

MOBILE ROBOT PATH PLANNING BY MEANS OF GENETIC ALGORITHMS

P. Krček*, J. Dvořák*

Summary: *In this paper, we deal with mobile robot path planning in a two-dimensional continuous space in which known static polygonal obstacles are defined. The aim of the path planning is searching for a path from a start to a goal position without collisions with known obstacles minimizing such criteria as length, difficulty and risk of the path. We investigate possibilities of using genetic algorithms for solving this problem and describe various problem specific genetic operators and fitness functions. We study also an ability of proposed algorithm to adapt a previous solution to changes of start or goal position and changes of environment. Results of computational experiments are presented.*

1. Úvod

Plánování cesty představuje významnou součást problematiky navigace mobilních robotů. Úkolem plánování cesty robotu je nalezení cesty z počáteční do koncové pozice bez kolize se známými statickými překážkami minimalizující ohodnocení cesty určené zejména délkou cesty a případně dalšími aspekty, jako je obtížnost a riziko cesty. délku a obtížnost cesty. Prostředí robotu může být reprezentováno jako spojitý prostor nebo může být nějakým způsobem diskretizováno (např. pomocí grafových struktur nebo pravoúhlé mřížky). V tomto článku se zabýváme plánováním cesty autonomního mobilního robotu ve spojitém prostoru s překážkami polygonálního tvaru.

Pro plánování cesty robotu bylo vyvinuto velké množství metod. Jejich přehled je možno najít např. v práci Meyer & Filliat (2003) a v knize LaValle (2006). Pro plánování cesty lze použít např. metodu *potenciálů*, metodu *pravděpodobnostní cestovní mapy*, algoritmy *rychle mapujících náhodných stromů*, metody založené na *grafech viditelnosti* a *Voronioových diagramech*.

Plánování cesty představuje obecně složitý optimalizační problém. K řešení složitých optimalizačních algoritmů se velmi často používají heuristické evoluční metody, k nimž patří také *genetické algoritmy* (GA). Je proto přirozené, že se tyto algoritmy aplikují i na problém plánování cesty robotu (viz např. Burchardt & Salomon 2006, Gemeinder & Gerke 2003, Homaifar et. al. 2001, Hu & Yang 2004, Soltani et al. 2002, Zheng et al. 2004). Nedostatkem raných aplikací genetických algoritmů na tuto problematiku bylo použití klasických GA s binární reprezentací chromozomu pevné délky a pouze dvěma základními genetickými operátory (pro křížení a mutaci). Pozdější úspěšnější aplikace modifikovaly klasický GA zavedením nebinární reprezentace, proměnné délky chromozomu a především problémově

* Ing. Petr Krček, RNDr. Jiří Dvořák, CSc.: Ústav automatizace a informatiky, Fakulta strojního inženýrství VUT v Brně; Technická 2896/2; 616 69 Brno; tel.: +420.541 143 351; e-mail: krcek@fme.vutbr.cz

specifických genetických operátorů, doplněných případně nějakými heuristickými znalostmi. Většina aplikací GA používá pro reprezentaci prostředí robota dvoudimenzionální pravoúhlou mřížku, přičemž některé na základě toho omezují směry pohybu robota na horizontální, vertikální a případně diagonální. Příkladem GA pro spojitě prostředí je algoritmus popsáný v práci Homafair et. al. (2001). Při řešení problémů velkého rozsahu GA překonávají limity jiných plánovacích metod, pokud jde o dobu výpočtu nebo potřebu na paměť. GA jsou schopné pokrýt rozsáhlý prostor prohledávání při relativně malých nárocích výpočetní zdroje. Další předností GA je schopnost adaptace nalezeného řešení na spojitě se měnící dynamické prostředí a generovat řešení v reálném čase. Tuto schopnost demonstrují např. práce Burchardt & Salomon (2006) a Zheng et al. (2004). Nedostatkem GA je, že nezaručují nalezení globálně optimálního řešení.

2. Genetické algoritmy a plánování cesty robota

Práce Homafair et al. (2001) popisuje evoluční plánovací systém pracující ve spojitěm prostoru a používající kódování pomocí reálných čísel. Základní tvar chromozomu se skládá ze tří genů obsahujících souřadnice startovního, cílového a vnitřního uzlu cesty (takto vypadají chromozomy po vygenerování počáteční populace, po křížení a po mutaci. Operace křížení a mutace pracují s vnitřním genem (jádretem). Souřadnice jádra potomka se získají tak, že se náhodně při použití rovnoměrného rozdělení stanoví hodnoty z intervalů určených dvojicemi odpovídajících souřadnic jader rodičů a k nim se pak připočítají náhodně vygenerované hodnoty z normálního rozdělení. Mutace se pak aplikuje na první nebo druhou souřadnici jádra potomka. Na základní tvar chromozomu se pak aplikuje opravný operátor, který do chromozomu doplňuje opravné uzly tak, aby nedocházelo ke kolizím cesty s překážkami. Cesta je přitom tvořena úsečkami spojujícími sousední uzly chromozomu. Pro výběr jedinců ke křížení je použita fuzzy turnajová selekce, která je založena na těchto kritériích: délka cesty jako součet délek jednotlivých lineárních segmentů, součet změn ve směrnících jednotlivých úseků a průměrná změna směrnice cesty.

