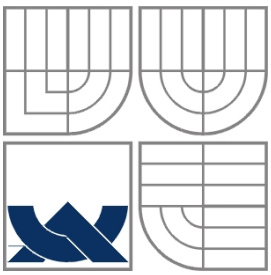


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

PLÁNOVÁNÍ CESTY MOBILNÍHO ROBOTA POMOCÍ GENETICKÉHO ALGORITMU

MOBILE ROBOT PATH PLANNING BY MEANS OF GENETIC ALGORITHM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — PETR SIPTÁK
AUTHOR

VEDOUCÍ PRÁCE — RNDr. JIŘÍ DVOŘÁK, CSc.
SUPERVISOR

BRNO 2009

Abstrakt

Má práce se zabývá plánováním cesty mobilního robota pomocí genetických algoritmů. V první části jsou popsány známé přístupy k problematice a ve druhé části popisuji vlastní řešení pomocí jazyka C#, které jsem pojal jako názorné vysvětlení chodu genetických algoritmů.

Abstract

My thesis deals with the mobile robot path planning by means of genetic algorithms. The first part describes common approaches to the topic and in the second part I describe my own solution using language C# which I conceived as a schematic illustration of how genetic algorithms work.

KLÍČOVÁ SLOVA

Robot, plánování cesty, genetický algoritmus

KEYWORDS

Robot, path planning, genetic algorithm

OBSAH:

1 Úvod

Plánování cesty ve známém prostředí se stalo za posledních deset let velice žádaným odvětvím robotiky, zejména díky rozvoji různých navigačních systémů. Úkolem této bakalářské práce bylo vytvořit program, který pro mobilního robota najde ve známém dvourozměrném prostředí s danými překážkami souvislou bezkolizní cestu ze zadaného startu do daného cíle.V první části jsou popsány některé dosavadní přístupy k této problematice, dále je popsáno vlastní řešení v prostředí Microsoft Visual Studio a jazyk C#. K vyhledávání cesty jsou použity genetické algoritmy a algoritmus A*.

Genetické algoritmy jsou založeny v podstatě na Darwinově teorii o vývoji druhů. Byly popsány již v 70. létech minulého století a jsou s úspěchem využívány v situacích, kde klasické exaktní postupy selhávají. Hlavní myšlenka spočívá v aplikaci přírodních vývojových procesů jako dědičnost, křížení, mutace, přirozený výběr na řešení zadané úlohy.

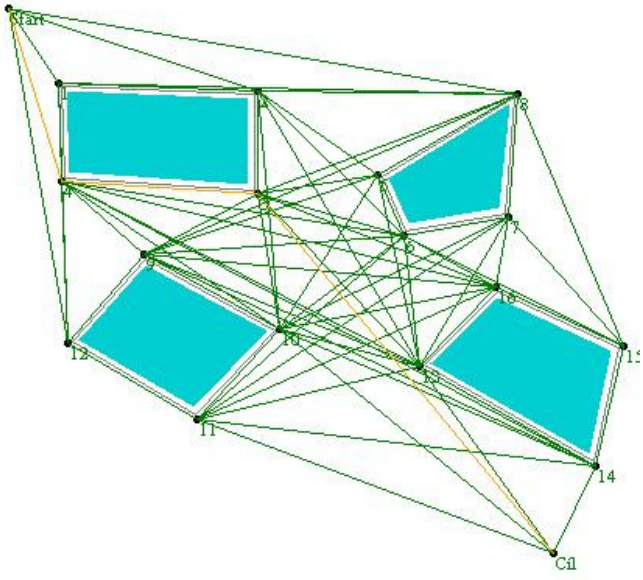
Algoritmus A* byl popsán v roce 1968 a vyhledává nejlepší první cestu ze zadaného startu do cíle. Využívá heuristickou funkci, která v závislosti na součtu vzdáleností od startu a cíle jednotlivých bodů determinuje pořadí těchto bodů.

2 Analýza dosavadních přístupů k plánování cesty robota

Důležitým nástrojem pro nalezení cesty robota je vhodná reprezentace prostoru. Nejpoužívanější přístup je zjednodušit problém na vyhledání cesty grafem. [3]

Graf viditelnosti

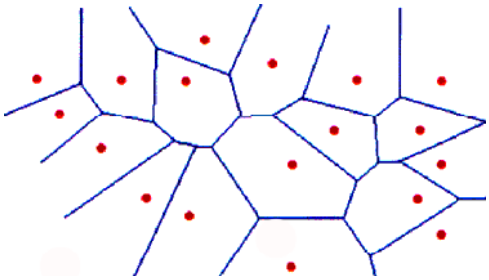
Graf viditelnosti je graf, jehož vrcholy jsou počáteční a cílový bod a vrcholy všech překážek. Vybrány jsou pouze ty spojnice jednotlivých vrcholů, které neprotínají překážky. Problém plánování cesty je tedy převeden na vyhledání co nejkratší cesty grafem mezi počátečním a cílovým bodem.



