

Robotická evoluce

Jan Černý

Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University

Seminář strojového učení a modelování



Obsah

- 1 Úvod
- 2 Metody
 - Robotická simulace
 - Evoluční algoritmus
 - Kontrolní algoritmy
- 3 Výsledky
- 4 Další vývoj
- 5 Závěr



Motivation

- Chodící roboti mají lepší mobilitu v komplexním terénu
- Plánování jejich pohybů je ale mnohem složitější
- Pohyb může být složen z pohybových primitivů
- Jejichž návrh je složitý
- Potřeba systému schopného automaticky navrhovat pohybové primitivy



Současný přístup

- Ruční návrh pohybu
- Central Pattern Generators
 - Jednoduché neuronové sítě
 - Periodický signál s různou fází
- HyperNEAT
 - Nepřímé kódování ANN
 - Dokáže zohledňovat symetrie a tvar robota



Současný přístup

- Ruční návrh pohybu
- Central Pattern Generators
 - Jednoduché neuronové sítě
 - Periodický signál s různou fází
- HyperNEAT
 - Nepřímé kódování ANN
 - Dokáže zohledňovat symetrie a tvar robota



Současný přístup

- Ruční návrh pohybu
- Central Pattern Generators
 - Jednoduché neuronové sítě
 - Periodický signál s různou fází
- HyperNEAT
 - Nepřímé kódování ANN
 - Dokáže zohledňovat symetrie a tvar robota



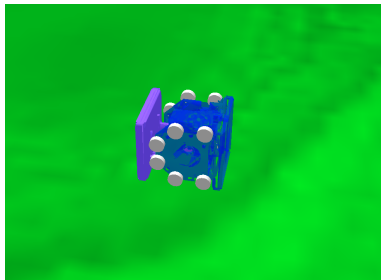
SIMBRION REPLICATOR

- SYMBRION a REPLICATOR
- Projekty společně vyvíjené na mnoha evropských univerzitách
- Výzkum multi-robotických organismů
- Zaměřeno na biologicky inspirované přístupy



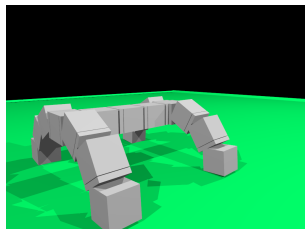
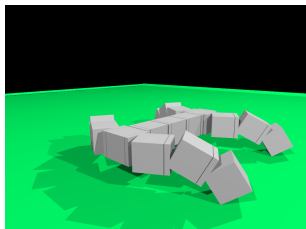
SSSA Robot

- Scuola Superiore Sant'Anna
- 8 kol pro pohyb na rovném povrchu
- 4 propojovací sloty
- Pohyblivé rameno s jedním stupněm volnosti
- Rozměry: 12x12x12 cm
- Hmotnost: 1.2 Kg



Complete Robotic Organism

- Trup složený z pěti bloků
- Čtyři končetiny, každá ze tří bloků
- Dva organismy s různou konfigurací končetin



Proč?

- Reální roboti jsou pomalí a nedostupní a drazí
- Vyžadují přítomnost člověka který řeší problémy
- V době experimentu se ještě nevráběli
- Řešením je použití počítačové simulace
- Simulace může být rychlejší než reálný svět
- Nedochází k přehřátí poruchám a není nutné vyměňovat baterie
- Většina problémů je řešitelná restartem



Proč?

- Reální roboti jsou pomalí a nedostupní a drazí
- Vyžadují přítomnost člověka který řeší problémy
- V době experimentu se ještě nevyroběli
- Řešením je použití počítačové simulace
- Simulace může být rychlejší než reálný svět
- Nedochozí k přehřátí poruchám a není nutné vyměňovat baterie
- Většina problémů je řešitelná restartem



Jak?

- Základem robotického simulátoru je simulace fyziky
- Další součástí je detekce kolizí
- Model světa a robota
- 3D rendering není komponenta simulátoru, ale hodí se
- Komunikační rozhraní stejné jako u reálných robotů
- Ovládací software by neměl rozpoznat zda řídí simulaci nebo robota
- Nedosažitelný cíl - simulace je pouze numerický výpočet s omezenou přesností



Jak?

