

**AKADÉMIA OZBROJENÝCH SÍL
GENERÁLA MILANA RASTISLAVA ŠTEFÁNKA**

**OPTIMALIZÁCIA KYBERNETICKÝCH PROCESOV
POMOCOU EXPERTNÝCH SYSTÉMOV**

Študentská vedecká odborná práca

Súťažiaci: Ľuboslav IVANKO

Konzultant: kpt. Ing. M. TURČANÍK, PhD.

Liptovský Mikuláš

2005

OBSAH

ÚVOD.....	3
1 KYBERNETIKA A ROBOTIKA	4
2 ZNALOSTNÉ A EXPERTNÉ SYSTÉMY	4
3 KYBERNETICKÝ SYSTÉM SERGEJ.....	6
4 PROGRAMOVÉ RIEŠENIE EXPERTNÉHO SYSTÉMU.....	8
4.1 Použité znalosti	10
4.2 Testovanie vlastností jednotlivých metód	17
ZÁVER	20
BIBLIOGRAFIA A BIBLIOGRAFICKÉ ODKAZY	21

Úvod

Prostredie, v ktorom pôsobia organizácie, sa stáva čoraz zložitejším a komplexnejším. K efektívnemu rozhodovaniu sú potrebné čoraz lepšie znalosti, a to nielen „učebnicové“ teoretické, ale najmä praktické, získané skúsenosťou a dlhodobou praxou. Práve táto skupina znalostí je vlastná ľuďom, ktorých nazývame experti. Je žiadúce transformovať znalosti expertov do perzistentnej, znovupoužiteľnej, uchovávateľnej a odovzdávateľnej formy. Jednou možnosťou takejto reprezentácie vedomostí sú expertné systémy.

Expertné systémy predstavujú nový spôsob využívania výpočtovej techniky. Vznikli ako dôsledok uplatňovania poznatkov a skúseností nadobudnutých v oblasti výskumov umelej inteligencie a vedy o programovaní.

Rozpracovaním a realizáciou Turingových teoretických vývodov o možnosti používať počítače na modelovanie tých procesov, ktoré vytvárajú jednu z najvýznamnejších prejavov človeka, *myslenie*, vznikla vedná disciplína nazvaná *umelá inteligencia*. Vďaka nej je možné charakterizovať súčasnú etapu využívania počítačov ako etapu prechodu od spracovania údajov k spracovaniu poznatkov.

1 Kybernetika a robotika

Kybernetika sa zaoberá kvantitatívnymi a štrukturálnymi zákonitosťami riadenia, oznamovania a kontroly samoregulujúcich sa sústav. Pod sústavou je potrebné si predstaviť takú, ktorá je schopná reagovať na podnety z okolia i zvnútra sústavy samotnej. Zvláštnym prípadom takejto sústavy môže byť aj živý organizmus.

Kybernetika je teda disciplína, ktorá skúma riadiace a regulačné procesy v biologickej sfére, v technike a v spoločnosti a navrhuje modely na znázornenie, transformáciu a spracovanie informácií. Všetky automatické zariadenia na spracovanie dát sú v tomto zmysle kybernetickými strojmi a samotná informatika je náuka o kybernetických strojoch a metódach. [Kyb05]

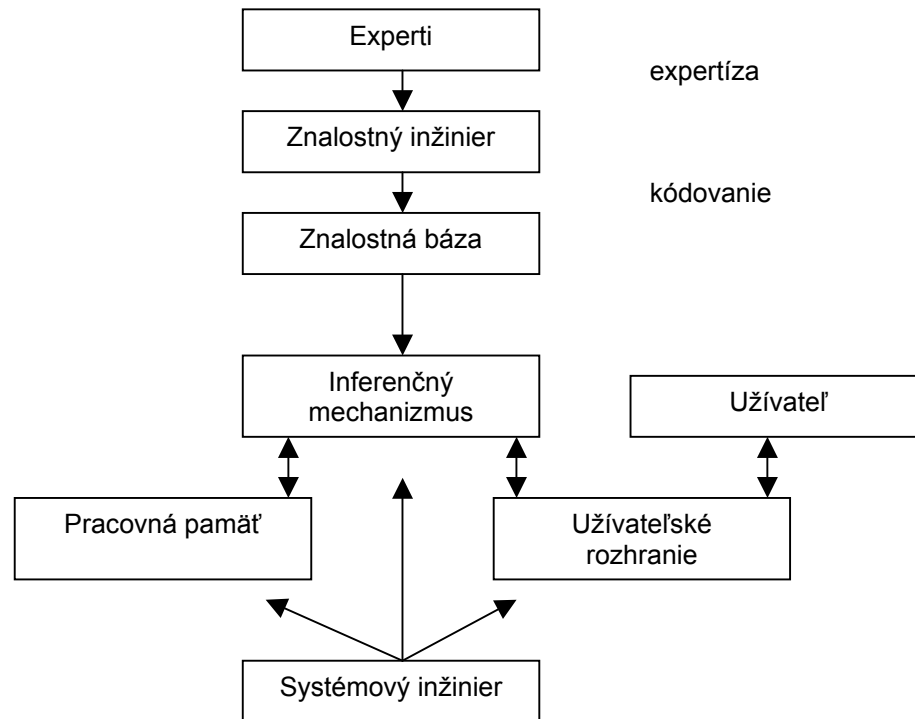
Robotika je vedné odvetvie, ktoré sa zaoberá robotmi. Definícia robota však už nie je tak jednoduchá. Najjednoduchšia definícia pravdepodobne znie: „Robot je stroj, ktorý je možné preprogramovať.“ Práve fakt, že robot je preprogramovateľný je veľmi dôležitý. Ďalšia dôležitá vlastnosť robota je to, že by mal byť schopný robiť skoro všetko to čo človek.

Robotika sa zaoberá mechanickým a elektrickým inžinierstvom, teóriou riadenia, programovaním a v súčasnosti aj umelou inteligenciou. Najväčší dôraz však kladie na matematické modelovanie pohybu fyzikálnych telies. [Sel92]

2 Znalostné a expertné systémy

Výskum v oblasti umelej sa zameriava na skúmanie prostriedkov počítačovej (symbolovej) reprezentácie poznatkov a znalostí ako aj metód ich používania. Vedie to k rozvoju programových prostriedkov, ktorými je možné obohacovať spôsobilosť počítača riešiť problémy postupmi založenými na využívaní znalostí. Expertné systémy sú jedným z konkrétnych prejavov realizácie takýchto spôsobilostí na súčasných počítačoch.

- Expertný systém je počítačový systém hľadajúci riešenie problému v rozsahu určitého súboru tvrdení alebo istého zoskupenia znalostí, ktoré boli formulované expertmi pre danú špecifickú aplikačnú oblasť.
- Expertný systém je systém založený na reprezentácii poznatkov expertov, ktoré využíva pri riešení zadaných problémov.
- Expertný systém je systém kooperujúcich programov na riešenie vymedzenej triedy úloh, v jednotlivých problémových oblastiach zvyčajne riešených expertmi.



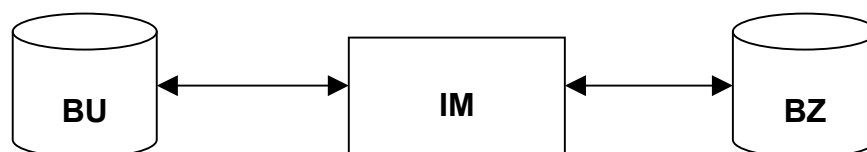
Obr. 1 Štruktúra expertného systému

Štruktúru expertného systému, jeho vznik a prevádzku znázorňuje obrázok 1. *Experti* z danej oblasti dávajú expertízou dohromady súbor znalostí. Znalostný inžinier tieto znalosti dá do formy zrozumiteľnej programu. Inferenčný mechanizmus bude potom vyberať vhodné znalosti a bude radiť užívateľom.

