



# Posudek oponenta závěrečné práce

<b>Oponent práce:</b>	Mgr. Jan Starý, Ph.D.
<b>Student:</b>	Štěpán Tupý
<b>Název práce:</b>	Multi-agentní hledání cest s více cíli pomocí výrokové splnitelnosti
<b>Obor / specializace:</b>	Znalostní inženýrství
<b>Vytvořeno dne:</b>	12. června 2023

## Hodnotící kritéria

### 1. Splnění zadání

- ▶ [1] zadání splněno
- [2] zadání splněno s menšími výhradami
- [3] zadání splněno s většími výhradami
- [4] zadání nesplněno

### 2. Písemná část práce

70/100 (C)

Práce je sepsána poměrně přehledně, obsahuje nicméně řadu věcných, logických i jazykových nedostatků.

V úvodu čteme "... [problém] je výpočetně náročný i pro jednoho agenta. V práci se budu soustředit právě na tuto variantu MAPF." (str. 1);  
zadání přitom spočívá v multi-agentní úloze.

Pojem "účelová funkce" (str. 4) je zdá se překladem "utility function", což je poněkud zavádějící překlad:  
utility function nemáme pro nějaký účel, je to funkce, která "měří užitek" (či naopak náklady).

Podle poznámky k obrázku 1.2 "by musel agent 1 obejít skoro celý ovál" (str. 6),  
v opačném směru ale přeci musejí obejít skoro celý ovál všichni ostatní agenti.

Na str. 11 čteme, že výkon solveru závisí na "... celkové struktuře booleovské formule, která nejde jendoduše zachytit",  
přitom booleovskou (výrokovou) formuli zachycuje např. binární strom; ostatně sami budeme takové formule konstruovat.

Vyloženě rušivě působí křovácký první pád ve formulacích jako "stačí vypočítat jedna cesta" (str. 17 a jinde),

"taková cesta jde najít", "jako heuristická funkce lze využít cena", "jeho cena lze nahradit jinou cenou" atd,  
místo kanonické pádové vazby koho/co vypočítat, nahradit atd.  
Podobně též nadužívané "tak" ve formulacích "pokud konflikty neobsahuje, tak je to platné řešení".

Na str. 14 se za "přípustnou" funkci bere taková, která "nikdy neodhadne vyšší cenu než je cena opravdová".  
V tom případě je konstantní minus patnáct přípustná funkce pro odhad ceny, což zřejmě není záměr.

Zvláštní je i zdůvodnění "jelikož ID je framework, tak jde použít i s jinými algoritmy" (str. 14) - kdyby ID nebyl "framework" (whatever that means), tak by se použít nedal?  
Podobně na str. 15 jsou "využity heuristické znalosti o doméně" - o kterou "doménu" se jedná  
a co se tím myslí se nedozvíme; myslí se zřejmě prostě informace ze zadání.

Z formulace "přestože je toto kódování nejjednodušší ... podporuje jednotkovou propagaci" (str. 20)  
není jasné, jestli snad "jednoduchost" nepovažujeme za nevýhodu.

Ve formuli 3.2 zřejmě chybí podmínka ( $v \neq w$ ), totiž jde o dvě různé hrany.

Smělé tvrzení, že "nevadí", pokud agentovi dovolíme vyskytovat se ve více vrcholech zároveň (str. 22)  
by zřejmě stálo za zevrubnější vysvětlení. Zároveň není pravda, že vytvořená formule obsahuje jen  
konjunkce, disjunkce a implikace (obsahuje i negace), ani to není důvodem,  
proč "jde lehce převést do CNF" (to jde jakákoli formule).

V sekci 3.5 (str. 22) se hovoří o MDD-SAT, ale teprve o dvě strany dále se dozvíme, co znamená M.D.D.

Na str. 25 se za výhodu SMT-CBS označuje schopnost "rychle dotvořit úplnou formuli",  
tu ale přeci známe předem, pokud chceme.

Podle 3.6.2 je SMT-CBS "výkonný na různých zadáních MAPF". Bylo by dosti smutné,  
kdyby fungoval jen pro jedno zadání, myslí se zřejmě různé typy zadání (tvar grafu,  
rozmístění agentů a cílů).

V definici 4.1 se pro graf  $G=(V,E)$ , podmnožinu  $U \subseteq V$  a vrchol  $v$  zavádí značení  $H_P(v, U)$   
pro Hamiltonovskou cestu,  
která začíná v daném vrcholu a navštíví všechny vrcholy z  $U$ . To je dosti zmatečná notace:  
takových cest přeci existuje více,  
která z nich je tedy  $H_P(v, U)$ ?

