VĚDECKÉ SPISY VYSOKÉHO UČENÍ TECHNICKÉHO V BRNĚ

Edice PhD Thesis, sv. 461

ISSN 1213-4198

Ing. Jaroslav Hodál

Využití případového usuzování pro navigaci robotu

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta strojního inženýrství
Ústav automatizace a informatiky

Ing. Jaroslav Hodál

VYUŽITÍ PŘÍPADOVÉHO USUZOVÁNÍ PRO NAVIGACI ROBOTU

USING CASE-BASED REASONING FOR ROBOT NAVIGATION

Zkrácená verze Ph.D. Thesis

Obor:	Inženýrská mechanika
Školitel:	RNDr. Jiří Dvořák, CSc.
Oponenti:	doc. Ing. Pavel Bělohoubek, CSc. pplk. doc. Ing. Alexandr Štefek doc. Ing. František V. Zbořil, CSc.
Datum obhajoby:	10. 12. 2007

KLÍČOVÁ SLOVA

Navigace robotu, plánování cesty, grafové algoritmy, případové usuzování, případový graf.

KEYWORDS

Robot navigation, path planning, graph algorithms, case-based reasoning, case-graph.

Rukopis disertační práce je uložen na oddělení vědy a výzkumu Fakulty strojního inženýrství VUT v Brně, Technická 2, 616 69 Brno.

© Jaroslav Hodál, 2008
ISBN 978-80-214-3645-9
ISSN 1213-4198

OBSAH

1 ÚVOD 5

2 NAVIGACE ROBOTU 5

2.1 Reprezentace prostředí 6
2.1.1 Topologické modely prostředí 6
2.1.2 Metrické modely prostředí 6
2.2 Rozdělení metod pro plánování cesty 7
2.3 Globální plánování cesty 7
2.3.1 Plánování pomocí map cest 8
2.3.2 Metoda pravděpodobnostní cestovní mapy 9
2.3.3 Plánování rychle mapujícími náhodnými stromy 9
2.3.4 Metody rozkladu prostředí do buněk 9
2.3.5 Metoda pole potenciálu 9
2.3.6 Plánování cesty genetickými algoritmy 10
2.3.7 Metody inspirované chováním hmyzích kolonií 10
2.3.8 Další metody plánování cesty robotu 10
2.4 Lokální řízení pohybu 11

3 PŘÍPADOVÉ USUZOVÁNÍ 11

3.1 Případová báze 12
3.2 Hybridní systémy 12
3.3 Údržba případové báze 12

4 VYUŽITÍ CBR PŘI NAVIGACI ROBOTU 13

4.1 CBR pro globální navigaci 13
4.2 CBR pro lokální navigaci 14
4.3 CBR v integrovaných systémech 14

5 CÍLE DISERTAČNÍ PRÁCE 15

6 NÁVRH PLÁNOVÁNÍ CESTY S VYUŽITÍM CBR 16

6.1 Reprezentace prostředí 16
6.2 Reprezentace cest 16
6.3 Případová báze 17
6.4 Plánování cesty bez použití CBR 17
6.5 CBR plánování cesty 18
6.6 Údržba případové báze 19

7 IMPLEMENTACE 19

7.1 Reprezentace případové báze 20

7.2 Vnitřní reprezentace prostředí _____ 20
7.3 Metody plánování cesty bez CBR _____ 21
7.4 Plánování cesty případovým usuzováním _____ 21
7.5 Přidávání cest do případového grafu _____ 22
7.6 Údržba případové báze _____ 23
8 POPIS A VYHODNOCENÍ EXPERIMENTŮ 23
8.1 Doba CBR plánování v závislosti na velikosti případového grafu 23
8.2 Test četnosti výběru metody plánování _____ 24
8.3 Porovnání Dob plánování _____ 25
8.4 Porovnání kvality naplánovaných cest _____ 25
9 ZÁVĚR 26
10 VÝBĚR Z POUŽITÉ LITERATURY 28
11 PUBLIKACE AUTORA K TÉMATU DISERTACE 30
AUTOROVO CV 31
ABSTRACT 32

1 ÚVOD

Jedním z typických úkolů, kterými se robotika zabývá, je navigace autonomních mobilních robotů. V situaci, kdy navigace probíhá v dynamickém nebo v částečně či úplně neznámém prostředí, se tato úloha značně komplikuje. Vyžaduje shromažďování informací o prostředí, v němž se robot pohybuje, zaznamenávání změn tohoto prostředí a následné přizpůsobování se zjištěným změnám. Vzrůstá tedy potřeba robotů, které by dokázaly reagovat na změny v prostředí, aniž by k tomu potřebovaly zásahy lidského operátora. Při řešení tohoto typu navigačních úloh lze uvažovat o použití metod případového usuzování jakožto zvláštního případu strojového učení

Stručně lze říci, že CBR systémy řeší nové problémy tak, že se snaží adaptovat úspěšná řešení podobných problémů řešených již v minulosti na podmínky nového problému. Z toho je také patrné, že CBR systémy pro svoji práci uchovávají předchozí úlohy i jejich řešení, a to ve formě tzv. případů. Případ jako takový sestává z jakési situace, se kterou se systém již setkal, a z řešení této situace, které systém použil. Uchováváním předchozích zkušeností pro potřeby dalšího usuzování dochází k učení se z minulých situací. Systém se tak stává efektivnější a kompetentnější pro rozhodování.

Výhody plynoucí z použití případového usuzování budou nejvýraznější při aplikaci na roboty, které se pohybují v méně proměnlivém prostředí a často opakují podobné úkoly. V takových podmínkách však pracují průmyslové roboty asi nejčastěji.

Výsledky dosažené v této práci by měly být východiskem pro začlenění metod případového usuzování do navigačních metod robotů.

Tato práce vznikala postupně v rámci vědecko-výzkumných záměrů MSM 0021630518 "Simulační modelování mechatronických soustav", MSM 262100024 „Výzkum a vývoj mechatronických soustav“ a MSM 261100009 „Netradiční metody studia komplexních a neurčitých systémů“.

2 NAVIGACE ROBOTU

Cílem navigace autonomního mobilního robotu je nalezení a absolvování co možná nejkratší a zároveň co nejméně náročné cesty mezi dvěma zadanými body v daném prostředí, aniž by došlo ke kolizím s překážkami, které se v tomto prostředí mohou vyskytovat. Můžeme rozlišit dvě hlavní úrovně navigace: globální navigaci a lokální navigaci. Těžištěm globální navigace je starost o absolutní pozici v prostředí a přesuny mezi zadanými místy. Lokální navigace zjišťuje relativní pozice vůči okolním objektům a řeší správnou interakci s nimi.

2.1 REPREZENTACE PROSTŘEDÍ

Za posledních třicet let výzkumů v oblasti plánování cest bylo navrženo množství různých algoritmů (viz např. Latombe 1991, LaValle 2006). Výběr vhodného algoritmu ale většinou závisí na modelovaném prostředí. Modely prostředí jsou obvykle popsány pomocí mapy. Mapová reprezentace je přirozeným a také nejčastěji využívaným způsobem popisu, neboť vychází přímočaře z intuitivního vnímání problému. Mapy lze rozdělit do dvou hlavních kategorií: *topologické* (trasové) a *metrické* (mřížkové, spojité).

2.1.1 Topologické modely prostředí

Topologické modely prostředí představující analytický popis prostředí bývají reprezentovány různými grafy. Vrcholy grafů reprezentují určitá místa v prostředí a hrany pak vztahy mezi nimi. Do této kategorie patří např. *grafy viditelnosti*, *grafy tečen* nebo *Voronoiovy diagramy*. Pro nalezení nejkratší (nejlevnější) cesty v grafu lze použít množství algoritmů. Grafové plánování cest bývá obvykle založeno na různých modifikacích *Dijkstrova algoritmu*. Obecně lze ale využít i další algoritmy hledání cest v grafech.

2.1.2 Metrické modely prostředí

Mřížková mapa je množina polí (buněk), o kterých víme, zda jsou či nejsou průchozí. Pokud je mapa plochá, pak se často používají dvourozměrné obdélníkové sítě. Pracovat lze ale i s jinými tvary buněk. V každém případě je nutné zadané prostředí pokrýt vhodnou sítí a tím ho rozložit do buněk. Buňky jsou většinou konstruovány tak, aby v celé své ploše obsahovaly vždy jen jediný typ prostředí. U jednodušších modelů prostředí tedy mohou obsahovat buď volný prostor nebo překážku. Složitější modely obvykle přidávají buňkám další informace (typ terénu, počet použití, četnost výskytu neznámých překážek, aj.).

Pro každou buňku jsou také jednoznačně určeny buňky sousední (jejich hranice mají alespoň jeden společný bod). Cestu pak tvoří posloupnost buněk, přičemž přes každou z nich je definována dílčí cesta. Metody rozkladu prostředí do buněk lze rozdělit do dvou skupin: metody pro *exaktní rozklad* a metody pro *aproximativní rozklad*.

Exaktní rozklad do buněk rozděluje pouze prostor neobsazený překážkami do nepřekrývajících se buněk jednoduchého tvaru. Jednoduchý tvar buněk se používá kvůli snazšímu výpočtu přímé dílčí cesty přes tuto buňku. Dílčí cesty začínají a končí v bodech ležících na společné hranici buňky a jejích sousedů. S každým sousedem (za souseda není považována překážka) má buňka jeden takovýto spojovací průjezdní bod. Aby se poloha spojovacích bodů dala snadno vypočítat, používají se co nejjednodušší tvary hranic. Průjezdní cestou je spojena každá dvojice spojovacích bodů v buňce. Překážky žádné buňky neobsahují.

Aproximativní rozklady rozloží do buněk celé prostředí včetně překážek. Každé buňce se stanoví typ prostředí stejný v celé buňce. Pokud tedy do prostoru buňky

zasahuje část některé překážky, rozšíří se překážka přes celou buňku. V případě, že budou mít skutečné překážky obecné tvary, může dojít k jejich aproximaci (konkrétně rozšíření).

Nejběžnější aproximativní rozklad je proložení prostředí pravoúhlou mřížkou. Velikost buněk se určuje podle tvaru překážek nebo rozměrů robotu. Cesta je pak tvořena posloupností středů navzájem sousedících buněk. Obvykle se předpokládá, že ke změnám směru jízdy může docházet právě jen v těchto středových bodech.

Různými reprezentacemi prostředí pomocí map cest a metod rozkladu do buněk se podrobně zabývá práce (Meyer 2003). Nejpřesnějším, ale také výpočetně nejnáročnějším metrickým popisem prostředí je spojitý prostor.

2.2 ROZDĚLENÍ METOD PRO PLÁNOVÁNÍ CESTY

Pro plánování cesty autonomních mobilních robotů existuje velké množství postupů, které je možné rozdělit do 3 kategorií: analytické metody (např. dynamické programování, síťová analýza aj.), konekcionistické modely (neuronové sítě a učící se systémy) a další metody umělé inteligence.