V článku Soltani et al. (2002) je provedeno srovnání genetického algoritmu s Dijkstrovým algoritmem a s algoritmem A*, které pracují nad dvourozměrnou mřížkou. Genetický algoritmus dosahuje lepších výsledků jak v kvalitě cesty (to je dáno také tím, že počet směrů pohybu robota není omezen na osm jako u zbývajících dvou algoritmů), tak i v době výpočtu. Chromozom má proměnnou délku a je tvořen startem, cílem a vnitřními uzly cesty, v nichž robot mění směr pohybu. Každý uzel je určen dvojicí celočíselných souřadnic. Genetický algoritmus pracuje pouze se základními operátory, žádné speciální operátory zde nejsou zavedeny. Funkce fitness je konstruována pomocí délky cesty, bezpečnosti (buněk mřížky jsou přiřazeny různé hodnoty bezpečnosti) a viditelnosti.

Další práci, ve které je metoda plánování navržena nad dvourozměrnou mřížkou, je Gemeinder & Gerke (2003). Chromozom má proměnnou délku a reprezentuje cestu určenou jako posloupnost sousedících buněk mřížky. Počet možných směrů pohybu je omezen na osm. Křížení je jednobodové, přičemž bod křížení se náhodně vybere jako průsečík rodičovských cest. Mutace se provádí tak, že se náhodně vyberou dva bod cesty a segment cesty mezi těmito body se nahradí dvěma lineárními segmenty spojujícími uvedené body s náhodně vybraným bodem ležícím mimo cestu. Součástí těchto dvou operátorů jsou také opravné akce, které z cesty odstraňují nežádoucí jevy jako např. uzavřené smyčky. Dalšími operátory jsou operátor napínání cesty a operátor obcházení překážek.

Prostředím robotu v práci Hu&Yang (2004) je dvojrozměrná mřížka, jejíž buňkám jsou přiřazeny celočíselné indexy. Cesta je složena z přímých segmentů spojujících buňky mřížky. Do chromozomu je cesta kódována jako posloupnost indexů buněk, přičemž první představuje start, poslední cíl a vnitřní indexy představují krajní body segmentů. Počet vnitřních indexů není pevný. Autoři zavádějí šest problémově specifických operátorů: křížení, mutaci, opravu uzlu, opravu segmentu, mazání uzlu a zlepšení cesty. Hodnotící funkce vychází ze součtu délek segmentů a z délek úseků, kterými segmenty protínají překážky.

Burchardt & Salomon (2006) plánují pomocí GA cestu pro všesměrový robot, který je určen pro týmový fotbalový zápas. Cesta, která je reprezentována chromozomem, je plánována v dynamickém prostředí, kde překážky tvoří ostatní roboty. Autoři používají operátory křížení a mutace. Chromozom je proměnné délky a jeho geny představují celočíselné souřadnice vnitřních uzlových bodů cesty. Algoritmus je optimalizován pro běh v mikrokontroléru, kde v režimu on-line neustále plánuje cestu až do dosažení cíle. Fitness funkce zohledňuje délku cesty a pokutu závislou na míře kolize.

Zheng et al. (2004) řeší problém koordinovaného plánování dráhy pro skupinu bezpilotních letadel. Každý chromozom koresponduje s dráhou letu jednoho letadla. Každý uzel chromozomu obsahuje tři souřadnice průsečíku lineárních segmentů, ze kterých je dráha složena, stavovou proměnnou, která určuje, zda je proveditelný průlet průsečíkem a následujícím segmentem a odkaz, který ukazuje na další uzel téhož chromozomu a definuje tak v podstatě následující segment. Autoři používají operátor křížení a pro tuto úlohu speciálně navržených šest operátorů mutace (změna souřadnic uzlu, vložení uzlu, rušení uzlu, výměna uzlu, vyhlazování cesty a speciální operátor změny předposledního uzlu s cílem dosáhnout vhodný úhel pro přistání). Hodnota funkce fitness vychází z hodnotící funkce, která je závislá na délkách segmentů, průměrných nadmořských výškách segmentů a mírách pronikání segmentů do nebezpečných oblastí.