Stejná námitka platí pro Steinerův strom  $T_S(U)$  ve značení z definice 4.2:  
V pseudokódu na str. 31 se na řádku 24 a 38 počítá cena Steinerova stromu - kterého?  
V komentáři po definici 4.2 se naopak uvažuje  $\min\{Cena(T_S)\}$ , rozumí se zřejmě  
minimum přes všechny takové stromy.

Podle téhož komentáře je hledání minimálního Steinerova stromu NP-těžké - jaký smysl  
tedy má používat je místo Hamiltonovských cest?

Procházením všech možných pořadí daných cílů vnášíme do algoritmu prohledávání všech permutací, které roste faktoriálně, tedy rychleji než exponenciálně - stálo by zřejmě za komentář, proč tím nezvyšujeme složitost.

V popisu algoritmu 4.1.1 se vzdálenost značí  $N.g$ ; dříve se jako  $g()$  označují cíle (goals). Nešťastně též působí značení "N.cesta.cesta" v pseudokódu - první tečka je odkaz na člena struktury, druhá značí konkatenaci (s danou cestou).

Podle komentáře za definicí 4.3 "existence cesty jde potom zesponu odhadnout aproximovanou cenou Steinerova stromu" - myslí se zřejmě cena cesty (jak se zespona odhadne existence?), ale cena cesty snad bude typicky nižší než cena stromu.

Pokud ke kódující formuli přidáme podmínky 4.2 (agent musel z vrcholu vystoupit některou z hran), nemusíme už zdá se kromě počátečních a koncových pozic agentů ve formuli vůbec mluvit o vrcholech: totiž pohybem po hranách je přeci zároveň plně určen i pohyb po vrcholech.

Podle porovnání 4.3 "Steinerovy stromy špatně odhadují opravdovou cenu Hamiltonovských cest" - vrací se tedy otázka, proč je vůbec zkoušíme použít, když samo jejich hledání je navíc NP-těžké.

Popis grafů na obrázku 5.1 je vysloveně zmatečný: osa "instance MG-MAPF" neudává počet instancí, jak by se čtenář mohl domnívat, nýbrž pořadí té které instance vyřešené daným algoritmem. To znamená, že 100. instance jednoho algoritmu je jiná instance než 100. instance druhého, ostatně jiný algoritmus mohl (při daném časovém omezení) vyřešit jinou sadu instancí, čímž celé srovnání padá.

Kterých 200 (resp. 300) instancí to v kterém případě bylo se nedozvíme. V adresáři assignments je 1369 zadání: proč netestujeme s pomocí každého algoritmu všechna?

I samotné časové údaje jsou statisticky na pováženou: jakou přesnost má měření na jednotky sekund na úlohách od zlomků po jednotky a desítky sekund? Na úlohách běžících hodinu by takový výkyv byl zanedbatelný.

Líné kódování (5.3) zdá se ani nezaručuje deklarovaný cíl, totiž že agent navštívil své cíle: "řešení extrahované z odhodnocení formule  $H(\dots)$  nemusí pokrývat všechny cílové vrcholy". Jaký smysl má se o takové "řešení" vůbec pokoušet? Spoléháme na šťastnou náhodu, že třeba agent prošel všude tam, kde projít měl? V komentáři k výsledkům experimentů se potom chabý výkon při takovém kódování vysvětluje "zvýšenou režii spojenou s průběžnou konstrukcí formule" - přidat ke stávající formuli další klauzuli ale žádnou zvláštní "režii" nevyžaduje, jde nejspíše o to, že při "řešení", které řešením ani není, počítáme spoustu instancí (neúplného) SAT problému zbytečně.

O grafech 5.2 a 5.3 platí stejná námitka jako výše: např. 300 instancí vyřešených (během minuty)  
pomocí SMT-HCBS přeci není týchž 300 (z 350) instancí vyřešených pomocí SMT-HCBS-r, těžko na nich  
tedy porovnávat výkon. Graf 5.2 vpravo dole pochází zdá se z experimentu, který běžel s každým algoritmem  
nejdéle dvacet minut - co nám brání nechat experiment běžet 24 hodin a získat tak porovnání na mnohem větších  
(a tedy zajímavějších) instancích? To se týká i grafu 5.2 vpravo dole: pokud už se šesti agenty algoritmus  
během minuty nevyřeší jedinou instanci, proč jej nenecháme běžet déle, aby vůbec bylo co porovnávat?  
Experimenty zachycené na grafech 5.3 pak údaje o úspěšnosti neobsahují; v textu se uvádí, že ze zadání  
s počty agentů a cílů od 1 do 10 doběhly (myslí se zřejmě opět během minuty) "skoro všechny".  
Kolik bylo kterých? Proč nezvýšíme počet agentů až tam, kde úspěšnost začne klesat, jako v 5.2?  
Pokud na nejtěžších z těchto zadání docházela paměť (jak je možné, že nestačí 8GB?), která zadání to byla, kolik jich bylo pro ten který algoritmus a proč jsou potom výsledky nadále porovnatelné?  
Obecně: kapitola o těchto testech je nejslabší částí textu.

