



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

VIZUALIZACE ADAPTIVNÍHO MRAVENČÍHO ALGORITMU

VISUALISATION OF ADAPTIVE ANT COLONY OPTIMIZATION ALGORITHM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Vojtěch Tichý

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Martin Appel

BRNO 2017

Zadání bakalářské práce

| | |
|-------------------|--|
| Ústav: | Ústav mechaniky těles, mechatroniky a biomechaniky |
| Student: | Vojtěch Tichý |
| Studijní program: | Aplikované vědy v inženýrství |
| Studijní obor: | Mechatronika |
| Vedoucí práce: | Ing. Martin Appel |
| Akademický rok: | 2016/17 |

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Vizualizace adaptivního mravenčího algoritmu

Stručná charakteristika problematiky úkolu:

V rešerší části bude cílem popsat možnosti vizualizace prohledávacích a optimalizačních metod, porovnat mravenčí algoritmus s jinými optimalizačními metodami.

V praktické části bude cílem naprogramovat mravenčí algoritmus tak, aby se přizpůsoboval měnícímu se prostředí. Algoritmus bude optimalizovat vynaloženou energii mravenců cestou za potravou při cestě terénem.

Cíle bakalářské práce:

- Popsat možnosti vizualizace prohledávacích a optimalizačních metod.
- Porovnat mravenčí algoritmus s jinými optimalizačními algoritmy.
- Naprogramovat mravenčí adaptivní algoritmus.
- Vytvořit prezentační model.
- Vytvořit vhodnou vizualizaci mravenčího algoritmu pro použití ve výuce.

Seznam doporučené literatury:

ZELINKA, I.: Umělá inteligence v problémech globální optimalizace. 1. vyd. Praha: BEN - technická literatura, 2002, 189 s. : grafy, tab. ISBN 80-7300-069-5.

KNOFLÍČEK, J.: Analýza různých přístupů k řešení optimalizačních úloh. Brno, 2013, Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií.

ZELINKA, I.: Evoluční výpočetní techniky: principy a aplikace. Praha: BEN - technická literatura, 2009. ISBN 978-80-7300-218-3.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2016/17

V Brně, dne

L. S.

prof. Ing. Jindřich Petruška, CSc.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

Abstrakt

Bakalářská práce se zabývala naprogramováním a vytvořením modelu pro edukaci a pochopení fungování optimalizace mravenčí kolonií (ACO – Ant Colony Optimization), která byla upravena tak, aby byla schopna se adaptovat na změnu terénu. Práce byla dále zaměřena na porovnání několika dalších přírodou inspirovaných optimalizačních metod a vytyčení jejich využití v praktických situacích.

Abstract

Bachelor thesis was dealing with programming and creating model for education and understanding of Ant Colony Optimization functioning, which was modified to be able to adapt on the change of terrain. Thesis was further focused on comparing several optimization methods inspired by nature and demarcation their utilization in practical situations.

Klíčová slova

Optimalizace, Heuristická funkce, Adaptivní mravenčí algoritmus, Optimalizace mravenčí kolonií, Matlab, Kinect

Keywords

Optimization, Heuristic function, Adaptive ant algorithm, Ant colony optimization, Matlab, Kinect

Bibliografická citace

TICHÝ, V. *Vizualizace adaptivního mravenčího algoritmu*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2017. 33 s. Vedoucí bakalářské práce Ing. Martin Appel.

Čestné prohlášení

Prohlašuji, že jsem bakalářskou práci na téma “Vizualizace mravenčího adaptivního algoritmu“ vypracoval samostatně s použitím odborné literatury a pramenů, které jsou uvedeny v seznamu použitých zdrojů.

V Brně dne 25. května 2017

Vojtěch Tichý

Poděkování

Mé poděkování patří Ing. Martinu Appelovi, který mi vyhověl při každém problému a měl vždy velice cenné připomínky k mé práci. Dále bych chtěl poděkovat své rodině a přátelům za jejich podporu a vstřícnost.

Obsah

| | |
|---|--------|
| Úvod..... | - 11 - |
| 2 Optimalizace..... | - 12 - |
| 2.1 Heuristická funkce | - 12 - |
| 2.2 Mravenčí algoritmus | - 12 - |
| 2.2.1 Princip fungování ACO..... | - 13 - |
| 2.2.2 Double Bridge experiment | - 13 - |
| 2.3 Některé další optimalizační metody inspirované přírodou | - 15 - |
| 2.3.1 Včelí Algoritmus | - 15 - |
| 2.3.2 Evoluční Algoritmus | - 15 - |
| 2.4 Porovnání a možnosti optimalizačních metod | - 16 - |
| 2.4.1 Vlastnosti..... | - 16 - |
| 2.4.2 Využití v praxi | - 16 - |
| 2.4.3 Shrnutí využití optimalizačních metod..... | - 17 - |
| 2.5 Možnosti vizualizace optimalizačních metod..... | - 17 - |
| 2.5.1 Napodobení situace v přírodě..... | - 17 - |
| 2.5.2 Hodnotící funkce | - 18 - |
| 3 Vizualizace adaptivního ACO..... | - 19 - |
| 3.1 MATLAB a KINECT | - 19 - |
| 3.1.1 Senzory Kinect s adaptérem pro Windows | - 19 - |
| 3.1.2 Vektorové prostředí Matlab..... | - 19 - |
| 3.2 Stanovení problému | - 20 - |
| 3.3 Program adaptivního ACO..... | - 20 - |
| 3.3.1 Inicializace..... | - 20 - |
| 3.3.2 Adaptivní ACO..... | - 20 - |
| 3.3.3 Prohledávací vlastnosti..... | - 22 - |
| 3.3.4 Vizualizace pohybu a feromonů..... | - 22 - |
| 3.4 Konstrukce a provedení modelu..... | - 23 - |
| 3.5 Hledání ideálních parametrů | - 24 - |
| 3.5.1 Konstanty α a β | - 24 - |
| 3.5.2 Množství ukládaného feromonu a jeho vypařování | - 24 - |
| 3.5.3 Heuristická funkce a počet mravenců..... | - 24 - |

| | | |
|-------|---|--------|
| 4 | Výsledky | - 25 - |
| 4.1 | Zvolení parametrů | - 25 - |
| 4.1.1 | Hodnoty ze simulace | - 25 - |
| 4.1.2 | Experimentálně určené hodnoty | - 27 - |
| 5 | Závěr | - 28 - |
| 6 | Seznam použitých zdrojů | - 29 - |
| 7 | Seznam použitých zkratk a symbolů | - 31 - |
| 7.1 | Zkratky | - 31 - |
| 7.2 | Symboly | - 31 - |
| 7.3 | Funkce v prostředí Matlab | - 31 - |
| 8 | Seznam obrázků a grafů | - 32 - |
| 8.1 | Seznam obrázků | - 32 - |
| 8.2 | Seznam Grafů | - 32 - |
| 9 | Seznam Příloh | - 33 - |
| 9.1 | Přílohy na CD | - 33 - |

Úvod

V dnešním světě je velmi silná touha po nalezení nejjednoduššího, nejlevnějšího a někdy i nejspolehlivějšího řešení v mnoha odvětvích dnešní společnosti. Od základních věcí jako je úspora energie v domácnostech, přes ušetření paliva při cestách a rozumném investování, až po vyřešení složitých problémů nebo dekodování složitých šifer.

Jedním z mnoha způsobů, které pomáhají řešit takové problémy, je optimalizace. Je to velice efektivní způsob, jak nalézt ideální a nejvhodnější řešení pro široké odvětví problémů, které nalezneme v dnešní době téměř ve všech systémech, které musí reagovat na neschopnost nalézt exaktní řešení.

Mezi způsoby optimalizace patří také algoritmy inspirované přírodou a právě tím je i Optimalizace Mravenčí Kolonií. Díky ní bude v této práci snaha nalézt nejvhodnější řešení pro dopravu mezi dvěma místy co nejefektivnějším a nejrychlejším způsobem. Zadání je ztíženo tím, že prostředí, ve kterém se bude tato ideální cesta hledat, je proměnné, a proto musí být algoritmus připraven tak, aby se dokázal vzdát původní cesty, kterou již označil za nejlepší, a hledal lepší řešení.

Tato optimalizace bude prováděna ve vektorově založeném prostředí MATLAB, ve kterém bude celý optimalizační problém řešen. Pro lepší vysvětlení a pochopení této optimalizační metody bude vytvořen vizualizační model, na kterém bude zobrazeno, jak tato optimalizační metoda pracuje.

V této práci bude i ukázáno, jak postupovat při implementování této metody pro podobný problém, který chceme optimalizovat.

2 Optimalizace

Optimalizační algoritmy jsou mocným nástrojem pro řešení mnoha problémů inženýrské praxe. Obvykle se používají tam, kde je řešení daného problému analytickou cestou nevhodné či nereálné [1]. Optimalizace je využívána ve všech odvětvích řízení proměnných situací, kde je vyžadováno „vlastního“ rozhodování v reálném čase za co nejkratší čas tak, aby bylo nalezeno globální optimum problému.

