pokud je pro řešení konkrétního problému k dispozici vhodná a-priori heuristická informace, která může být využita v mravenčím algoritmu. Významným přínosem k úspěšnému řešení problému mravenčím algoritmem je také lokální vyhledávání [12].

Je těžké označit některou z uvedených variant mravenčích algoritmů za lepší než ostatní, protože jejich úspěšnost (rychlost výpočtu, kvalitu nalezených řešení) vždy vyhodnocujeme v kontextu řešeného problému. V oblasti metaheuristických algoritmů a tedy i v oblasti mravenčích algoritmů platí tzv. "No Free Lunch" teorém [21], který stručně řečeno říká, že neexistuje jediný algoritmus či jediné nastavení nějakého algoritmu, které by přinášelo největší efektivitu a rychlost a nejlepší výsledky pro všechny aplikační oblasti a všechny druhy problémů. Vyšší efektivita jednoho algoritmu, jeho varianty, či konkrétního nastavení parametrů pro jednu úlohu je zpravidla vykoupena nižší efektivitou pro jinou úlohu (která bude mít jiné charakteristiky, tzn. prohledávaný prostor všech řešení bude vypadat jinak než prostor řešení první úlohy, na který jsme algoritmus nastavili). Teorém populární formou říká, že neexistuje metaheuristický algoritmus, který by řešil všechny problémy (není žádný oběd zdarma) zdůvodňuje nutnost experimentálního hledání vhodných metod a kombinací parametrů pro různé aplikační problémy (práce s tím spojená je cenou, kterou za pomyslný oběd platíme). V češtině by se dalo No Free Lunch teorém shrnout rčením "bez práce nejsou koláče".

K dispozici je porovnání výkonnosti variant mravenčích algoritmů pro některé konkrétní problémy. Dorigo a Stützle porovnali výkonnost mravenčího systému a dalších variant mravenčích algoritmů na problému obchodního cestujícího [12]. Z provedených experimentů vyplývá, že modifikace základního mravenčího systému vedou ke kvalitnějším řešením problému obchodního cestujícího než samotný mravenčí systém. Co do kvality nalezeného řešení si nejlépe vedl max-min mravenčí systém, který ale produkoval v počátečních fázích vyhledávání horší řešení než ostatní zkoumané varianty mravenčích algoritmů. V [12] je také ukázáno, že přidání lokálního vyhledávání a přítomnost heuristické a-priori informace zlepšuje výsledky vyhledávání.

4 Aplikace mravenčích algoritmů

Mravenčí algoritmy jsou vhodné k řešení problémů, které lze snadno vyjádřit jako hledání cesty v grafu.

4.1 Problém obchodního cestujícího

První algoritmus optimalizace pomocí mravenčích kolonií byl demonstrován na problému obchodního cestujícího (Travelling Salesman Problem). Problém obchodního cestujícího je známý NP-těžký problém. Neexistuje tedy přesný algoritmus, který by dokázal řešit rozsáhlejší instance tohoto problému v přijatelném čase. Naopak meta-heuristické algoritmy dokáží najít dostatečně kvalitní přibližná řešení i velkých instancí problému obchodního cestujícího a dalších NP-těžkých problémů v rozumném čase.

V problému obchodního cestujícího se snažíme najít optimální trasu, která pomyslného obchodníka provede zadanou množinou měst tak, že každé z nich (s výjimkou města, ze kterého vyjíždí) navštíví právě jednou a její cena (tzn. délka cesty) bude minimální. Pohledem teorie grafů je problém obchodního cestujícího snahou najít Hamiltonovskou kružnici minimální délky ve orientovaném grafu $G = (N, A)$ kde každá hrana $(i, j) \in A$ má přiřazenu délku (váhu) d_{ij} . V případě symetrického problému obchodního cestujícího platí, že $d_{ij} = d_{ji}$, což v případě asymetrické verze problému není zaručeno. Optimálním řešením problému obchodního cestujícího je taková permutace π_{opt} vrcholů množiny N pro kterou platí:

$$\pi_{opt} = \min_{\pi \in \Pi} \left(\sum_{i=1}^{n-1} d_{\pi(i)\pi(i+1)} + d_{\pi(n)\pi(1)} \right) \quad (17)$$