Expertný systém je realizovaný sústavou kooperujúcich programov, ktorých činnosť sa opera o špecifické údajové štruktúry.

Vzájomné väzby medzi modulmi a ich programovými celkami vytvárajú *architektúru expertného systému*. V každom expertnom systéme je možné rozlíšiť tri základné zložky tvoriace jeho *minimálnu konfiguráciu*. Sú to:

- inferenčný (odvodzovací, riešiaci) mechanizmus (IM)
- báza údajov resp. báza faktov (BU)
- báza znalostí (BZ)



Obr. 2 Základné zložky expertného systému

Inferenčný mechanizmus je tvorený systémom kooperujúcich programov zabezpečujúcich procedurálnu zložku činnosti expertného systému. Báza znalostí a báza údajov sú pasívne údajové štruktúry. Toto členenie odráža kľúčovú myšlienku deklaratívneho (situačného) programovania, tvoriaceho jeden zo základných princípov expertných systémov. Báza znalostí a báza údajov nemusia byť vzájomne oddelené a v niektorých systémoch sú tieto pasívne systémy realizované jedinou údajovou štruktúrou.

Báza znalostí však predstavuje štruktúru údajov reprezentujúcich všeobecne platné a prijímané poznatky o pravidlách a zákonitostiach, kým báza znalostí je nositeľkou konkrétne zadaných či odvodených faktov, alebo predpokladaných údajov o špecifickom probléme, ktorý je predmetom riešenia. V praxi sa mnohokrát uchováva báza znalostí a báza údajov fyzicky rôzne. Kým báza údajov je zvyčajne v operačnej pamäti, báza znalostí sa často nachádza na diskovom pamäťovom médiu. [Pop89]

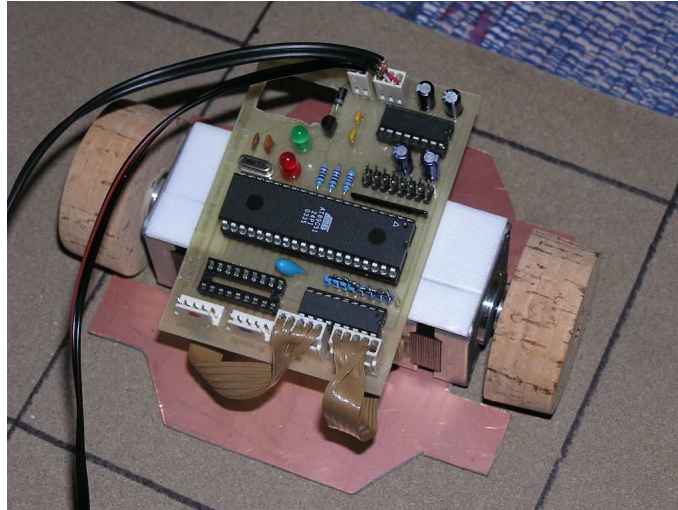
3 Kybernetický systém sergej

Úloha, na ktorej je demonštrovaná činnosť a vlastnosti expertného systému je veľmi triviálna. Jedná sa o hľadanie cesty v bludisku, ktoré pozostáva zo štvorcov. Hľadanie prebieha v nezmapovanom bludisku a cestu (pokiaľ možno optimálnu) hľadá jednoduchý robot. Robot je jednoduché dvojkolesové vozidlo, so sústavou senzorov, prostredníctvom ktorých robot získava informácie z prostredia. Keďže sa jednalo iba o veľmi jednoduchú demonštráciu, pohyb robota je obmedzený na krok vpred, krok vzad, otočenie doľava a doprava. Robot sa teda v pomyselnom bludisku pohybuje iba v rozsahu štvorcov.

Návrh robota vychádza z predstavy, že v pamäti robota budú uložené len základné funkcie pre pohyb v jednotlivých smeroch a návrat hodnôt snímačov nadradenému systému. Nadradeným systémom, ktorý vykonáva "myslenie" je osobný počítač. Toto rozdelenie úloh je tu najmä kvôli komfortu, ktorý poskytuje programovanie vo vyššom programovacím jazyku na osobnom počítači, a taktiež preto aby bolo možné ľahko demonštrovať proces "myslenia" prostredníctvom obrazovky osobného počítača.

Na osobnom počítači je vytvorený program vo vyššom programovacom jazyku, v ktorom je implementovaný jednoduchý expertný systém. Program je vytvorený v programovacom jazyku C++ v prostredí MS Windows.

Komunikácia medzi robotom a počítačom prebieha po sériovom kanáli cez rozhranie RS232.



Obr. 3 Robot Sergej

Pre svoj pohyb využíva robot dvojicu krokových motorov. Sú to motory typu KP4M4 s krokom 3.6 stupňa a impedanciou 150 ohmov na vinutie. Stredové konce jednotlivých vinutí sú spojené a vyvedené ako samostatný vývod. Tento typ motorov sa používal v starších 5,25 palcových disketových mechanikách a v robotovi je použitý najmä kvôli nízkej nadobúdacej cene. Budenie krokových motorov je zabezpečené budičmi ULN2803 firmy SGS-Thomson Microelectronic. Jedná sa o integrovaný obvod skladajúci sa z ôsmich Darlingtonových polí.

Ako senzory sú použité mechanické mikrosplínače, opäť kvôli čo najväčšej jednoduchosti. Senzory sú umiestnené na prednej aj zadnej časti robota, teda robot spoľahlivo určí prekážku pri pohybe vpred aj vzad. Senzory sú umiestnené na troch miestach vpredu a troch miestach vzadu takže robot dokáže detekovať aj prekážku, ktorá neleží priamo v imaginárnom štvorci po ktorých sa robot pohybuje.

Vnútnú logiku robota zabezpečuje mikroprocesor rady 8051 firmy ATMEL typu 89C51, v ktorom je napevno napálený ovládací program, vďaka ktorému robot dokáže robiť elementárne pohyby a hlásiť stav svojich senzorov. Celý systém ešte obsahuje integrovaný obvod MAX232 firmy MAXIM, čo je prevodník logických úrovní z rozhrania RS – 232C (V24) na TTL logiku a naopak.

Ako už bolo spomenuté v úvode, robot nie je autonómna jednotka, a pre svoju *rozumnú* činnosť musí komunikovať s nadradeným počítačom. Oddelenie robota a počítača má význam v tom, že je možné kedykoľvek nahradiť robota novším modelom, pričom nie je nutné meniť ovládací program. Rovnako je možné jednoducho doplniť v programe nové vlastnosti (napr. modifikovať bázu znalostí) bez nutnosti robiť zásahy do koštrukcie robota.