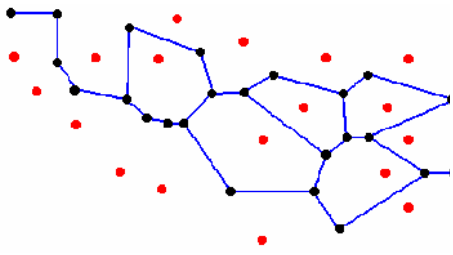
Obr. 1 Graf viditelnosti

Retrakční metoda

Tato metoda používá k nalezení cesty Voronoiův diagram. Okraje Voronoiova diagramu představují cesty, které jsou stejně vzdálené od dvou nejbližších překážek a jeho vrcholy jsou body, kde se setkávají tři a více takových cest. Řešení je nalezení nejkratší cesty grafem.



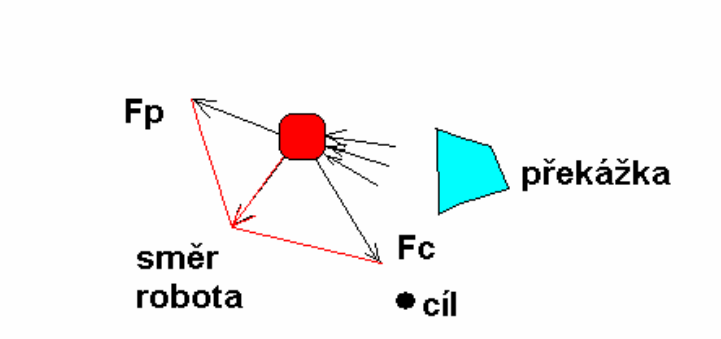
Obr.2 Voronoiův diagram [5]



Obr.3 Vlastní graf [5]

Metody pole potenciálů

Tyto metody používají myšlenku imaginárních sil působících na robota. Překážky působí na robota odpudivou silou, zatímco cíl na něj působí přitažlivou silou. Suma těchto sil, výsledná síla R, určuje následný směr a rychlost cesty. Jeden z důvodů popularity těchto metod spočívá v jejich jednoduchosti a eleganci. Při těchto metodách však není zaručeno, že nalezená cesta bude nejkratší nebo nejbezpečnější.



Obr. 4

Rozložení prostoru do jednoduchých oblastí

Jedním z nejstarších přístupů plánování cesty je rozložení prostoru, kudy se robot může pohybovat, do jednoduchých oblastí zvaných *buňky* a konstrukce neorientovaného grafu, tzv. souvislého grafu. Tento graf představuje vztah sousednosti mezi buňkami. Obvykle je pro vyhledání cesty použit Dijkstrův vyhledávací algoritmus nebo algoritmus A*.

Algoritmus A*

Algoritmus A*využívá heuristickou funkci, která v závislosti na součtech vzdáleností od startu a cíle jednotlivých bodů determinuje pořadí těchto bodů. Pro každý uzel tedy pracuje s těmito třemi hodnotami:

g(x): skutečná vzdálenost ze startu do aktuálního uzlu

h(x): vzdálenost od aktuálního uzlu k cíli

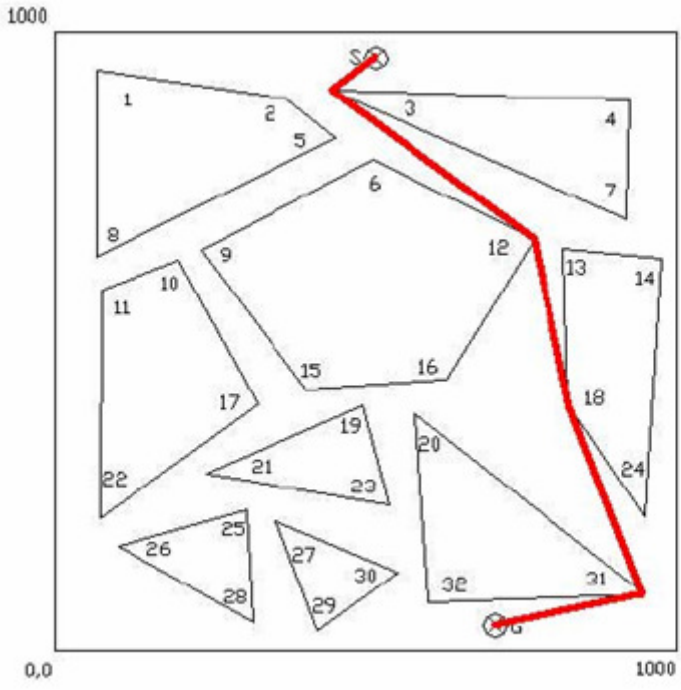
f(x): součet g(x) a h(x)

Při prohledávání bude tedy vybrán ten uzel, který má nejmenší hodnotu hodnotící funkce f(x). Protože však funkce h(x) není známá, nahrazujeme ji funkcí h*(x), která vyjadřuje odhad vzdálenosti z daného uzlu do cíle. Bývá tedy nazývána heuristickou funkcí a je pro efektivitu prohledávání velmi podstatná. Do fronty cest se jako první uloží cesta tvořená počátečním bodem. Z fronty se postupně berou cesty, pokud jejich poslední bod je stejný jako cíl, tak je cesta řešením. Jinak jsou vytvořeny nové cesty spojením této cesty a sousedních bodů. Tyto nové cesty jsou zařazeny do fronty v pořadí podle vzdálenosti od cíle. Body, kterými algoritmus už jednou prošel se uloží do pole uzavřených bodů a cesty, které končí tímto bodem, nejsou dále zpracovávány.