Optimalizaci pomocí mravenčích kolonií lze velmi jednoduše použít k řešení problému obchodního cestujícího [12]. Pro problém obchodního cestujícího o velikosti $n = |N|$ vytvoříme feromonovou matici o velikosti $n \times n$. Hodnota feromonu τ_{ij} představuje vhodnost cestování z města i do města j . Jako heuristická informace slouží obrácená hodnota vzdálenosti mezi městy i a j , tedy $\eta_{ij} = \frac{1}{d_{ij}}$. V případě, že mezi městy i a j neexistuje spojnice, zvolíme pro η_{ij} konstantní malou hodnotu. Obecný mravenčí algoritmus pro řešení problému obchodního cestujícího pak vypadá: Výše uvedené schéma reprezentuje obecný návod pro aplikaci mravenčí optimalizace na problém obchodního cestujícího.

```

1 Inicializuj feromony v  $P$  na malou náhodnou nebo konstantní hodnotu;
2  $iterace = 0$ ;
3 while Není nalezena dostatečně dobrá cesta nebo není dosaženo limitu iterací do
4   |   Rozmístí mravence do počátečních měst (která jsou zvolena podle nějakého
5   |   vhodného kritéria);
6   |   foreach  $k$  do
7   |     |   Jdi vpřed  $n$  kroků v grafu. Vybírej směr podle pravděpodobnostního
8   |     |   pravidla a tak, aby se žádné město v cestě neopakovalo;
9   |     |   Spočítej délku trasy nalezené mravencem;
10  |     |   Nech  $k$  uložit na jeho cestu v  $P$  feromony;
11  |   end
12 end

```

Algorithm 2: Algoritmus ant systému v pseudokódu.

Volba trasy, ukládání feromonů a vypařování se pak řídí podle konkrétního mravenčího algoritmu, který je k řešení problému použit. Porovnání některých variant mravenčích algoritmů je možno najít např. v [16]. Více o aplikacích mravenčích algoritmů na další NP-těžké problémy lze najít např. v [16, 12].

4.2 Další aplikace mravenčích algoritmů

Optimalizace pomocí mravenčích algoritmů byla použita pro celou řadu praktických úloh. V této podkapitole přiblížíme několik ilustrativních příkladů použití mravenčí optimalizace.

Blum, Bautista a Pereira [22] použili variantu mravenčí optimalizace k řešení problému návrhu jednoduché výrobní linky s cílem minimalizovat počet výrobních stanovišť. Výrobní linka je chápána jako posloupnost výrobních stanovišť, které jsou propojeny transportním systémem. Výroba probíhá vykonáním sady úkonů, které jsou vzájemně provázané a které mají přiřazenu určitou časovou náročnost. Úkony jsou prováděny na výrobních stanovištích. Cílem je navrhnout takovou výrobní linku, která by splnila všechny podmínky specifikované v zadání problému a zároveň využívala minima výrobních stanovišť. Blum a kolegové ukázali, že upravená mravenčí optimalizace dokáže vyřešit větší počet z referenční sady zadání v kratším čase než jiné algoritmy (heuristické metody a jiné mravenčí metody).

Metodu pro koordinaci krizového řízení inspirovanou mravenčími algoritmy vyvinuli Tatomir a Rothkrantz [23]. Navržený systém se zaměřil na organizaci pomoci poskytované lékaři v místě katastrofy a poskytl podporu prioritizaci zásahu u jednotlivých obětí a směřování lékařů mezi jednotlivými zraněnými. Celá úloha byla formulována jako grafový problém a řešení pomocí systému mravenčí kolonie bylo experimentálně vyhodnoceno za pomoci počítačových simulací. Ukázalo se, že řešení pomocí mravenčích

algoritmů je efektivnější než řešení hladovými algoritmy (Greedy Algorithms).