- Základem robotického simulátoru je simulace fyziky
- Další součástí je detekce kolizí
- Model světa a robota
- 3D rendering není komponenta simulátoru, ale hodí se
- Komunikační rozhraní stejné jako u reálných robotů
- Ovládací software by neměl rozpoznat zda řídí simulaci nebo robota
- Nedosazitelný cíl - simulace je pouze numerický výpočet s omezenou přesností



Jak?

- Vstupem každého motoru je funkce
- Funkce lze vytvářet pomocí GP
- Robot je symetrický
- Symetrii lze využít k zjednodušení problému
- K nalezení symetrie lze použít další EA



Jak?

- Vstupem každého motoru je funkce
- Funkce lze vytvářet pomocí GP
- Robot je symetrický
- Symetrii lze využít k zjednodušení problému
- K nalezení symetrie lze použít další EA



Jak?

- Vstupem každého motoru je funkce
- Funkce lze vytvářet pomocí GP
- Robot je symetrický
- Symetrii lze využít k zjednodušení problému
- K nalezení symetrie lze použít další EA



Jak?

- Vstupem každého motoru je funkce
- Funkce lze vytvářet pomocí GP
- Robot je symetrický
- Symetrii lze využít k zjednodušení problému
- K nalezení symetrie lze použít další EA



Dekompozice problému

- Problém lze řešit jako dvě pod-úlohy
 - Hledání pohybového vzoru jedné nohy
 - Hledání optimální koordinace končetin
- Implementace - Koevoluce dvou populací

GP modul - evoluce populace jedinců reprezentujících pohyb jedné nohy robota

EA modul - jedinci reprezentují koordinaci končetin

Koevoluce - propojení obou algoritmů



- Generační evoluční model s *ramped-half-and-half* inicializací
- Genetické operátory:
 - funkce: sin, cos, +, -, *, /, unární -, min, max
 - terminály: čas, π , konstanty $\langle -1, 1 \rangle$
- Každý segment se může otáčet od $-\frac{\pi}{2}$ do $\frac{\pi}{2}$
- Tělo je zafixované a není součástí evoluce



- Steady state evoluční model.
- Re prezentace - vektor n skupin parametrů

$$[d_1 \dots d_i, \varphi_1 \dots \varphi_i]$$

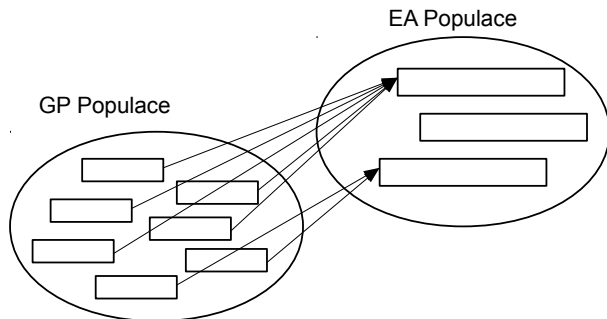
určujících koordinaci končetin.

- d_i směr pohybu končetiny i
- φ_i fáze pohybu končetiny i
- Genetické operátory:
 - Turnajová selekce
 - Jednobodové křížení pro binární parametry
 - Blend crossover pro reálné parametry



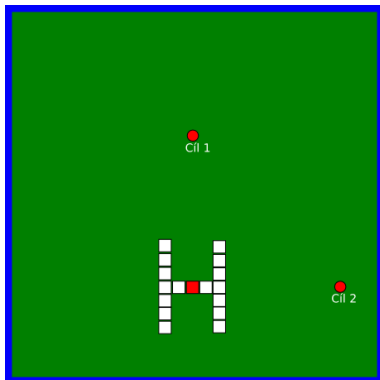
Koevoluce

- Jedinci z obou algoritmu nemají samostatně význam, je nutné je spárovat
- Hledání párů je výpočetně náročné, není možné testovat každého s každým
- Párování se mění během evoluce
- GP modul na EA modul je mapován 1 na 1
- EA modul na GP modul je mapován 0 na n



Experimentální ohodnocení

- Všechny experimenty v simulovaném prostředí
- Robot vždy začíná ze stejné pozice
- Úkolem je přesunout se na zadaný bod
- Je vyžadován pohyb který připomíná chůzi
- Cíl je nad zemí
- Penalizace pokud se tělo dotkne země



Kontrolní algoritmy

- Pro kontrolu bylo implementováno několik alternativních přístupů:
 - Central Pattern Generators [CPG] (periodické signály v různé fázi)
 - GP bez koevoluce (12 podstromů)
 - CPG s využitím symetrie