3. Návrh systému pro plánování cesty pomocí genetických algoritmů

Prostředí se uvažuje jako dvourozměrný spojitý prostor, v němž se vyskytují polygonální překážky popsané svými vrcholy. Předpokládáme holonomický robot, který je pro jednoduchost reprezentován jako bod v prostoru. Pro uvažování rozměrů holonomického robotu by bylo nutné použít metodu zvětšení překážek, nebo upravit použitou metodu rozpoznávání kolizí.

Chromozom vyjadřuje cestu v prostoru a jeho geny představují uzlové body, ve kterých cesta mění svůj směr. Každý gen je přitom tvořen dvojicí souřadnic x a y . U všech chromozomů platí, že počáteční gen reprezentuje pozici startu a poslední gen pozici cíle. Cesta je tudíž tvořena úsečkami spojujícími sousedící geny (uzly) chromozomu. Chromozom má proměnlivou délku v závislosti na složitosti cesty.

Počáteční populace se získá náhodným generováním, avšak do chromozomů jsou generovány pouze takové uzly, které leží mimo překážky. Délka každého chromozomu se určí jako součet dané minimální velikosti a náhodně zvoleného čísla z daného rozsahu. Velikost počáteční populace je dána a zůstává stejná i v následujících generacích.

Jako operátor výběru rodičů pro křížení je použita binární turnajová selekce. Z populace jsou náhodně vybrány dva různé chromozomy a z nich se jako rodič vybere chromozom s větší hodnotou fitness. Totéž se opakuje pro druhého rodiče. Celkem se vybere

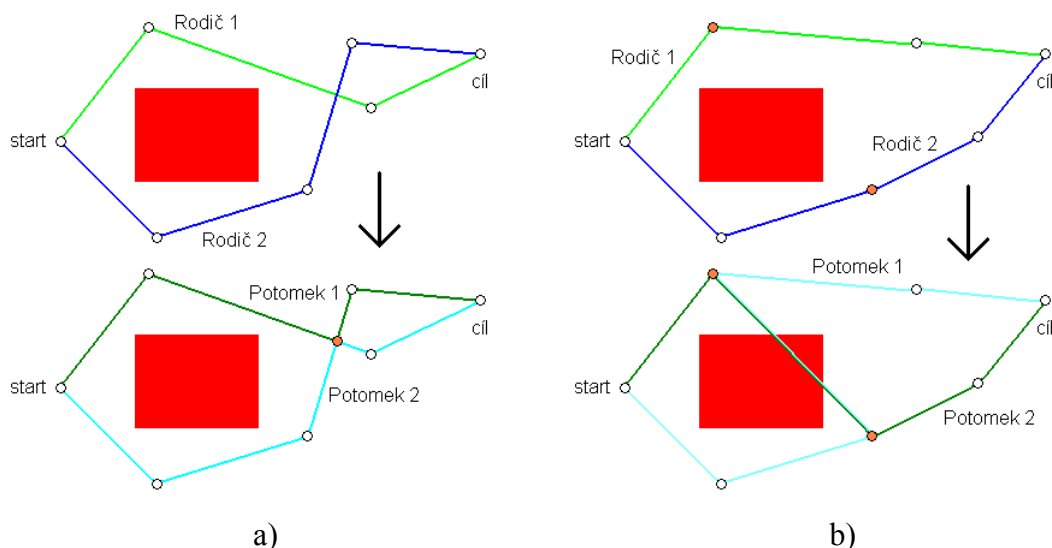
tolik rodičů, kolik požadujeme potomků v nové generaci. S ohledem na to, že výsledkem křížení jsou vždy dva potomci, je nutné, aby to byla sudá hodnota.

Hodnota fitness zohledňuje počet průchodů překážkami, délku cesty jako součet euklidovských vzdáleností dvojic po sobě následujících genů a počet uzlů cesty. Hodnota fitness i -tého chromozomu je určena následující fitness funkcí

$$f(i) = k_1 \frac{n_i - m_i - 1}{n_i - 1} + k_2 \frac{\sqrt{(x_{i,1} - x_{i,n_i})^2 + (y_{i,1} - y_{i,n_i})^2}}{\sum_{j=1}^{n_i-1} \sqrt{(x_{i,j} - x_{i,j+1})^2 + (y_{i,j} - y_{i,j+1})^2}} + k_3 \frac{v - n_i}{v - 2},$$

kde n_i je počet genů i -tého chromozomu, m_i je počet kolizních segmentů i -tého chromozomu, $x_{i,j}, y_{i,j}$ jsou souřadnice j -tého genu i -tého chromozomu, v je maximální počet genů v chromozomu a k_1, k_2, k_3 jsou kladné váhové koeficienty.

