



# Posudek oponenta závěrečné práce

**Oponent práce:** Ing. Václav Blažej  
**Student:** Jan Šmolík  
**Název práce:** Stabilita Hedonických her se strukturovanými preferencemi  
**Obor / specializace:** Teoretická informatika  
**Vytvořeno dne:** 7. června 2021

## Hodnotící kritéria

### 1. Splnění zadání

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

Úkolem bylo prozkoumat koncepty stability pro nedávno definovanou variantu Hedonických her a pokust se o dokázání existence či neexistence (případně návrh algoritmu pro nalezení) stabilního rozdělení pro instance takové hry, případně za pomoci strukturálních omezení.

Práce sestává z příjemného úvodu do teorie, úvodu do konkrétního problému, vlastního přínosu práce, a popisu pomocného kódu.

Přestože student nedosáhl až k samotnému důkazu tvrzení, na které se zaměřil, vytvořil mnoho argumentů (podložených vlastní implementací), které značně podporují kýžené tvrzení.

### 2. Písemná část práce

95 /100 (A)

Kapitola 1 uvede čtenáře do teorie hedonických her. Text se věnuje relevantním částím teorie po vhodné délce a dodává dobré množství konkrétních příkladů. Poskytnuté koncepty plynule přechází do kapitoly 2, která popisuje nový koncept v tomto odvětví, který byl představen v r. 2020. Kapitola 2 také detailněji vysvětluje různé pojmy stability a strukturálních omezení instancí problému, včetně známých poznatků o nich. Konec kapitoly 2 je čtenáři jasný, co víme, co nevíme, a co by se dalo zjistit -- student potom dodává základní definici parametrizované složitosti s tím, že budeme problém studovat pro fixní velikost cílových skupin (jeden z parametrů).

Vlastní přínos studenta přichází na straně 19, kde v kapitole 3 popisuje myšlenky, které vedly k randomizovanému algoritmu řešící konkrétní problém. Algoritmus je založen na tom, že mnoho "spokojených" rozdělení je "core stabilní", tedy algoritmus

randomizovaně najde spokojené rozdělení, a pokud je core stabilní, tak končí, jinak zkusí hledat znovu. Toto základní tvrzení je sice podpořeno měřeními (odměřené případy vykazují kolem 90%), ale není dokázané. Další důležitý koncept je, že na začátku můžeme instanci 'zmenšit', protože dovedeme odvodit část řešení -- toto tvrzení mi přijde dokázané pouze částečně (vyjasnění je otázkou k obhajobě). Dále práce obsahuje měření: pravděpodobnost nalezení řešení pro různě veliké instance s různými nastaveními parametrů, průměrný počet restartů algoritmu před nalezením řešení.

Kapitola 4 uvádí 'nové poznatky', vycházející ze studentovy práce: konkrétně našel single-peaked instanci roommate diversity problému pro velikosti místností 5 s deseti agenty takovou, že má prázdné core. Toto je značný přínos, protože bylo dokázáno, že pro instance s velikostí místností 2 mají core neprázdný -- student tedy omezil otázku neprázdného core na případy s velikostí místnosti 3 a 4, ke kterým dodává domněnku podpořenou jeho měřeními.

Nakonec student krátce popisuje jeho pomocný program.

Rešeršní část je moc pěkná. Část s vlastními výsledky často obsahuje tvrzení ve stylu 'Je tedy velice pravděpodobné, že ...', která jsou podložena pouze slabým pozorováním chování programu -- nejsou podložena důkazem. Popis programu je přiměřený, hlavně kvůli tomu, že práce nebyla primárně implementační.

### 3. Nepísemná část, přílohy

90/100 (A)

Zdrojový kód je v pořádku, bez zjevných chyb a bez leaků. Detailněji jsem kontroloval zásadní části algoritmu, a vypadají správně. Avšak mám k přílohám mnoho menších výhrad:

Příloha zbytečně obsahuje složky ``.idea`` a ``__MACOSX``, a soubor ``.DS_store``. Makefile funguje při úplném překompilování, ale nepřekompiluje příslušné soubory po změně pouze hlavičkového souboru. V příloze chybí zdrojové kódy práce (tex, obrázky, atd.).

Program komunikuje s uživatelem pomocí interfacu ve stylu "zadejte 0 nebo 1", což mi přijde nevhodné, hlavně je s tím mnoho zbytečné práce -- lepší by bylo zpracovávat argumenty spuštění, nebo mít oddělené programy pro různé účely. Nyní interface nereaguje správně na EOF, a při doporučeném ukončení přes Ctrl-C signál neodchytí a spadne.

Program lze řádově zrychlit zapojením několika běžných optimalizačních technik: Doporučil bych upravit ``set`` na ``unordered_set`` tam, kde není důležité pořadí prvků uvnitř tohoto kontejneru. Dále mít v ``Makefile`` kompilaci s optimalizací alespoň ``-O2``. A protože program stráví mnoho času v metodě ``comb`` (kde generuje seznam kombinací), tak lze by šlo použít DP, tj. předpočítat výsledky a rovnou je vracet.

### 4. Hodnocení výsledků, jejich využitelnost

90/100 (A)

Hlavním přínosem vidím omezení případů instancí, které mají neprázdný core, tj. protipříklad pro velikost místnosti 5, a podporné argumenty pro menší místnosti. Podporné argumenty jsou založeny na programu -- ale mohou sloužit jako dobrý výchozí bod pro jejich dokázání. Na bakalářskou práci je přínos netriviální, a drobek posunulo

poznání v nové oblasti hedonických her. Kdyby se argumentace dotáhla do důkazů a dořešil se případ pro velikosti místností 3 a 4, mohl by být z práce pěkný článek.

## Celkové hodnocení

95 /100 (A)

Text práce pěkně uvádí do problematiky a navazuje přínosem studenta, který díky pomocné implementaci zjistil netriviální výsledek pro nové odvětví hedonických her. Program také podporuje řadu studentových domněnek, ale pro jejich dokázání je potřeba ještě kus práce.

Text práce má minimum chyb, zato se snaží často nepodloženě argumentovat, že je něco pravděpodobné -- což spíše pokládám pozorování chování programu, než řádné teoretické argumentaci. K přílohám mám sice značné množství drobných výhrad, ale ty nebyly hlavní náplní této práce.

Doporučuji tuto práci k obhajobě.

## Otázky k obhajobě

Na straně 24 obrázku 3.1 je graf závislosti průměrného počtu vrácení v algoritmu na počtu červených agentů. Průměrná hodnota (nakreslené křivky) říká velice málo, jak vypadá rozložení dat pro fixní počet červených agentů? Konkrétně: jsou hodnoty shromážděné u průměru, nebo je většina 0 a pár zbylých dat je hodně velikých, a proto je průměr vyšší.

Lemma 3.1 na straně 19 zmiňuje způsob, jak můžeme do řešení problému výběru koalice rovnou přidat koalici takové, že všichni její členové ji považují za ideální. Toto tvrzení je podané jako ekvivalence, tedy "... agenty můžeme umístit spolu do jedné koalice ... a zbývá nám vyřešit menší instanci problému ...". Je jasné, že tito členové nemohou být členové blokující koalice, protože nemohou nic preferovat více (toto je jeden směr implikace). Ovšem chybí odůvodnit směr druhý, tedy: pokud existuje core-stabilní řešení, tak i po přidání této koalice nějaké najdu. Teoreticky by mohlo přidání koalice znemožnit vytvoření core stabilního řešení -- je tento druhý směr vidět triviálně?, nevyužíváte ho?, nebo v práci chybí?

## **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. 26/2017, 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.