Pro vlastní plánování cesty se historicky vyvinuly dva základní přístupy: algoritmy reaktivní, zpětnovazební, s přímou odezvou (*reactive architecture*) a algoritmy využívající symbolické mapy prostředí (*deliberative architecture*). Zatímco reaktivní algoritmy nezávisejí na reprezentaci prostředí a poskytují odezvu v reálném čase díky jednoduchým výpočtům, deliberativní algoritmy naopak na volbě reprezentace prostředí závisejí značně a obvykle mají pomalejší odezvu s proměnlivou dobou zpoždění. Deliberativní algoritmy však představují vyšší úroveň inteligence, narozdíl od nižší úrovně algoritmů reaktivních (Arkin 1998).

Z tohoto hlediska lze metody pro řízení pohybu mobilních robotů obecně také rozdělit do dalších dvou kategorií: *globální* plánování cesty a *lokální* řízení pohybu.

2.3 GLOBÁLNÍ PLÁNOVÁNÍ CESTY

Pro mobilní autonomní roboty je typickou formou globálního plánování hledání cesty mezi dvěma body ve dvoudimenzionálním prostoru. S určitými úpravami se ale některé z těchto metod používají i pro roboty manipulační.

Globální plánování se využívá především v systémech s bohatými informacemi o prostředí. Tyto informace jsou použity ke konstrukci cest (nebo jejich částí), na nichž se nevyskytují známé překážky a minimalizuje se také výskyt překážek neznámých. Obvykle využívají nějakou hodnotící funkci a hledání nejlepší cesty se pak řeší hledáním optima této funkce.

Metody globálního plánování mají také své nevýhody. Obvykle vyžadují velmi přesný model prostředí, který v mnoha aplikacích nelze vyhotovit nebo je obtížné ho udržovat a doplňovat. Navíc je nutné počítat i s překážkami nepředvídatelnými a předem neznámými.

Nebudeme-li předpokládat možný výskyt neznámých překážek (dopravní prostředky, osoby), je hlavním přínosem globálních plánů cest ověření, zda je možné

dosáhnout zadaného cíle z místa startu. Existence globálně naplánované cesty nemůže s jistotou garantovat, zda bude použitelná, ani zda v danou chvíli bude nějaká cesta existovat.

2.3.1 Plánování pomocí map cest

Topologické reprezentace prostředí (mapy cest, grafy) lze vytvářet pomocí grafů viditelnosti, grafů tečen nebo Voronoiových diagramů. Mapy cest využívají k vyhledávání cest obvykle grafové algoritmy. Mezi základní postupy patří hledání naslepo (*blind search*), hledání metodou rozděl a panuj (*divide and conquer*), prohledávání do šířky (*breadth-first search*), prohledávání do šířky v obou směrech (*bi-directional breadth-first search*), prohledávání do hloubky (*depth-first search*) a prohledávání do hloubky s postupným zanořováním (*iterative depth-first search*). Tyto algoritmy v praxi nemusejí garantovat nalezení cesty, i když existuje, kvůli používanému limitu počtu kroků. Nejčastěji jsou využívány pokročilejší algoritmy – Dijkstrův algoritmus nebo algoritmus A*.

Dijkstrův algoritmus

Dijkstrův algoritmus slouží k nalezení nejkratší (nejlevnější) cesty v grafu. Funguje pouze nad grafy, které mají nezáporně ohodnocené hrany.

Popis algoritmu lze nalézt např. v (Demel 2002). Vstupem je graf *G*, výchozí vrchol r a nezáporné ohodnocení hran $a : E(G) \to \mathbf{R}$. V algoritmu se používá pomocná množina vrcholů *D* taková, že pro všechny vrcholy $v \in D$ je hodnota $U(v)$ nejkratší dosud nalezené cesty z výchozího vrcholu *r* do vrcholu *v* rovna vzdálenosti $u(r,v)$. V každém kroku se z množiny $V \setminus D$ vybírá vrchol x s nejnižší hodnotou $U(x)$ a pro koncové uzly hran (x,y), kde $y \in (V \setminus D)$ se upravují hodnoty $U(y)$ tak, aby platilo $U(x) + a(e) \geq U(y)$.

Dijkstrův algoritmus s heuristikou

Protože v rozsáhlých grafech bývá klasický Dijkstrův algoritmus poměrně pomalý, používá se v takových případech Dijkstrův algoritmus s heuristikou. Ten kromě nejkratší dosud nalezené cesty $U(x)$ používá pro výběr vrcholu také hodnotu heuristické funkce $h(x)$. Heuristická funkce $h : V \to \mathbf{R}$ představuje dodatečné ohodnocení vrcholů. Přitom musí pro každou hranu (x,y) splňovat podmínku $a(x,y) \geq h(x) - h(y)$. Při výběru vrcholu x z množiny $V \setminus D$ se tedy hledá vrchol s nejnižší hodnotou $U(x) + h(x)$.

A* (A-star)

Kombinace Dijkstrova algoritmu a best-first search má za cíl zrychlení procesu hledání nejkratší cesty. Umožňuje kombinovat obě kritéria výběru vrcholu *x*, tj.

$U(x)$ i $h(x)$ v libovolném poměru podle vzorce $f(x) = k \cdot U(x) + (1-k) \cdot h(x)$, kde $k \in \langle 0, 1 \rangle$.

Pro k=1 dostáváme Dijkstrův algoritmus a pro k=0 best-first search. Pro k=0,5 a heuristiku $h(x)$, která je dolním odhadem vzdálenosti, se výsledný algoritmus označuje symbolem *A**. Pokud je odhad vzdálenosti v heuristice vždy kratší než je skutečná vzdálenost, nalezne i A* nejkratší cestu do cíle.

2.3.2 Metoda pravděpodobnostní cestovní mapy

Algoritmus využívající pravděpodobnostní cestovní mapy (*PRM*, *probabilistic roadmaps*) pracuje s grafem cest (*roadmap*), který si vytváří stochasticky v první fázi své činnosti. Generují se různé konfigurace robotu, které jsou přidávány do grafu cest jako vrcholy. Algoritmus se je pokouší propojit s ostatními vrcholy přímými cestami, které jsou pak přidávány do grafu jako hrany. Takto připravený graf cest se při dotazu na nalezení cesty doplní o vrcholy startu a cíle, které se napojí na stávající graf. Nalezení cesty pak lze provést některým z příslušných grafových algoritmů. Při nedostatečném pokrytí prostoru však tato metoda nemusí nalézt řešení, i když existuje. Přehled různých přístupů využívajících pravděpodobnostní cestovní mapy a jejich srovnání lze nalézt např. v práci (Geraerts 2006).

2.3.3 Plánování rychle mapujícími náhodnými stromy

Za speciální případ metody pravděpodobnostní cestovní mapy lze považovat tzv. rychle mapující náhodné stromy (*RRT*, *rapidly-exploring random trees*). Tato metoda byla poprvé představena v (LaValle 1998) jako metoda navržená pro plánování cesty se zaměřením na *neholonomní plánování* robotů s vysokým počtem stupňů volnosti. Robot je charakterizován přívlastkem neholonomní tehdy, jestliže jsou na pohyb robotu kladena nějaká dodatečná omezení (vyjma těch která jsou spojena se způsobem modelování prostředí).

Základní myšlenkou plánování rychle mapujícími náhodnými stromy je inkrementální vytváření prohledávajícího stromu, kterým se snažíme rychle a rovnoměrně prohledat konfigurační prostor. Strom se vytváří iteračně tak, že se v každé iteraci rozroste směrem k náhodně vygenerované konfiguraci robotu o nový vrchol. Vrcholy stromu představují konfigurace robotu a jeho hrany akce robotu.

2.3.4 Metody rozkladu prostředí do buněk

Pro nalezení cesty v takto modelovaném prostředí použijeme opět grafové algoritmy. Graf sestavíme z vrcholů představujících buňky a hran reprezentujících jejich společné hranice.

2.3.5 Metoda pole potenciálu

Tato metoda využívá teorii elektrického pole (Agirrebeitia 2005) Vychází z poznatku, že pokud do elektrického pole tvořeného kladným nábojem vložíme

další kladný náboj, na náboj začne působit odpudivá síla a začne se přemísťovat do bodu, ve kterém bude potenciál menší. Elektrické náboje se přitom mohou pohybovat pouze mezi takovými místy elektrického pole, jejichž potenciál je různý. Pro potřeby této metody je scéna nahrazena uměle vytvořeným elektrickým polem. Překážky jsou reprezentovány jako kladně nabitá tělesa. Poloha cíle je nahrazena jediným bodem v poli, ve kterém je potenciál roven nule (nebo minimu). Robot představuje kladně nabitou částici, která se přemísťuje z počátečního bodu do bodu s nejnižším potenciálem.

2.3.6 Plánování cesty genetickými algoritmy

Genetické algoritmy se snaží využívat při řešení úloh principy známé z evoluční biologie. Řešení úlohy se hledá pomocí postupného vývoje tzv. *populace* tvořené jedinci (*chromozomy*). Pro každého jedince je vypočtena hodnota *fitness* funkce, která vyjadřuje kvalitu řešení reprezentovaného tímto jedincem. V průběhu vývoje populace se vytvářejí generace nových jedinců z generace předchozí (*dědičnost*). Cílem tohoto vývoje je získat jedince s co nejlepší hodnotou *fitness*. K vytváření nových jedinců se používají různé operace. Mezi nejpoužívanější patří *křížení*, *mutace* a *selekce*. Plánování cesty robotu genetickými algoritmy se věnuje poměrně hodně prací, např. (Gemeinder 2003, Burchardt a Salomon 2006, Zheng et al. 2004).

2.3.7 Metody inspirované chováním hmyzích kolonií

Metody plánování cest mohou využívat i mechanismy fungování sociálních skupin živočichů (rojová inteligence). Na základě pozorování skutečných mravenčích kolonií byl vyvinut přístup nazývaný ACO (*ant colony optimization*). Velký počet jednoduchých umělých mravenců pomocí vzájemné komunikace hledá dobré řešení složitých optimalizačních úloh. Umělé feromony představují společnou paměť, která umožňuje celému společenství jednoduchých mravenců nacházet nová řešení. Kose a Acan (2004) popisují použití algoritmu ACO pro řešení úlohy WFAR (*wall-following autonomous robot*). Práce (Tambouratzis 2007) popisuje další variantu algoritmu ACO nazvanou POOCA (*progressive optimization of organized colonies of ants*). Ta kombinuje ACO s rozšířením aktivace okolí vítězného uzlu pomocí samoorganizujících map (SOM).