Genetické algoritmy

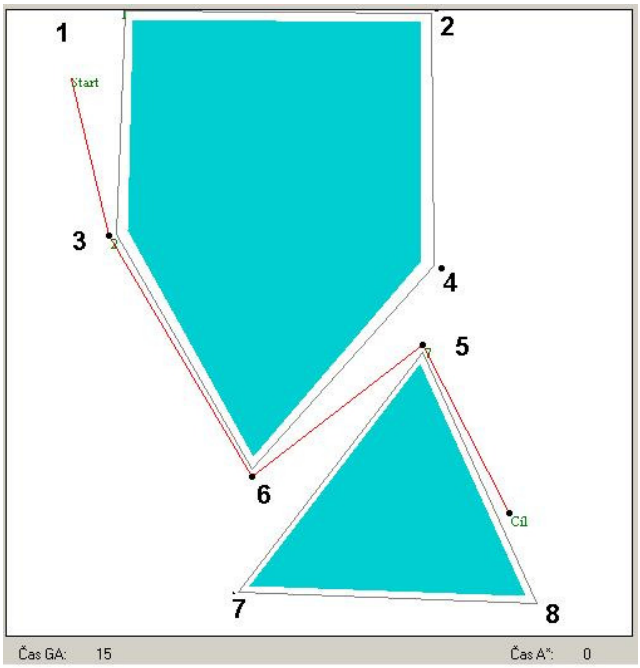
Tato metoda byla objevena v sedmdesátých létech minulého století a spočívá v aplikaci Darwinovy teorie o přirozeném výběru druhů na řešení složitých situací, kde selhávají klasické matematické a fyzikální postupy. Spočívá v tom, že několik nejzdatnějších jedinců v daném prostředí má nejvyšší pravděpodobnost přežití a tendenci k dalšímu rozmnožování, zatímco slabší jedinci jsou z generace odstraněni. Typ prostředí, ve kterém se robot pohybuje, můžeme rozdělit na diskrétní a spojité. V diskrétním prostředí je scéna tvořena sítí jednotlivých buněk a překážky jsou složeny z jedné nebo více buněk. Spojité prostředí je více podobné realitě a překážky jsou tvořeny vlastními vrcholy a hranami. Každá cesta je reprezentována jako chromozom a jednotlivé body cesty jako geny. Tyto chromozomy mohou mít pevnou nebo proměnnou délku. K vlastnímu procesu jsou používány jak klasické operátory tak specifické. Mezi základní operátory patří selekce, sloužící k výběru jedinců vhodných ke křížení, vlastní křížení a mutace, které umožňují pestrost populace. Ke specifickým operátorům patří vložení nebo odebrání genu. Ke každému chromozomu je přiřazena hodnota fitness, která určuje jeho kvalitu. Po vytvoření první náhodné populace a následné aplikaci operátorů se částečně nebo úplně nahradí minulá populace novou až do ukončení běhu programu.

Zajímavé řešení nabízí metoda použitá v práci [3], kde je definována cesta jako posloupnost jedniček a nul. Při vytvoření prostředí se vrcholy překážek očíslují zleva doprava a shora dolů. Pokud výsledná cesta prochází určitým vrcholem, je na jeho pozici dané pořadím vrcholu generována 1. Na ostatních vrcholech je dána 0. Na obr. 5 cesta prochází body 3,12,18,31, což odpovídá chromozomu 00100000000100000100000000000010.



Obr. 5 Graf se 32 vrcholy a optimální cesta generovaná GA [3]

Problém však nastane v případě prostředí, kde následující vrchol bude výše (nebo vlevo) než předchozí vrchol, který zřejmě bude očíslován nižším indexem, jako v případě cesty z bodu 6 na bod 5. Tato cesta by byla binárně reprezentována jako 00101100, což by znamenalo cestu Start-3-5-6-Cíl, nikoli Start-3-6-5-Cíl, jak by bylo správné (obr. 6). Úspěšnost tedy závisí na způsobu očíslování vrcholů.



Obr.6

3 Návrh řešení

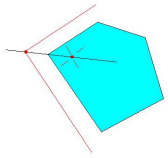
K řešení jsem použil metodu genetických algoritmů. První krok je náhodné vytvoření počáteční populace cest.. Jako body určující cestu robota používám buď náhodně vygenerované body nebo přímo vrcholy zvětšených překážek, takže se významně omezuje vyhledávací prostor. Druhá změna spočívá v možnosti proměnného nastavení počtu bodů k mutaci, což umožňuje nalezení cesty i ve složitějších prostředích. Jako další metodu program používá algoritmus A*, který je v závěru porovnán s genetickými algoritmy.

3.1 Prostředí

Robot a překážky se nacházejí ve spojitém dvojrozměrném prostředí, polohy překážek, startu a cíle jsou předem dány a v čase se nemění. Pro cestu robota se používá zvětšený model překážek o jeho šířku, aby se nedotýkal překážek.

Zvětšení překážek

Zvětšování překážek vzniká posunutím jednotlivých vrcholů překážky. Kolem každého vrcholu se vytvoří dva body na přímce půlící úhel dvou sousedních stran překážky a jejich vzdálenost od původního vrcholu je zadána uživatelem. Následně se vybere bod, který neleží uvnitř překážky.



