

Typované funkcionální genetické programování

Diplomová práce

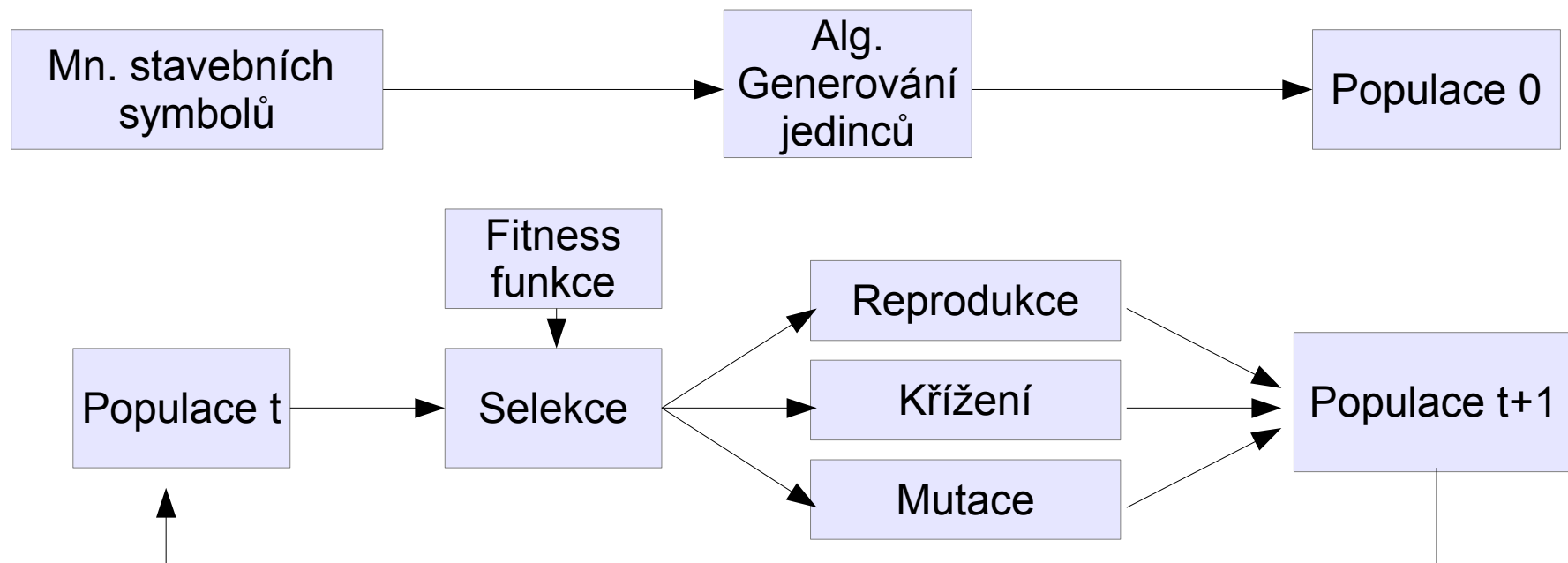
Tomáš Křen

Cíl

- Navrhnout a implementovat prototyp systému řešící úlohu genetického programování nad typovaným funkcionálním jazykem.
- Simply typed lambda calculus
- Zobecnění standardního GP (*Koza, 1992*)
- Hlavní důraz na generování jedinců
- Sekundární důraz na křížení

Co je to Genetické Programování

- Prohledávací metaheuristika inspirovaná přirozeným výběrem v přírodě
- Jedinec/řešení je syntaktický strom programu



Co přináší typy

- Standardní GP omezuje symboly stavební množiny tím, že požaduje aby to byly operace, konstanty a proměnné nad jediným typem A
 - tzn. konstanty a proměnné typu A
 - funkce typu $A \rightarrow \dots \rightarrow A \rightarrow A$ (neboli $(A \times \dots \times A) \rightarrow A$)
- Typované GP toto omezení odstraňuje