2.3.8 Další metody plánování cesty robotu

Pro hledání hladkých cest vhodných především pro neholonomní roboty lze využít také algoritmus PSO (*particle swarm optimization*), který podobně jako ACO patří mezi algoritmy rojové inteligence. Tento algoritmus je popsán v (Li 2005).

Kromě výše popisovaných metod plánování cesty robotu existuje i řada dalších, k nimž patří například různé heuristické algoritmy, expertní systémy nebo neuronové sítě. Vzhledem k zaměření práce nebylo v tomto přehledu rozebíráno případové usuzování, protože je mu věnována samostatná kapitola 3.

2.4 LOKÁLNÍ ŘÍZENÍ POHYBU

Lokální metody řízení se využívají především pro řízení pohybu v neznámých prostředích. Pohyb založený výhradně na lokálním chování však často vede k problémům. Stává se že robot „uvízne“ v místě s nepříznivými podmínkami, protože mu chybí schopnost chápání a čtení mapy na vyšší úrovni jakou má například člověk se svým heuristickým způsobem myšlení. Čistě lokální chování je vhodné pouze pro roboty, které se pokaždé pohybují jen jednou vždy v novém prostředí (např. záchranářský robot). Nedostatky lokálních metod pak obvykle musí eliminovat přítomnost lidského operátora.

Z uváděných principů je tedy zřejmé, že kombinací metod globálních a lokálních lze při navigaci robotů dosáhnout poměrně dobrých výsledků, protože globální metody představující vyšší úroveň řízení poskytují předem plány cest a metody lokální slouží k okamžitému přizpůsobení se situaci při střetnutí s nečekanými prvky prostředí. Navíc umožňují robotu fungovat i ve chvílích, kdy ještě nemá o prostředí žádné informace.

Roboty s vyšším řízením se hodí především do prostředí, v němž se pohybují opakovaně při plnění různých úkolů (např. poslíček s poštou). Pokud je robot navíc schopen uchovávat informace z předchozích úloh, může si průběžně doplňovat znalosti o prostředí a vylepšovat tak výsledky dosahované globálním plánováním a také tuto činnost výrazně zrychlovat.

3 PŘÍPADOVÉ USUZOVÁNÍ

Případové usuzování (*case-based reasoning*, *CBR*) představuje moderní přístup k řešení problémů. CBR systémy řeší nový problém adaptováním již známých řešení podobných dříve řešených problémů (*případů*) a tím dochází k žádoucímu efektu učení se z předchozích zkušeností. Rozhodování na základě využití dříve řešených případů je silný a v praktickém životě často používaný způsob řešení problémů. Přehled základních pojmů vztahujících se k případovému usuzování a popis některých z hlavních metodologických přístupů na tomto poli lze nalézt např. v (Aamodt 1994).

Základní myšlenkou případového usuzování je předpoklad, že podobné úlohy budou pravděpodobně mít také podobná řešení. Nejprve tedy navrhneme strukturu případu včetně jeho řešení, pak shromáždíme v minulosti řešené problémy a umístíme je do tzv. případové báze, což je jistý druh znalostní báze. Dále pak tyto případy využíváme k řešení podobných zadání tak, že je různě upravujeme (adaptujeme). Úspěšně řešené případy následně také přidáváme do případové báze.

Funkce běžného CBR systému je popsána tzv. obecným cyklem CBR. V této souvislosti se často hovoří o tzv. cyklu 4R. Typicky totiž definujeme následující

čtyři kroky (Aamodt 1994) – nalezení (*retrieve*), použití (*reuse*), ověření (*revise*) a uchování (*retain*). Tyto kroky představují zjednodušený popis fungování CBR systému: nalezení nejvíce podobného případu nebo případů, použití informací a znalostí z nalezeného případu k vyřešení nového problému, ověření navrženého řešení a uchování částí této zkušenosti tak, aby byly využitelné v budoucnosti pro řešení dalších problémů.

3.1 PŘÍPADOVÁ BÁZE

Případová báze bývá obvykle implementována podobně jako klasická databáze. Hlavní rozdíl spočívá v druhu informací, které se z nich snažíme získat. V databázi hledáme konkrétní záznam, v případové bázi ten nejpodobnější. K tomu potřebujeme další znalosti a mechanismy, z nichž hlavní roli bude hrát podobnostní míra.

Narozdíl od běžných přístupů jsou CBR systémy schopny využívat předchozí zkušenosti. To jim umožňuje rychlejší hledání řešení, protože nemusejí prohledávat oblasti, které přinášejí častěji neuspokojivá řešení. Rozhodnutí činí na základě informací uchovávaných v případové bázi. Dodatečné prostředky nezbytné k udržování báze jsou obvykle nižší než ty, které by byly spotřebovány zbytečným vyhledáváním v neperspektivních oblastech.

3.2 HYBRIDNÍ SYSTÉMY

Hybridní systémy využívají kromě mechanismů CBR také alternativní metody hledání řešení. Ty musí být schopné řešit celou úlohu úplně od začátku. K tomu mohou posloužit např. genetické algoritmy. Hybridní systémy jsou pak schopné řešit i problémy, které pomocí aktuální případové bázi řešitelné nebudou. Báze může být zpočátku zcela prázdná. K jejímu zaplnění počátečními hodnotami poslouží alternativní metody, na kterých bude systém zpočátku závislý. Postupně budou ale metody CBR převládat, i když alternativní metody nevytlačí zcela. Mohou nastat i jiné situace, kdy nemáme dostatek informací k navržení řešení jen pomocí báze případů nebo dokonce navržené řešení zamítneme a musíme pak využít alternativních postupů k nalezení alespoň nějakého řešení.

Louis a Li (1997) například kombinují případové usuzování s genetickým algoritmem. Výsledná kombinace využívá výhod obou přístupů. Genetický algoritmus dodává robustnost a adaptivní učení, zatímco případové usuzování zrychluje celý systém. Využití genetického algoritmu v kombinaci s CBR pro navigaci robotu popisuje i Šeda (2003). V práci (Ram 1997) se v kombinaci s případovým usuzováním používá posilované učení a v práci (Kruusmaa 1998a) má doplňující funkci pravděpodobnostní hledací algoritmus.

3.3 ÚDRŽBA PŘÍPADOVÉ BÁZE

Efektivitu CBR systému je možné zvýšit výběrem vhodné strategie pro ohodnocení kvality a následný výběr uchovávaných případů. Uchovávání všech

případů je nevýhodné, protože velké množství velmi podobných případů nepřináší více užitečných dodatečných informací. Po nějakém čase můžeme dosáhnout limitu velikosti případové báze. Proto má smysl zabývat se nějakou výběrovou strategií, zvláště pak u systémů určených pro dlouhodobější provoz.

Strategie lze rozdělit na pasivní a aktivní. Pasivní nerozlišují kvalitu uchovávaných případů. Příkladem je nahrazování nejstarších případů novými. Tento přístup by ale mohl vést ke zhoršování navrhovaných řešení, protože bychom průběžně ztráceli i velmi dobré případy.

Aktivní výběrové strategie se jeví jako mnohem efektivnější. Jistým aktivním přístupem disponuje i strategie, která po dosažení limitu případů v případové bázi spíše než nejstarší nahrazuje nejméně úspěšné resp. nejméně používané případy. Dalšími často používanými kritérii bývají náklady na použití případu a případně riziko vyplývající z jeho použití. Další možnou strategií je nezapamatovávat si celé případy, ale pouze indexy případů, z nichž byly odvozeny pomocí adaptačního procesu, protože tento adaptační proces může být na základě stejných informací kdykoliv zopakován.

4 VYUŽITÍ CBR PŘI NAVIGACI ROBOTU

Navigace autonomního robotu probíhá ve dvou úrovních. Vygenerovat cestu ze startovní do cílové pozice bez kolizí se známými překážkami má na starosti tzv. globální navigace. Pohyb robotu po plánované cestě tak, aby se vyhnul neočekávaným překážkám, případně hledání cest, které nelze z různých příčin globálně naplánovat, řídí lokální navigace. Případové usuzování lze použít pro navigaci robotu na obou úrovních.

4.1 CBR PRO GLOBÁLNÍ NAVIGACI

Díky případové bázi lze při plánování cesty využívat dříve nalezené cesty nebo jejich části. Místo hledání zcela nové cesty stačí nalézt nejpodobnější cesty nebo jejich části a adaptovat je tak, aby byly použitelné pro řešení aktuálního zadání. Podobnost problémů při navigaci bývá obvykle chápána jako vzájemná blízkost bodů startu a zároveň bodů cíle nového problému a zapamatovaného problému.

Určitým nedostatkem některých CBR systémů je, že nerozeznávají vícenásobné případy a části případů. Využitím pouze celých případů při plánování cesty je značně redukována úroveň využití předchozích zkušeností. Tento přístup, kdy jsou za případy brány celé cesty, je využíván například v (Kruusmaa 1998a, Kruusmaa 1998b).

Haigh a Shewchuk (1994) navrhují ve spojitém prostředí postupy pro určení množiny případů, které společně tvoří dobrý základ pro novou cestu a přitom mohou obsahovat i části cest. Každý případ je aproximován přímkovým segmentem ve dvourozměrném grafu, přičemž segmenty se navzájem mohou protínat pouze v koncových bodech. Pokud navržená cesta protíná některé z existujících segmentů

grafu, jsou cesta i segmenty, se kterými se kříží, rozděleny na menší případy v bodech průsečíků tak, aby byly splněny výše uvedené požadavky. Výsledný graf autoři nazvali případový graf (*case graph*) a jednotlivé segmenty tohoto grafu jsou případové segmenty.

4.2 CBR PRO LOKÁLNÍ NAVIGACI

Případové usuzování lze použít i pro výběr lokálního chování robotu. Můžeme například v případové bázi uchovávat postupy, které byly použity pro vyhnutí se překážce společně s vlastnostmi této překážky. Systém pak může na základě těchto informací navrhnout efektivní způsob překonání podobné překážky.

Ram a Santamaria ve své práci (Ram 1997) představují systém SINS (*self-improving navigation system*). Ten je schopen dynamického výběru, modifikace a vyhledávání vhodného chování robotu. Je založen na hybridní metodě zvyšování výkonu reaktivního řídícího systému, který kombinuje spojité případové usuzování a posilované učení.

V práci (Likhachev 2002) je popsán přístup pro optimální parametrizaci chování v rámci CBR metodologie pro autonomní navigační úlohy. Popisovaná metoda rozšiřuje modul případového usuzování tak, že mu umožňuje učit se nové a optimalizovat existující případy. CBR modul pracuje na reaktivní úrovni. Slouží především k výběru sady parametrů pro aktuálně zvolené chování, které se nejlépe hodí pro současné prostředí.