D'Acerno a kolegové [24] navrhli mravenčí algoritmus pro lepší a rychlejší simulaci transportních systémů a experimentálně ověřili jeho kvalitu.

K plánování údržby flotil dopravních prostředků využili mravenčí algoritmy Abrahão a Gualda [25]. Problém plánování údržby spočívá v nalezení optimální sekvence preventivních prohlídek tak, aby byla maximalizována životnost (tedy minimalizována poruchovost) a využití vozidel. Problém, formulovaný jako grafová úloha, byl řešen řadou mravenčích algoritmů a nejlepší varianta a nastavení algoritmu byly nalezeny v simulovaných počítačových experimentech.

Michael Eley se věnoval aplikaci mravenčích algoritmů na problém plánování zkoušek [26]. Tento problém spočívá v optimálním naplánování zkoušek v rámci omezeného počtu časových period tak, aby zatížení studentů nebylo příliš velké. Eley použil jednoduchý mravenčí algoritmus a hybridní variantu max-min mravenčího systému s lokálním vyhledáváním založeném na gradientním algoritmu (Hill Climbing) a na sadě testovacích úloh ukázal že již velmi jednoduché varianty řešení založené na mravenčích algoritmech vedou k lepším řešením než heuristické metody.

5 Budoucnost mravenčích algoritmů

Mravenčí algoritmy jsou jednou z nejúspěšnějších bio-inspirovaných metaheuristik. Jejich úspěch podpořil studium biologických systémů a návrh podobných algoritmů napodobující chování živočichů, např. včel [27] či světlušek [28, 29].

Mravenčí algoritmy vznikly jako zjednodušený počítačový model vzorů chování hmyzích kolonií. Dnes, dvacet let po představení mravenčích algoritmů, již víme, že představy, ze kterých vycházejí, jsou samy o sobě značně zjednodušené a více méně nepřesné [30]. Reálná mravenčí kolonie je stochastický stroj, ve kterém hrají interakce mnohem důležitější úlohu, než jsme se domnívali. Chování jednotlivých mravenců vychází ze sítí interakcí (Networks of Interactions) zahrnujících jak mravence v rámci jediné kolonie, tak jejich okolí. Deborah Grodon [30] ukázala, že komplexní chování kolonie vychází z interakce jejích členů, potřeb, které jsou v daný moment vyhodnoceny jako nejdůležitější, a interakce s okolím kolonie. Pozorování ukázalo, že mravenčí kolonie jsou mnohem dynamičtější, než se dříve předpokládalo. Například jednotlivé úkoly, vykonávané specializovanými jedinci jako bojovníci či dělníci, jsou v okamžiku, kdy to mravenci na základě četnosti interakcí s jedinci vykonávajícími různé úkoly vyhodnotí jako potřebné, okamžitě přebírány ostatními mravenci. Stává se tak, že mravenčí kuklu místo dělníka přenáší neohrabaný bojovník, nebo že se útočníkům postaví zranitelní farmáři. Neméně významnou roli hraje pro fungování opravdových mravenčích kolonií interakce s okolními společenstvími mravenců stejných i jiných druhů. Tyto poznatky jsou velmi podnětné pro další studium mravenčích algoritmů.

Mravenčí kolonie je distribuovaný, vysoce paralelní multiagentní systém. Algoritmy mravenčí optimalizace byly vyvinuty tak, že je možno je implementovat sekvenčně, kdy v program postupně simuluje chování jednoho mravence po druhém. V dnešní době, kdy se paralelní výpočetní prostředí stávají stále dostupnějšími, je možno nezávislé části

výpočtů mravenčích algoritmů provádět skutečně paralelně a tím je výrazně zrychlit. Mravenčím algoritmům se tak naskýtá šance být nasazeny v situacích, kdy je nutno výpočet provádět v řádu sekund či rychleji. Perspektivním ekosystémem pro paralelní varianty mravenčích algoritmů jsou mnohojádrové výpočetní akcelerátory vycházející z grafických procesorů (GPGPU), distribuovaná výpočetní prostředí či cloud. Všechny tyto platformy nabízejí šanci revidovat stávající modely mravenčích výpočtů, dále je rozšířit a inovovat.