3.2 Chromozom

Chromozom se skládá z genů a představuje jedno řešení problému, v tomto případě cesty od startu do cíle, a to i přes překážky za cenu penalizace. Každý chromozom obsahuje stejný první a poslední gen, který označuje start a cíl. Chromozomy můžeme reprezentovat binárně nebo diskrétně.

Binární reprezentace genů

Binární reprezentace spočívá v uložení cesty jako kombinaci jedniček a nul reprezentující, zda robot navštíví daný bod. Takže cesta uložená jako 100101 by znamenala, že robot půjde přes body 1,4,6. Možnou nevýhodou je, že robot nemůže jít z vyššího bodu na nižší.

Diskrétní reprezentace genů

Diskrétní reprezentace je ukládání cesty robota jako posloupnost bodů, které navštíví. Body můžeme v chromozomu ukládat jako indexy bodů (cesta z minulého příkladu by vypadala následovně [1,4,6], čímž se ovšem omezíme v možnostech mutace) nebo jako jejich souřadnice (tzn. [(10,10), (45,64), (450,450)]. V mém řešení jsem zvolil diskrétní reprezentaci a jednotlivé geny ukládám jako souřadnice.

3.3 Počáteční populace

Větší populace má větší šanci k nalezení cesty za cenu snížení rychlosti. Počáteční populace je náhodně vygenerovaná. Počet genů v chromozomu (bez startu a cíle) je dán náhodně od nuly do celkového počtu vrcholů minus jedna.V programu se dá velikost počáteční populace zadávat uživatelem.

3.4 Hodnota fitness

Každý chromozom má hodnotu fitness, která určuje jeho kvalitu. Nejčastěji se jedná o délku cesty, kterou chromozom reprezentuje, ale mohou být zohledněny i další charakteristiky. Délka cesty se zvyšuje v případě kolize cesty s překážkou o penalizační hodnotu (v programu o 5000).

3.5 Selekce rodičů

Selekce vybírá jedince ke křížení. Zdálo by se logické, že pro výběr rodičů by byly nejvhodnější chromozomy s nejlepší fitness funkcí. Ukázalo se však, že se pak hledaná cesta soustřeďuje jen do jedné části prohledávaného prostředí, protože se snižuje genetická pestrost chromozomů. Proto je lépe použít nějaká pravděpodobnostní pravidla. Existuje více druhů selekce.

Turnajový výběr

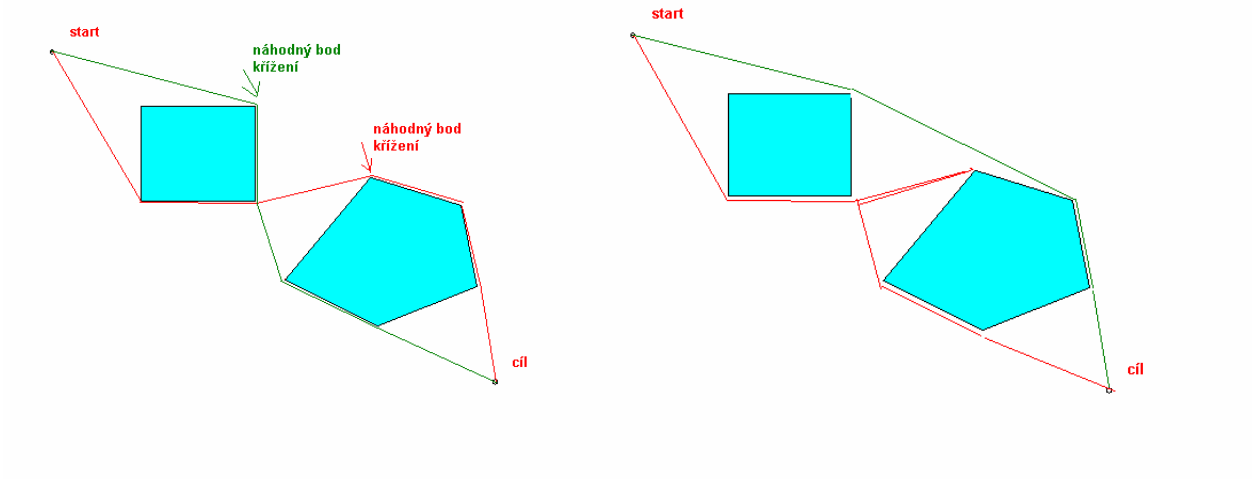
Ze všech chromozomů vybereme náhodně K chromozomů. Hodnota K říká, o jaký stupeň výběru se jedná. Z těchto chromozomů pak vybereme ten nejsilnější.V programu je použit turnajový výběr druhého stupně tzv. binární turnajový výběr.

Proporcionální výběr

Šance na výběr chromozomu se v tomto způsobu selekce odvíjí od velikosti jeho hodnoty fitness.

3.6 Křížení

Křížení kombinuje jednotlivé části rodičů a tím vytvoří nové chromozomy vedoucí k lepším řešením. Existuje více druhů křížení, zde jsem vybral bodové křížení. Pomocí selekce jsou vybráni dva jedinci ke křížení. Náhodně se vygeneruje jeden gen z každého chromozomu. Křížení dvou chromozomů vyprodukuje dva nové chromozomy, z nichž každý obsahuje unikátní kombinaci složenou z částí chromozomu rodičů.