Práce (Urdiales 2006) se zabývá využitím CBR pro lokání navigaci mobilních robotů v dynamickém prostředí.. Případy obsahují orientaci robotu, směr k cíli, informace ze senzorů a výsledný směr pohybu a jsou doplněny vahami, které se adaptivně mění na základě frekvence a úspěšnosti použití případů. Pomocí shlukovacího algoritmu se vytvářejí třídy podobných případů a prototypy těchto tříd tvoří případovou bázi. Případy jsou získávány učením na základě pozorování a učením na základě vlastní zkušenosti. Navržený přístup k lokální navigaci je ověřován na čtyřkolovém robotu se sonarovými senzory.

Práce (Micarelli 2001) popisuje použití CBR v systému rozpoznávání charakteristických rysů interiérového prostředí pro lokální navigaci robotu v tomto prostředí.

4.3 CBR V INTEGROVANÝCH SYSTÉMECH

Jak již bylo zmíněno, existují dvě hlavní skupiny metod navigace robotu. Metody globální navigace (deliberativní metody) a metody lokální navigace (reaktivní, zpětnovazební metody). Deliberativní plánovač sestavuje celou cestu předtím, než je použita, a obvykle pracuje s některými zjednodušujícími předpoklady. Reaktivní

plánovač hledá sadu chování na nejnižší úrovni podle vývoje okolní aktuální situace. Činí rychlá rozhodnutí o nutných okamžitých akcích, přičemž svůj výběr každou chvíli znovu přehodnocuje.

Supic a Ribaric (2001) spojují globální a lokální navigační úlohy pomocí postupného případového usuzování (*stepwise case-based reasoning*, *SCBR*). SCBR nabízí vhodný rámec pro systém autonomní navigace, v němž se případy cest používají pro podporu vyšší úrovně plánování v modulu plánování cest a stavové případy se používají pro podporu stavového modulu na nižší úrovni.

Jiný způsob integrace obou navigačních úrovní využívajících případové usuzování byl navržen v práci (Fox 2000) pro RUPART, což je doručovací robot pohybující se chodbami kancelářské budovy a převážející předměty a zprávy mezi různými místy. Toto prostředí je modelováno pomocí obdélníkové mřížky. Jak plány cest tak i zpětnovazební chování mohou být ukládány v případech a hledány pomocí stejných indexů a podobnostních ohodnocení.

V práci (Srinivasan 2006) je popsáno použití CBR pro účinné plánování herních strategií v robotickém fotbale. Případy popisují různé herní situace a odpovídající chování robotického týmu. Systém na základě rozpoznání aktuální herní situace vyhledává v případové bázi nejpodobnější případy a na základě nich konstruuje optimální týmové chování.

5 CÍLE DISERTAČNÍ PRÁCE

Cílem disertační práce je zkoumat možnosti využití metod případového usuzování pro navigaci autonomního robotu. V rámci této práce by měly být realizovány tyto úkoly:

- Podrobná analýza metod CBR použitelných pro navigaci robotu. Cílem je vytvoření přehledu ve studované problematice.
- Studium možností použití metod bez CBR, které by sloužily pro srovnání s metodami CBR a pro jejich doplnění v případě systému s prázdnou bází případů.
- Nalezení reprezentace případů s využitím částí případů vhodné pro ukládání dat v případovém grafu.
- Vytvoření softwarového systému pro plánování cesty robotu. Implementace metod CBR a vybrané metody pro hledání cest bez báze případů do tohoto systému.
- Posouzení využitelnosti a vhodnosti metod CBR pro navigaci robotu pomocí srovnávacích experimentů.

6 NÁVRH PLÁNOVÁNÍ CESTY S VYUŽITÍM CBR

6.1 REPREZENTACE PROSTŘEDÍ

Uvažujeme pravoúhlou obdélníkovou mřížku $[1, m]\times[1, n]$, kde m je počet řádků a n počet sloupců mřížky. Každá buňka c této mřížky je určena dvojicí souřadnic v mřížce $c = (x, y)$, kde $x \in \{1,2,...,m\}$, $y \in \{1,2,...,n\}$. Buňky obsahující předem známé překážky jsou inicializovány konstantní hodnotou. Buňky odpovídající volnému prostoru nesou hodnotu 1, což znamená že jsou bez jakéhokoliv rizika. Na vstupu je ale možné uvažovat i mapu obsahující obecná rizika. Taková počáteční mapa může být připravena například tehdy, pokud riziko zahrnuje předem známou kvalitu terénu.

Při průjezdech robotu plánovanými cestami může v případě výskytu komplikací každá buňka získávat vyšší ohodnocení rizikovosti. Podobně při úspěšném průjezdu buňkou se hodnota rizika sníží. Díky těmto mechanismům si robot udržuje svoji „živou“ privátní mapu. Do rizika by bylo možné zahrnout i další vlivy prostředí (různý terén, blízkost překážek, ap.).

Povolené směry pohybu robotu v tomto prostředí jsou pouze horizontální, vertikální a diagonální. Vzdálenost mezi buňkami $c_i = (x_i, y_i)$ a $c_j = (x_j, y_j)$ lze definovat těmito vzorci:

$$d(c_i, c_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

$$d(c_i, c_j) = |x_i - x_j| + |y_i - y_j|$$

$$d(c_i, c_j) = (\sqrt{2} - 1)\cdot \min\{|x_i - x_j|, |y_i - y_j|\} + \max\{|x_i - x_j|, |y_i - y_j|\},$$

přičemž poslední z nich odpovídá situaci, kdy jsou povoleny pouze horizontální, vertikální a diagonální směry pohybu. Pro výpočet vzdáleností je v navrhovaném systému možné využívat kterýkoliv z uvedených tří vzorců. První se používá hlavně pro výpočet odhadů vzdáleností, druhý z nich je vhodný spíše pro interiérová prostředí (tato vzdálenost bývá někdy nazývána manhattanská). Pomocí třetího jsou počítány skutečné délky plánovaných cest. Podle typu prostředí pak lze kterýkoli využít také pro výpočet délky ideální cesty.

6.2 REPREZENTACE CEST

Cesta $P(c_s, c_g)$ ze startovní buňky c_s do cílové buňky c_g je specifikována jako posloupnost $P(c_s, c_g) = \{c_s, c_{k1},..., c_{ki},..., c_{kn}, c_g\}$, jejímiž vnitřními buňkami c_{ki} jsou jen ty buňky, v nichž dochází ke změně směru pohybu nebo kde se tato cesta protíná s jinou cestou. Cesta je chápána jako posloupnost lineárních segmentů (c_i, c_j). Každému lineárnímu segmentu je přiřazena hodnota funkce nákladů F charakterizující jeho průjezdnost.

Funkci nákladů lze konstruovat několika způsoby. Z intuitivního chápání úlohy vychází vzorec $F(c_i, c_j) = l(c_i, c_j) \cdot r(c_i, c_j)$, kde l je délka lineárního segmentu mezi body c_i a c_j, r je pak riziko průjezdu tímto segmentem. Riziko každého použitého segmentu se na celkových nákladech podílí v takové míře, jak je segment dlouhý. Na základě tohoto hodnocení je vybírána nejvhodnější cesta s ohledem na zvolenou míru rizika. Výpočet rizika r lze ovšem provádět různými způsoby v závislosti na druhu aplikace. Může jít o prostý součet rizik buněk, kterými segment prochází

$$r(c_i, c_j) = \frac{1}{2} r(c_i) + \sum_{c \in I(c_i, c_j)} r(c) + \frac{1}{2} r(c_j),$$

kde $I(c_i, c_j)$ je množina všech vnitřních buněk hrany (c_i, c_j). Tento způsob výpočtu rizika byl, v kombinaci s předchozím vzorcem pro výpočet funkce nákladů, nakonec také použit v implementaci. V situacích, kdy bude důležité vyhnout se co nejvíce riziku (např. kvůli poškození robotu), bude vhodnější použít maximální riziko ze všech buněk segmentu.

Samotnou funkci nákladů lze ale konstruovat i dalšími způsoby, např. jako vážený součet délky segmentu a rizika průjezdu tímto segmentem. Pomocí vhodně zvolené funkce nákladů lze výběr cest orientovat spíše k rychlým cestám nebo naopak ke spolehlivým.

6.3 PŘÍPADOVÁ BÁZE

Všechny zapamatované cesty jsou ukládány do případové báze. Ta se používá při plánování cest pomocí případového usuzování. Aby se zvýšila využitelnost zapamatovaných informací, nepoužívají se jako případy pouze celé cesty, jako je tomu například v (Kruusmaa 1998a, Kruusmaa 1998b), ale lze používat i jednotlivé části cest. Za část cesty je považován každý lineární segment cesty ohraničený dvěma body, což mohou být koncové body cesty, body změny směru jízdy nebo body křížení cesty s jinou cestou. Všechny segmenty cest uchovávané v případové bázi lze s výhodou sestavit do podoby případového grafu podobně jako je tomu v (Haigh 1994), kde je tento graf používán ve spojitém prostředí.

Chápání případové báze jako grafu umožňuje převést složitou fázi hledání nejpodobnějšího případu v cyklu případového usuzování na snáze řešitelné hledání nejlevnější cesty v grafu, při němž lze využít známé algoritmy.

6.4 PLÁNOVÁNÍ CESTY BEZ POUŽITÍ CBR

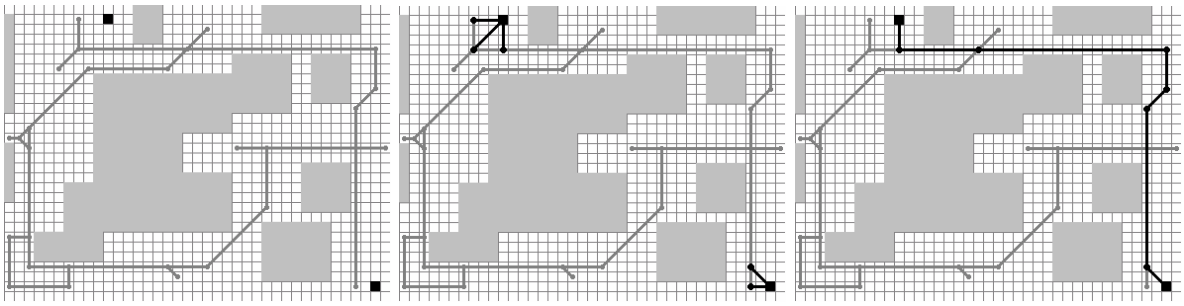
Pro situace, kdy nelze cestu naplánovat pomocí postupů využívajících případové usuzování, je nutné dovybavit systém metodami pracujícími na jiném principu. Využívají se především tehdy, kdy případová báze neobsahuje žádnou cestu, která by ústila v blízkosti zadaných bodů startu a cíle. Lze je využít i v případě, že cesta naplánovaná pomocí případového usuzování nebude dostatečně kvalitní.