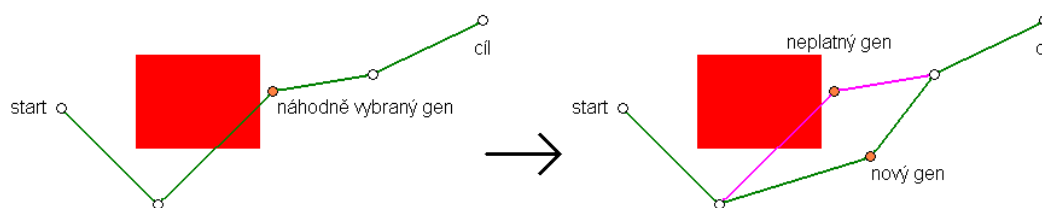
Křížení se pro každý pár rodičů provádí s danou, obvykle velmi vysokou pravděpodobností. Pokud tedy není operátor křížení pro nějaký pár vynechán, nejprve se určí všechny geometrické průsečíky cest těchto dvou chromozomů. Je-li počet těchto průsečíků větší než nula, pak se náhodně vybere jeden z nich. Tento průsečík slouží jako dělicí místo, které rozdělí chromozom na dvě části. Potomci potom vzniknou tak, že se spojí prvá část chromozomu prvního rodiče se druhou částí druhého rodiče a prvá část chromozomu druhého rodiče se druhou částí prvního rodiče. Mezi obě části se do potomků vloží nový gen, reprezentující vybraný průsečík (obr. 1a). Byl-li však počet průsečíků roven nule, pak se dělicí místo v obou rodičovských chromozomech určí jako náhodně vybraný gen. Vkládání nového genu se v tomto případě již neprovádí (obr. 1b). V obou případech se jedná se o variantu jednobodového křížení, operátor je možné však snadno modifikovat pro křížení vícebodové.



Obr. 1. Princip operátoru křížení

Operátor mutace se na rozdíl od operátoru křížení provádí již na potomcích a často s velmi nízkou pravděpodobností. Během mutace se náhodně vygeneruje nový gen, který leží

mimo uvažovanou cestu a mimo překážky. Dále se náhodně určí jeden gen z chromozomu a tento gen je novým genem nahrazen (obr. 2).



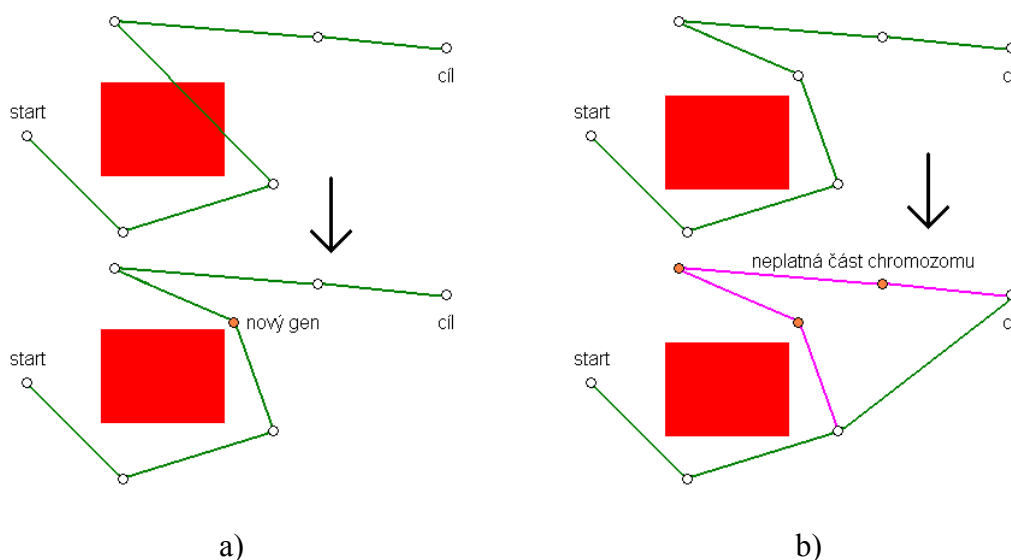
Obr. 2. Princip operátoru mutace

Po operátoru mutace se na potomcích provádí opravný operátor vložení genu, opět s danou pravděpodobností. Jestliže nějaký segment cesty je v kolizi s překážkou, vloží se nový gen mezi krajní geny tohoto segmentu tak, aby byla odstraněna kolize. Princip je zobrazen na obr. 3a.

Vhodným protikladem operátoru vložení genu je operátor odstranění genu. S danou pravděpodobností se tento operátor provádí jako poslední. Tento operátor zjednodušuje cestu a odstraňuje smyčky v cestě tím, že vyjme všechny nadbytečné geny v chromozomu. Princip je zobrazen na obr. 3b.