Algoritmů, které se zabývají optimalizací, je mnoho. Podstatou je co nejlepší a nejrychlejší nalezení řešení. Optimalizační metody se dají matematicky rozdělit podle typu problematiky na lineární a nelineární nebo na diskrétní a spojité funkce.

Další možné rozdělení může být podle jejich deterministického přístupu k problému. Tedy záleží na jejich opakovatelnosti.

2.1 Heuristická funkce

Heuristika je založená na vlastním pseudonahodilém způsobu prohledávání prostoru a hledání možných řešení složitých algoritmů při využití jistého druhu zkušenosti, náhody nebo stylu prohledávání [2]. V Optimalizaci Mravenčí Kolonií je tato vlastnost velmi důležitá, protože díky ní se dokáží mravenci pohybovat v prostoru s jistou orientací podle „zkušeností“ ostatních mravenců, kteří již tuto cestu absolvovali. Ovšem heuristická vlastnost musí být velice pečlivě nastavena, aby mravenci buďto nevolili stále stejnou trasu, nebo se od ní stále odchylovali. Při prohledávání je vždy potřebné, aby zde byli „průzkumníci“ a „dělníci“. Zvolení vhodné heuristické funkce lze obejít tím, že se mravenci rozdělí do těchto podskupin a podle nich se chovají.

Výhoda i jistého druhu nevýhoda je při volbě správné heuristické funkce v tom, že se mravenci při prohledávání nemusí tak vzdálit od dřívější cesty, proto ideální optimalizace trvá trochu déle než může být požadováno.

Může však nastat to, že při různých typech problémů bude nalezena ideální (optimalizovaná) cesta. To nelze zaručit hlavně proto, že heuristika je založena na náhodě a zkušenosti, které mohou být jakousi dočasnou odchylkou v prostoru.

2.2 Mravenčí algoritmus

Mravenčí algoritmus, neboli Algoritmus Mravenčí Kolonií (dále jen ACO), je optimalizační metoda, která pramení z chování mravenců, kteří tvoří kolonie čítající tisíce až miliony jedinců, a jejich způsobu komunikace a vyhledávání potravy ve svém okolí. Jejich velice specifickým způsobem komunikace je komunikace skrze prostředí. Mravenci se při běžném pohybu pohybují zcela nahodile. V momentu nalezení potravy vylučují při zpáteční cestě do mraveniště látky – feromony, na které ostatní mravenci dokáží zareagovat a tuto feromonovou stopu následovat. Této komunikaci mezi mravenci se říká nepřímá komunikace prostředím. Mravenci se tak dokáží přizpůsobovat prostředí a jeho změnám velice rychle.

Velmi silným aspektem komunikace skrze feromonovou stopu je intenzita a také pomíjivost feromonu, který může být překonán feromonem intenzivnějším nebo místem, kde se mravenci sami často pohybují a tak je zde feromon nahromaděný.

Dalším podstatným faktorem této prostorové orientace je vypařování feromonu, díky které může mravenec označenou cestu najít a držet se jí. Ovšem v samotném algoritmu funguje pouze jako známka toho, že když je zvolená cesta téměř bez feromonu, je cesta nezajímavá, a proto se touto cestou mravenec spíše nevydá.

2.2.1 Princip fungování ACO

Celý princip a fungování této optimalizační metody spočívá v jedné pravděpodobnostní rovnici, která velmi věrně napodobuje způsob, jakým jednají mravenci [3]. Parametrů rovnic pro algoritmus je vcelku několik. Ty, které nejsou obsaženy v rovnici, jsou například vypařování feromonu nebo počet iterací. Rovnice je ukázána zde:

$$p_{ik} = \frac{(f_{ik})^\alpha \cdot (h_{ik})^\beta}{\sum_j^N (f_{jk})^\alpha \cdot (h_{jk})^\beta} \quad (2.1)$$

V této rovnici je zohledněn vliv feromonu, náročnost vybrané cesty a heuristická funkce. Ta je zde proto, aby mravenec dokázal zvolit novou cestu, a tím se otvírá možnost nalezení nové a lepší cesty.

Každý ze členů této rovnice má svůj význam. f je množství feromonu na zvolené cestě od bodu i do bodu k , h je již zmíněná heuristická funkce, kterou může představovat konstanta nebo klidně složitější závislá funkce, α a β jsou koeficienty vlivu feromonu či heuristické vlastnosti na pravděpodobnost a p_{ik} je pravděpodobnostní hodnota pro zvolenou cestu od bodu i do bodu k [3].

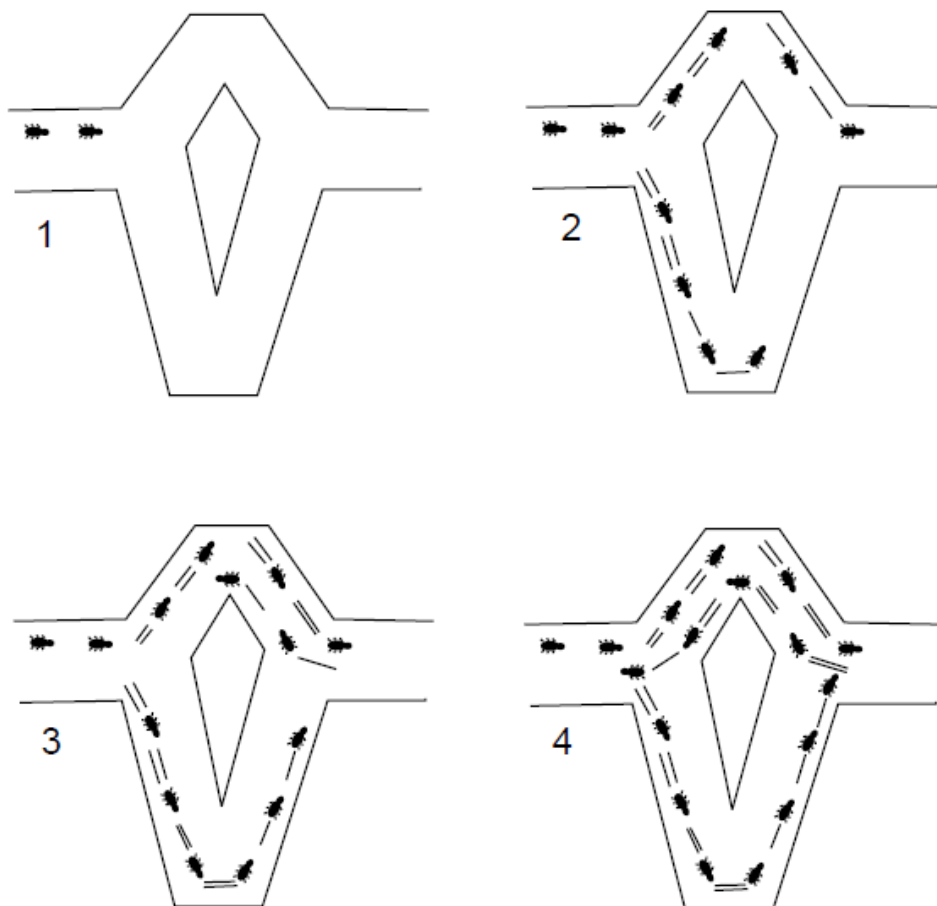
Koeficienty α a β musí být zvoleny, protože pro různé optimalizační úlohy mohou fungovat jiné parametry. To platí také pro množství pokládaného feromonu, počtu iterací, počtu mravenců, či pro hodnotu rychlosti vypařování feromonu. Všechny tyto parametry je potřeba vhodně nastavit na specifický druh problému.

2.2.2 Double Bridge experiment

Základní princip chování tohoto algoritmu lze dobře vysvětlit na „Double bridge“ experimentu. Ten je zakreslen v následujícím obrázku. Na něm je zobrazen most, který se uprostřed rozděluje na dvě odlišné části - na viditelně kratší a delší úsek. Pro lepší vysvětlení je „Double bridge“ rozdělen do 4 mezikroků, kterými můžeme popsat chování mravenců [4]:

- 1) Mravenci vcházejí na most a přichází k bodu, kde se musí rozhodnout, kterou cestou se mají vydat. Jejich rozhodnutí naprosto nezávisí na feromonech, protože místem procházejí poprvé. Proto se mravenci vydávají oběma variantami poměrně stejně bez znalosti jejich kvality.

- 2) Na dalším obrázku můžeme vidět, že mravenci, kteří zvolili kratší cestu, již našli potravu a vrací se zpět do mraveniště, zatímco mravenci na delším úseku k ní teprve přicházejí.



Obrázek 2.1 Double bridge experiment ve čtyřech krocích [4].