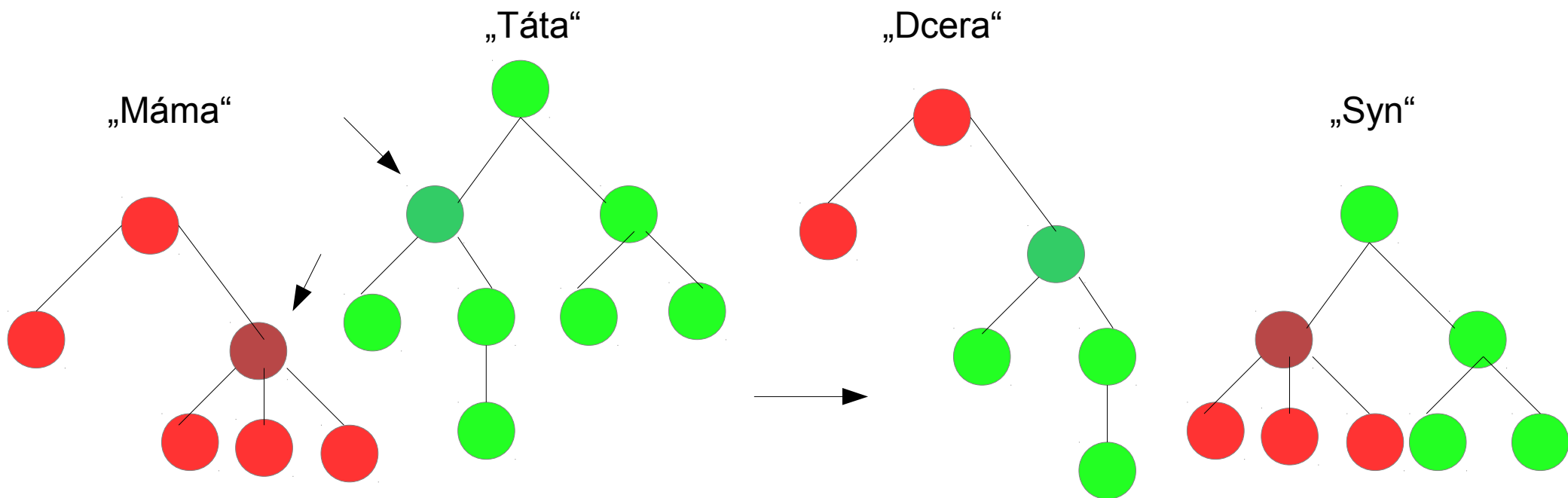
Podářilo se

- navrhnout metodu pro generování jedinců parametrizovanou jednoduchou prohledávací strategií.
 - Strategie pro standardní *Ramped half-and-half* generování
 - *Systematická strategie* generující jedince systematicky podle velikosti
 - Nová *Geometrická strategie* vykazující pěkné vlastnosti jako redukce bloatu
- navrhnout několik vylepšení založených na teoretickém pozadí, které se ukázaly mít kladný vliv na výkon systému
 - η -normalizace
 - Reprezentace jedinců jako @-stromy
- navrhnout zobecnění standardního podstromy prohazujícího křížení založené na algoritmu *eliminace abstrakcí*

Generování jedinců

- Použito rozšíření A^* algoritmu.
- Ve stavovém prostoru „nekompletních“ lambda termů se hledají kompletní lambda termy.
- A^* rozšířen o „prohledávací strategii“, která má možnost zahodit některé následníky stavu a zmenší tak prohledávaný prostor.
- Použitá metoda generuje jedince v dlouhé normální formě, čímž brání generování ekvivalentních jedinců ve smyslu β a η redukcí.

Standardní křížení



Křížení typovaných lambda termů

- Prohazují se jen podstromy stejného typu
- Hlavní problém: volné proměnné v prohazovaných podstromech
- Řešení: proměnných se zbavíme pomocí *eliminace abstrakcí*
 - Převeďte term na ekvivalentní term neobsahující žádné proměnné ani lambda abstrakce (pomocí kombinátorů S,K,I)

Výsledky

- Tři měřené experimenty (+ jeden konceptuální)
- Použité metriky
 - Pravděpodobnost nalezení korektního řešení
 - Průměrná velikost jedince
 - Počet jedinců, které je potřeba vyhodnotit aby bylo nalezeno korektní řešení s pravděpodobností 99%
 - Doba běhu