Pro plánování bez použití CBR lze využít Dijkstrův algoritmus a algoritmy z něj odvozené. Graf, který bude v takovém případě konstruován, bude jako vrcholy obsahovat všechny buňky mapy neobsazené překážkami, přičemž hrany budou vyjadřovat vzájemné sousedství buněk.

6.5 CBR PLÁNOVÁNÍ CESTY

Pro plánování cesty pomocí případového usuzování se využívá případová báze konstruovaná v podobě případového grafu. V tomto grafu se cesta mezi zadaným startem a cílem hledá opět pomocí Dijkstrova algoritmu. Kromě případového grafu se při plánování cesty využívají také další informace, které robot shromáždil při svém pohybu v prostředí. Jsou to především rizika průjezdu jednotlivými buňkami mapy sdružená do funkce nákladů.

V situaci, kdy je případová báze reprezentována pomocí případového grafu, může adaptační fáze probíhat před hledáním nejpodobnějším případu, protože adaptační proces není vázán na znalost nejpodobnějšího případu. Pokud dočasně doplníme případový graf o spojovací úseky do všech vytypovaných blízkých bodů ještě před samotným hledáním cesty, bude pro nalezení cesty v grafu stačit jediné použití hledacího algoritmu. Samotný adaptační proces často využívá některou z metod plánování bez použití CBR.



a b c

Obr. 1. Případový graf, start a cíl (a), navržená (b) a adaptovaná cesta (c)

Naplánované řešení je dále ověřeno praktickým využitím cesty. Pokud se robot cestou střetne s neznámými překážkami, opustí režim sledování naplánované cesty a pokusí se překážku v lokálním režimu objet. Pak pokračuje ve sledování původní cesty. Pokud robot úspěšně dorazí do cíle, může být absolvovaná cesta přidána do případového grafu.

Získaná ověřená cesta se může od navržené cesty lišit v místech, kde se robot setkal s některou z neznámých překážek. Místa a počty těchto setkání si robot uchovává v podobě informací v privátní mapě. Při opakovaných výskytech překážek na některém úseku má tato statistika za následek zhoršení hodnocení daného úseku a plánování nových cest je pak spíše orientováno na použití bezpečnějších úseků, jsou-li dostupné.

Při plánování cesty případovým usuzováním nemusí být vždy možné cestu nalézt (nelze se napojit na případový graf, je málo informací v bázi případů, ap.) V takové

situaci je automaticky nasazena doplňková metoda plánování nevyužívající CBR. Plánování bez CBR ale může být, kvůli zachování schopnosti systému hledat nová řešení, upřednostněno i ve chvíli, kdy nějaká podobná cesta existuje. Systém je tak nucen hledat i nová řešení místo toho, aby jen konzervativně využíval stará zapamatovaná řešení.

6.6 ÚDRŽBA PŘÍPADOVÉ BÁZE

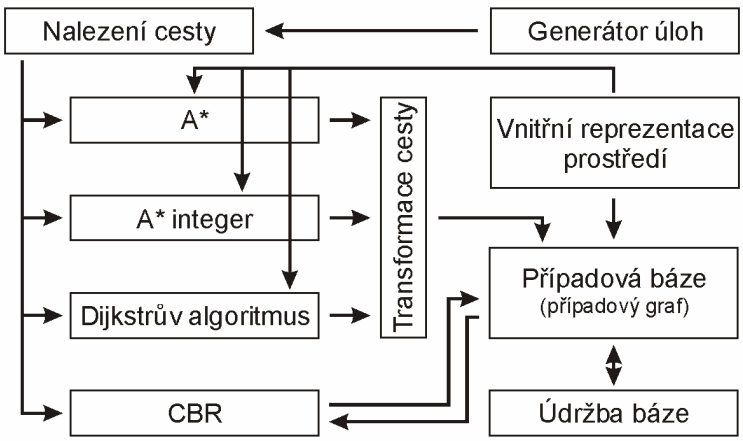
Přibývající množství nových vrcholů a hran v případovém grafu bude postupně zpomalovat plánování cest případovým usuzováním. Proto je potřeba udržovat případový graf rozumně ohraničený jak v počtu hran, tak i v počtu vrcholů.

Jednodušší formou údržby případového grafu bude vypouštění nadbytečných vrcholů, které zdánlivě leží uprostřed hrany.

Další metodou údržby případového grafu může být mazání hran (celých úseků cest). Některými strategiemi takovéto údržby v případě celých cest se zabývá (Kruusmaa 1998a, Kruusmaa 1998b). Vzhledem k povaze navrhované aplikace se jeví jako optimální strategie zapomínání nejhorších případů. Ta také dobře koresponduje s ostatními principy případového usuzování. Hlavní myšlenkou tedy je odstraňovat z případové báze ty případy, které mají špatné ohodnocení. Ohodnocení kvality případů (cest resp. jejich úseků) je ale možné chápat několika způsoby. Minimálně je to hodnocení podle rizika průjezdu, ale za špatný případ lze považovat i případ, který je jen zřídka využíván.

7 IMPLEMENTACE

Aby bylo možné experimentálně ověřit popisované metody případového usuzování, bylo do testovací aplikace nezbytné implementovat efektivní reprezentaci případové báze v podobě případového grafu, metody správy a údržby



Obr. 2. Struktura aplikace

případové báze, metodu plánování cest využívající případové usuzování, další vybrané metody plánování cest a nástroje pro experimentální srovnávání metod

plánování cesty. Ostatní součásti popisované v kapitole 6 (především fáze ověření navrženého řešení použitím cesty) jsou již silně závislé na konkrétní realizaci skutečného robotu, a proto nejsou do implementace zahrnuty. Pro naplánované experimenty nejsou podstatné, protože se na časových nákladech plánování cesty nijak nepodílejí a neovlivňují ani kvalitu plánovaných cest.

Z vedeného schématu je patrné, že se přirozeným centrálním prvkem stala případová báze. Vstupují do ní naplánované cesty ať už přímo při plánování případovým usuzováním nebo po transformaci z ostatních metod. Pro její údržbu je navrženo několik metod. Samotná případová báze pak využívá i informace o prostředí pro výpočet ohodnocení případů.

7.1 REPREZENTACE PŘÍPADOVÉ BÁZE

Případová báze je reprezentována případovým grafem sestaveným z lineárních úseků zapamatovaných cest. Základem případového grafu je seznam buněk (souřadnic), které tvoří vrcholy případového grafu. V různých okamžicích bude potřeba rychlým způsobem hledat mezi vrcholy případového grafu právě podle hodnot mapových souřadnic, proto jsou souřadnice každé použité buňky *c* z dvourozměrného prostoru transformovány na jednorozměrnou celočíselnou souřadnici. Souřadnice vrcholů jsou v seznamu seřazeny podle velikosti, což umožňuje velmi rychlé hledání vrcholů, testování jejich přítomnosti v seznamu a přidávání nových vrcholů pomocí dichotomického hledání (*binary search*). Pořadová hodnota slouží pro přístup k dalším informacím o případovém grafu.

Spolu s každým vrcholem je do případového grafu přidáván také prostor pro informace o hranách, jejichž jedním koncovým bodem vrchol může být. Jde o osm globálních souřadnic sousedních vrcholů, osm informací o riziku hrany a osm informací o používání hrany.

Seřazení všech popisovaných informací (sousedé, rizika, použití) kopíruje seřazení vrcholů v seznamu. Tento seznam tak funguje i jako směrovač pro přístup k datům o hranách.

7.2 VNITŘNÍ REPREZENTACE PROSTŘEDÍ

Doplňující zdroj informací pro všechny metody plánování cesty představuje vnitřní reprezentace prostředí. Ta má podobu privátní mapy robotu. Privátní mapa je stejně jako zadaná mapa prostředí dvourozměrné pole se čtvercovými buňkami. Se zadanou mapou prostředí je rozměrově shodná a obsahuje také všechny informace o známých překážkách, které byly k dispozici na počátku. Kromě nich však obsahuje i průběžně aktualizované informace o zjištěných výskytech neznámých překážek.

Každá buňka je ohodnocena reálnou hodnotou z intervalu $\langle 1, 2 \rangle$, přičemž hodnota 2 označuje buňku obsazenou překážkou. Všechny ostatní hodnoty znamenají, že použití této buňky je spojeno s určitým rizikem, které je tím větší, čím je vyšší její ohodnocení.

7.3 METODY PLÁNOVÁNÍ CESTY BEZ CBR

Pro plánování cest v situacích, kdy nebude možné použít případové usuzování jsou implementovány tři doplňující algoritmy: Dijkstrův algoritmus, A* a A* s celočíselnou aproximací heuristiky (dále označovaný jako A_i*).

Základní schéma těchto algoritmů však není samo o sobě pro programové zpracování příliš vhodné. Množinu vrcholů $V \setminus D$, z níž se vybírá vrchol s nejnižší hodnotou nejkratší dosud nalezené cesty (resp. její kombinace s hodnotou heuristické funkce), je potřeba reprezentovat nějakým vhodným způsobem tak, aby bylo možné ji rychle prohledávat. Jako vhodná se jeví reprezentace této množiny haldou. Výhodou třídění dat haldou je velmi malá závislost času třídění na vstupních datech. Halda udržuje data průběžně uspořádaná, a proto dostáváme nejmenší (největší) prvek automaticky na jejím vrcholu. Navíc lze haldu velmi efektivně zobrazit do lineárního pole, což snižuje paměťové nároky při jejím třídění a umožňuje její snazší programové zpracování.

Algoritmus A_i* je modifikací algoritmu A*, v heuristice je ale místo klasické eukleidovské vzdálenosti použita celočíselná aproximace vzdálenosti. Algoritmus A_i* využívá předem připravenou mapu celočíselných vzdáleností, k jejíž konstrukci lze použít 4, 8 nebo 16-směrovou propagační metodu. Použití celočíselných aproximací pouze pro 4, 8 resp. 16 nejbližších buněk znamená maximální chybu odhadu vzdálenosti ve výši přibližně 41%, 8% resp. 3%.

Výsledkem těchto plánovacích algoritmů je posloupnost sousedících buněk mapy, které tvoří naplánovanou cestu.

7.4 PLÁNOVÁNÍ CESTY PŘÍPADOVÝM USUZOVÁNÍM

Pro plánování cesty případovým usuzováním se také používá Dijkstrův algoritmus s haldou. Prohledávaným grafem je ale tentokrát případový graf. Aby bylo možné hledat cestu mezi zadaným startem a cílem, je nejprve nutné dočasně doplnit případový graf tak, aby se jak start, tak i cíl staly jeho vrcholy. Jak již bylo v podkapitole 6.5 popsáno, jedná se o adaptační krok cyklu případového usuzování. Je však prováděn v předstihu před krokem nalezení nejpodobnějšího případu, protože tak dochází k výpočetním úsporám. Navíc princip adaptace není nijak vázán na znalost řešení, a proto jeho provedením již v této chvíli nedochází ke zhoršení výsledného řešení.