- 3) Mravenci se nyní vrací po vlastní příchozí cestě, a tím, že na ní zanechávají feromon, ji zesilují. Na druhé straně k tomu nedochází, protože se zde mravenci ještě nevrací zpět, a je zde proto méně feromonu než na kratší straně.
- 4) V posledním kroku mravenci přichází zpět do mraveniště a jdou opět hledat potravu. Když dojdou ke křižovatce, volí kratší cestu kvůli tomu, že na této cestě je více feromonu, protože po delší trase neprošlo tolik mravenců jako na kratší ve stejném čase, a tedy na ní není feromon tak silný.

Tento experiment poukazuje na jednoduchost a praktičnost algoritmu. Nejčastější využití ACO nalézá v optimalizaci sítí, rozvodů a podobně založených oblastí.

2.3 Některé další optimalizační metody inspirované přírodou

2.3.1 Včelí Algoritmus

Tato optimalizace patří mezi nejnověji využívané algoritmy. Umělá včelí kolonie (Artificial bee colony - ABC) byla vyvinuta v roce 2005 [5]. Jak už z názvu optimalizace vyplývá, tak je inspirována včelí kolonií.

Včela prozkoumává své okolí a hledá květinu. Při nálezů zjistí množství nektaru a tuto informaci uchová. Při nalezení další květiny porovná množství nektaru s předchozí a rozhoduje se, zda u této květiny zůstane a bude čerpat z ní nebo se vrátí zpět nebo bude ještě prohledávat okolí. Tímto způsobem najde nejlepší zdroj potravy a z něj čerpá, dokud nenalezne lepší řešení, lepší zdroj.

Výhodou této optimalizace je počet řídicích parametrů. Mimo základní parametry jako je počet iterací či počet agentů, má ABC pouze jeden řídicí parametr, ale i přes tuto vlastnost vykazuje ABC velmi dobrá řešení různých optimalizačních problémů [6].

2.3.2 Evoluční Algoritmus

Jak už z názvu vyplývá, jedná se o algoritmus, který se časem samostatně mění a vyvíjí neboli prochází evolucí. Evoluční algoritmus (EA) vychází přímo z evoluční teorie, kterou prosazovali nejvíce Darwinovci a která je mezi lidmi považována za pravdivou [7].

EA obsahuje 5 základních prvků, které formují optimalizační metodu tak, aby opravdu fungovala. Jsou to: populace, mutace, křížení, elitismus, generace [1].

Populace je vygenerovaná nebo již upravená množina jedinců, se kterou se pracuje v celém algoritmu [4]. Každý jedinec je trochu jiný (má jiné parametry či vlastnosti). Při průběhu tohoto typu optimalizace se populace obměňuje podle dalších prvků algoritmu.

Aby byl algoritmus účinný, musí proběhnout buďto mutace nebo křížení. Mutace je spíše náhodné pozměnění nového jedince tak, aby lépe dokázal splnit svůj úkol.

Elitismus je prosté vybrání nejlepších jedinců, kteří nejlépe optimalizují danou situaci. Počet vybraných jedinců pak záleží pouze na tom, jaké jsou požadavky.

Křížení není nutný prvek, ale díky němu lze velmi efektivně urychlit optimalizaci. Spočívá ve vybírání jedinců, kteří mají dobré výsledky, ale v některých částech se příliš vychylují od požadovaného výsledku. Proto se pro novou generaci vyberou někteří jedinci, u kterých dojde k vhodné kombinaci jejich vlastností – tedy ke křížení. Tento prvek je velice dobře pozorovatelný u křížení rostlin, zvířat, či při vytváření slitin pro dosažení lepších vlastností nebo vzhledu.

Generace je cyklus, ve kterém se objevují všechny prvky. Generace se vždy skládá z nových jedinců, kteří prošli mutací či křížením [4]. Vyskytují se zde i jedinci, kteří byli označeni elitismem za nejlepší. Ti mají důležitější postavení v generaci

obzvláště díky tomu, že pokud ob stojí v elitismu znovu, pak je buďto nalezen optimální výsledek nebo byla daná generace příliš odkloněna od požadovaného výsledku.

Kombinací v EA je mnoho a dají se velmi dobře měnit a vytvářet spoustu variací.

2.4 Porovnání a možnosti optimalizačních metod

2.4.1 Vlastnosti

Každá z těchto metod, které byly uvedeny, mají svůj vlastní styl fungování. Velké rozdíly jsou především ve způsobu spolupráce jednotlivých členů skupiny a také počtem nastavitelných parametrů.

Co se týká první vlastnosti (spolupráce), tak v EA se jedná o sdílení informace po dokončení jednoho úseku a pak sdílení, vylepšování, či změna dalšího postupu. U metody ACO je to komunikace skrze prostředí, kde umísťují feromony a u ABC jde o hodnotu, kterou má samotný cíl cesty.

Pokud jde o počet parametrů, když nebudeme počítat obvyčejné parametry, kde je počet iterací nebo počet jedinců, tak EA jich má zřejmě nejvíce. Jsou zde parametry např. jak často dojde ke křížení, kdy dojde pouze k mutaci nebo kdy zůstane elitní jedinec. Ve velmi těsném závěsu je ACO, které má jen o něco méně nastavitelných parametrů. Zde to je počet jedinců, koeficienty v pravděpodobnostní rovnici, rychlost vypařování feromonu a další. ABC je mezi těmito optimalizačními metodami nejméně náročná na volbu parametrů. Mimo obvyčejné parametry se volí pouze jeden.

2.4.2 Využití v praxi

Co se týče využití, je zde vidět, že každá z uvedených optimalizačních metod se věnuje oblasti, která velmi blízce připomíná samotný děj v přírodě. Z toho lze vyvodit, že je veliký problém posoudit, která optimalizační metoda je celkově nejlepší, nejrychlejší či nejpřesnější. Dají se posoudit nejlépe na jednoduchém problému, který nemá příliš složité řešení, ale i zde bychom našli spoustu rozdílů.

ABC vykazuje ideální vlastnosti pro kontrolu a správu bezdrátových sítí. Jak je vysvětleno v kapitole 2.2.1, algoritmus je zaměřen na množství obsahu pylu ve květině. Tedy to se dá představit např. pro wifi připojení, kdy je k jednomu routeru připojeno několik uživatelů [8]. ABC zde funguje jako senzor slabého připojení (málo pylu), aby posílil pro daného uživatele signál, i třeba na úkor ostatních uživatelů [8].

EA je využíváno velmi často pro velice složité procesy či mechanismy, kdy se snaží dosáhnout dostatečného výsledku při co nejnižších nákladech. Jde tedy například o vylepšování průmyslových technik výroby, sloučitelnost postupů při výrobě nebo samotné provedení úkonu při průmyslové výrobě. EA je v řešení těchto problémů velice zdatný, neboť postup výroby jednoho výrobku obsahuje mnohdy nespočet jednoduchých úkonů a technik.

ACO, stejně jako ABC, je uplatňovaná v telekomunikačních sítích, ale zde spíše pro kabelové spoje. To ovšem nemusí být vždy pravda. ACO zde pomáhá nalézt nejlepší cestu, a tak usměrňuje signál procházející síťovou smyčkou. ACO optimalizace

funguje tak, že se informace zanechává právě v síťových uzlech, kde tak mohou co nejrychleji dopravit informaci na místo určení [9].

2.4.3 Shrnutí využití optimalizačních metod

Z předchozího textu je tedy naprosto jasné, že někdy se optimalizační metody nedají ani porovnávat. To je vidět obzvláště u EA, které se na rozdíl od zbylých metod, nevěnuje rozvodným telekomunikačním sítím. I přesto však u všech zmíněných metod je naprosto stejný cíl: dosáhnout co nejlepšího výsledku nejjednodušeji a co nejrychleji.

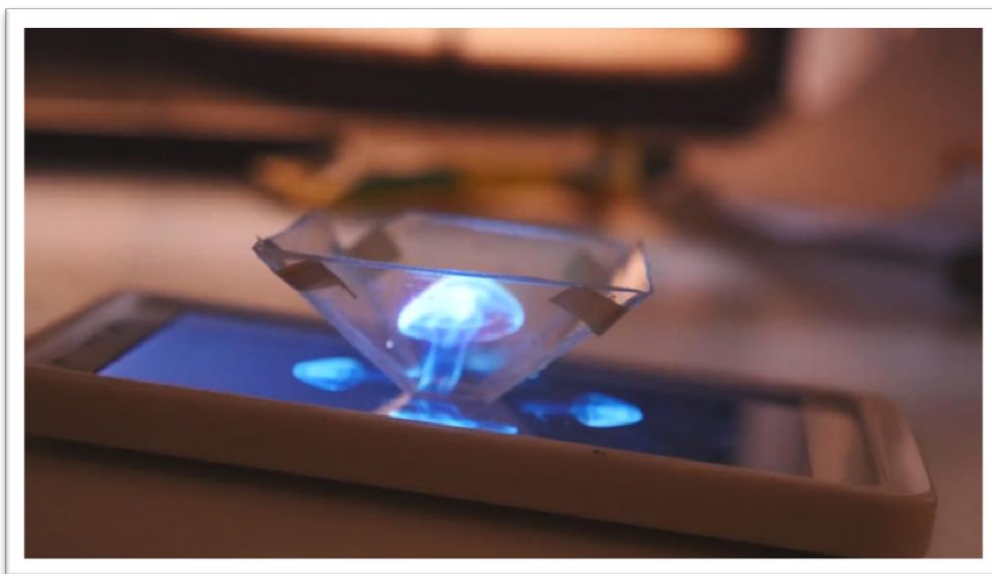
Další společná a velice zásadní vlastnost je správné nastavení parametrů. To vyžaduje spíše metodu pokus – omyl, protože se ironicky dobrá optimalizace odvíjí od správně nastavených parametrů. Tedy optimalizace je sice velmi užitečná a praktická pomůcka v dnešní době, ale za každou dobře provedenou optimalizací stojí spousta času strávená při zkoušení a testování ideálních parametrů.