Obr.7 Křížení

3.7 Mutace

Mutace chromozomu ovlivňuje jen nově vytvořené potomky a zaručuje tvorbu nových genů, které v populaci ještě nemusí být obsažené.

Jednobodová mutace

Náhodně se vybere jeden gen v chromozomu a ten se zamění za jiný.

Vícebodová mutace

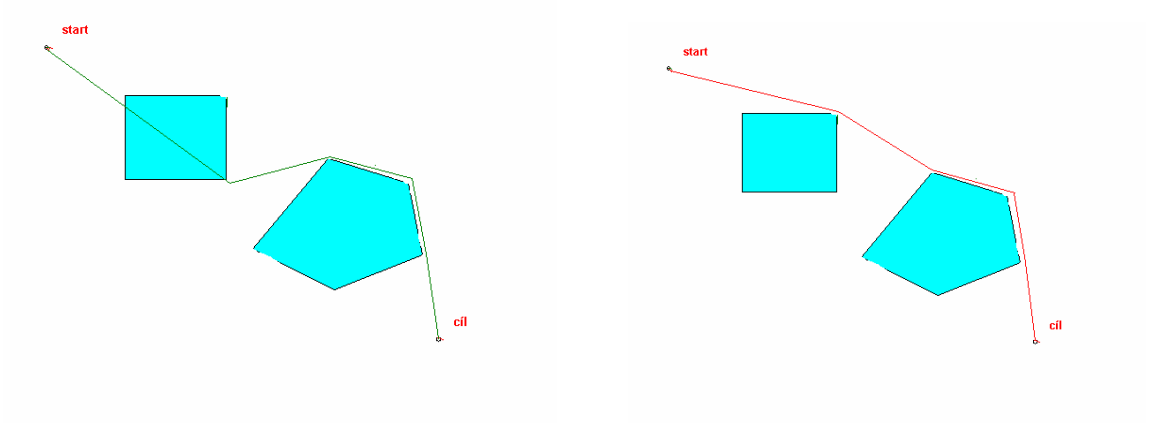
Projede se každý gen v chromozomu a s určitou pravděpodobností se náhodně zamění za jiný.

Způsob mutace

Jednotlivé geny můžou mutovat buď tím způsobem, že se náhodně vybere jiný bod z předem připraveného grafu nebo se určí nový naprosto náhodný bod kdekoliv v prostoru. V programu je použita vícebodová mutace, kdy počet genů a způsob mutace je nastavitelný uživatelem.

Šance mutace

Uživatel si může zvolit s jakou pravděpodobností dojde k záměně každého genu za jiný.



Obr.8 Mutace

Šance přidání

Uživatel si může zvolit s jakou pravděpodobností dojde k přidání genu.

3.8 Nová populace

Nová populace se může vytvořit buď z naprosto nových jedinců, nebo se může zachovat většina původní populace. Velikost populace generací většinou zůstává stejná.

Generační výměna

Stará populace je celá smazána a jako nová populace jsou bráni pouze potomci. U této metody někdy zůstává nejsilnější jedinec ze staré populace zachován.

Inkrementální náhrada

Do nové populace se dostane většina jedinců z minulé populace. Několik nejslabších jedinců je nahrazeno potomky. Inkrementální náhrada je použita v programu.

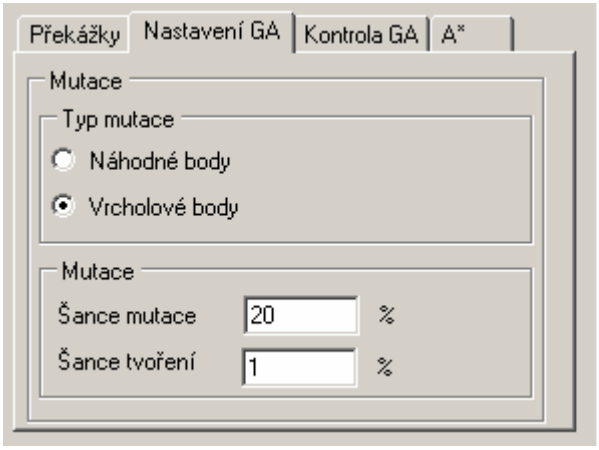
3.9 Evoluce

Začátek evoluce začíná vytvořením počáteční populace. U všech jedinců této populace se určí jejich hodnota fitness. Jeden krok evoluce spočívá v tom, že se selekcí vyberou dva vhodné chromozomy ke křížení. Noví jedinci vzniklí křížením zmutují a je vytvořena nová populace.