Reference

- [1] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press, 1996.
- [2] I. Zelinka, Z. Oplatková, M. Šeda, P. Ošmera, and F. Včelař, *Evoluční výpočetní techniky: Principy a aplikace*. Praha: Technická Literatura BEN, 2008.
- [3] A. Engelbrecht, *Computational Intelligence: An Introduction, 2nd Edition*. New York, NY, USA: Wiley, 2007.
- [4] Y. Zhang, A. Kandel, T. Y. Lin, and Y. Yao, *Computational Web Intelligence: Intelligent Technology for Web Applications*. World Scientific Press, 2004.
- [5] J. Deneubourg, J. Pasteels, and J. Verhaeghe, “Probabilistic behaviour in ants: A strategy of errors?” *Journal of Theoretical Biology*, vol. 105, no. 2, pp. 259 – 271, 1983. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0022519383800071>
- [6] J. Pasteels, J. Deneubourg, and F. les treilles, *From individual to collective behavior in social insects: les Treilles Workshop*, ser. Experientia: Supplementum. Birkhaeuser, 1987. [Online]. Available: <http://books.google.cz/books?id=xd5JAQAIAAJ>
- [7] C. Detrain, J. Deneubourg, and J. Pasteels, *Information processing in social insects*. Birkh "auser Verlag, 1999. [Online]. Available: <http://books.google.cz/books?id=N9iw1nLPaacC>
- [8] S. Goss, S. Aron, J. Deneubourg, and J. Pasteels, “Self-organized shortcuts in the Argentine ant,” *Naturwissenschaften*, vol. 76, no. 12, pp. 579–581, Dec. 1989. [Online]. Available: <http://dx.doi.org/10.1007/BF00462870>
- [9] M. Ebling, M. DiLoreto, M. Presley, F. Wieland, and D. Jefferson, “An Ant Foraging Model Implemented on the Time Warp Operating System,” in *Proceedings of the SCS Multiconference on Distributed Simulation*, 1989, pp. 21–26.
- [10] M. Dorigo, “Optimization, Learning and Natural Algorithms,” Ph.D. dissertation, Politecnico di Milano, Italy, 1992.

- [11] M. Dorigo, V. Maniezzo, and A. Colorni, “Ant system: optimization by a colony of cooperating agents,” *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29–41, Feb. 1996. [Online]. Available: <http://dx.doi.org/10.1109/3477.484436>
- [12] M. Dorigo and T. Stützle, *Ant Colony Optimization*. Cambridge, MA: MIT Press, 2004.
- [13] A. Colorni, M. Dorigo, and V. Maniezzo, “Distributed Optimization by Ant Colonies,” in *European Conference on Artificial Life*, 1991, pp. 134–142.
- [14] M. Dorigo and C. Blum, “Ant colony optimization theory: A survey,” *Theoretical Computer Science*, vol. 344, no. 2-3, pp. 243–278, Nov. 2005. [Online]. Available: <http://dx.doi.org/10.1016/j.tcs.2005.05.020>
- [15] C. Blum and D. Merkle, *Swarm Intelligence: Introduction and Applications*. Springer Publishing Company, Incorporated, 2008.
- [16] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. New York, NY, USA: Oxford University Press, Inc., 1999. [Online]. Available: <http://portal.acm.org/citation.cfm?id=328320>
- [17] A. Abraham, H. Guo, and H. Liu, “Swarm intelligence: Foundations, perspectives and applications,” in *Swarm Intelligent Systems*, ser. Studies in Computational Intelligence, N. Nedjah and L. de Macedo Mourelle, Eds. Springer, 2006, vol. 26, pp. 3–25.
- [18] M. Beekman, G. A. Sword, and S. J. Simpson, “Biological foundations of swarm intelligence,” in *Swarm Intelligence*, ser. Natural Computing Series, C. Blum and D. Merkle, Eds. Springer, 2008, pp. 3–41.
- [19] B. Bullnheimer, R. F. Hartl, and C. Strauss, “A New Rank Based Version of the Ant System,” *Central european journal of operations research*, vol. 7, no. 1, pp. 25–38, 1999.
- [20] L. M. Gambardella and M. Dorigo, “Ant-q: A reinforcement learning approach to the traveling salesman problem,” in *ICML*, 1995, pp. 252–260.
- [21] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *Evolutionary Computation, IEEE Transactions on*, vol. 1, no. 1, pp. 67–82, August 2002. [Online]. Available: <http://dx.doi.org/10.1109/4235.585893>
- [22] C. Blum, J. Bautista, and J. Pereira, “Beam-aco applied to assembly line balancing,” in *ANTS Workshop*, ser. Lecture Notes in Computer Science, M. Dorigo, L. M. Gambardella, M. Birattari, A. Martinoli, R. Poli, and T. Stützle, Eds., vol. 4150. Springer, 2006, pp. 96–107.