2.5 Možnosti vizualizace optimalizačních metod

2.5.1 Napodobení situace v přírodě

Pro dobrou představu, jak samotná optimalizační metoda pracuje, je zřejmě ideálním postojem se přímo zaměřit na inspiraci z přírody, ze které tyto zde popsané metody čerpaly.

Pro EA a ACO není takový problém zrealizovat velmi věrohodně tyto algoritmy, jak vypadají v přírodě, ale u ABC je to velmi obtížné hlavně z důvodu, že včely létají a technologie, které by dokázaly let včely v prostoru dobře vizualizovat, nejsou zatím na takové úrovni. Existují sice již některé holografické zařízení, ale o kvalitním napodobení zatím můžeme jen debatovat. Možná současná vizualizace skrze jednoduchý kosý jehlan je zobrazena na obr. 2.2.



Obrázek 2.2 Možné jednoduché zobrazení ABC – Hologram [10].

ACO bude velice jednoduché napodobit, protože se mravenci pohybují po zemi a prostorové vnímání může být zobrazeno několika funkcemi, mezi kterými je samotná projekce na již zformovaný terén nebo jen projekce, kde rozdíly mezi body zobrazují barevné odstíny. Feromon může být také zobrazován jako po čase mizící bod.

Pro Evoluční algoritmy bude nejjednodušší vizualizační metodou také projekce prostředí. Zde bude rozdíl od Mravenčího algoritmu v tom, že se musí poukazovat, co se po překonání jednoho úseku děje s jedincem a populací. Možné řešení je při použití jedince jako malou figurínu, která bude nejprve zobrazena jako malé dítě, později jako dospělý jedinec a jeho poslední fází bude zobrazení starce. Systém křížení by se dal pojmout tak, že každý jedinec bude například jinak oblečen a při křížení dojde ke směsi buďto přímo barev (příkladem může být kombinace žluté a modré barvy, kdy vznikne zelená) nebo jen kombinace části oděvu pro nového jedince.

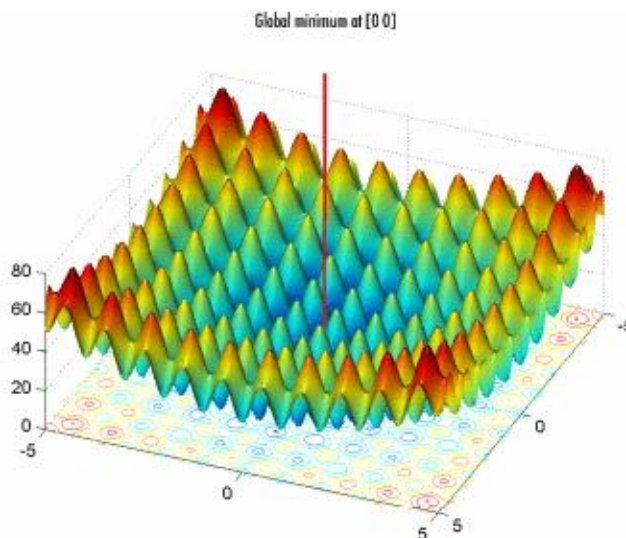
2.5.2 Hodnotící funkce

Hodnotící funkce mají spíše testovací vlastnosti, kde se ukazuje, zda-li je například funkce schopna překonat lokální minimum nebo lokální maximum. Jednou z těchto funkcí je i Rastriginova funkce:

$$F(x, y) = 20 + x^2 + y^2 - 10 \cdot [\cos(2\pi x) + \cos(2\pi y)] \quad (2.2)$$

Na této funkci tedy můžeme dobře “vizualizovat” všechny optimalizační metody, které jsou v této práci uvedeny. Díky tomuto vizualizačnímu prvku tedy můžeme porovnat přednosti a nedostatky všech optimalizačních metod.

Odhadovat, která z optimalizačních metod bude lépe působit v Rastriginově funkci, by bylo zbytečné. Metody se mohou vždy chovat odlišně a vykazovat jiné výsledky při opakovaném testování. Grafické zobrazení Rastriginovy funkce je na obr. 2.3.



Obrázek 2.3 Rastriginova funkce [11].

3 Vizualizace adaptivního ACO

Tato kapitola se zabývá implementací ACO algoritmu a jeho vytvoření a fungování. Podmínky pro spuštění jsou velice náročné, protože pro synchronizaci zařízení Kinect a prostředí Matlab je zapotřebí OS Windows 10 a software Matlab 2016a. To je způsobeno adaptérem ke Kinectu, který není přímo určen pro počítač, ale pro Xbox.

3.1 MATLAB a KINECT

3.1.1 Senzory Kinect s adaptérem pro Windows

Toto zařízení se nejčastěji prodává jako příslušenství pro interaktivní hry, kde je možné hru ovládat za pomoci gest a pohybů. Zařízení Kinect obsahuje dva typy senzorů, kameru s rozlišením 1080p, která zachycuje obraz, a hloubkový infračervený senzor pro snímání vzdálenosti od objektů před ním, který má záběrnou oblast 512x424 pixelů a dokáže rozpoznat 2 body na vzdálenost od 0,5 m až 4,5 m [12].

Aby bylo možné spárovat toto zařízení s počítačem, je třeba využít adaptér Kinect pro Windows. Zařízení Kinect je na obr. 3.1.



Obrázek 3.1 Zařízení Kinect se zapnutými hloubkovými senzory.

3.1.2 Vektorové prostředí Matlab

Matlab je vektorové prostředí, které je přizpůsobeno pro řešení inženýrských a vědeckých problémů. Matematický jazyk Matlab je světově nejlepším způsobem, jak

pracovat s výpočetní matematikou. Grafické rozhraní umožňuje pozorovat a získávat potřebné hodnoty. Knihovna s rozsáhlým množstvím nástrojů umožňuje jednoduše porozumět a vytvořit potřebný algoritmus [13].

Pro potřeby této práce je také využíván toolbox *Image Aquisition*, který umožňuje propojení Matlabu a senzorů Kinect. Také dokáže tyto senzory ovládat a získávat z nich potřebná data.

3.2 Stanovení problému

Chceme vytvořit model, kde budeme moci pozorovat, jak se v dané situaci chovají mravenci při pohybu v terénu. Na to by stačila pouze dobrá platforma, ale pro lepší představu bude terén realizován pískem v krabici, která bude snímána hloubkovými senzory, a tak bude možné prostředí měnit prohrábnutím rukou v písku, čímž se terén bude měnit. Velmi složitou částí tohoto problému není nastavení samotného algoritmu, ale vytvořit jej tak, aby dokázal změnit svou původní cestu. To ještě umocňuje potřebu ideálního nastavení parametrů. Proto část práce bude zaměřena na nalezení ideálních parametrů.

Dále je třeba zvolit umístění projektoru, Kinect senzorů a synchronizovat je spolu s umístěním krabice s pískem, aby odpovídala realita s promítanými informacemi.

3.3 Program adaptivního ACO

Program byl nastaven tak, aby se dokázal sám rozhodovat a volit ideální cestu, podle svých parametrů. Další potřebné podúseky programu jsou například prohledávání okolí, čtení a vykreslování terénu nebo rychlost mravence do kopce či z kopce. Následné podkapitoly obsahují funkce a názvosloví z prostředí MATLAB.

3.3.1 Inicializace

Pro správnou funkčnost se musí vše nastavit tak, aby se program mohl rozběhnout. Na prvních řádcích ve skriptu se inicializuje zobrazování terénu, kde se zobrazují vrstevnice a pohybující se mravenci. Dále kvůli rychlosti provedení jsou zde také stanovené všechny používané proměnné, aby se velikost maticí během průběhu zbytečně neměnily.

Zde jsou stanovené všechny parametry. Experimentálně je zde nastavena pozice pro promítání vrstevnic a celkové nastavení promítacích parametrů. Způsob zobrazení výškových hodnot v terénu bude proveden funkcí `contour`, která vykresluje barevně rozlišené vrstevnice.