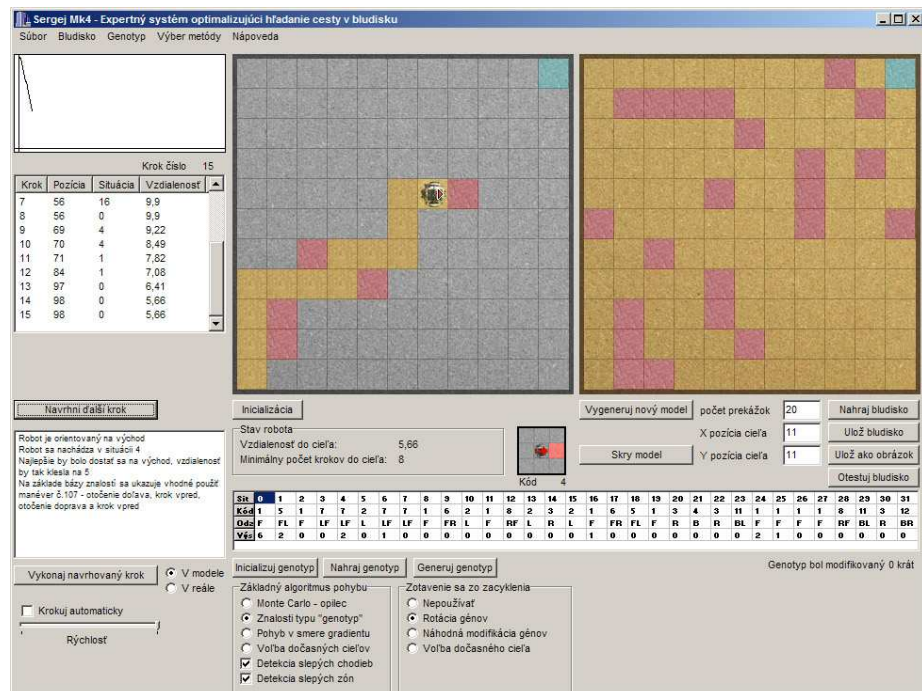
Množstvo údajov, ktoré posielajú počítač robotovi a naopak robot počítaču je veľmi malé a tieto údaje prichádzajú v nepravidelných časových úsekoch – podľa potreby. Komunikácia

je teda realizovaná nie po bitoch, ale po znakoch. Každý pokyn robotovi znamená jeden prenesený znak po sériovom porte, a každé hlásenie robota o úspešnom či neúspešnom vykonaní požadovanej akcie, či zopnutí niektorého zo senzorov je oznamované nadradenému počítaču taktiež jedným znakom. Jednou z výhod tejto komunikácie po znakoch je to, že otvára možnosti použitia robota na rôzne účely, bez nutnosti hlbšie skúmať komunikačné protokoly. Robota je takto dokonca možné jednoducho ovládať aj cez obyčajný hyperterminál.

Nadradený počítač a robot medzi sebou komunikujú prenosovou rýchlosťou 2400 b/s, 8 údajových bitov, 1 stop bit, bez parity a riadenia toku. Prenosová rýchlosť je na dnešné komunikačné zariadenia relatívne nízka, avšak pre požiadavky systému plne postačuje a s použitým typom procesora na strane robota je táto rýchlosť veľmi ľahko technicky realizovateľná.

4 Programové riešenie expertného systému

Program je vytvorený vo vývojovom prostredí C++ Builder pod MS Windows. Obsahuje v sebe nielen možnosť priamo optimalizovať pohyb robota v bludisku, ale aj simulačné prostriedky, nutné na to aby bolo možné pozorovať činnosť jednotlivých algoritmov, a sledovať účinnosť jednotlivých metód – druhov znalostí.



Obr. 4 Užívateľské rozhranie expertného systému v prostredí MS Windows

Štruktúra programu vychádza z architektúry expertných systémov. V programe je oddelená báza znalostí, báza údajov a výber vhodných znalostí a vytváranie znalostí, ktoré nie sú implicitne obsiahnuté v báze znalostí má na starosť inferenčný mechanizmus.

Pre lepšiu demonštráciu som sa rozhodol v programe implementovať viacero typov znalostí. Tieto predstavujú vlastne algoritmy hľadania cesty k cieľu. Program využíva priamy chod inferenčného mechanizmu, a teda hľadá riešenie v rozsahu známych faktov. Akonáhle robot (príp. simulácia) zistí nový fakt o probléme, poznačí si tento fakt do bázy údajov a hľadá nové riešenie v rámci nových faktov.

Program teda pracuje v dvoch fázach. V prvej fáze navrhne na základe doterajších údajov a príslušných znalostí ďalší krok a druhej fáze navrhovaný krok vykoná. Ak sa tieto dve fázy budú opakovať dostatočne dlho program nájde riešenie.

Báza údajov je reprezentovaná statickým dvojrozmerným poľom, ktorého prvkami sú celočíselné hodnoty. Pole má rozmer 13x13, čo vyplýva z rozmeru reálneho fyzického bludiska, ktoré bolo použité. Hranice poľa sú pri inicializácii označené ako prekážka, čo taktiež zodpovedá reálnemu fyzickému bludisku, ktoré má po obvode hranicu.

2	2	2	2	2	2	2	2	2	2	2	2	2
2	0	0	0	0	0	0	0	0	0	0	0	2
2	0	0	0	0	0	0	0	0	0	0	0	2
2	0	0	0	0	0	0	0	0	0	0	0	2
2	0	0	0	0	0	0	0	0	0	0	0	2
2	0	0	0	0	0	0	0	0	0	0	0	2
2	0	0	0	0	0	0	0	0	0	0	0	2
2	0	0	0	0	0	0	0	0	0	0	0	2
2	0	0	0	0	0	0	0	0	0	0	0	2
2	2	2	0	0	0	0	0	0	0	0	0	2
2	1	1	0	0	0	0	0	0	0	0	0	2
2	1	1	1	0	0	0	0	0	0	0	0	2
2	2	2	2	2	2	2	2	2	2	2	2	2

Obr. 5 Vnútoraná reprezentácia bázy údajov pomocou statického dvojrozmerného poľa

Hodnoty prvkov poľa môžu nadobúdať tieto hodnoty:

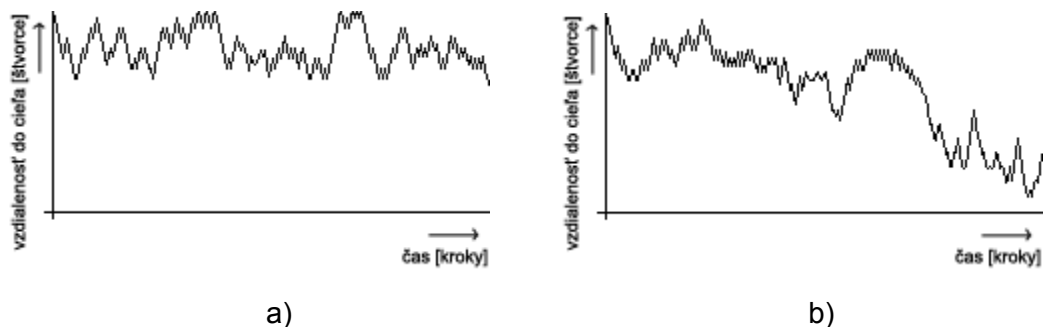
- 0 – políčko je nezmapované, systém nevie či je políčko voľné, alebo je na ňom prekážka
- 1 – políčko je zmapované a je voľné
- 2 – políčko je zmapované a je na ňom prekážka
- 12 – políčko je voľné, alebo obsadené, avšak systém usúdil, že cesta do cieľa nevedie cez toto políčko

Ako znalosti program využíva nasledujúce metódy a ich kombinácie:

- náhodný pohyb
- znalosti typu genotyp
- pohyb v smere gradientu
- voľbu dočasných cieľov
- detekciu a zotavenie sa z uviaznutia (zacyklenia)
- detekciu „slepých chodieb“ a „slepých zón“

4.1 Použité znalosti

Algoritmus náhodného pohybu vychádza z metód pre stochastické modelovanie metódou Monte Carlo, i keď samotná metóda Monte Carlo rieši úplne iný problém. V každom okamihu sa robot pohne náhodne zvoleným smerom, pričom šanca pre každý jednotlivý smer je presne 25%. Pri tomto algoritme nemožno hovoriť o nejakých znalostiach, pretože sa jedná o čisto stochastický pohyb. Aby algoritmus predsa len vykazoval nejakú inteligenciu, je určená obmedzujúca podmienka, aby sa robot nikdy nepresunul na políčko, na ktorom je prekážka. Je zrejmé, že metóda je stále príliš slabá a nemôže nájsť riešenie v prijateľnom čase.