- [23] B. Tatomir and L. J. M. Rothkrantz, “Ant based mechanism for crisis response coordination,” in *ANTS Workshop*, ser. Lecture Notes in Computer Science, M. Dorigo, L. M. Gambardella, M. Birattari, A. Martinoli, R. Poli, and T. Stützle, Eds., vol. 4150. Springer, 2006, pp. 380–387.
- [24] L. D’Acerno, B. Montella, and F. D. Lucia, “A stochastic traffic assignment algorithm based on ant colony optimisation,” in *ANTS Workshop*, ser. Lecture Notes in Computer Science, M. Dorigo, L. M. Gambardella, M. Birattari, A. Martinoli, R. Poli, and T. Stützle, Eds., vol. 4150. Springer, 2006, pp. 25–36.
- [25] F. T. M. Abrahão and N. D. F. Gualda, “Fleet maintenance scheduling with an ant colony system approach,” in *ANTS Workshop*, ser. Lecture Notes in Computer Science, M. Dorigo, L. M. Gambardella, M. Birattari, A. Martinoli, R. Poli, and T. Stützle, Eds., vol. 4150. Springer, 2006, pp. 412–419.
- [26] M. Eley, “Some experiments with ant colony algorithms for the exam timetabling problem,” in *ANTS Workshop*, ser. Lecture Notes in Computer Science, M. Dorigo, L. M. Gambardella, M. Birattari, A. Martinoli, R. Poli, and T. Stützle, Eds., vol. 4150. Springer, 2006, pp. 492–499.
- [27] *Application of the Bees Algorithm to the Training of Learning Vector Quantisation Networks for Control Chart Pattern Recognition*, vol. 1, 2006. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1684627
- [28] X.-S. Yang, “Firefly algorithm, lévy flights and global optimization,” in *SMAI Conf.*, M. Bramer, R. Ellis, and M. Petridis, Eds. Springer, 2009, pp. 209–218.
- [29] ———, “Firefly algorithms for multimodal optimization,” in *SAGA*, ser. Lecture Notes in Computer Science, O. Watanabe and T. Zeugmann, Eds., vol. 5792. Springer, 2009, pp. 169–178.
- [30] D. M. Gordon, *Ant Encounters: Interaction Networks and Colony Behavior*, ser. Primers in complex systems. Princeton University Press, 2010. [Online]. Available: <http://books.google.com/books?id=MabwdXLZ9YMC>
- [31] M. Dorigo, L. M. Gambardella, M. Birattari, A. Martinoli, R. Poli, and T. Stützle, Eds., *Ant Colony Optimization and Swarm Intelligence, 5th International Workshop, ANTS 2006, Brussels, Belgium, September 4-7, 2006, Proceedings*, ser. Lecture Notes in Computer Science, vol. 4150. Springer, 2006.