3.3.2 Adaptivní ACO

Program je napsán tak, aby dokázal reagovat na změny okolí. Proto každý mravenec při svém pohybu dělá několik základních kroků. Nejprve zkontroluje, zda již nenalezl cíl, následně provede sérii pohybových kroků, které nejsou kvůli zrychlení zobrazování vidět. Ty zrychlí program tak, že místo více kroků se zobrazí pouze jeden větší. Jejich počet závisí na náročnosti a uražené vzdálenosti. Během těchto malých

kroků si hlídá, zda-li již na tento bod během svého pohybu k potravě nedorazil, kontroluje, zda-li není na okraji, svou energii, obtížnost okolních bodů a množství feromonů na nich umístěných. Po této kontrole se provede shrnutí možností a jejich ohodnocení skrze pravděpodobnostní rovnici. Náhodným číslem vybereme jednu z těchto možností, kam se má mravenec vydat. Tato hodnota přepíše původní pozici a tuto pozici mravenec ohodnotí určitým množstvím feromonu, které závisí na náročnosti terénu mezi těmito dvěma body, které mravenec právě překonal. Pokud mravenec „spotřeboval“ již svou energii, tak se vykreslí současná pozice. Pokud se ještě tak nestalo, opakuje se opět proces výběru dalšího kroku. Níže v obr. 3.2 je uvedena část pseudokódu, kde jsou ukázány základní části adaptivního ACO algoritmu.

Energie mravence je hodnota, kterou má každý mravenec stejnou. Pokud například mravenec musí překonat veliké převýšení, je mu ubrána velká energie, stejně jako v realitě. Proto neujde takovou vzdálenost jako mravenec, který jde takřka po rovné ploše. Hodnota energie mravence byla zvolena intuitivně na základě potřeb a dobrého zobrazení při vizualizaci.

Po každém postoupení po vyčerpání energie u všech mravenců se odpařují feromony. Tahle akce je pouze na jednom řádku, kdy se hodnota všech feromonů násobí hodnotou menší než jedna, ale blízkou jedné. Hodnota feromonů tedy klesá pomalu, ale jelikož klesá velmi často, musí být odpařování velmi malé.

Ve chvíli, kdy mravenec dorazí k potravě, se mravenec otáčí zpět a jde do mraveniště stejnou cestou, jakou přišel. Tato vlastnost je sice velmi jednoduše proveditelná, ale dělá problém při změně prostředí, na kterou již mravenec nijak nereaguje. Proto při zpáteční cestě mravenec neprodukuje již tolik feromonu.

```

WHILE počet iterací
  FOR každý mravenec postupně
    IF nalezena potrava
      WHILE počet kroků v jedné iteraci
        FOR Zjištění možných směrů, kam může mravenec jít
          Kontrola, zda-li byl bod již navštíven nebo se jedná o stěnu
          Výpočet náročnosti v každém směru
          Pravděpodobnostní funkce pro každou možnost, pokud
            prošla přes kontrolu
          END
          Volba kroku na základě pravděpodobnostních hodnot a
            náhodného čísla
          Odečtení spotřebované energie
          Posun mravence na novou pozici
          Přidání feromonu na novou pozici
          IF zda-li již není mravenec vyčerpán
            END
          END
        END
      END
    END
  END
  Vypařování feromonů
END

```

Obrázek 3.2 Část pseudokódu ACO – hlavní *while* smyčka.

3.3.3 Prohledávací vlastnosti

Mravenci mají schopnost hlavně na začátku prohledávat okolí a nalézt potravu. Tato část kódu není nijak náročná na vytvoření, ale spíše záleží, jakým stylem se chce prohledávat. Zde bylo zaimplementováno jednoduché prohledávání v přímkách pod různým úhlem, které jsou přesně rozdělené. Mravenci mají pouze 3 vlastnosti: reagují na prostředí tím, že mají různou rychlost při pohybu, dále se při nalezení okraje „odrazí“ pod stejným úhlem a při nalezení jídla jdou přímo do mraveniště nejkratší cestou.

Nevýhodou tohoto postupu prohledávání je hlavně nejistota, že bude nalezena potrava. Proto u této metody je zapotřebí dostatečné množství mravenců. Možná náhrada by byla náhodným prohledáváním s určitým omezením pohybu, aby se nemohlo stát, že se mravenec nehýbe.

3.3.4 Vizualizace pohybu a feromonů

Tato část obsahuje ve své podstatě pouze několik klíčových funkcí, které jsou součástí Matlabu v jeho knihovnách. Stavebním kamenem je funkce *plot*, která vykresluje mravence. To se děje také s pomocí příkazu *drawnow*, který tak říká, že vše, co je změněno, bude obnoveno, ale ne přikleslo do původního okna.

Další nezbytný využívaný příkaz je *contour*. Ten, jak už vyplývá z anglického překladu, vykresluje do okna vrstevnicovou mapu, která slouží jako vizualizační pomůcka při promítání na prostředí. Tato funkce vykreslí body, které jsou vedle sebe a mají stejnou hodnotu nebo je mezi nimi hodnota současné vrstevnice, kterou vykreslujeme. Vykresleno bude několik čar zobrazujících stejné hodnoty vedle sebe a budou vytvářet spojitě uzavřené smyčky. Rozdíly mezi vrstevnicemi jsou rozlišeny barevně.

Jelikož jsou hloubkové senzory nepřesné a často zde dochází k jisté odchylce, kdy se prostředí nemění, bylo zapotřebí přidat filtraci těchto dat, aby byly hodnoty ustálenější. To bylo provedeno díky funkci *filter2*, která funguje tak, že hodnoty, které jsou přímo snímány, jsou mezi sebou prokládány „jádre“m, což je matice o libovolné čtvercové velikosti s nejvyšší hodnotou uprostřed matice. To dokáže propojit lépe sousední hodnoty mezi sebou. To vytváří při pronásobování snižování rozdílů mezi sousedními hodnotami, protože funkce *filter2* velké skokové hodnoty eliminuje. Tento filtrační systém je ještě vylepšen o průměrování hodnot z filtru, aby se změna vrstevnic jevila plynulejší. Kvůli tomu se vrstevnice nevykreslují při každé iteraci. To poskytuje také rychlejší fungování programu, jelikož vykreslení vrstevnic je velmi náročné na procesor.

Filtrace také probíhá proti vlivům z okolí, tedy kdy snímá ruku, která mění prostředí. To je implementováno jednoduchou dvojitou *for* smyčkou, kdy se kontroluje hodnota každého bodu, zda-li není příliš rozdílný od předchozí hodnoty.

Feromony se mohou zobrazovat také, ale mají zde pouze funkci porozumivací. Jsou zobrazovány formou funkce *scatter*, která zobrazuje body jako malé kroužky. Zde je možné přímo nastavit pozici, velikost a barvu každého bodu. Při inicializaci se zařadí mezi ostatní vykreslovací funkce a na závěr každé iterace proběhne kontrola feromonu,

zda-li jsou zde nenulové hodnoty a za pomoci funkce *find* nalezne pozice a hodnoty všech feromonů, které se pak vykreslí s daným barevným odstínem.

3.4 Konstrukce a provedení modelu

Jedním z cílů bakalářské práce je vytvoření modelu k vizualizaci mravenčího adaptivního algoritmu. Hlavní prvek modelu je v první řadě počítač, ve kterém musí být, jak už bylo zmíněno dříve, OS Windows 10 a Matlab verze 2016a. Další nosnou věcí je senzor Kinect a Projektor, který je připojen k počítači HDMI kabelem.

Další částí konstrukce je krabice 65x70x19 cm, která zde slouží jako prostor, kde je uložen písek, který bude symbolizovat terén. Písek byl vybrán hlavně z toho důvodu, že vytváří pouze málo prudké kopce a je velmi jednoduché jej prohrábnout holou rukou.

Nad touto krabicí je vytvořena jednoduchá konstrukce z hliníkových profilů, které jsou spojeny jednoduchými spojkami přímo určenými pro tento typ profilů. Profily mají na všech bočních stranách drážku, díky které se profily mohou velice jednoduše spojovat. K této konstrukci je přichycen projektor, Kinect a oba míří na krabici. Projektor je s Kinectem synchronizován tak, aby byl zapnut program a stačí, aby se krabice posunula podle promítané plochy na správné místo. Celá konstrukce je zobrazena na Obrázku 3.3:



Obrázek 3.3 Nosná konstrukce s projektorem a Kinectem + krabice s pískem.

Pro lepší pochopení algoritmu je možnost, že se feromony budou promítat do snímaného prostředí. To je popsáno výše. Tento způsob volby je zde kvůli edukačním důvodům, aby se na něm mohlo přesně tento princip demonstrovat.