Obr. 6 Grafy charakterizujúce náhodný pohyb

Grafy na obrázku 6 charakterizujú pohyb robota v bludisku pri použití metódy náhodného pohybu vychádzajúcej z metódy Monte Carlo. Ako testovacie bludisko bol použitý súbor TestLab2.lab. Obrázok a) charakterizuje pohyb v bludisku bez detekcie slepých zón a chodieb, obrázok b) charakterizuje pohyb s detekciou slepých zón a chodieb.

Ako ukazujú grafy, algoritmus nedokázal nájsť cestu bludiskom v prijateľnom čase. Z grafu však možno pozorovať vplyv detekcie slepých zón a chodieb, vďaka čomu sa dostal robot k cieľu rýchlejšie. Je nutné podotknúť, že keďže algoritmus je stochastický výsledky nie

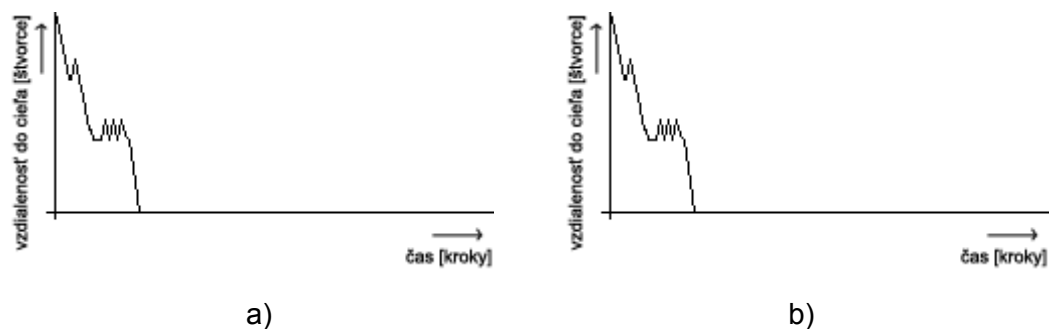
sú reprodukovateľné a v každom pokuse dosiahne robot iné výsledky. Ďalšia dôležitá vlastnosť, ktorá platí pri všetkých stochastických metódach, je že algoritmus nikdy neuviazne v lokálnom extréme. Keďže algoritmus nie je deterministický, nemôže pri ňom dôjsť k zacykleniu. Táto metóda teda určite nájde riešenie, aj keď možno v nekonečnom čase (po nekonečnom počte krokov).

Preblematiku použitia evolučných algoritmov na optimalizáciu pohybu robota v bludisku riešil vo svojej práci Böhmer. [Boh05] Genotyp použitý v spomínanej práci sa skladá z 32 génov, ktoré popisujú všetky elementárne situácie v ktorých sa môže robot nachádzať. Výsledky Böhmerovej práce sú aplikované aj v tomto expertnom systéme ako jedna z báz znalostí. Použitý je genotyp, ktorý podľa Böhmerových testov vykazoval najlepšie vlastnosti.

Pri reprezentácii znalostí genotypom boli niektoré gény genotypu ešte vhodne modifikované, aby viac zodpovedali znalostiam experta. Aby dokázal algoritmus vyriešiť ešte väčšiu skupinu bludísk, je nutné použiť aj detekciu zacyklenia (v prípade uviaznutia v lokálnom extréme) a metódy, ktoré dostanú systém zo zacyklenia.

Pri genotype je možnosť manipulovať len s jednotlivými génmi, takže zotavenie prebieha modifikáciou génov, a to buď rotáciou alebo náhodne. Pri rotácii sa gén ktorý zapríčinil zacyklenie cyklicky nahradí iným povoleným génom. Pri náhodnej modifikácii je gén, ktorý zapríčinil zacyklenie nahradený úplne náhodne (teda nový gén môže byť aj zakázaný).

Metóda vychádza z predpokladu, že k modifikácii génov nebude dochádzať až tak často, a veľká časť pôvodnej *kvalitnej* genetickej informácie ostane zachovaná. V prípade zložitejšieho bludiska, nijaká konfigurácia genotypu nedokáže problém hľadania cesty vyriešiť, kvôli obmedzeniam vyplývajúcim z dĺžky génu.



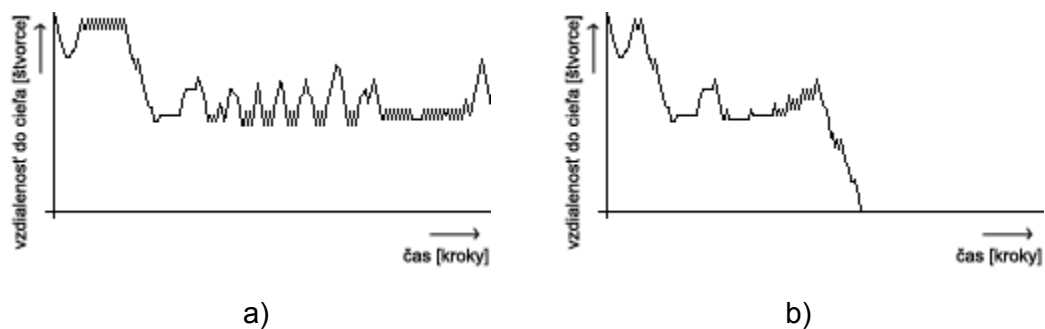
Obr. 7 Grafy charakterizujúce pohyb využívajúci znalosti typu genotyp

Grafy na obrázku 7 charakterizujú pohyb robota v bludisku pri použití znalostí typu genotyp so zotavením sa zo zacyklenia rotáciou génov. Ako testovacie bludisko bol použitý

súbor TestLab2.lab. Obrázok a) charakterizuje pohyb v bludisku bez detekcie slepých zón a chodieb, obrázok b) charakterizuje pohyb s detekciou slepých zón a chodieb.

Z obrázku 7 je vidieť, že algoritmus dokázal nájsť cestu bludiskom v prijateľnom čase. Vplyv detekcie slepých zón a chodieb sa neprejavil, pretože testovacie bludisko TestLab2 je veľmi jednoduché. Zotavenie zo zacyklenia prebiehalo deterministicky cyklickou rotáciou génov. V prípade náhodnej modifikácie génov by výsledok mohol a nemusel byť lepší.