4 Popis programu

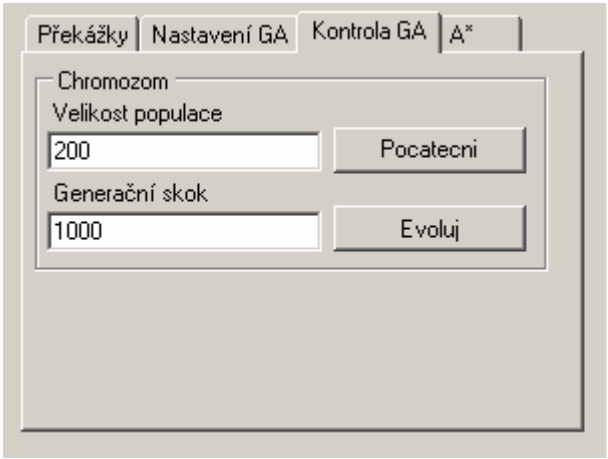
K řešení je použito prostředí Microsoft Visual Studio a jazyk C#. Snažil jsem se o co nejmenší počet zadávaných dat a tím i zjednodušení ovládání.

Po spuštění programu se automaticky objeví přednastavené pozice startu a cíle, které se dají případně měnit pomocí myši. Překážky se kreslí pomocí myši a tlačítka enter, které dokončuje, popř. začíná překážku. Zvětšení překážky se děje automaticky o uživatelem nastavený počet pixelů.



Obr. 9 Záložka "Nastavení GA"

Na záložce nastavení GA obr. 9 si můžeme vybrat, zda se k vytvoření chromozomu použijí jako geny vrcholy jednotlivých překážek nebo náhodně vygenerované body v prostoru mimo překážky. Šance mutace určuje, s jakou pravděpodobností chromozomy zmutují a šance tvoření určuje, s jakou pravděpodobností se přidají nové geny.



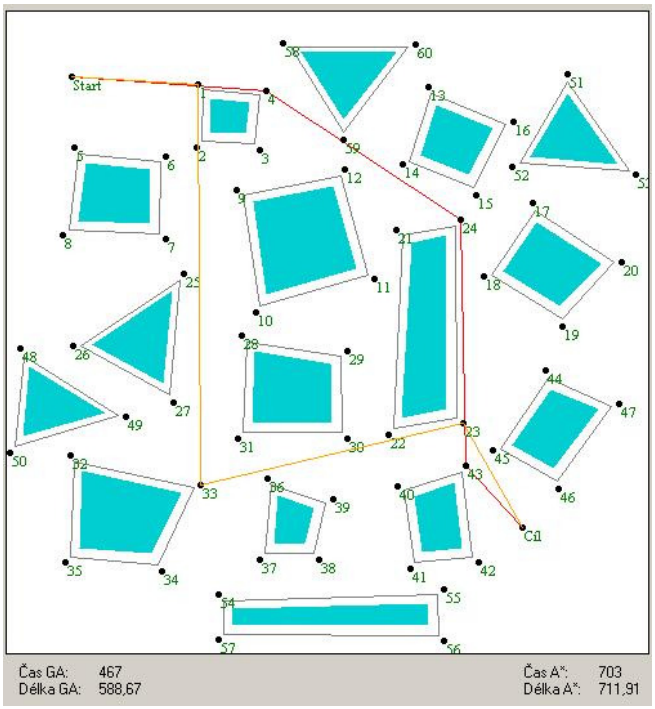
Obr.10 Záložka "Kontrola GA"

Na záložce Kontrola GA obr.10 lze zadat počet chromozomů, které se náhodně vygenerují pro první populaci a počet generací na jedno stisknutí tlačítka Evoluj.

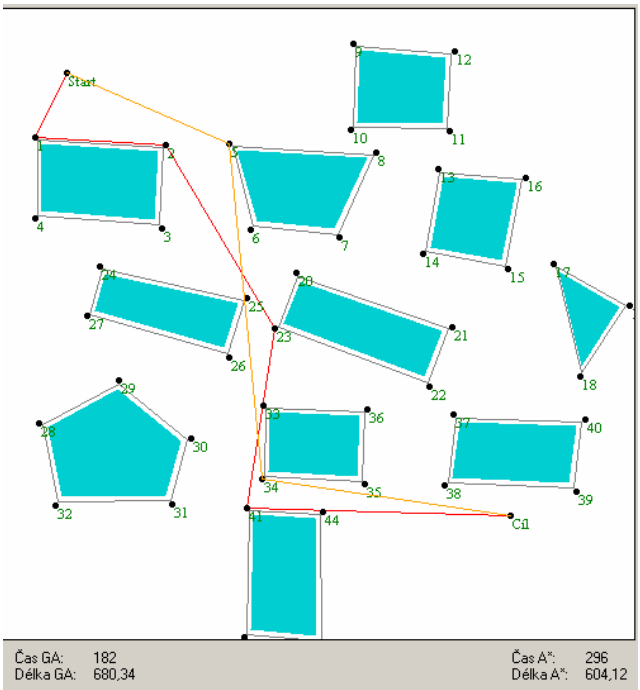
Poslední záložka A* vytvoří cestu pomocí dříve popsaného algoritmu. Tato cesta se zobrazuje žlutou barvou, výsledné cesty pomocí GA červenou a jednotlivé chromozomy během výpočtu odstínem šedé podle fitness. Pod oknem prostředí robota jsou uvedeny jednotlivé dosažené časy v milisekundách a délka označené cesty. Níže je posuvník, který umožňuje pohybovat se mezi generacemi a názorně pozorovat, které chromozomy zaniknou, zmutují nebo se zkříží. Poslední okno slouží k označení cesty z nejmenším fitness, případně označení a zvýraznění kterékoli cesty během celého výpočtu.