### 3. Nepísemná část, přílohy

80 /100 (B)

Hlavním SW výsledkem práce je implementace popsaných algoritmů napsaná nad SAT solverem Glucose (verze 4.2.1).

Ta prokazatelně funguje na obligátních zadáních (agenti v rohu čtverce, dvě křížící se cesty, protínající se cykly, atd).

V kódu je jasně vyčleněn vlastní kód od použitého Glucose.

Limit na jednu minutu běhu je zdá se zadržovaný do kódu:

bylo by sympatické mít takový timeout jako \_\_možnost\_\_ pro uživatele.

### 4. Hodnocení výsledků, jejich využitelnost

80 /100 (B)

Práce je zjevně příspěvkem v živém výzkumu aktuálních výpočetních problémů.

### Celkové hodnocení

80 /100 (B)

Přes všechny výše uvedené námitky považuji práci za mírně nadprůměrnou a navrhuji hraniční B.

### Otázky k obhajobě

1. Je pravda, že při "přísném" kódování, kdy agenti mohou chodit vždy po právě jedné hraně,

by se kódovací formule obešla bez výroků  $\chi(a, u, t)$ , rozumí se kromě počátečních a cílových pozic?

Jinými slovy, je pravda, že z takového popisu chození po hranách už plyne i jednoznačný popis

chození po vrcholech? Pokud ano, kolik výrokových proměnných by se tím ušetřilo?

2. Můžete uvést typickou instanci, kterou SMT-HCBS-L1 nebo L2 vyřeší rychleji nežli SMT-HCBS?

3. Při jakých počtech agentů klesne v experimentech 5.3 úspěšnost na nulu (tak jako v 5.2 klesne na nulu při pěti agentech)?

4. Proč necháváte algoritmy běžet jen jednu minutu?

## **Instrukce**

### **Splnění zadání**

Posudte, zda předložená ZP dostatečně a v souladu se zadáním obsahově vymezuje cíle, správně je formuluje a v dostatečné kvalitě naplňuje. V komentáři uveďte body zadání, které nebyly splněny, posudte závažnost, dopady a případně i příčiny jednotlivých nedostatků. Pokud zadání svou náročností vybočuje ze standardů pro daný typ práce nebo student případně vypracoval ZP nad rámec zadání, popište, jak se to projevilo na požadované kvalitě splnění zadání a jakým způsobem toto ovlivnilo výsledné hodnocení.

### **Písemná část práce**

Zhodnoťte přiměřenost rozsahu předložené ZP vzhledem k obsahu, tj. zda všechny části ZP jsou informačně bohaté a ZP neobsahuje zbytečné části. Dále posudte, zda předložená ZP je po věcné stránce v pořádku, případně vyskytují-li se v práci věcné chyby nebo nepřesnosti.

Zhodnoťte dále logickou strukturu ZP, návaznosti jednotlivých kapitol a pochopitelnost textu pro čtenáře. Posudte správnost používání formálních zápisů obsažených v práci. Posudte typografickou a jazykovou stránku ZP, viz Směrnice děkana č. 52/2021, článek 3.

Posudte, zda student využil a správně citoval relevantní zdroje. Ověřte, zda jsou všechny převzaté prvky řádně odlišeny od vlastních výsledků, zda nedošlo k porušení citační etiky a zda jsou bibliografické citace úplné a v souladu s citačními zvyklostmi a normami. Zhodnoťte, zda převzatý software a jiná autorská díla, byly v ZP použity v souladu s licenčními podmínkami.

### **Nepísemná část, přílohy**

Dle charakteru práce se případně vyjádřete k nepísemné části ZP. Například: SW dílo – kvalita vytvořeného programu a vhodnost a přiměřenost technologií, které byly využité od vývoje až po nasazení. HW – funkční vzorek – použité technologie a nástroje, Výzkumná a experimentální práce – opakovatelnost experimentů.

### **Hodnocení výsledků, jejich využitelnost**

Dle charakteru práce zhodnoťte možnosti nasazení výsledků práce v praxi nebo uveďte, zda výsledky ZP rozšiřují již publikované známé výsledky nebo přinášející zcela nové poznatky.

### **Celkové hodnocení**

Shrňte stránky ZP, které nejvíce ovlivnily Vaše celkové hodnocení. Celkové hodnocení nemusí být aritmetickým průměrem či jinou hodnotou vypočtenou z hodnocení v předchozích jednotlivých kritériích. Obecně platí, že bezvadně splněné zadání je hodnoceno klasifikačním stupněm A.