Gradient je vektorová veličina charakterizujúca priestorovú zmenu (spád) skalárnej veličiny. Skalárnou veličinou je v tomto prípade vzdialenosť do cieľa a *pohybom v smere gradientu* je myslený pohyb smerom, ktorým je zmena skalárnej veličiny maximálna – teda smerom čo najbližšie k cieľu. Môže sa stať, že takýchto smerov (políčok) je viac. V tomto prípade systém navrhne ísť smerom, ktorý ešte nebol zmapovaný. Takýmto spôsobom je aspoň čiastočne motivovaný k tomu, aby zmapoval čo najväčšiu časť poľa, pretože čím viac toho program vie a čím obsiahlejšia je báza údajov, tým menšia je množina možných riešení a o to ľahšie je nájsť konečné riešenie. Tento postup – snaha čo najviac sa priblížiť k cieľu, je veľmi jednoduchá a efektívna metóda, avšak hrozí riziko uviaznutia v lokálnom extréme a následného zacyklenia. Zotavenie zo zacyklenia v tomto prípade nastáva voľbou náhradného cieľa – najvhodnejšieho nezmapovaného políčka do ktorého je možné sa dostať. Postup voľby dočasných cieľov bude vysvetlený neskôr.



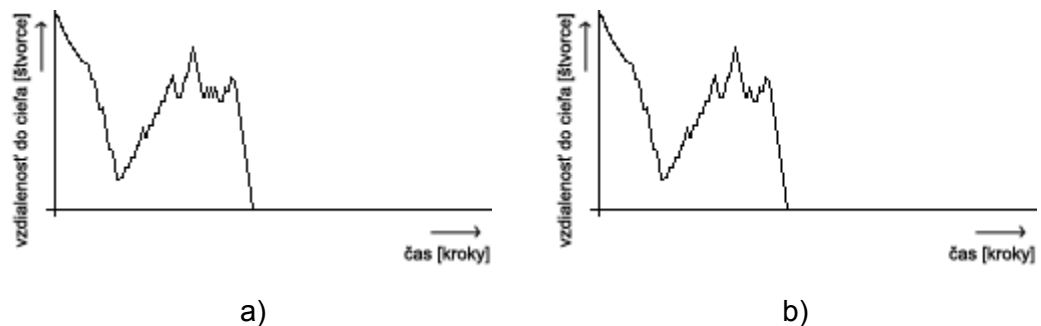
Obr. 8 Grafy charakterizujúce pohyb v smere gradientu

Grafy na obrázku 8 charakterizujú pohyb robota v bludisku v smere gradientu so zotavením sa zo zacyklenia voľbou dočasných cieľov čo najbližšie k robotovi. Ako testovacie bludisko bol použitý súbor TestLab4.lab. Obrázok a) charakterizuje pohyb v bludisku bez detekcie slepých zón a chodieb, obrázok b) charakterizuje pohyb s detekciou slepých zón a chodieb.

Obrázok 8 ukazuje chovanie sa systému v pomerne ľahkom bludisku. Použitie detekcie slepých chodieb má pri tejto metóde veľký význam, pretože práve slepé chodby predstavujú pre systém lokálne extrémy, v ktorých môže systém uviaznuť.

Všetky doteraz popisované metódy (okrem náhodného pohybu samozrejme) sa snažili nájsť riešenie na základe toho, že sa pokúšali čo najviac priblížiť k cieľu. Toto prináša so sebou riziko uviaznutia v lokálnom extréme a následného zacyklenia. Pozorovaním činnosti človeka – experta, pri riešení problému hľadania cesty bludiskom, je vidieť, že človek nikdy neuviazne v lokálnom extréme. Toto je spôsobené tým, že ak nájde cestu, ktorá nevedie k riešeniu, pokúsi sa hľadať iné riešenie. V okamihu keď človek (v tomto prípade je expertom aj obyčajný človek) nájde lokálny extrém prestáva sa sústrediť na to aby sa dostal čo najrýchlejšie do cieľa, ale snaží sa zmapovať si okolité prostredie aby si vybral nové potencionálne riešenie. Volí si teda *dočasný cieľ*. Implementovanie takéhoto typu znalostí do programu prináša ďalšie priblíženie sa k rozhodovaniu experta, a čo je najdôležitejšie – minimalizuje riziko uviaznutia. Práve tento algoritmus poskytuje najlepšie výsledky a dokáže spolu s detekciou slepých chodieb, vyriešiť hocijaké bludisko.

Pri voľbe náhradného riešenia sa expert môže vydať rôznymi cestami – volí dočasný cieľ podľa svojich kritérií. Jeden expert sa môže rozhodnúť ísť cestou, ktorá je k nemu najbližšie, iný zvolí cestu ktorá je najbližšie cieľu, rozhodovací proces tretieho môže byť úplne iný. Voľba dočasného cieľa je v programe riešená použitím rôznej metriky pre ohodnotenie *kvality* nového potencionálneho riešenia (ktoré závisí od voľby dočasného cieľa).



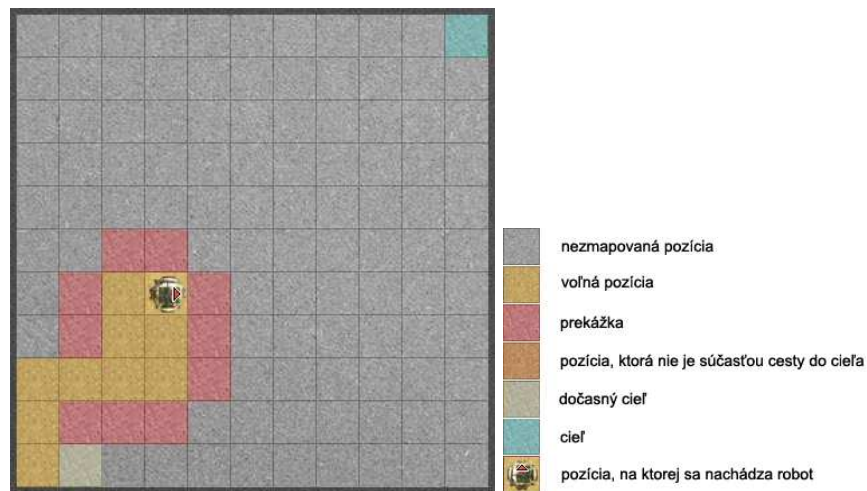
Obr. 9 Grafy charakterizujúce pohyb s voľbou dočasných cieľov

Grafy na obrázku 9 charakterizujú pohyb s voľbou dočasných cieľov čo najbližšie k robotovi. Ako testovacie bludisko bol použitý súbor TestLab6.lab. Obrázok a) charakterizuje pohyb v bludisku bez detekcie slepých zón a chodieb, obrázok b) charakterizuje pohyb s detekciou slepých zón a chodieb.

Veľmi dôležitou vlastnosťou pri hľadaní dočasných cieľov skutočnosť, že ak systém nemôže navrhnúť nijaký ďalší dočasný cieľ, dokáže správne určiť že bludisko nemá riešenie. Metódou dočasných cieľov je teda možné veľmi rýchlo vyriešiť aj problém určenia riešiteľnosti bludiska. Práve tento spôsob je použitý aj v tomto programe pri testovaní riešiteľnosti bludiska.

Expert pri hľadaní cesty v neznámom prostredí môže, v prípade, že je prostredie príliš komplexné a zložité usúdiť že uviazol v lokálnom extréme a pohybuje sa stále dookola v tej istej oblasti – zacyklil sa. Toto zistí obvykle na základe toho, že prostredie, v ktorom sa pohybuje sa mu zdá povedomé a dokáže prehlásiť „*tu som už predsa bol*“ a „*tieto kroky som už predsa raz podnikol*“. Ďalším faktom, ktorý ho môže utvrdiť v tom, že pravdepodobne uviazol v lokálnom extréme, môže byť skutočnosť, že jeho vzdialenosť do cieľa sa v poslednom čase príliš nemení. V takomto prípade pravdepodobne podnikne kroky, ktoré mu pomôžu sa z tejto situácie dostať.