5 Popis a výsledky experimentů

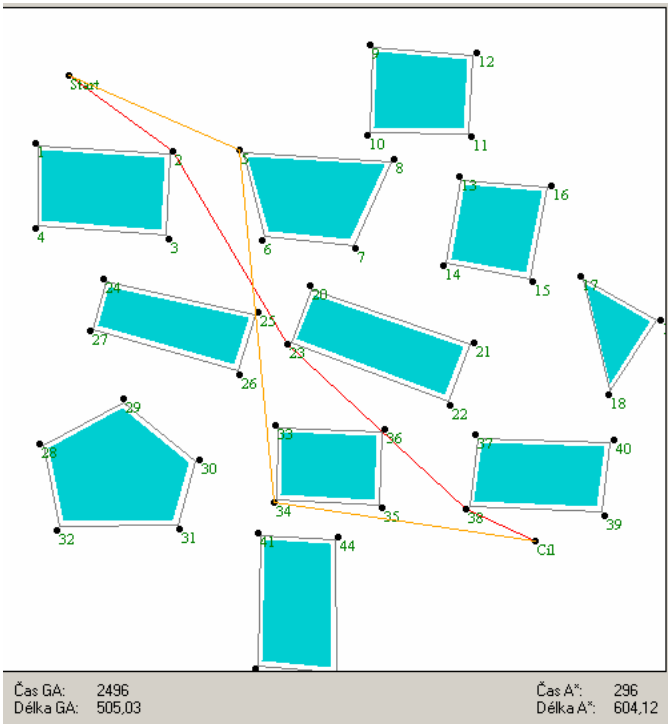
Na následujících obrázcích jsou porovnávány jednotlivé nalezené cesty v různých prostředích. Červené čáry jsou řešením vyhledání cesty pomocí genetických algoritmů (při použití vrcholů jednotlivých překážek nebo náhodných bodů), žluté pak pomocí algoritmu A*. Jednotlivé evoluce jsou zastaveny buď v okamžiku nalezení první bezkolizní cesty nebo po nalezení relativně nejkratší cesty.



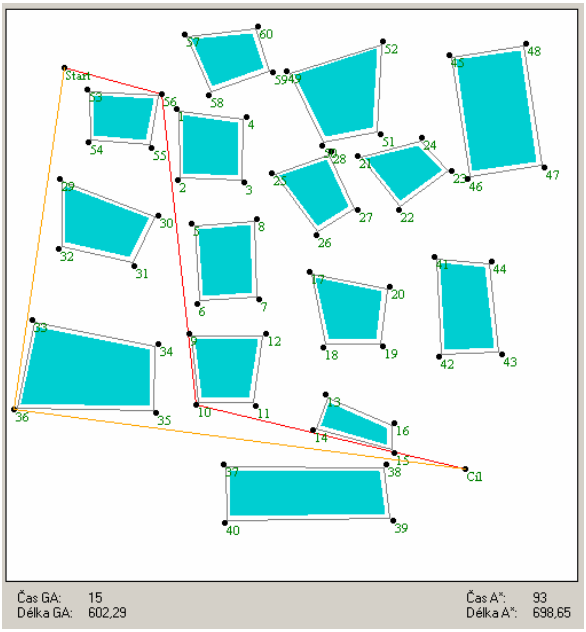
Obr. 11 Prostředí s 57 vrcholy, GA-vrcholové body



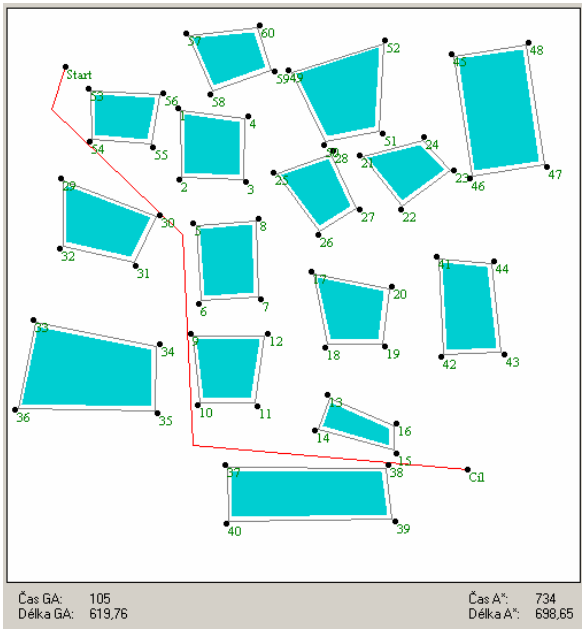
Obr. 12 Srovnání časů a délek GA a A* při nalezení první přípustné cesty