3.5 Hledání ideálních parametrů

Pro nalezení parametrů pro adaptivní ACO algoritmus bude vytvořena simulace, kdy se hledá na neměnném terénu nejlepší cesta. Všechny hodnoty, které nejsou v následujících podkapitolách, nejsou nijak stanovené. Obzvláště počet iterací není nijak důležitý, protože teoreticky by měl algoritmus neustále pracovat a neustále se přizpůsobovat novým situacím a změnám.

3.5.1 Konstanty α a β

Pro obě tyto konstanty je jasné, že hodnoty by měly být přibližně stejné. Je to způsobeno tím, že by měl být mravenec schopen se držet feromonové stopy a dokázat ji také opustit. Zkoumané hodnoty pro obě hodnoty budou v rozsahu od 1,5 do 2,3. Hodnoty by mohly být i vyšší nebo nižší, ale z důvodů, aby čísla nebyla zbytečně velká, se volí tato konvence čísel.

3.5.2 Množství ukládaného feromonu a jeho vypařování

Obzvláště důležitým parametrem je vypařování feromonu. Ten může velice posilovat a dokáže velmi ovlivňovat schopnost opustit feromonovou stopu. Tato hodnota se bude pohybovat těsně pod hranicí čísla 1, aby vypařování nebylo příliš rychlé. Jelikož se volí parametr vypařování, musíme najít také ideální množství feromonu, kterým jednotlivý mravenec ohodnotí současný krok. Tato hodnota je vzhledem k tomu, že je zde projev rychlosti, také velice zásadní. Hodnota položeného feromonu je jen mírně ovlivněna momentální náročností kroku mravence. Pro simulaci bude zvolen větší rozsah hodnot, protože jsem nevěděl, jak moc ho bude potřeba. Rozsah byl od 0,6 do 3.

3.5.3 Heuristická funkce a počet mravenců

Tyto dva parametry spolu nemají příliš společného, protože ovlivňují oba rozdílně průběh algoritmu. Heuristická funkce (v tomto případě je přesnější termín heuristická hodnota) je druh reakce na množství feromonu v terénu. Kdy je tato hodnota větší než feromonová složka, je šance, že bude tím pádem spíše preferovat hodnotu s feromonem, ale to díky této vlastnosti nemusí platit. Tak se může mravenec rozhodnout nenásledovat slabou feromonovou stopu, ale hledat lepší cestu.

Počet mravenců souvisí obzvláště s množstvím feromonů, které se ukládají do prostoru, protože více mravenců vyprodukuje více feromonů. Počet mravenců je hledán pro několik hodnot od 15 do 100. Stejně jako u všech předchozích nastavení velmi záleží na koordinaci parametrů navzájem.

4 Výsledky

4.1 Zvolení parametrů

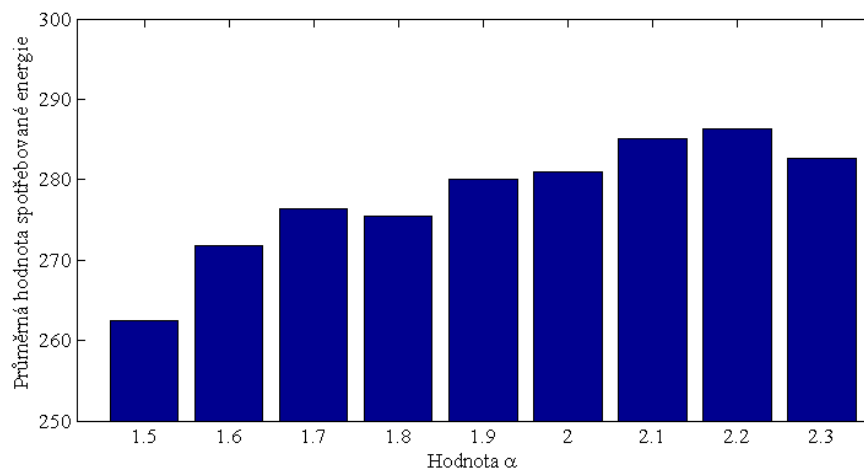
Byla provedena simulace v jednom neměnném prostředí, kde byla předem vybrána ideální cesta, po které se mají mravenci na konci pohybovat. Vzhledem k tomu, že mravenci mají být schopni reagovat na změny prostředí, je tato simulace pouze spíše pro určení správných parametrů pro udržení se na feromonové stopě.

Po zjištění výsledků z této simulace bylo usouzeno, že simulace byla spíše pro určení hodnot parametrů α , množství uloženého feromonu a jeho vypařování. Zbylé hodnoty více souvisí s heuristickou vlastností mravenců, a proto v neměnném prostředí se téměř nedají nastavit. Tato simulace proto nebyla zcela úspěšná, protože polovina hledaných parametrů nebyla vhodná pro adaptivní ACO. Kvůli tomu se přistoupilo po této simulaci k experimentálnímu řešení zbylých parametrů.

4.1.1 Hodnoty ze simulace

Simulace prozkoumávala vhodnou kombinaci hodnot na základě vynaložené energie mravence, což zahrnovalo nejen vzdálenost, kterou urazil, ale i obtížnost cesty. Z tohoto parametru bylo tedy vybíráno nejvíce. Další, ale již pouze kontrolní kritérium pro zvolení této cesty, jsou samotné kroky mravenců, díky nimž je možné si jejich pohyb opětovaně prohlédnout.

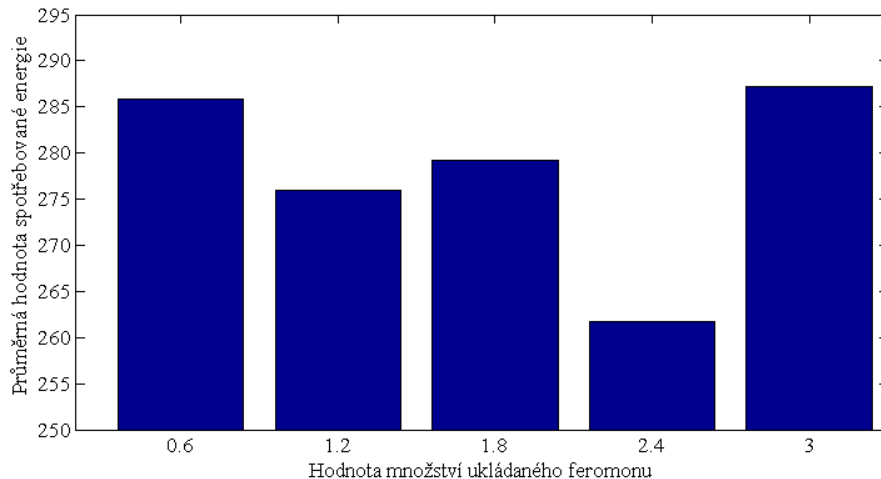
Pro hodnotu α byl zvolen rozsah pro simulaci 1,5 až 2,3. Výsledky simulace jsou zobrazeny v Grafu 4.1. Hodnoty byly porovnány vzhledem k průměrné spotřebované energii jednoho mravence.



Graf 4.1 Výsledky simulace při hledání ideálního parametru α .

Ideální hodnota parametru α podle simulace vyšla 1,5. To nám může říci, že algoritmus má velice silnou heuristickou funkci nebo hodnotu β . Nastavení parametru tedy bude na hodnotě 1,5.

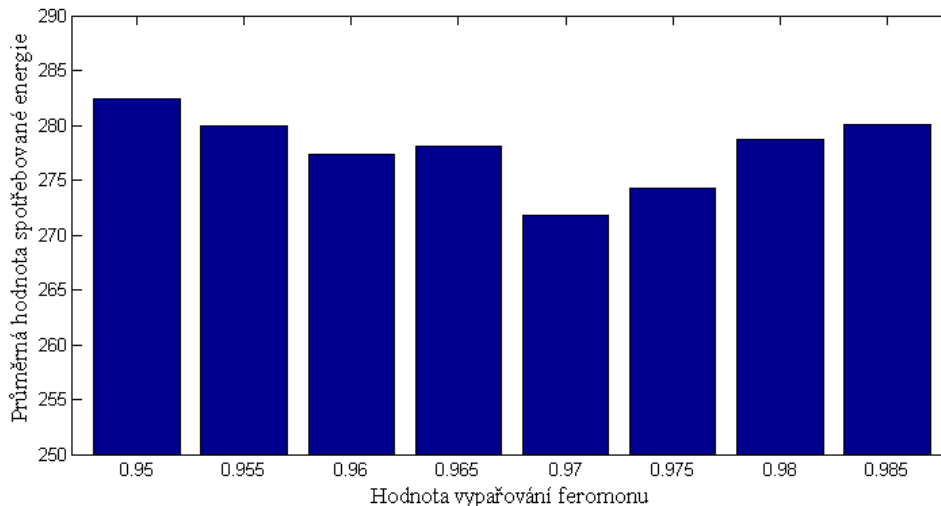
Při hledání množství ukládaného feromonu je dobré mít alespoň představu o tom, kolik bude mravenců v prostoru a jak moc rychlé bude vypařování tohoto feromonu. Pro simulaci byly vybrány hodnoty v rozsahu od 0,6 do 3. Výsledky simulace jsou v Grafu 4.2:



Graf 4.2 Výsledky simulace při hledání parametru množství feromonu.