Změna populace je prováděna inkrementálně. Do nové populace se přidají všichni potomci, odstraní se případné duplicitní chromozomy a poté se provede seřazení jedinců v populaci podle hodnoty fitness. Dále se provede odstranění chromozomů s nejnižší hodnotou fitness tak, aby velikost populace odpovídala velikosti počáteční populace. Tím je zajištěno, že nejlepší chromozomy staré populace postoupí do populace nové.

Kritérium ukončení algoritmu není pevně stanoveno, vždy záleží na charakteru úlohy. Nejsme-li časově omezeni, je možné algoritmus zastavit až po daném počtu generací, nebo zastavit až když bude rozdíl dvou průměrných fitness hodnot celých, po sobě následujících populací menší než nějaká konstanta. Pro nalezení vhodného řešení však postačuje navržený algoritmus zastavit v okamžiku, kdy populace obsahuje několik přípustných jedinců.



Obr. 3. Principy operátorů vložení uzlu a odebrání uzlu

4. Experimentální ověření

Pro ověření a experimentování s navrženým genetickým algoritmem byl vytvořen programový systém. Pomocí následujících experimentů byla testována schopnost adaptace populace navrženého algoritmu. Jednak byla zkoumána možnost použití části již naučené populace z nějakého předchozího řešení jako část počáteční populace nové úlohy, a za druhé byla zkoumána možnost použití naučené populace aktuálního řešení pro přeplánování své cesty v důsledku výskytu náhodné překážky. Oba experimenty byly provedeny ve scéně 1000×1000 jednotek se zadanými překážkami. Parametry genetického algoritmu byly nastaveny experimentálně takto:

- počet chromozomů v populaci: 100
- počet potomků v jedné generaci: 24
- min. délka počátečního chromozomu: 3
- rozsah počátečního chromozomu: 0–14
- koeficienty funkce fitness: $k_1 = 10$, $k_2 = 1$, $k_3 = 1$
- pravděpodobnost operátoru křížení: 0,95
- pravděpodobnost operátoru mutace: 0,05
- pravděpodobnost operátoru přidání uzlu: 1
- pravděpodobnost operátoru odebrání uzlu: 1

Kritériem ukončení genetického algoritmu bylo ve všech případech nalezení dvaceti různých přípustných chromozomů (nalezené cestě odpovídá chromozom s největší fitness), nebo nalezení prvního přípustného chromozomu o dvou genech (přímá spojnice startu a cíle).

První experiment byl založen na následujících krocích:

1. Zvolíme vhodně start s i cíl c tak, aby předpokládaná naučená populace této úlohy pokrývala co možná největší oblast scény.
2. Spustíme navržený genetický algoritmus.
3. Po ukončení algoritmu určíme v populaci množinu B , kterou tvoří 30 chromozomů s nejlepším fitness.
4. Zvolíme velikost okolí startu s a cíle c .
5. V okolí startu s náhodně zvolíme nový start s' . V okolí cíle c zvolíme podobně cíl c' .
6. Vygenerujeme počáteční populaci, do které přidáme chromozomy z množiny B a nastavíme u nich aktuální start s' a cíl c' . Poté populaci seřadíme podle fitness a odebereme z ní nejhorší chromozomy tak, aby velikost populace odpovídala velikosti po počátečním vygenerování.
7. Spustíme GA s touto počáteční populací. Po ukončení algoritmu zaznameneáme počet generací, které byly během tohoto kroku vygenerovány.
8. Pro srovnání nyní spustíme GA s celou náhodně vygenerovanou počáteční populací. Po ukončení algoritmu si opět zaznameneáme počet generací, které byly během tohoto kroku vygenerovány.
9. Kroky 5. až 8. opakujeme pro 50 nových startů a cílů.
10. Kroky 4. až 9. opakujeme pro různé velikosti okolí.

Kroky 1. až 10. byly desetkrát opakovány pro různé s i c . Obr. 4 zobrazuje takto získané závislosti průměrných hodnot zaznamenaných počtů generací na velikosti okolí startu s a cíle c . Závislost GA 2 odpovídá údajům z kroku 7, závislost GA 1 údajům z kroku 8. Obr. 5 zobrazuje podobné závislosti, avšak buďto platí, že v okolí startu s je nyní vygenerován nový start s' a nový cíl c' leží mimo okolí cíle c , nebo naopak platí, že v okolí cíle c je nyní vygenerován nový cíl c' a nový start s' leží mimo okolí startu s .

Z tohoto experimentu plyne, že použití části již naučené populace z nějakého předchozího řešení jako část počáteční populace nové úlohy snižuje potřebný počet generací pro nalezení řešení, a tudíž snižuje časovou náročnost algoritmu. Významného snížení časové náročnosti lze však dosáhnout pouze v případě, že start i cíl leží v okolí původního startu i cíle. Navíc musí být velikosti okolí menší než 200 jednotek.