Obr.13 Srovnání časů a délek GA a A* při nalezení zřejmě nejkratší cesty



Vrcholové body

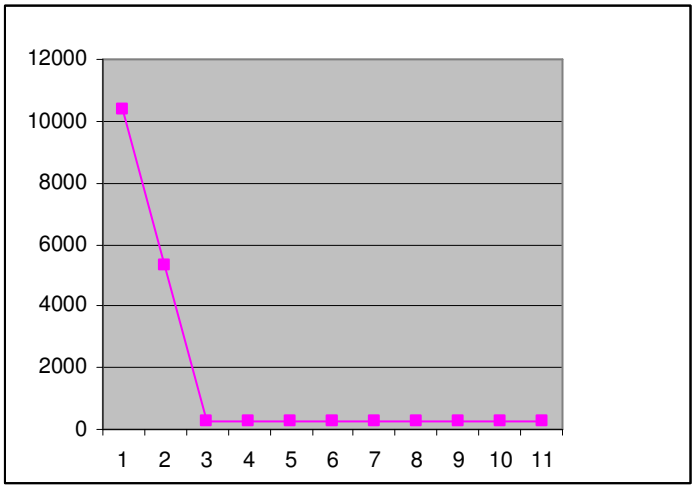


Náhodné body

Obr. 14 Srovnání časů a délek GA při nalezení první přípustné cesty

Chromozomy vytvořené pomocí vrcholových bodů se jeví jako kratší a rychlejší než pomocí náhodně generovaných bodů už z toho důvodu, že změny směru nastávají právě v ideálních místech již v počátečních populacích na rozdíl od náhodných bodů, kde se k tomu algoritmus musí teprve "dopracovat".

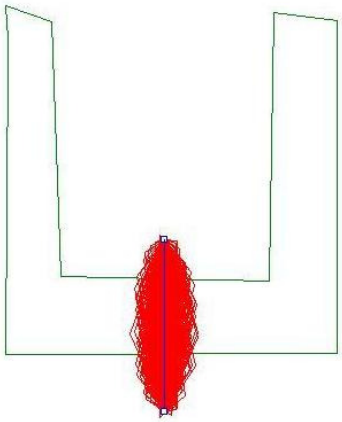
Počet generací	fitness
100	10355
200	5298
300	275
400	275
500	263
600	262
700	260
800	250
900	250
1000	250
1100	249



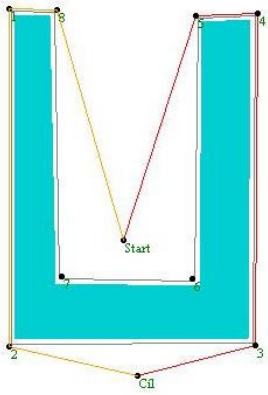
Obr.15 Typický vývoj funkce fitness v závislosti na počtu generací a její graf

Zastavení běhu programu je dáno počtem nastavených generací, kdy uživatel může sám určit jestli výsledná cesta je dostatečně kvalitní nebo je zapotřebí k vylepšení cesty dalších generací.

Na následujících obrázcích je patrná výhoda vícebodové mutace nastavitelné uživatelem



Obr.16



Obr. 17

Na obr. 19 je situace, kdy program použitý v práci [4] nenalezl cestu pomocí GA bez obcházení a všechny červené cesty procházejí překážkou. Možným důvodem nenalezení přípustné cesty je pouze dvoubodová mutace, takže nelze nalézt cestu pomocí dvou nových bodů, aby neprocházela překážkou. Tento problém je v práci [4] vyřešen pomocí algoritmu obcházení překážek. Podobná situace je vyřešena na obr. 20 pomocí vícebodové mutace(červená čára).

7 Závěr

V této práci jsem se zabýval plánováním cesty pomocí genetických algoritmů a algoritmu A* v prostředí polygonálních překážek. Vytvořený program při běžných prostředích najde cestu pomocí algoritmu A*, pomocí GA vznikají problémy pouze u složitějších prostředí typu spirálovitého bludiště. Porovnávané časy jsou u GA jen orientační, protože je zadáván počet generací, takže přijatelná cesta může být generována již dříve. Časy cesty jsou významně ovlivněny náhodně generovanou počáteční generací, takže mohou být vyšší ve stejném prostředí. Vliv má i nastavená hodnota pravděpodobnosti mutace, kdy u běžného prostředí je lépe ponechat šanci mutace nižší, u složitějších vyšší a po nalezení první bezkolizní cesty pravděpodobnost mutace snížit a pokračovat v evoluci do nalezení nejvhodnější cesty. Naproti tomu algoritmus A* si poradí ve všech schůdných prostředích, hledá však první přijatelnou cestu.

Nespornou výhodou genetických algoritmů je také jejich adaptabilita na změnu překážek, případně pozice startu a cíle, kdy lze použít již vygenerované populace jako počáteční. Mají zároveň tolik možností nastavení navzájem propojených faktorů, že při podrobném otestování všech kombinací je možné dojít k cíli všemi prostředími.

8 Seznam použité literatury

[1]HODÁL, J.: Využití případového usuzování pro navigaci robotu, VUT FSI ,2007
[2]MEYER, J., FILLIAT, D.: Map based navigation in mobile robots, 2003
[3]LOO, C.-K.: Mobile robot path planing using hybrid genetic algorithm and traversability vectors method, 2004
[4]GROSS, T.: Plánování cesty mobilního robota, VUT FSI, 2007
[5]ŠTĚPÁNEK, P.: Plánování dráhy robota pomocí GA, VUT FSI,2006
[6]KRČEK, P; DVOŘÁK, J.: Plánování cesty mobilního robota
[7]BŘEZINA, T.: Prostředky umělé inteligence, 2004
[8] http://en.wikipedia.org/