Kontrolní algoritmy

- Pro kontrolu bylo implementováno několik alternativních přístupů:
- Central Pattern Generators [CPG] (periodické signály v různé fázi)
- GP bez koevoluce (12 podstromů)
- CPG s využitím symetrie

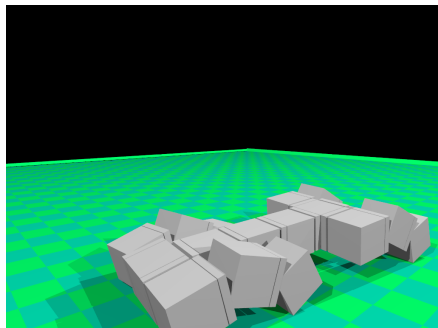
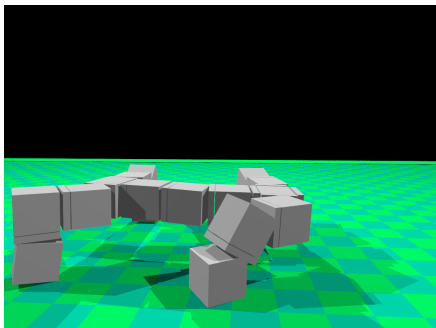


Nastavení experimentů

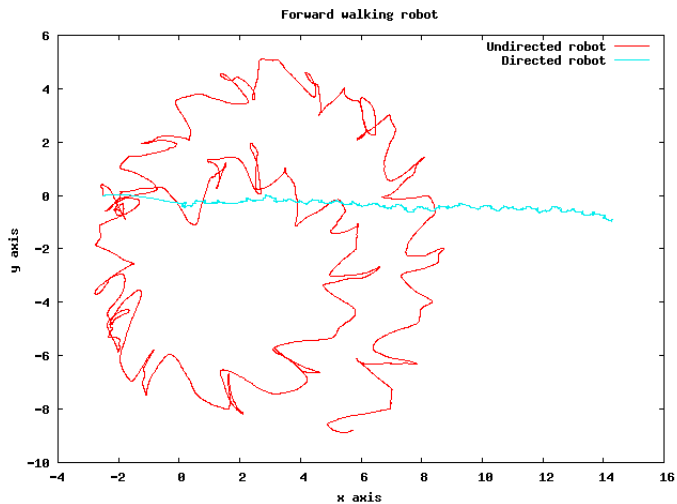
- Genetické programování
 - Velikost populace: 500 jedinců
 - Počet generací: 200
 - Velikost turnaje: 7
- Evoluční algoritmus
 - Velikost populace: 50 jedinců
 - Velikost turnaje: 3
- Kontrolní algoritmy
 - Velikost populace: 600 jedinců
 - Počet generací: 250



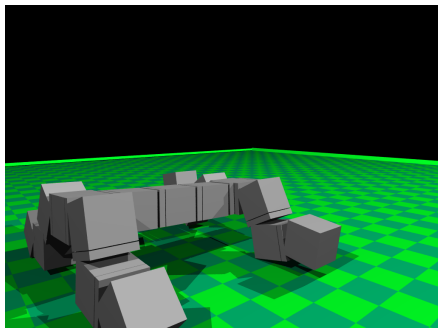
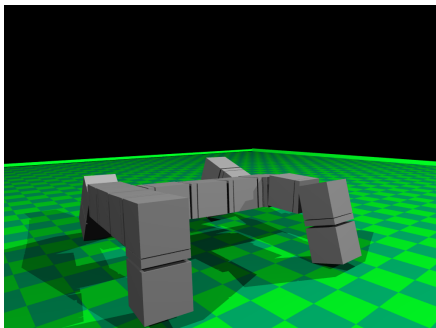
Robot plaz