Artificial Ant problém

$\sigma = \text{AntAction}$

$\Gamma = \{ \quad l : \text{AntAction},$

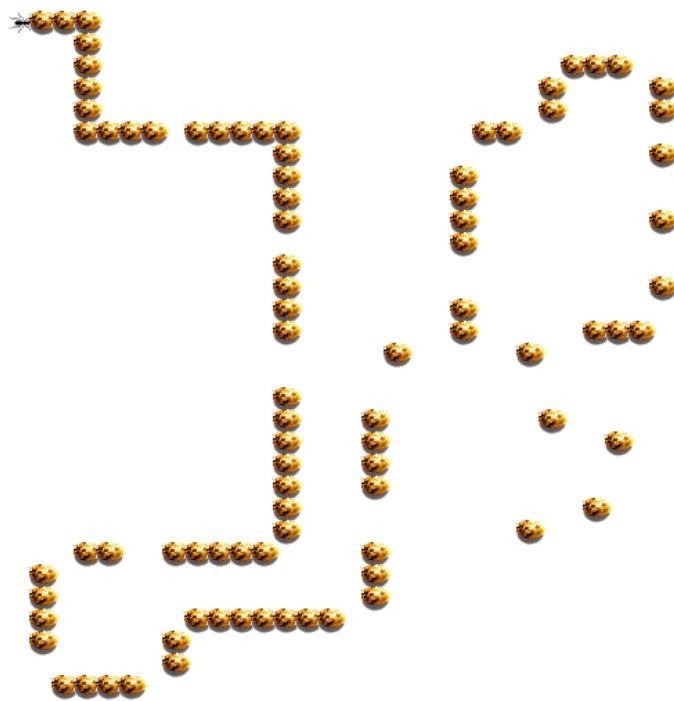
$\quad r : \text{AntAction},$

$\quad m : \text{AntAction},$

$\quad ifa : \text{AntAction} \rightarrow \text{AntAction} \rightarrow \text{AntAction},$

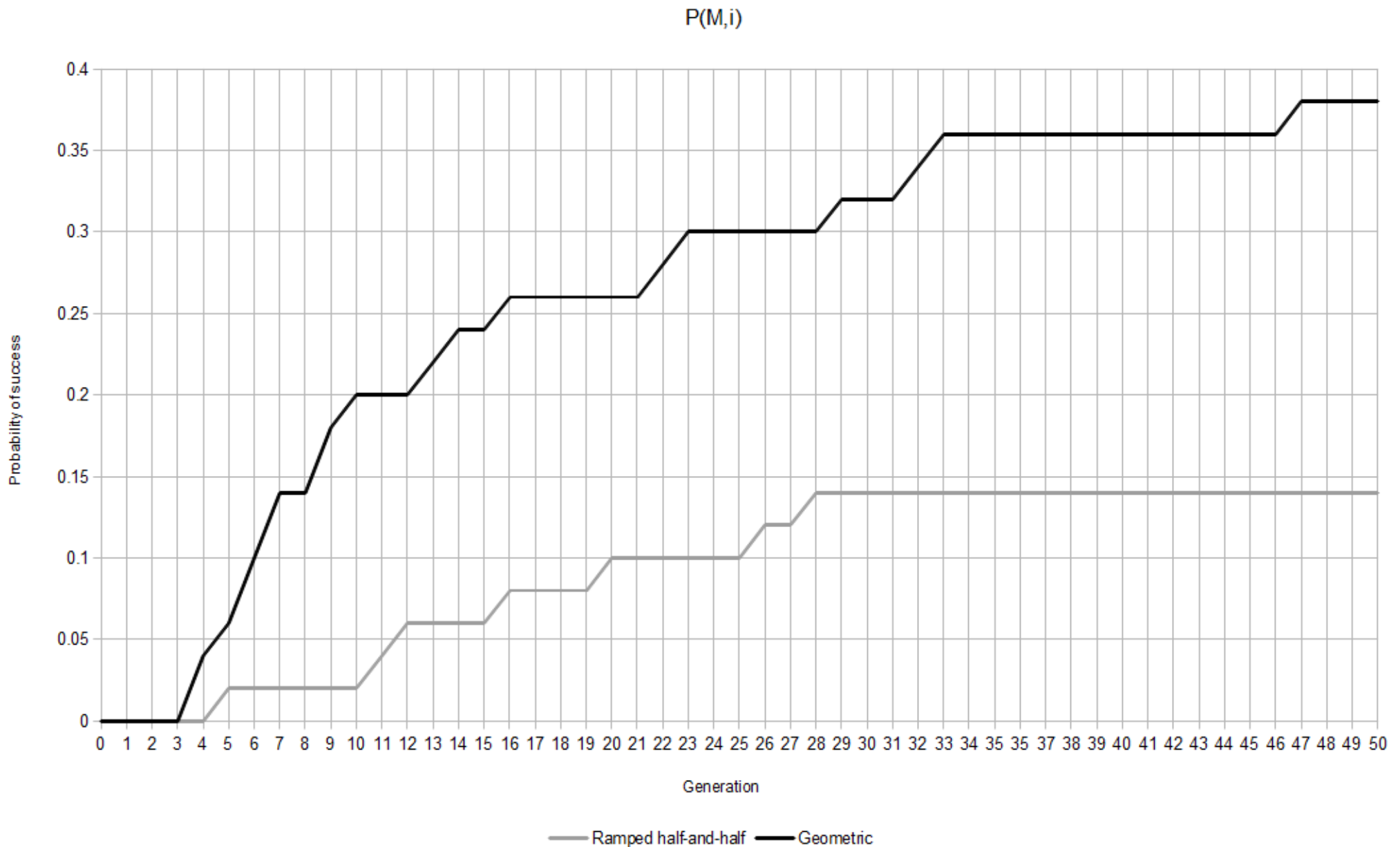
$\quad p2 : \text{AntAction} \rightarrow \text{AntAction} \rightarrow \text{AntAction},$