Program sa snaží kopírovať rozhodovanie experta a pracuje podobne. V operačnej pamäti si systém vedie „denník“ o tom, kde sa v ktorom kroku robot nachádzal. Tento denník možno považovať za súčasť bázy údajov, pretože aj tieto informácie pomôžu nájsť systému riešenie. Na základe denníka dokáže systém určiť či už v pozícii, v ktorej sa nachádza už niekedy nebol, a či sa do tejto pozície nedostal krokmi ako niekedy v minulosti. V takomto prípade dokáže usúdiť, že došlo k zacykleniu a vykoná akciu, ktorá ho z tejto situácie dostane.

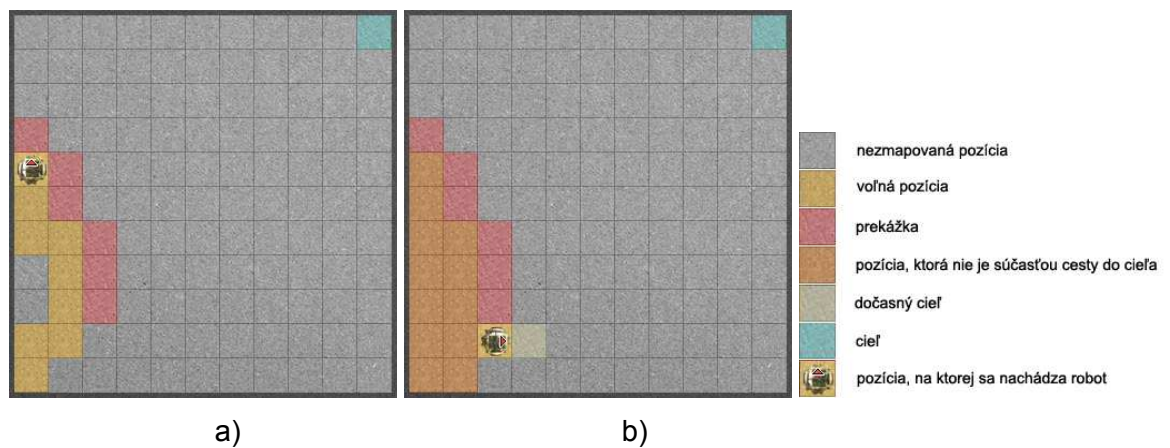


Obr. 10 Príklad triviálneho uviaznutia v lokálnom extréme.

Obrázok 10 demonštruje najjednoduchší typ uviaznutia v lokálnom extréme a následného zacyklenia. Robot sa snažil dostať do cieľa avšak uviazol v lokálnom minime. Na základe toho, že si „uvedomil“, že sa stále pohybuje v jednej oblasti zvolil nový dočasný cieľ. Zacyklenie tohoto typu väčšinou nastáva pri znalostiach typu genotyp, alebo pri pohybe v smere gradientu. Pri silnejších metódach dochádza k zacykleniu iba v omnoho komplexnejších situáciách a riziko, že k nemu dôjde je veľmi malé.

Pri pohybe bludiskom, si expert často dokáže uvedomiť, že určitá cesta nevedie k riešeniu, pretože je to slepá chodba a vylúči ju z množiny prípustných ciest vedúcich do cieľa. Ak expert správne vylúči všetky nevhodné cesty, musí mu ostať správna cesta – ak existuje. Kybernetický systém, ak chce úspešne konkurovať živému expertovi si musí takéto skutočnosti uvedomovať tiež. Záznam o tom, že určitá cesta nevedie do cieľa si urobí v báze údajov, a políčka označí podobne ako obsadené, a nebude sa do nich už v budúcnosti vracieť.

V tejto chvíli je dobré poznamenať, že tu získava expertný systém oproti človeku obrovskú výhodu, pretože si dokáže zapamätať úplne všetky slepé chodby, ktoré objaví. Rýchlosť s akou tieto chodby objaví je tiež mnohonásobne vyššia ako u živého experta a teda najmä pri zložitejších a rozsiahlejších bludiskách ho táto schopnosť stavia do výhodnejšej pozície oproti živému expertovi.



Obr. 11 Detekcia slepých chodieb

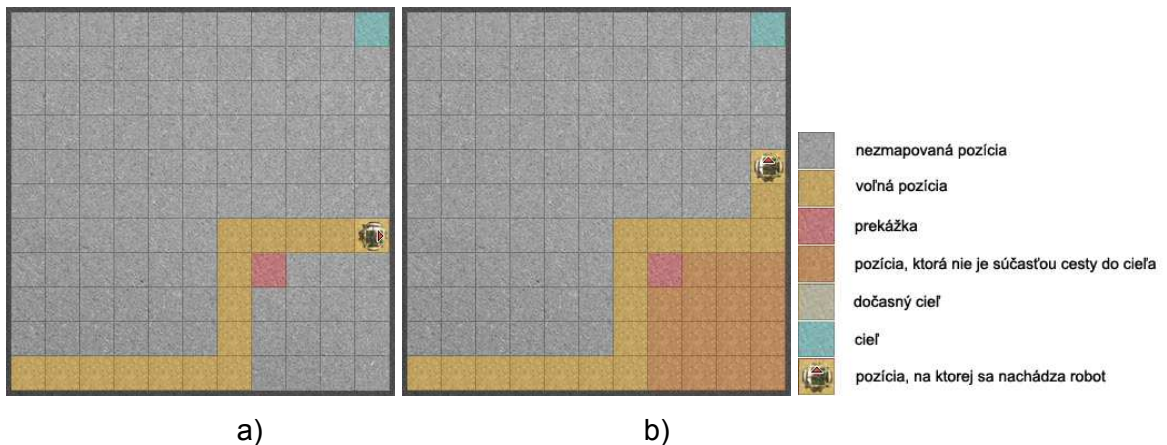
Na obrázku 11 je vidieť ako prebieha rozhodovanie systému v prípade objavnia slepej chodby. Na obrázku a) robot dorazil do slepej chodby. Na obrázku b) je vidieť, že pri ceste naspäť si robot označil slepú chodbu, a v budúcnosti sa už do nej nevráti.

Expert, pri hľadaní cesty bludiskom občas objaví celé oblasti, ktoré môže považovať za oblasti, ktorými určite nevedie cesta k cieľu. Takéto oblasti sa budú vyskytovať najmä v bludiskách, ktoré sú ohraničené zo všetkých štyroch strán a to najmä v oblastiach rohov. Keďže expertný systém by mal čo najvernejšie kopírovať činnosť experta, musí byť aj program byť schopný odhaliť tieto oblasti. V prípade, že program takúto zónu objaví, poznačí to do bázy údajov a políčka v zóne označí podobne ako obsadené a nebude sa do nich už v budúcnosti vracieť.

Program na rozdiel od experta označuje aj zóny vnútri bludiska, ktoré nie sú potrebné pre nájdenie cesty bludiskom. Takéto riešenie síce ze určitých okolností môže cestu do cieľa

predĺžiť, avšak výrazne zmenší množinu, v ktorej expertný systém bude hľadať riešenie. V mnohých prípadoch použitie detekcie slepých zón zníži počet krokov potrebných k nájdeniu riešenia a použitie tejto techniky má určite význam.

Kombináciou detekcie slepých chodieb a zón dokáže systém správne vylúčiť prakticky všetky „neužitočné“ cesty, čím sa celé bludisko značne zjednoduší (veľakrát ostane len jediná možná cesta) a proces hľadania cesty sa stáva triviálnym.