Z tohoto grafu je velice zřejmý nejideálnější výsledek simulace. Hodnota ukládání feromonu do prostoru mravencem tedy bude 2,4.

Posledním úspěšně simulovaným parametrem je vypařování feromonu. Hodnota, jak už zde bylo poznamenáno, musí být o málo nižší než jedna. Zvolené simulované parametry budou v rozmezí od 0,950 do 0,985. Výsledky jsou i nyní zobrazeny v závislosti na Průměrné spotřebované energii jednoho mravence v Grafu 4.3:



Graf 4.3 Výsledky simulace pro parametr rychlosti vypařování feromonů.

Z těchto výsledků jsme stanovili rychlost vypařování na hodnotu 0,97. Ostatní hodnoty buďto vykazovaly příliš rychlé vypařování feromonů, nebo naopak příliš ovlivňovaly ostatní mravence a to způsobilo jejich setrvání na jedné neideální cestě.

Tyto všechny hodnoty nemusí být naprosto přesné, protože pro úplnou přesnost je třeba provést velice širokou simulaci a provést ji opakovaně pro získání lepších výsledků. Na provedení takto rozsáhlých simulací nebyl čas, neboť taková simulace by zabrala i na rychlých procesorech několik týdnů.

4.1.2 Experimentálně určené hodnoty

Zbylé hodnoty, které jsme se snažili simulací alespoň odhadnout, se nepodařilo určit. Proto byly zvoleny některé hodnoty, které byly získány a jevíly se jako lepší. Ty byly zkoumány, jak pracují. Nešlo však hodnoty zkoumat jednotlivě, proto se všechny níže popisované experimenty prováděly kombinovaně, tedy pro každou kombinaci alespoň jednou.

Počet mravenců musel být stanoven mezi hodnotami 30 až 60, protože při velkém množství se příliš nevypařovaly feromony a tím, že byl velký počet mravenců, je stále někdo obnovoval na neideální cestě. Po několika pokusech se hodnota stanovila na 40 mravenců, pro dobré vlastnosti obzvláště opustit feromonovou stopu.

Heuristická funkce je základním kamenem, na kterém stojí vlastnost mravenců opustit feromonový vliv. Dlouho byl problém určit nefunkčnost samotného algoritmu, proto bylo spuštěno více simulací. Nakonec se ukázalo, že hodnota heuristické funkce musí alespoň přibližně korespondovat s hodnotami množství feromonu, kterým mravenec hodnotí bod v prostoru a s hodnotou parametru α . Zde se ukázalo, že heuristická funkce, v tomto případě číslo, musí mít alespoň hodnotu 10. Hodnota byla po několika zkušebních provezech zvolena na hodnotu 15.

Pro poslední parametr β se po vyzkoušení všech kombinací nakonec zachovala nejlepší hodnota při simulaci. To také záviselo na hodnotě heuristické funkce, která nejlépe za těchto parametrů pracovala. Hodnota parametru β byla tedy zvolena na 2,1.

5 Závěr

V rešerší části byly srovnány některé optimalizační metody, které jsou velmi často pro své výhody používány. Při porovnání jejich vlastností a výhod bylo nesporně jasné, že každá z těchto tří metod má naprosto odlišné využití. To je pozorovatelné i v přírodě, ze které jsou tyto optimalizace inspirovány. Každý způsob má jiné předpoklady, jiné cíle a jiné problémy a jejich řešení. Co se týče nejpoužívanější z těchto metod, kterou je beze sporu EA, tak má obrovsky široké pásmo využití díky několika možným obměnám, přídatným parametrům a specializacím.

Vizualizační stránka těchto parametrů a jejich stylu optimalizace se nejlépe ukazuje na testovacích funkcích, které nejsou proměnné a mají jasné hodnoty. Je tedy možné porovnávat tyto metody.

V praktické části se pracovalo na naprogramování adaptivního mravenčího algoritmu, který má být schopen reagovat na změnu prostředí a zvládat tak upuštění od současné ideální cesty. Pro dosažení bylo potřeba určit co nejlépe optimalizační parametry.

Po provedení simulace, která proběhla, byly získány jen částečné výsledky, protože algoritmus po dosazení do programu nepracoval správně. Proto byly některé hodnoty zvoleny intuitivně na základě experimentálních pokusů.

Po provedení všech potřebných kroků byl adaptivní mravenčí algoritmus zprovozněn. Funkčnost algoritmu je však na malé úrovni, protože parametry nejsou ideálně nastaveny a algoritmus má nedostatky, které je třeba odstranit. Ty jsou zde však záměrně kvůli rychlejšímu zpracování dat a kvůli lepším prohledávacím vlastnostem mravence při objevování lepších cest.

Jinak však algoritmus funguje optimálně. Byl vytvořen reálný model, na kterém lze dobře pozorovat vlastnosti ACO a jeho chování při změnách terénu. Model byl vytvořen tak, aby bylo jednoduché jej nastavit pro správné spuštění.

Pro využití k edukativním účelům je model přizpůsoben zvláště tím, že je možné zobrazovat spolu s pohybujícími se mravenci i jejich ukládané feromony v prostředí. To slouží k lepšímu pochopení výběru cesty pro pozorovatele. Pro použití ve výuce je model v současné době téměř připraven. Pro plné nasazení by bylo třeba model více zdokonalit v rámci konstrukce a také, jak už bylo zmíněno, nalézt lepší parametry.

Tuto práci lze rozšířit obzvláště v oblasti získávání ideálních parametrů, kde by se hlavně odkazovalo na systém jejich nalezení. Dále je možnost rozšíření v oblasti porovnávání optimalizačních metod, kdy se samotné metody mohou srovnávat v jedné z testovacích funkcích nebo třeba i v modelu pro ACO, tedy je i možnost vhodným způsobem přizpůsobit model pro jiné optimalizace.

6 Seznam použitých zdrojů

- [1] ZELINKA, Ivan. *Evoluční výpočetní techniky: principy a aplikace*. 1. vyd. Praha: BEN - technická literatura, 2009. ISBN 978-80-7300-218-3.
- [2] KOMÍNEK, Jan. *Heuristické algoritmy pro optimalizaci*. Brno, 2012. Diplomová práce. Vysoké Učení Technické v Brně, Fakulta strojního inženýrství, Ústav Matematiky.
- [3] TUREK, Luboš. *Použití ACO algoritmu na řešení jednoduché substituční šifry*. Praha, 2012. Bakalářská práce. Univerzita Karlova v Praze, Matematicko-Fyzikální Fakulta.
- [4] POPEK, Miloš. *Řešení optimalizačních úloh inspirované živými organismy*. Brno, 2010. Diplomová práce. Vysoké Učení Technické v Brně, Fakulta Informačních Technologií, Ústav Inteligentních Systémů.
- [5] Artificial Bee Colony Algorithm-based Parameter Estimation of *Fractional-order Chaotic System with Time Delay*. IEEE/CAA JOURNAL OF AUTOMATICA SINICA. 2017, 4(1), 107-113.
- [6] Artificial Bee Colony (ABC) Algorithm. The Faculty of Engineering of *Erciyes University [online]*. Erciyes, Turkey: The Faculty of Engineering of Erciyes University, 2010 [cit. 2017-05-22]. Dostupné z: <http://mf.erciyes.edu.tr/abc/>
- [7] KNOFLÍČEK, Jakub. *Analýza různých přístupů k řešení optimalizačních úloh*. Brno, 2013. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií, Ústav informačních systémů.
- [8] Artificial bee colony *algorithm for dynamic deployment of wireless sensor networks*. Turk J Elec Eng & Comp Sci [online]. 2012, 20(2), 8 [cit. 2017-05-22]. Dostupné z: <http://journals.tubitak.gov.tr/elektrik/issues/elk-12-20-2/elk-20-2-6-1101-1030.pdf>
- [9] SIM, *Kwang a Weng SUN*. *Multiple ant-colony optimization for network routing [online]*. Tokyo, Japonsko, 2002 [cit. 2017-05-22]. Dostupné z: <http://ieeexplore.ieee.org/document/1180890/>
- [10] *How To Make 3D Holograms With Your Smartphone*. In: www.boredpanda.com [online]. Vilnius, Lithuania: Bored Panda, 2016 [cit. 2017-05-23]. Dostupné z: <http://www.boredpanda.com/3d-hologram-smartphone-diy-device-mrwhosetheboss/>
- [11] MIŠKAŘÍK, *Kamil*. *Včelí algoritmus*. Brno, 2010. Bakalářská práce. Vysoké Učení Technické v Brně, Fakulta strojního inženýrství, Ústav automatizace a informatiky.
- [12] Kinect Hardware. *Windows Dev Center [online]*. Redmond, Washington, USA: Microsoft, 2017 [cit. 2017-05-22]. Dostupné z: <https://developer.microsoft.com/en-us/windows/kinect/hardware>