Napojování bodu startu a cíle je nutné provádět nejen do vrcholů případového grafu, ale obecně do kteréhokoliv jeho bodu. Při rozšiřování případového grafu tedy mohou pro každý bod startu i cíle nastat tři možnosti: bod již je vrcholem případového grafu, bod leží na některé hraně případového grafu, ale není jeho vrcholem nebo bod leží zcela mimo hrany případového grafu.

Největší problém představuje třetí z uvedených možností, kdy přidávaný bod není ani vrcholem případového grafu, ani neleží na žádné jeho hraně. Takový bod je nutné k případovému grafu připojit pomocí nové části cesty. Obecně však nestačí

jedno napojení na případový graf, protože nelze předem rozhodnout, kam bude nejlepší se napojit.

Se všemi nalezenými nebo vytvořenými vrcholy bude přidávaný vrchol propojen hranou. Pokud ale nelze nalézt pro některý bod žádné napojení na případový graf, adaptace a tím i celé plánování cesty pomocí případového usuzování selhává a je automaticky nasazena některá z alternativních metod plánování cesty.

Po rozšíření případového grafu o body startu a cíle a případném doplnění všech adaptačních změn již lze přistoupit k samotnému hledání nejlepší cesty Dijkstrovým algoritmem. I hledání cesty v případovém grafu může skončit nezdarem. Pokud graf sestává z několika navzájem izolovaných částí, přičemž start a cíl jsou součástí dvou různých částí, nebude možné cestu nalézt.

I když ale nějaké řešení nalezneme, nemusí se jednat o řešení dostatečně kvalitní. Kvalitu řešení lze posuzovat podle různých kritérií, ale asi nejpřirozenějším bude délka nalezené cesty vůči délce cesty ideální. Vzájemný poměr těchto délek závisí na prostředí a bude různý např. pro obecnou krajinu s malými, náhodně rozestavěnými překážkami a pro interiérové prostředí budovy s místnostmi. Pokud by měl robot umět rozlišovat různá prostředí, bylo by vhodné, aby tento parametr také podléhal učení.

Pokud tedy nastane některá z uvedených dvou situací, následuje opět automaticky nové naplánování cesty některou z alternativních metod. Ve druhém případě tento mechanismus realizuje jednání, které raději volí riziko hledání nové cesty, než aby byla použita cesta špatná. Právě díky tomuto jednání se postupně mohou řešení v případové bázi vylepšovat.

Po dokončení hledání jsou z případového grafu postupně odstraněny všechny adaptační úpravy. Pokud byla úspěšně nalezena cesta mezi zadanými body, je přidána do případového grafu.

7.5 PŘIDÁVÁNÍ CEST DO PŘÍPADOVÉHO GRAFU

Přidávání nových cest do případového grafu je asi nejkomplikovanější činností související s uchováváním případů v případové bázi. Vzhledem ke zvolené reprezentaci prostředí a cest totiž existuje velké množství různých vzájemných poloh dílčích úseků cest a to si vyžaduje poměrně velkou skupinu nejrůznějších testů při přidávání do případového grafu. Úseky nově přidávané cesty se mohou různě vzájemně křížit s původními hranami případového grafu a tím vznikají nové vrcholy, dochází k různému dělení hran, apod.

První krok při přidávání nové cesty je rozdělení cesty na jednotlivé lineární úseky (části případů). Koncovými body těchto lineárních úseků jsou start, cíl a body, ve kterých dochází ke změně směru pohybu. S každým takto vzniklým lineárním úsekem se dále pracuje jako se samostatným případem. Přidávání cesty se pak řeší jako dávkově prováděné přidávání jednotlivých úseků.

Většinu operací při přidávání úseků je výhodnější provádět s lineární reprezentací mapy, protože tím lze ušetřit některé výpočty, které by se jinak musely provádět při

dvourozměrné reprezentaci. Proto jsou vstupní informace o koncových bodech nejprve převedeny na jednorozměrné souřadnice.

7.6 ÚDRŽBA PŘÍPADOVÉ BÁZE

Program nabízí tři způsoby údržby případové báze. První možností je odstranění nadbytečných vrcholů. To jsou ty vrcholy, z nichž vedou pouze dvě hrany a to v navzájem protilehlých směrech. Tento způsob údržby tak redukuje množství vrcholů i hran případového grafu.

Druhá možnost, jak udržovat případový graf rozumně ohraničený, je zapomínání zřídka používaných hran. Při tomto způsobu údržby lze využít dvě různá kritéria – mez počtu použití a procento počtu použití.

Poslední možností údržby je zapomínání nejhorších hran (s nejvyšším rizikem). Tentokrát jsou k dispozici tři možná kritéria – počet, mez rizika a procentní poměr rizika.

8 POPIS A VYHODNOCENÍ EXPERIMENTŮ

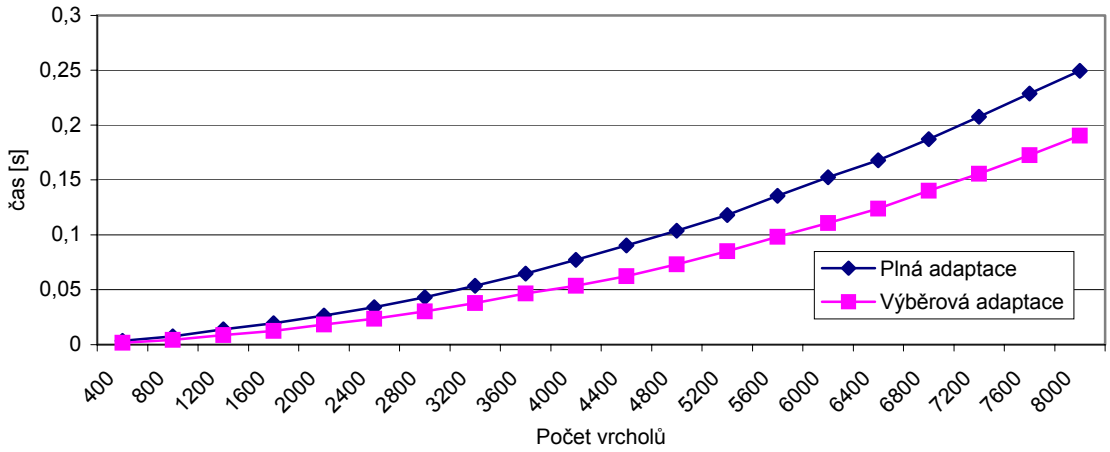
Veškeré experimenty byly prováděny na osobním počítači s procesorem Mobile Intel Pentium 4 – M, 1,70GHz a 768MB RAM. Experimenty byly zaměřeny na ověření několika klíčových charakteristik výkonu metod případového usuzování. Většina z nich se však odvíjí od počtu vrcholů případového grafu a volitelných hodnot některých dalších parametrů. Protože se při experimentálním plánování jedná o úlohu s mnoha parametry, byly pro některé testy voleny takové hodnoty parametrů, které se při provádění jiných experimentů ukázaly jako nejvhodnější. Ve všech testech jsou používány mapy s informacemi o riziku.

8.1 DOBA CBR PLÁNOVÁNÍ V ZÁVISLOSTI NA VELIKOSTI PŘÍPADOVÉHO GRAFU

V tomto testu byly pevně voleny startovní a cílová buňka tak, aby mezi nimi bylo možné naplánovat cestu bez použití alternativních metod. Byl měřen čas potřebný pro 1000 naplánování téže cesty. Dosažené časy jsou průměrem z 10 testů s různě volenými starty a cíli. Celý test byl prováděn pro obě implementované metody adaptace – plnou a výběrovou.

Obě metody adaptace hledají adaptační hrany do maximální délky dané rozměrem adaptačního okolí. Při volbě plné adaptace se hledají adaptační hrany napojující start (cíl) na kteroukoliv hranu případového grafu procházející adaptačním okolím. Výběrová adaptace redukuje objem výpočtů výběrem hran případového grafu, které začínají nebo končí pouze v určené výběrové oblasti mapy.

Při výběrové adaptaci bylo v experimentech používáno adaptační okolí o velikosti 5% velikosti mapy a pro výběrovou oblast byl použit trojnásobek velikosti adaptačního okolí.

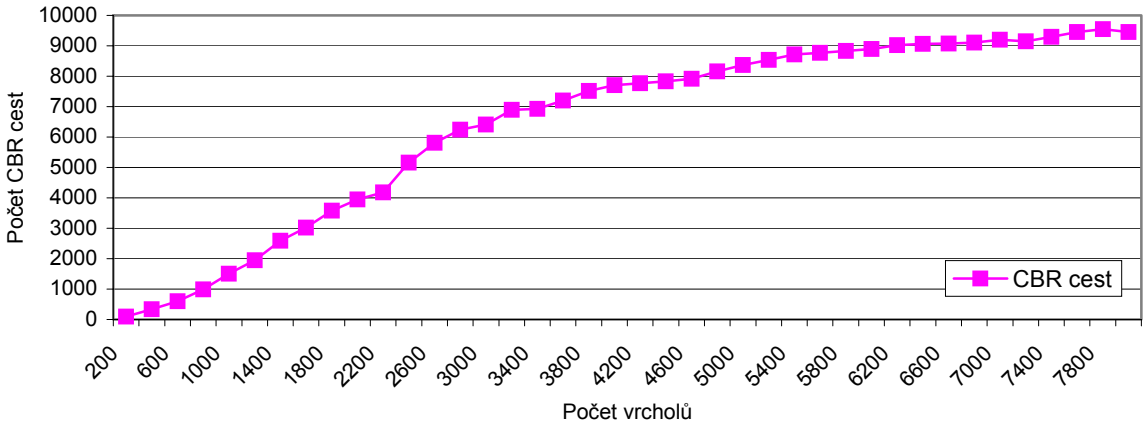


8.2 TEST ČETNOSTI VÝBĚRU METODY PLÁNOVÁNÍ

Četnost použití metody případového usuzování závisí opět na pokrytí prostředí případovým grafem. Čím více bude prostředí pokryto, tím pravděpodobněji půjde použít případové usuzování. V tomto testu byly na mapě prostředí o rozměrech 200×200 konstruovány případové grafy s postupně rostoucím počtem vrcholů. Každý z nich byl použit pro naplánování série 10000 cest s náhodnými body startu a cíle. Získané hodnoty jsou průměrem ze tří sérií testů s různými případovými grafy.