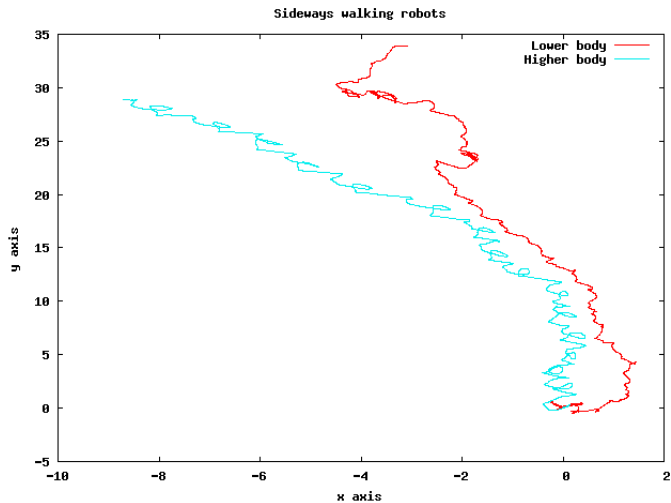
Trajektorie plaz



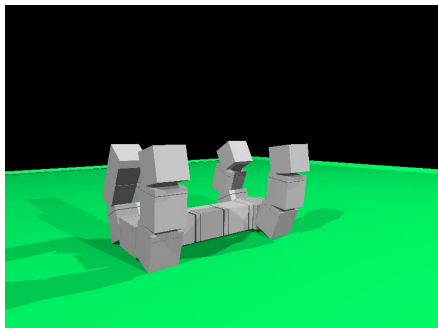
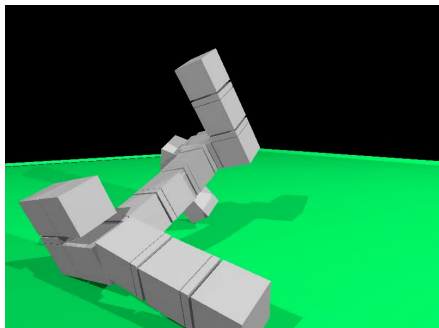
Robot krab



Trajektorie krab

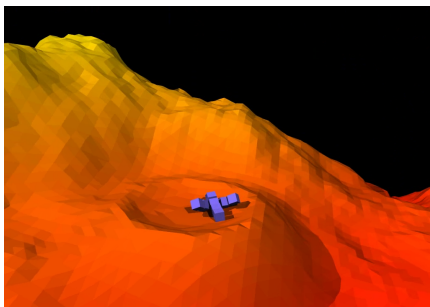


Kontrolní experimenty



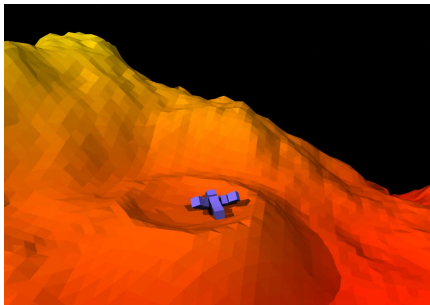
Další pohyby

- Pomocí změny fitness funkce je možné učit robota jiným pohybům
- Například chůze do zatáčky, úkroky, otáčení a další činnosti
- Změnou terénu je možné učit robota pohyb ve složitém prostředí
- Například:



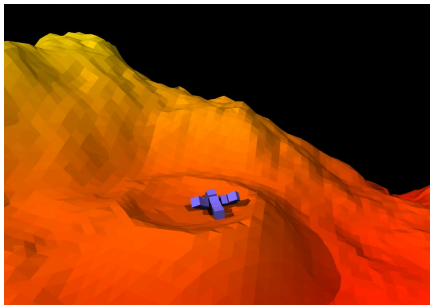
Další pohyby

- Pomocí změny fitness funkce je možné učit robota jiným pohybům
- Například chůze do zatáčky, úkroky, otáčení a další činnosti
- Změnou terénu je možné učit robota pohyb ve složitém prostředí
- Například:



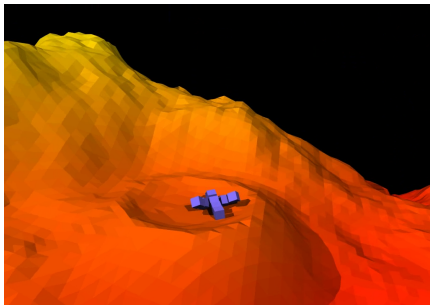
Další pohyby

- Pomocí změny fitness funkce je možné učit robota jiným pohybům
- Například chůze do zatáčky, úkroky, otáčení a další činnosti
- Změnou terénu je možné učit robota pohyb ve složitém prostředí
- Například:



Další pohyby

- Pomocí změny fitness funkce je možné učit robota jiným pohybům
- Například chůze do zatáčky, úkroky, otáčení a další činnosti
- Změnou terénu je možné učit robota pohyb ve složitém prostředí
- Například:



Plánování

- Učení složitějších akcí pomocí EA je neefektivní
- Složité akce jsou potřeba

Plánování



Plánování

- Učení složitějších akcí pomocí EA je neefektivní
- Složité akce jsou potřeba

Plánování



Plánování

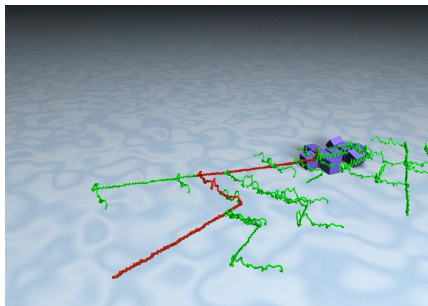
- Učení složitějších akcí pomocí EA je neefektivní
- Složité akce jsou potřeba

Plánování



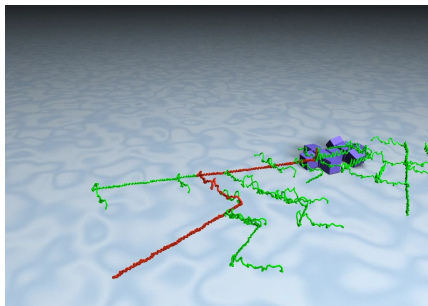
Plánování

- Vytváří se strom stavů
- Hledání nejkratší cesty pomocí A*
- Prohledávání celého prostoru možných vstupů - je jich moc
- Použití menšího množství náhodně generovaných vstupů - nejsou efektivní
- Použití připravených pohybových primitivů pro plánovač



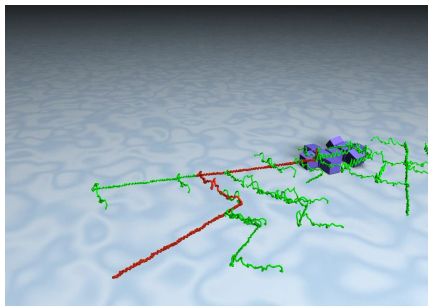
Plánování

- Vytváří se strom stavů
- Hledání nejkratší cesty pomocí A*
- Prohledávání celého prostoru možných vstupů - je jich moc
- Použití menšího množství náhodně generovaných vstupů - nejsou efektivní
- Použití připravených pohybových primitivů pro plánovač



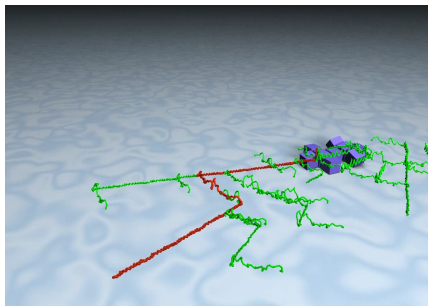
Plánování

- Vytváří se strom stavů
- Hledání nejkratší cesty pomocí A*
- Prohledávání celého prostoru možných vstupů - je jich moc
- Použití menšího množství náhodně generovaných vstupů - nejsou efektivní
- Použití připravených pohybových primitivů pro plánovač



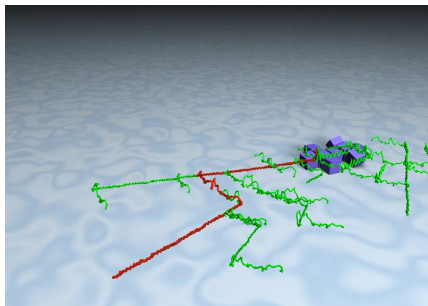
Plánování

- Vytváří se strom stavů
- Hledání nejkratší cesty pomocí A*
- Prohledávání celého prostoru možných vstupů - je jich moc
- Použití menšího množství náhodně generovaných vstupů - nejsou efektivní
- Použití připravených pohybových primitivů pro plánovač



Plánování

- Vytváří se strom stavů
- Hledání nejkratší cesty pomocí A*
- Prohledávání celého prostoru možných vstupů - je jich moc
- Použití menšího množství náhodně generovaných vstupů - nejsou efektivní
- Použití připravených pohybových primitivů pro plánovač



Závěr

- EA jsou vhodné pro hledání pohybových primitivů
- S pomocí plánování lze tyto primitivy využít pro složitější úkoly
- Stále dost nezodpovězených otázek