Obr. 12 Detekcia slepých zón

Obrázok 12 zobrazuje ako prebieha detekcia slepej zóny. Na obrázku a) robot objavil slepú zónu. Na obrázku b) systém robot zónu zaznačil, akonáhle sa bezpečne vzdialil od tejto zóny, a v budúcnosti sa už do nej nevráti.

Pri každej voľbe dočasného cieľa, či už v prípade uviaznutia, alebo len pri voľbe nového podcieľa, je dôležité tento nový cieľ zvoliť čo najvhodnejšie. Tento nový cieľ bude pravdepodobne nezmapované políčko, také do ktorého je možné sa dostať. Nezmapované bude preto, aby bol systém motivovaný k prehľadávaniu bludiska a dopĺňal nové informácie do bázy údajov. V prípade, že takýchto políčok bude viac, je nutné vybrať to najvhodnejšie. Každý potencionalný nový cieľ je nutné nejakým spôsobom ohodnotiť, a nakoniec vybrať ten najvhodnejší.

Ak je ako metrika použitá vzdialenosť k robotovi vyberá sa ako dočasný cieľ to políčko, ktorého vzdialenosť k robotovi je najmenšia. V prípade, že je takýchto políčok viac vyberá sa to, ktoré je z nich najbližšie k cieľu. Tento typ metriky pre dočasný cieľ predstavuje najnižie riziko uviaznutia v lokálnom extréme, keďže do cieľa, ktorý je blízko robota je väčšinou možné dostať sa bez problémov.

Ak je ako metrika použitá vzdialenosť do cieľa vyberá sa ako dočasný cieľ to políčko, z ktorého je vzdialenosť do cieľa najmenšia. V prípade, že je takýchto políčok viac vyberá sa to, ktoré je z nich najbližšie k robotovi. Tento typ metriky pre dočasný cieľ zaisť najrýchlejší

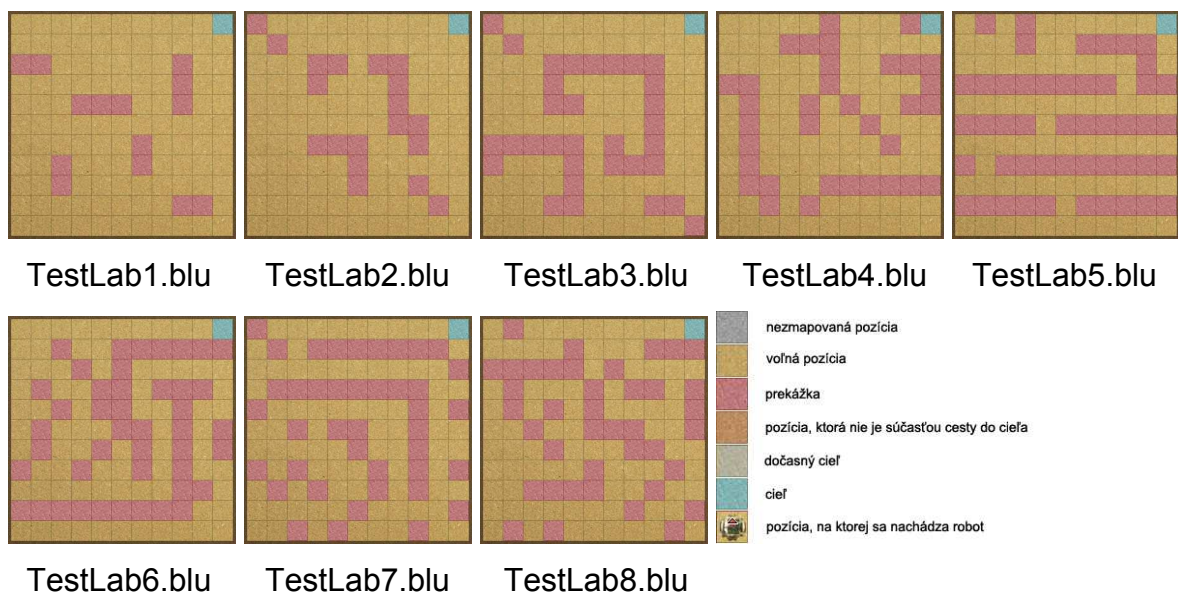
pohyb k skutočnému cieľu, avšak zároveň prináša väčšie riziko uviaznutia v lokálnom extréme, keďže nový dočasný cieľ môže byť príliš ďaleko od súčasnej pozície robota a robot sa nemusí do ňo vždy jednoducho dostať.

Ak je ako metrika použitý stred medzi robotom a cieľom, tak potenciálne dočasné ciele sú ohodnotené podľa ich vzdialenosti k robotovi plus ich vzdialenosti k cieľu. Najvhodnejšie políčka sú teda blízko k robotovi, a zároveň blízko k cieľu. Toto predstavuje kompromis medzi prvými dvoma spôsobmi určovania metriky.

Posledným spôsobom je náhodné určenie dočasného cieľa. Táto metóda zvolí dočasný cieľ bez ohľadu na jeho vzdialenosť k cieľu či k robotovi. Takáto voľba väčšinou prinesie horší výsledok ako deterministické určenie dočasného cieľa, avšak práve vďaka faktoru náhody môže tento spôsob znížiť riziko uviaznutia v lokálnom extréme.

4.2 Testovanie vlastností jednotlivých metód

Pri návrhu jednotlivých metód resp. znalostí nebolo možné presne určiť, ktorá metóda bude tá najefektívnejšia a bude použitá vo finálnom expertnom systéme. Aby bolo možné ohodnotiť efektívnosť jednotlivých metód, jednotlivé algoritmy a ich kombinácie boli podrobené sérii testov. Metódy boli testované na ôsmich bludiskách rozličnej náročnosti.



Obr. 13 Bludiská použité v testoch

Po otestovaní všetkých alternatív bol vyhodnotený priemerný počet krokov ktoré robot, pri určitej metóde potreboval na prejdenie jedného bludiska a hodnoty boli zipsané do tabuľky najvhodnejších metód.

Tab. 1 Tabuľka najvhodnejších metód

Poradie metódy	Algoritmus	Priemerný počet krokov potrebných k prejdeniu bludiska
1	Voľba dočasných cieľov čo najbližšie k robotovi s detekciou slepých chodieb	75,8
2	Voľba dočasných cieľov čo najbližšie k robotovi s detekciou slepých chodieb a zón	76,5
3	Voľba dočasných cieľov medzi robotom a cieľom s detekciou slepých chodieb	77,8
4	Voľba dočasných cieľov čo najbližšie k cieľu s detekciou slepých chodieb a zón	92,3
5	Voľba dočasných cieľov medzi robotom a cieľom s detekciou slepých chodieb a zón	95,1
6	Voľba dočasných cieľov čo najbližšie k cieľu s detekciou slepých chodieb	127,8
7	Pohyb v smere gradientu so zotavením sa voľbou dočasného cieľa čo najbližšie k cieľu s detekciou slepých chodieb a zón	165,7
8	Pohyb v smere gradientu so zotavením sa voľbou dočasného cieľa čo najbližšie k cieľu s detekciou slepých chodieb	175,3
9	Pohyb v smere gradientu so zotavením sa voľbou náhodného dočasného cieľa s detekciou slepých chodieb a zón	185
10	Pohyb v smere gradientu so zotavením sa voľbou dočasného cieľa medzi robotom a cieľom s detekciou slepých chodieb a zón	190,1
11	Pohyb v smere gradientu so zotavením sa voľbou náhodného dočasného cieľa s detekciou slepých chodieb	194
12	Pohyb v smere gradientu so zotavením sa voľbou dočasného cieľa medzi robotom a cieľom s detekciou slepých chodieb	199,1
13	Znalosti typu „Genotyp“ so zotavením sa voľbou dočasného cieľa medzi robotom a cieľom s detekciou slepých chodieb	203,3
14	Znalosti typu „Genotyp“ so zotavením sa voľbou dočasného cieľa medzi robotom a cieľom s detekciou slepých chodieb a zón	203,3
15	Pohyb v smere gradientu so zotavením sa voľbou dočasného cieľa čo najbližšie k robotovi s detekciou slepých chodieb a zón	208,2

V tabuľke 1 je uvedených 15 najlepších metód, ktoré dokázali vyriešiť všetkých osem testovacích bludísk. Priemerný počet krokov potrebných k prejdeniu bludiska bol vyrátaný ako priemer z počtu krokov, ktoré systém potreboval na vyriešenie bludiska pre jednotlivé testovacie bludiská.