- [13] Matlab. *Https://www.mathworks.com/* [online]. Natick, Massachusetts, USA: The MathWorks, Inc., 2017 [cit. 2017-05-22]. Dostupné z: <https://www.mathworks.com/products/matlab.html>

7 Seznam použitých zkratek a symbolů

7.1 Zkratky

| | |
|------|--------------------------------------|
| ACO | Optimalizace Mravenčí Kolonií |
| ABC | Umělé Včelstvo |
| EA | Evoluční Algoritmus |
| HDMI | High-Definition Multimedia Interface |
| OS | Operační systém |

7.2 Symboly

| | |
|----------|---|
| α | Mocnitel feromonové části pravděpodobnostní funkce ACO |
| β | Mocnitel heuristické části pravděpodobnostní funkce ACO |
| f | Feromony |
| h | Heuristická funkce |
| p | Pravděpodobnostní funkce |

7.3 Funkce v prostředí Matlab

| | |
|--------------------------|--|
| Image Aquisition Toolbox | Zpracovávání dat ze zařízení |
| Plot | Zobrazení grafu vykreslením dvojic bodů |
| Drawnow | Vykreslení změn v již zobrazeném okně |
| Contour | Vykreslení vrstevnic |
| Filter2 | Filtrace dat prokládáním jiné matice |
| Scatter | Vykreslení grafu s pomocí bodů (kružnic) |
| Find | Pro nalezení nenulových hodnot v matici |
| For | smyčka, která má stanovený počet iterací, které v ní proběhnou |

8 Seznam obrázků a grafů

8.1 Seznam obrázků

| | |
|--|--------|
| Obrázek 2.1 Double bridge experiment ve čtyřech krocích [4]..... | - 14 - |
| Obrázek 2.2 Možné jednoduché zobrazení ABC – Hologram [10]..... | - 17 - |
| Obrázek 2.3 Rastriginova funkce [11]..... | - 18 - |
| Obrázek 3.1 Zařízení Kinect se zapnutými hloubkovými senzory..... | - 19 - |
| Obrázek 3.2 Část pseudokódu ACO – hlavní <i>while</i> smyčka..... | - 21 - |
| Obrázek 3.3 Nosná konstrukce s projektorem a Kinectem + krabice s pískem . | - 23 - |

8.2 Seznam Grafů

| | |
|---|--------|
| Graf 4.1 Výsledky simulace při hledání ideálního parametru α | - 25 - |
| Graf 4.2 Výsledky simulace při hledání parametru množství feromonu | - 26 - |
| Graf 4.3 Výsledky simulace pro parametr rychlosti vypařování feromonů | - 26 - |

9 Seznam Příloh

Příloha 1... Konstrukce celého modelu – Pohled Shora

Příloha 2... Vizualizace prostředí a mravenců na písku

Příloha 3... Adaptér Kinect pro Windows

Příloha 4... Vizualizace adaptivního ACO s feromony

9.1 Přílohy na CD

Příloha 5... Model s barevným vykreslením terénu (video formát .mp4)

Příloha 6... Model s vrstevnicemi a feromony (video formát .mp4)

Příloha 7... Program ACO (Matlab formát .m)

Příloha 8... Program ACO se zobrazováním vrstevnic (.m)

Příloha 9... Program ACO se zobrazováním feromonů (.m)

Příloha 10.. Program ACO s vrstevnicemi a feromony (.m)

Příloha 11.. Program ACO s barevným vykreslením terénu a feromony (.m)

Příloha 12.. Funkce angle_fi (.m)

Příloha 13.. Funkce odraz (.m)

Příloha 14.. Funkce Initial plotting (.m)

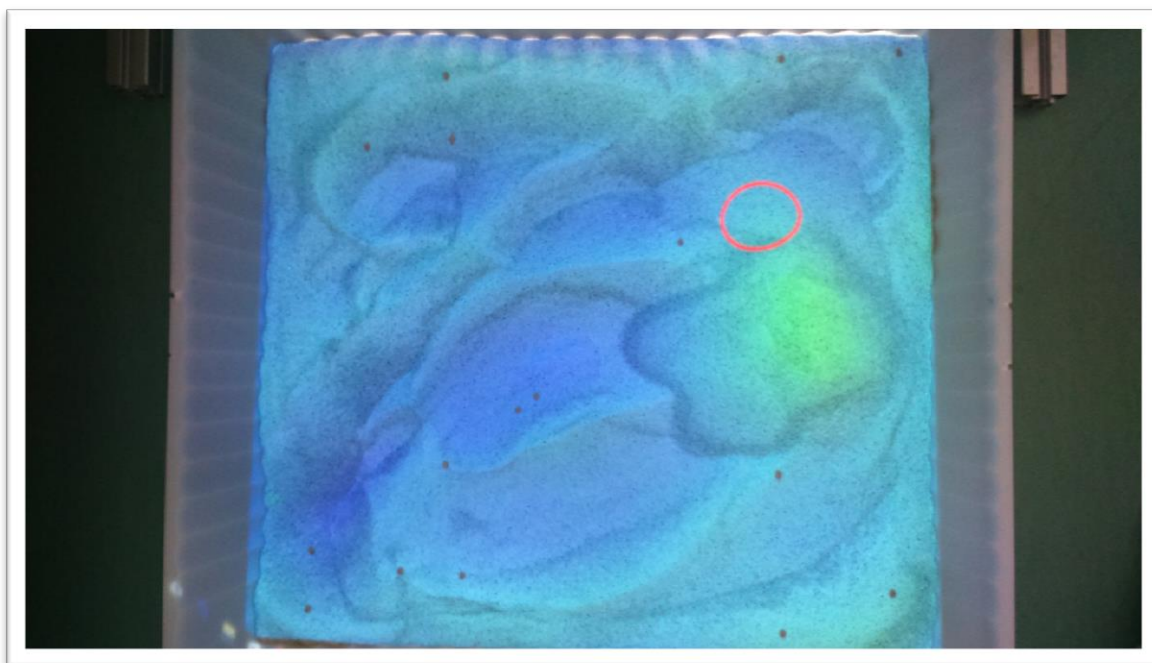
Příloha 1

Konstrukce celého modelu – pohled shora



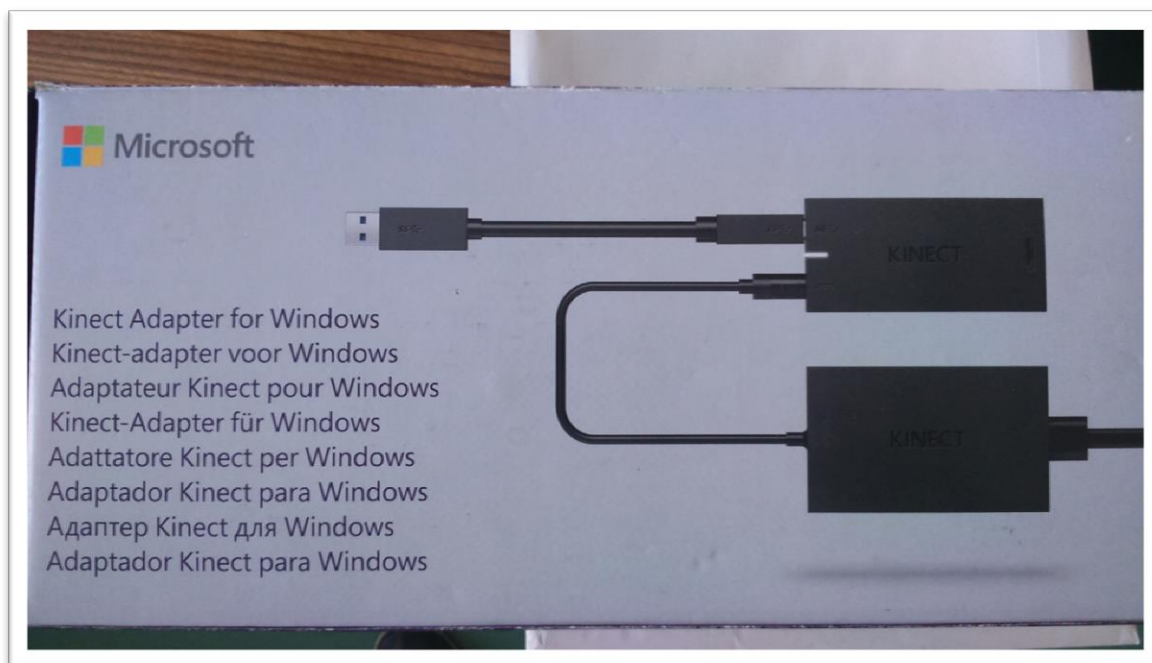
Příloha 1

Vizualizace prostředí a mravenců na písku



Příloha 2

Adaptér Kinect pro Windows



Příloha 3

Vizualizace adaptivního ACO s feromony