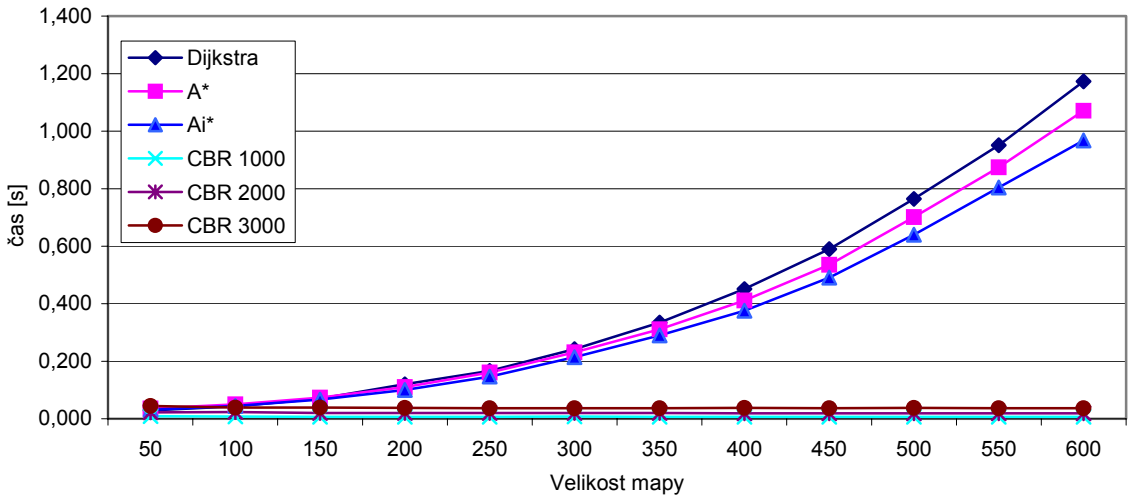
Výsledkem je počet cest, které bylo možné naplánovat pomocí případového usuzování. Neúspěchy přitom představují jednak cesty, kde nebylo možné v rámci zadaných adaptačních parametrů (velikost okolí 10, plná adaptace) nalézt napojení na případový graf, jednak cesty, které byly zamítnuty jako málo kvalitní na základě poměru jejich délky k délce ideální cesty (8-směrová metrika, poměr 2 a větší).

Z výsledků tohoto testu vyplývá, že již při přibližně 6% obsazení buněk vrcholy případového grafu (předpokládá se víceméně rovnoměrné rozložení vrcholů v mapě) lze asi 50% cest naplánovat pomocí případového usuzování. Při 10% pokrytí to bude už 75% úspěšných plánů a při 15% bude možné naplánovat případovým usuzováním 90% cest.



8.3 POROVNÁNÍ DOB PLÁNOVÁNÍ

Srovnání dob nutných k naplánování cesty bylo prováděno postupně na čtvercových mapách o rozměrech 50×50 – 600×600, obsahujících 20% překážek. V několika sériích byly voleny pevné body startu a cíle, mezi kterými bylo možné naplánovat cestu případovým usuzováním. Tatáž cesta byla plánována opakovaně 1000krát. Dosažené časy jsou průměrem všech sérií. Výsledky případového usuzování jsou uvedeny pro případové grafy s 1000, 2000 resp. 3000 vrcholy.



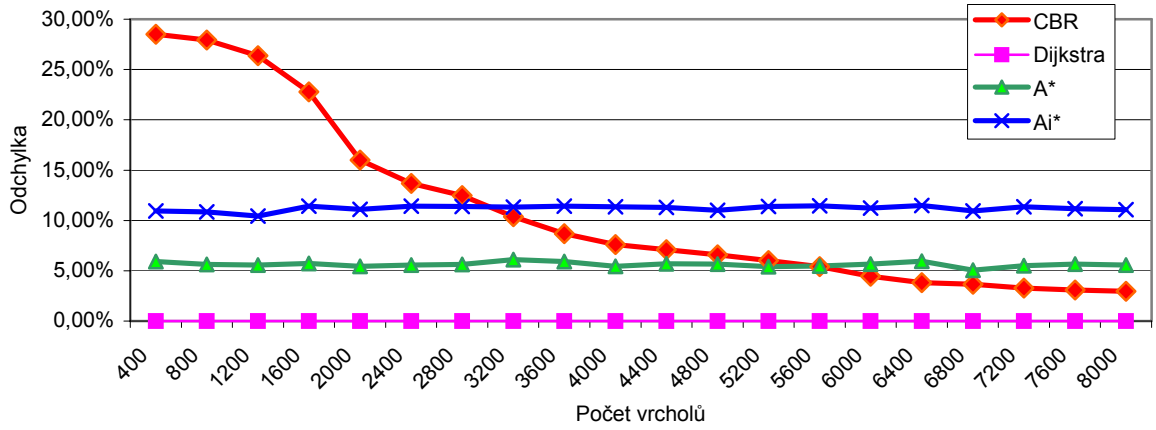
Je patrné, že rychlost plánování případovým usuzováním je ovlivněna velikostí mapy jen minimálně. Závisí především na počtu vrcholů a hran případového grafu. U alternativních metod plánování se ale časová náročnost exponenciálně zvyšuje. Pro konkrétní velikost mapy je tedy vhodné stanovit určitou hranici počtu vrcholů, z níž vyplyne i v podstatě konstantní doba plánování. Z výsledků prvního testu například vyplývá, že pro mapu 200×200 je maximální rozumnou hranicí cca 4000 vrcholů resp. 6000 vrcholů při použití plné resp. výběrové adaptace. Tato hranice by pak měla být udržována pomocí nástrojů údržby případové báze.

8.4 POROVNÁNÍ KVALITY NAPLÁNOVANÝCH CEST

Porovnání kvality plánovaných cest bylo provedeno opět ve vztahu k zaplněnosti případového grafu. Byly náhodně generovány takové startovní a cílové pozice, z nichž bylo možné provést plánování cesty i případovým usuzováním. Takových zadání bylo generováno 1000 pro každý případový graf. Mezi každou vygenerovanou dvojicí bodů byly cesty plánovány všemi čtyřmi metodami. U každé cesty byly sledovány tři ukazatele: náklady na cestu, délka cesty a počet změn směru jízdy. V každé sérii byla pro každý ukazatel vybrána metoda s minimální hodnotou, k níž pak byly dopočítány procentuální odchylky metod ostatních. Průměr těchto odchylek pro všech 1000 cest tvoří výsledek jednotlivých testů.

Náklady na cestu

Náklady jsou vypočteny jako součin délky průjezdu buňkami a rizika.



Při 5% pokrytí prostředí vrcholy je odchylka nákladů kolem 15% od ideálních nákladů dosažených Dijkstrovým algoritmem. Při 10% pokrytí mapy klesá odchylka pod 10% a při 15% pokrytí dokonce pod 5% od nejlepších dosažitelných nákladů.

Délka cesty

Vzhledem ke způsobu konstrukce heuristiky dosahoval téměř vždy nejlepších výsledků z hlediska délky cesty algoritmus A*. I v tomto testu, při přibližně 10% pokrytí mapy vrcholy případového grafu, klesá odchylka délky cest plánovaných případovým usuzováním pod 10%, což logicky koresponduje s výsledky předchozího kritéria.

Počet změn směru jízdy

U počtu změn směru jízdy je metoda případového usuzování značně závislá na výběru alternativní metody plánování. Prezentované výsledky byly dosaženy při použití případových grafů, k jejichž konstrukci se jako alternativní metoda používal Dijkstrův algoritmus. K jeho výsledkům se tak s postupným zaplňováním případového grafu lze přiblížit při cca 15% pokrytí mapy vrcholy případového grafu. Lepší výsledky s obdobným tvarem grafu lze obdržet při použití alternativní metody A*. V takovém případě dochází navíc k souběžnému poklesu délky cest, ale také k růstu nákladů podobně, jako je tomu u samotného algoritmu A*.

9 ZÁVĚR

Tato práce se zabývá posouzením využitelnosti metod případového usuzování pro řešení navigačních úloh autonomních mobilních robotů. Zaměřuje se především na globální plánování cest. Cesty jsou plánovány v prostředí reprezentovaném pravoúhlou mřížkou. Použití této reprezentace umožňuje rychlejší a jednodušší práci s případy za cenu malého zhoršení kvality plánovaných cest (cca 4%). Obecné principy popisované v této práci jsou však použitelné i pro spojité prostředí.

V rámci práce byla na základě současného stavu výzkumu této problematiky navržena a z větší části také implementována metoda pro plánování cest případovým usuzováním. Tato metoda využívá speciální formu případové báze v podobě případového grafu. Dále byly implementovány alternativní metody plánování cest pro podporu případového usuzování v situacích, kdy nelze využít případový graf. Jedná se o Dijkstrův algoritmus, algoritmus A* a jeho modifikaci s celočíselnou heuristikou nazvanou A_i*.

V prostředí se předpokládá pohyb holonomního robotu. Pro neholonomní roboty, jejichž omezení se týkají možných směrů pohybu, však stačí modifikovat použité grafové algoritmy tak, aby braly v úvahu i tato kinematická omezení.

V realizovaném prostředí byly provedeny experimenty pro ověření výkonu metod případového usuzování. Z těchto experimentů vyplynulo, že při rozumné velikosti případového grafu znamená použití případového usuzování značnou úsporu výpočetního času. Většina požadavků na přesnost, kvalitu i dobu plánování je splněna již při 5-15% rovnoměrném pokrytí prostředí vrcholy případového grafu.

Přestože byly v implementaci používány velmi efektivní postupy zpracování dat, je problematika konstrukce a údržby případového grafu mnohem rozsáhlejší, než bylo možné popsat a řešit v této práci. Lze uvažovat množství dalších metod pro vylepšení práce případového usuzování. Bylo by například možné automaticky propojovat blízké hrany případového grafu tak, aby se co nejrychleji propojily izolované nebo nevhodně spojené části případového grafu. Bylo by také vhodné dále optimalizovat adaptační proces, protože se při vyšším pokrytí prostředí vrcholy a hranami výrazně podílí na čase nezbytném pro plánování.

V práci byly pro srovnávací experimenty použity pro plánování cest tři grafové algoritmy. Obecně lze ale konstatovat, že všechny metody plánování cesty, kde se s cestou pracuje explicitně, lze kombinovat s CBR.

Parametrem, který je potřeba výrazněji měnit pro různá prostředí, je pouze poměr délky naplánované cesty k cestě ideální. Ten může mít pro interiérová prostředí hodnotu i několik desítek. Jedná se o parametr, jehož adaptivní změny by musely podléhat složitému učení, a proto je v současnosti snazší stanovit jej pro určitý typ prostředí pevně na základě empirických zkušeností.

Nejčastějšími prostředími, která díky častému opakování podobných úloh umožňují ideální využití popisovaných metod, jsou průmyslové a kancelářské provozy. Zde lze také očekávat jejich největší praktické využití.

10 VÝBĚR Z POUŽITÉ LITERATURY

AAMODT, A.; PLAZA, E. 1994. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications*, 1994, vol. 7, no. 5, pp. 39-59.