Metóda náhodného pohybu skutočne nedokáže priniesť uspokojivé výsledky. Nepoužíva prakticky žiadne relevantné znalosti, a to či nájde riešenie je iba otázkou náhody. Použitie takejto metódy nemá reálny význam.

Použitie znalostí vo forme genotypu prináša akceptovateľné výsledky iba pri veľmi jednoduchých bludiskách. Pri komplexnejších problémoch nedokáže metóda nájsť riešenie ani pri rôznej modifikácii génov. Toto je samozrejme spôsobené aj malou dĺžkou genotypu, ktorý dokáže popísať iba veľmi málo a aj to iba elementárnych situácií. Kvôli úplnosti bol *ideálny* genotyp nahradený aj úplne náhodným genotypom, avšak výsledky boli, podľa očakávaní, ešte horšie.

Metóda pohybu v smere gradientu dokázala vyriešiť veľkú časť bludísk. Jej najväčšou nevýhodou je však to, že veľmi často uviazne v lokálnom extréme a zotavenie zo zacyklenia trvá pomerne dlho. Kvôli tejto vlastnosti metóda nájde cestu bludiskom až po veľmi veľkom počte krokov.

Metóda voľby dočasných cieľov sa približuje najviac rozhodovaniu experta a aj preto prináša najlepšie výsledky. Táto metóda dokáže nájsť riešenie akéhokoľvek bludiska v prijateľnom čase. V kombinácii s detekciou slepých chodieb a zón ponúka najefektívnejšie riešenie problému hľadania cesty v bludisku z doteraz spomínaných metód.

Detekcia slepých chodieb má jednoznačne pozitívny vplyv na počet krokov nutných k prejdeniu bludiska, a zároveň znižuje riziko uviaznutia. Vďaka jej použitiu dokázali bludisko vyriešiť aj slabšie metódy. Detekcia slepých chodieb má v systéme neoceniteľný význam pri bludiskách v ktorých existuje iba jediná cesta do cieľa.

Detekcia slepých zón mierne zvyšuje pravdepodobnosť vyriešenia bludiska. Vplyv na počet krokov nutných k prejdeniu bludiska je len veľmi malý. V niektorých prípadoch sa dokonca počet krokov potrebných k prejdeniu bludiska dokonca zvýšil.

Keďže expertný systém má nahadiť funkciu experta a má robiť rozhodnutia minimálne tak kvalitné ako expert, systém bol porovnaný aj so živými ľuďmi. Nakoľko problém hľadania cesty bludiskom je veľmi triviálny, nebolo nutné hľadať špecialistov a program bol otestovaný ľuďmi z môjho okolia. Tri osoby boli požiadané aby sa pokúsili nájsť cestu tými istými neznámymi testovacími bludiskami, ktoré boli použité pri teste systému. V prípade jednoduchých bludísk boli výsledky približne rovnaké, ale najmä pri komplexnejších bludiskách program dokázal nájsť riešenie tak isto rýchlo, ba až rýchlejšie (v zmysle počtu potrebných krokov) ako človek. Toto však nebolo spôsobené len tým, že by znalosti programu boli kvalitnejšie ako znalosti človeka. Výhoda osobného počítača spočíva samozrejme v jeho výpočtovej sile, rýchlosti a veľkosti pamäti. Možno predpokladať, že pri bludiskách väčších rozmerov by sa rozdiel medzi človekom a programom prejavil ešte výraznejšie. Samozrejme v prospech expertného systému.

Záver

Práca si kládla za úlohu overiť možnosti použitia expertných systémov pre riešenie na prvý pohľad triviálnych problémov. Ukázalo sa, že aj takáto neštandardná aplikácia ES dokáže priniesť pozitívne výsledky, a možnosti nasadenia expertných systémov sú skutočne veľmi široké. Podarilo sa zostrojiť systém, ktorý je minimálne tak *inteligentný* ako človek. Aspoň v prípade problému hľadania cesty bludiskom. Je možné, že existujú ešte silnejšie algoritmy na riešenie tohoto druhu problému. V tomto systéme však šlo o snahu čo najvernejšie kopírovať myslenie človeka – experta. Analýzou myšlienkových pochodov, a rozhodnutí experta boli formulované jednoduché znalosti, ktoré boli neskôr implementované do programu.

Existujú samozrejme mnohé ďalšie znalosti, ktoré človek dokáže využiť avšak tieto do systému neboli implementované nakoľko je problém ich vhodne reprezentovať. Jedná sa najmä o rozpoznávanie tvarov, detekciu slepých zón nepravidelného tvaru a pokročilé heuristické metódy, teda rozhodnutia, ktoré sám expert nevie racionálne vysvetliť, avšak často prinášajú úspech.

Bibliografia a bibliografické odkazy

- [Kri96] KRISHNAMOORTHY, C.S. – RAJEEV, S.: Artificial Intelligence and Expert Systems for Engineers. Madras: CRC Press, 1996, 254 s. ISBN 0849391253
- [Pop89] POPPER, M. – KELEMEN, J.: Expertné systémy 1. vyd. Bratislava: Alfa, 1989, 360 s. ISBN 80-05-00051-0
- [Sel92] SELIG, J.M.: Introductory Robotics. Hertfordshire: Prentice Hall International, 1992, 157 s. ISBN 0-13-488875-8 (pbk.)
- [Boh05] BÖHMER M.: Návrh a optimalizácia kybernetického systému pomocou genetických algoritmov: Diplomová práca. Liptovský Mikuláš: Akadémia ozbrojených síl generála Milana Rastislava Štefánika Liptovský Mikuláš, 2005
- [Rac95] RAČANSKÝ V.: Umělá inteligence: Zápisy z prednášok. Brno:Universitas Masarykiana, 1995, 48 s.
- [Har04] Hardware server: RS232
<http://rs232.hw.cz/> (2004-09-11)
- [Joh04] JOHNSON J.: Working With Stepper Motors
<http://eio.com/jasstep.htm> (2004-05-13)
- [Kyb05] Kybernetika
http://karlik.gymkc.cz/vyuka/m/inf_minerva/kybernetika.html (2005-01-03)
- [Rez02] ŘEZÁČ K.: Krokové motory, princip funkce, metody řízení
<http://robotika.cz/articles/steppers/cs> (2002-28-10)
- [Wik05] Wikipedia – The Free Encyclopedia
<http://www.wikipedia.org> heslá: artificial intelligence, cybernetics, expert systems, robot, robotics, (2005)