Druhý experiment byl založen na následujících krocích:

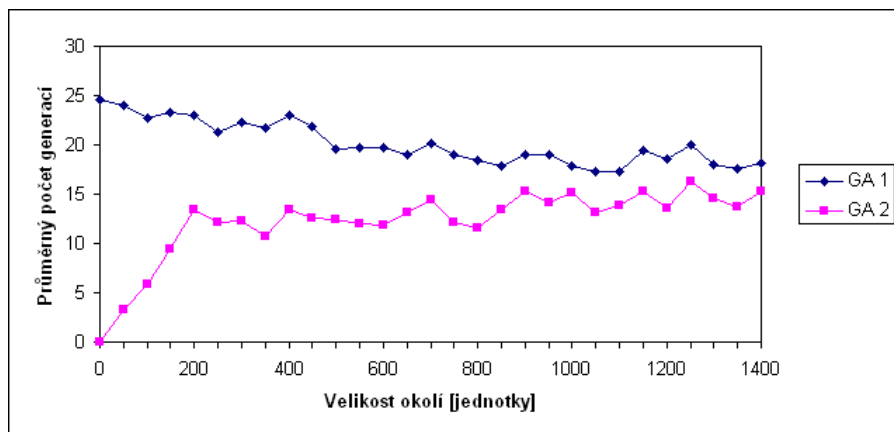
1. Náhodně vygenerujeme start i cíl.
2. Spustíme navržený genetický algoritmus.
3. Po ukončení algoritmu určíme v populaci množinu B , kterou tvoří 30 chromozomů s nejlepším fitness.
4. Náhodně vygenerujeme překážku tak, aby zasahovala do nalezené cesty.
5. Určíme nový start, který se nachází na nalezené cestě v místě před překážkou
6. Vygenerujeme počáteční populaci, do které přidáme chromozomy z množiny B a nastavíme u nich aktuální start s' . Poté populaci seřadíme podle fitness a odebereme z ní nejhorší chromozomy tak, aby velikost populace odpovídala velikosti po počátečním vygenerování.
7. Spustíme GA s touto počáteční populací. Po ukončení algoritmu zaznamenáme počet generací, které byly během tohoto kroku vygenerovány.
8. Pro srovnání nyní spustíme GA s celou náhodně vygenerovanou počáteční populací. Po ukončení algoritmu opět zaznamenáme počet generací, které byly během tohoto kroku vygenerovány.
9. Kroky 6. až 8. desetkrát opakujeme.
10. Kroky 4. až 8. opakujeme taktéž desetkrát.

V tabulce 1 jsou předloženy výsledky experimentu pro dvě cesty z obr. 6a a 7a. Označení metod je stejné jako v přecházejícím experimentu. Závislost GA 2 odpovídá údajům z kroku 7, závislost GA 1 údajům z kroku 8. Pro obě cesty jsou doplněny obrázky s přípustnými chromozomy z množiny B (obr. 6b, 7b). Na dalším obrázku jsou příklady nalezených cest uvažující dynamické překážky o velikosti 20×20 a 60×60 jednotek (obr. 8).

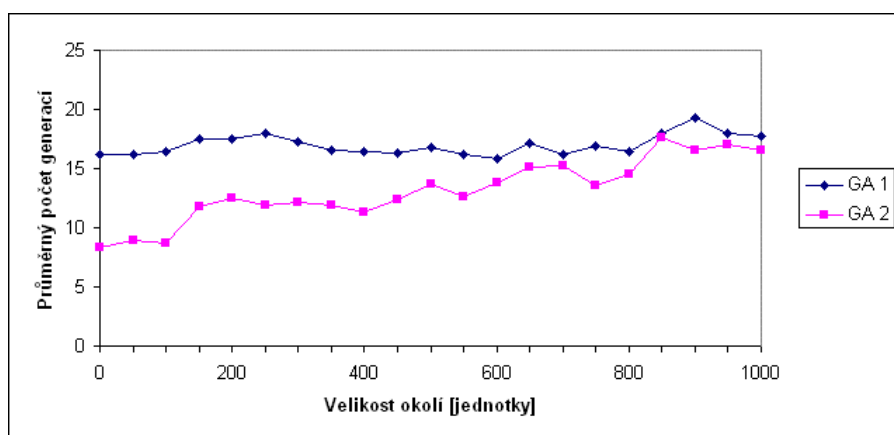
Tab. 1. Získané průměrné hodnoty počtu potřebných generací druhého experimentu

Původní cesta z obr.	Velikost náhodné překážky [jednotky]					
	20×20		40×40		60×60	
	GA 1	GA 2	GA 1	GA 2	GA 1	GA 2
6a	13, 878	5, 111	19, 922	10, 178	24, 733	23, 678
7a	40, 692	12, 813	42, 508	14, 205	53, 753	26, 606

I tento druhý experiment prokázal schopnost rychlé adaptace populace navrženého algoritmu. Při přeplánování cesty v důsledku výskytu náhodné překážky se jeví jako velmi vhodné začlenit do počáteční populace množinu B původního řešení. V některých případech je sice zlepšení počtu potřebných generací mizivé (cesta z obr. 6a, překážka 60×60), avšak v žádném z případů použití původní množiny B nezvětšilo počet potřebných generací a tím tedy nezhoršilo časovou náročnost algoritmu.