AGIRREBEITIA, J.; AVILÉS, R.; DE BUSTOS, I. F.; AJURIA, G. 2005. A New APF Strategy for Path Planning in Environments with Obstacles. *Mechanism and Machine Theory*, 2005, vol. 40, issue 6, pp. 645-658.

ARKIN, R.C. 1998. *Behavior-Based Robotics.* 1st ed. Cambridge, MA, USA : The MIT Press, 1998. 464p.

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 : 2006, pp. 1831-1836.

DEMEL, J. 2002. *Grafy a jejich aplikace*. 1. vyd. Praha : Academia, 2002. 257 s.

FOX, S. E. 2000. A Unified CBR Architecture for Robot Navigation. In Enrico Blanzieri and Luigi Portinale (Eds.) *Advances in Case-Based Reasoning. 5th European Workshop, EWCBR 2000. Trento, Italy, September 6-9*. Berlin (Germany) : Springer-Verlag, 2000, pp. 406-417.

GEMEINDER, M.; GERKE M. 2003. GA-Based Path Planning for Mobile Robot Systems Employing an Active Search Algorithm. *Applied Soft Computing*, 2003, vol. 3, pp. 149-158.

GERAERTS, R.; OVERMARS, M. H. 2006. Sampling and Node Adding in Probabilistic Roadmap Planners. *Robotics and Autonomous Systems*, 2006, vol. 54, issue 2, pp. 165-173.

HAIGH, K. Z.; SHEWCHUK, J. R. 1994. Geometric Similarity Metrics for Case-Based Reasoning. In *Case-Based Reasoning: Working Notes from the AAAI-94 Workshop, Seattle, WA, August*. Seattle : AAAI Press, 1994, pp. 182-187.

KOSE, M.; ACAN, A. 2004. Knowledge Incorporation into ACO-Based Autonomous Mobile Robot Navigation. In *Lecture Notes in Computer Science. Computer and Information Sciences - ISCIS 2004*. 2004, vol. 3280/2004.

KRUUSMAA, M.; SVENSSON, B. 1998a. Combined Map-Based and Case-Based Path Planning for Mobile Robot Navigation. In Vidyasagar M. (Ed): *Proceedings of International Symposium of Intelligent Robotic Systems*. Tata McGraw Hill : 1998, pp. 11-16.

KRUUSMAA, M.; SVENSSON, B. 1998b. Using Case-Based Reasoning for Mobile Robot Navigation. In *Proceedings of the 6th German Workshop on Case-Based Reasoning, March 6-8*. Berlin : 1998, 8 pp.

LATOMBE, J. C. 1991. *Robot Motion Planning*., Norwell, MA : Kluwer Academic Publishers, 1991.

LAVALLE, S. M. 1998. *Rapidly-Exploring Random Trees: A New Tool for Path Planning. TR 98-11*. Computer Science Dept., Iowa State University, 1998.

LAVALLE, S. M. 2006. *Planning Algorithms*. 2006, Cambridge University Press. 842 p.

LI, Y.; CHEN, X. 2005. Mobile Robot Navigation Using Particle Swarm Optimization and Adaptive NN. In *Lecture Notes in Computer Science. Swarm Intelligence and Intelligent Agents*. 2005, vol. 3612/2005.

LIKHACHEV, M.; KAESS, M.; ARKIN, R. C. 2002. Learning Behavioral Parameterization Using Spatio-Temporal Case-Based Reasoning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. 2002, Vol. 2, pp. 1282-1289.

LOUIS, S.J.; LI, G. 1997. Combining Robot Control Strategies Using Genetic Algorithms with Memory. In *Proceedings of the 6th International Conference Evolutionary Programming, Indianapolis, IN, April 13-16*. Berlin (Germany) : Springer-Verlag, 1997, pp. 431-441.

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*. 2003, vol. 4, pp. 283-317.

MICARELLI, A.; PANZIERI, S.; SCIAVICCO, L.; ULIVI, G. 2001. Landmark Recognition in Indoor Navigation by Fuzzy Maps and CBR. In *RAMSETE: Articulated and Mobile Robotics for Services and Technologies*. 2001, vol. 270/2001.

RAM, A.; SANTAMARÍA, J. C. 1997. Continuous Case-Based Reasoning. *Artificial Intelligence*. 1997, v. 90, No. 1-2, pp. 25-77.

SRINIVASAN, T.; AARTHI, K.; AISHWARYA MEENAKSHI, S.; KAUSALYA, M. 2006. CBRRoboSoc: An Efficient Planning Strategy for Robotic Soccer Using Case Based Reasoning. In *Proceedings of the International Conference on Computational Inteligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce (CIMCA'06)*. November 2006, pp. 113.

SUPIC, H.; RIBARIC, S. 2001. Adaptation by Applying Behavior Routines and Motion Strategies in Autonomous Navigation. In David W. Aha and Ian Watson (Eds.). *Case-Based Reasoning. Research and Development. 4th international Conference on Case-Based Reasoning, ICCBR 2001*, *Vancouver, BC, Canada, July 30 - August 2*. Berlin (Germany) : Springer-Verlag, 2001, pp. 517-530.

ŠEDA, M.; DVOŘÁK, J. 2003. Robot Navigation Using Genetic Algorithm and Case-Based Reasoning. In *Book of Extended Abstracts of the National Conference*

with International Participation Engineering Mechanics 2003. Svratka : Association for Engineering Mechanics, 2003, pp. 328-329 + 8 pp. on CD-ROM.

TAMBOURATZIS, T. 2007. Progressive Optimisation of Organised Colonies of Ants for Robot Navigation: An Inspiration from Nature. In *Lecture Notes in Computer Science. Adaptive and Natural Computing Algorithms*. 2007, vol. 4432/2007.

URDIALES, C.; PEREZ, E. J.; VÁZQUEZ-SALCEDA, J.; SÀNCHEZ-MARRÈ, M.; SANDOVAL, F. 2006. A Purely Reactive Navigation Scheme for Dynamic Environments Using Case-Based Reasoning. *Autonomous Robots*, August 2006, vol. 21, no. 1.

ZHENG, C.; DING, M.; ZHOU, C.; LI, L. 2004. Coevolving and Cooperating Path Planner for Multiple Unmanned Air Vehicles. *Engineering Applications of Artificial Intelligence*. 2004, vol. 17, issue 8, pp. 887-896.

11 PUBLIKACE AUTORA K TÉMATU DISERTACE

HODÁL, J.; DVOŘÁK, J. Podobnostní míry v případovém usuzování. In *Sborník přednášek k 6. ročníku konference Inteligentní systémy pro praxi, Luhačovice, 06.11.2001-07.11.2001*. AD&M, konferenční servis, 2001, pp.227-234. ISBN 80-238-7812-3.

HODÁL, J.; DVOŘÁK, J. Plánování cesty robota pomocí případového usuzování. In *Proceedings of XXVth International Autumn Colloquium Advanced Simulation of Systems ASIS 2003, Sv.Hostýn, 08.09.2003-10.09.2003*. MARQ, 2003, pp.173-178. ISBN 80-85988-88-7.

HODÁL, J.; DVOŘÁK, J., KRČEK, P. Systém pro plánování cesty robota případovým usuzováním. In *Proceedings of XXVIIth Internatinal Autumn Colloquium Advanced Simulation of Systems ASIS 2005, Sv. Hostýn, 06.09.2005-08.09.2005*. MARQ, 2005, pp.255-260. ISBN 80-86840-16-6.

HODÁL, J.; DVOŘÁK, J. The Use of Case-Based Reasoning to Robot Navigation. Internetový časopis *Cybernetic Letters*, 3/2007, http://cybletter.com/. ISSN 1802-3525.

HODÁL, J., DVOŘÁK, J. Using Case-Based Reasoning for Mobile Robot Path Planning. Přijato k publikaci v časopise *Engineering Mechanics*. ISSN 1802-1484.

AUTOROVO CV

Jméno: **Ing. Jaroslav Hodál**

Narozen: 17. 5. 1977 v Boskovicích, okr. Blansko

Bydliště: Hybešova 23, Blansko 678 01

Vzdělání

2000 – 2007: doktorské studium, Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav automatizace a informatiky

1995 – 2000: magisterské studium, Vysoké učení technické v Brně, Fakulta strojního inženýrství, obor Matematické inženýrství, studium zakončeno státní zkouškou 19. 6. 2000

1991 – 1995: Gymnázium Blansko

Dosavadní praxe

1998 – 1999: TECO Brno, programátor

2000 – 2003: VUT v Brně, výuka programování a expertních systémů.

od 2000: SPŠ a VOŠ Technická, Sokolská 1, Brno, učitel, předměty programování a výpočetní technika, internet.

ABSTRACT

This work deals with using case-based reasoning for autonomous mobile robot navigation. The navigation involves finding and passing as short as possible and also as easy as possible path between two selected points in a given environment without collisions with obstacles, which may be found in this environment.

The key role in choosing navigation methods is played by the representation of environment, in which the robot locomotes. Therefore the considerable place is devoted to different ways of environment modeling in the first part of this work. Topological models represented by visibility graphs, tangent graphs and Voronoi diagrams are described as well as metric (grid) models. Particularly grid representations created by exact and approximative decomposition methods of environment into the cells are described.

This work also deals with a division of path-planning methods. The main attention is devoted to global path-planning methods. The description of these methods includes also a survey of works, which use these methods for solving navigation tasks.

The main idea of case-based reasoning is a presumption that similar tasks probably also have similar solutions. New tasks are solved by adapting old proved solutions of similar tasks to conditions of the new task. Tasks and their solutions constitute so-called cases. Cases are stored in a case-base, which is one of the fundamental knowledge containers of case-based reasoning. The operation of case-based reasoning systems is described by a common CBR cycle. Four steps typically define the CBR cycle – retrieve, reuse, revise and retain.

Considering that similar tasks are often solved within path planning, it is possible to expect advantages by using case-based reasoning methods in these tasks. In this work is closely documented the contemporary state of research on the field of using case-based reasoning for robot navigation on global and local level.

The focal point of this work is the proposition of the path planning method based on case-based reasoning combined with graph algorithms in the environment represented by the rectangular grid. Paths or their parts constitute the cases. The case-base has form of a case-graph, which is formed from linear parts of the paths passed in the past. Vertices of the case-graph are created from the start and goal points as well as from points in which the path direction changes. During planning the path between new start and goal points, connections to the case-graph from these points are found in the adaptation phase of the CBR. Known graph algorithms can realize path finding in the case-graph subsequently.

Parts of the proposed method, which are necessary for verifying case-based reasoning operation, are implemented in the software appendix. Various experiments were performed by means of implemented test program. Results and conclusions of these experiments are also included in this work. On the basis of obtained results it is possible to say that case-based reasoning can significantly save computation costs particularly in large environments.