$\quad p3 : \text{AntAction} \rightarrow \text{AntAction} \rightarrow \text{AntAction} \rightarrow \text{AntAction} \}$



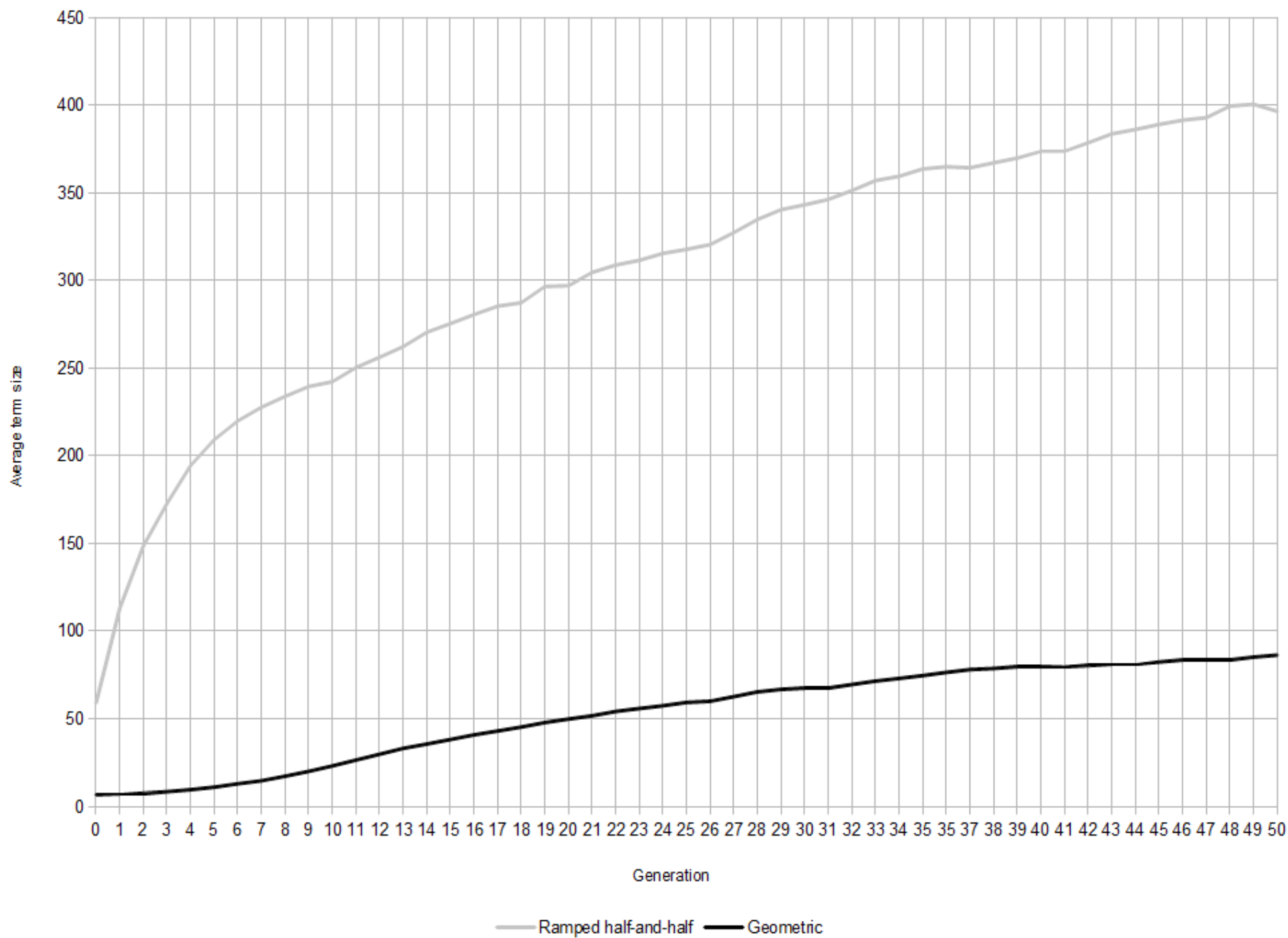
Artificial Ant problém

pravděpodobnost nalezení korektního řešení



Artificial Ant problém

Průměrná velikost jedince



Závěrem

- Experimenty mimo jiné ukázaly
 - Žádoucí vlastnosti Geometrické strategie
 - Ve všech sledovaných metrikách.
 - Výhodnost η -normalizace a „@-stromové“ reprezentace
 - Srovnatelné (o něco horší) výsledky v třetím experimentu oproti specializovanější metodě typovaného programování (Yu,2001)

Děkuji za pozornost.