Obr. 4. Závislosti pro případ, že nový start i cíl leží v okolí původního startu i cíle



Obr. 5. Závislosti pro případ, že v okolí původního startu a cíle leží jen nový start nebo jen cíl

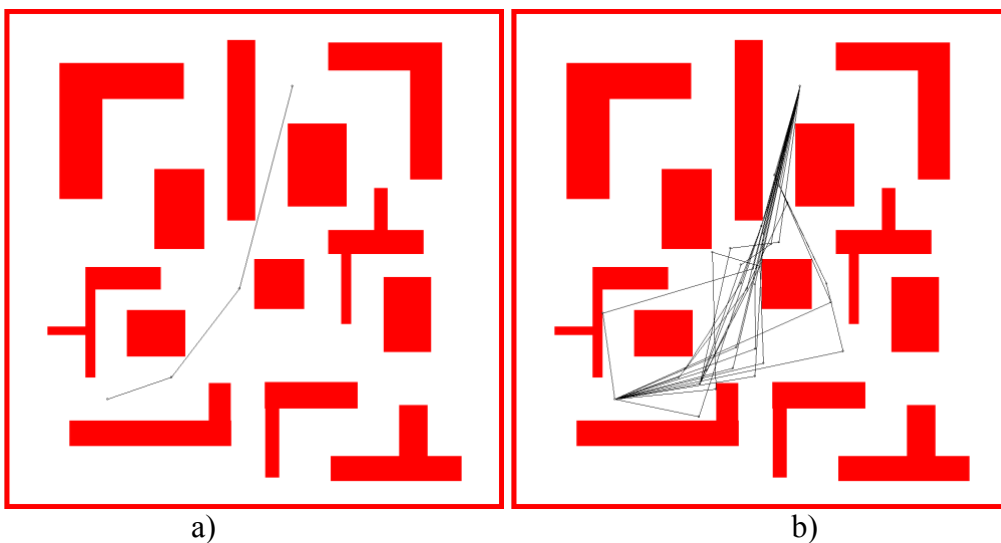
5. Závěr

Tento článek se zabývá plánováním cesty holonomického mobilního robotu ve dvou-rozměrném spojitém prostředí se zadanými polygonálními překážkami. V přehledové části článku, věnované plánování cesty pomocí GA, jsou uvedeny popisy publikovaných metod včetně používaných operátorů a funkcí fitness. V další části je představena navržená implementace systému pro plánování cesty pomocí GA. Popsané experimenty jsou věnovány zkoumání schopnosti rychlé adaptace populace navrženého algoritmu na změny zadání. Při znovupoužití vhodné části „naučené“ populace výsledky experimentů tuto schopnost potvrzují.

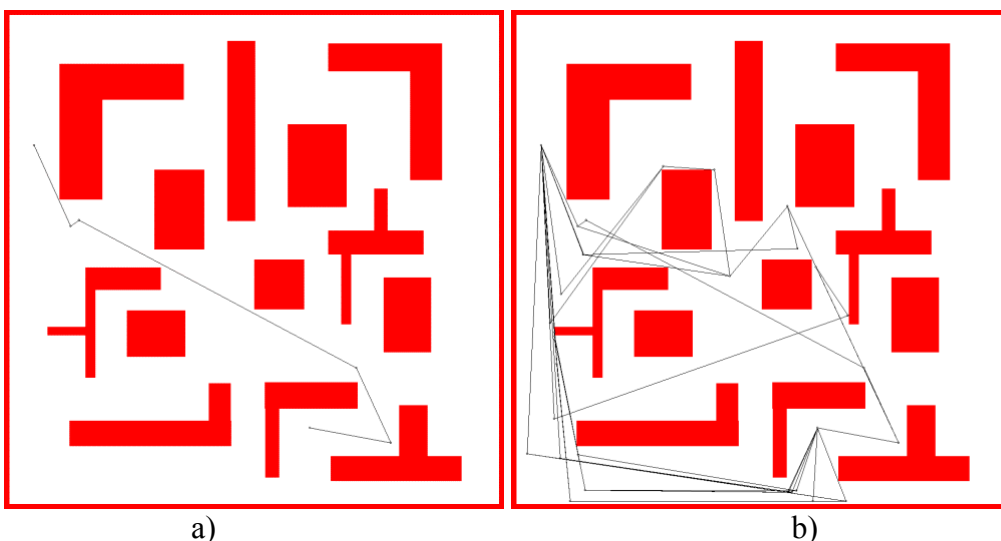
V dalším výzkumu se budeme věnovat dalším typům operátorů a tvarů funkcí fitness. Budeme rovněž zkoumat možnosti úpravy navrženého algoritmu pro plánování dráhy neholonomických robotů. Dále také předpokládáme, že „naučené“ části populací je možné uchovávat v bázi případů a vyhledávat je pomocí metody případového usuzování.

6. Poděkování

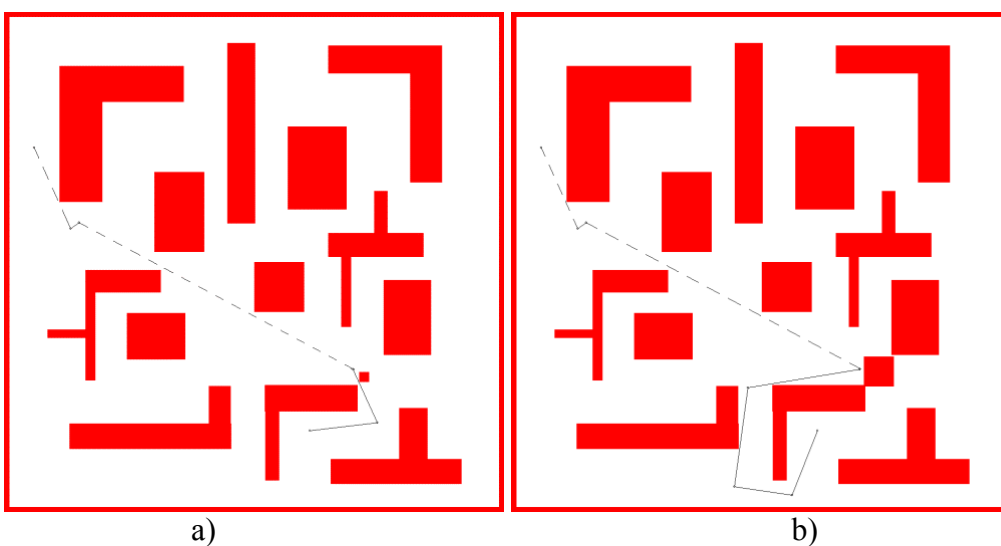
Tento článek byl zpracován v rámci vědecko-výzkumného záměru MSM 0021630518 "Simulační modelování mechatronických soustav".



Obr. 6. Nalezená počáteční cesta a přípustné chromozomy z její množiny B



Obr. 7. Nalezená počáteční cesta a přípustné chromozomy z její množiny B



Obr. 8 Příklady nalezených cest, při jejichž hledání byla použita množina B z obr. 7b

7. Literatura

- Burchardt, H. & Salomon, R. (2006) Implementation of Path Planning Using Genetic Algorithms on Mobile Robots. In *Proceedings of the IEEE World Congress on Computational Intelligence (WCCI 2006), Congress on Evolutionary Computation (CEC 2006)*, Vancouver, Canada, pp. 1831-1836.
- Gemeinder, M. & Gerke, M. (2003) GA-Based Path Planning for Mobile Robot Systems Employing an Active Search Algorithm. *Applied Soft Computing*, vol. 3, pp. 149-158.
- Homaifar, A., Tunstel, E., Dozier, G. & Battle, D. (2001) Genetic and Evolutionary Methods for Mobile Robot Control and Path Planning. In Ali Zillouchian and Mo Jamshidi (eds.). *Intelligent Control Systems Using Soft Computing Methodologies*, CRC Press LLC, Boca Raton.
- Hu, Y & Yang, S.X. (2004) A Knowledge Based Genetic Algorithm for Path Planning of a Mobile Robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, New Orleans, vol. 5, pp. 4350-4355.
- LaValle, S.M. (2006) *Planning Algorithms*. Cambridge University Press, <http://planning.cs.uiuc.edu/>
- Meyer, J.-A. & Filliat, D. (2003) Map-Based Navigation in Mobile Robots: II. A Review of Map-Learning and Path-Planning Strategies. *Cognitive Systems Research*, vol. 4, pp. 283-317.
- Soltani A. R., Tawfik H., Goulermas J. Y. & Fernando T. (2002) Path Planning in Construction Sites: Performance Evaluation of the Dijkstra, A*, and GA Search Algorithms. *Advanced Engineering Informatics*, vol. 16, pp. 291-303.
- Zheng, C., Ding, M., Zhou, C. & Li, L. (2004) Coevolving and Cooperating Path Planner for Multiple Unmanned Air Vehicles. *Engineering Applications of Artificial Intelligence*, vol. 17, issue 8, pp. 887-896.