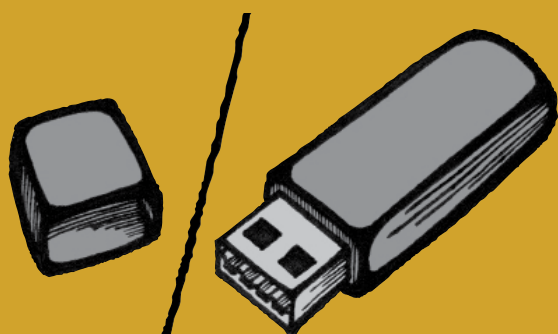


FAKULTA APLIKOVANÝCH VĚD
ZÁPADOČESKÉ UNIVERZITY
V PLZNI

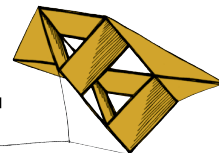
SBORNÍK

STUDENTSKÁ VĚDECKÁ KONFERENCE



BAKALÁŘSKÉ
STUDIJNÍ PROGRAMY

23.5.2024



Bakalářské studijní programy

Sborník rozšířených abstraktů

Název: SVK FAV 2024 – bakalářské studijní programy

Editor: Jan Rendl

Ilustrace na obálce: Jana Trávníčková

Vydavatel: Západočeská univerzita v Plzni, Univerzitní 8, 306 14 Plzeň

Datum vydání: květen 2024

ISBN 978-80-261-1227-3

Tato publikace neprošla redakční ani jazykovou úpravou.

Sponzoři konference



Obsah

Sekce – Fyzika, matematika, geomatika, mechanika

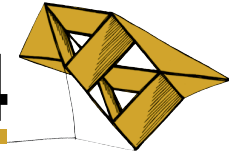
Stanovení ztrátového součinitele kónických redukcí různých rozměrů prostřednictvím numerických simulací	
Eisenhammer Tomáš	6
Studium hydrofilních a hydrofobních povrchů	
Kussainov Jangir	8
Neautonomní dynamické systémy a jejich aplikace	
Mužík Jaroslav	10

Sekce – Informatika, kybernetika

Vizualizace alokací paměti a jejich duplicit	
Faragula Štěpán	13
Kompresce konektivity trojúhelníkových sítí se známou geometrií s využitím neuronových sítí	
Havlík Viktor	15
Rozšíření možností Czech Salivary Gland Database pro klinickou praxi a pro analýzu dat	
Jelínek Vojtěch	17
Estimating optimal encoder parameters for a priority-driven connectivity compression algorithm for triangle meshes with known geometry	
Kafara Stanislav	19
Porovnání paralelní implementace B algoritmu v Javě a C/C++	
Pernička Petr	21
Porovnání algoritmů pro dělení silniční sítě	
Roy Lucie	23
Automatic Creation of Summaries of Historical Documents	
Tran Václav	25

Sekce

Fyzika, geomatika, matematika, mechanika



Stanovení ztrátového součinitele kónických redukci různých rozměrů prostřednictvím numerických simulací

Tomáš Eisenhammer¹

1 Úvod

Určení tlakových ztrát v redukci je nezbytné například při návrhu potrubních systémů tepelných elektráren. Tlakové ztráty jsou totiž spojeny s disipací energie, která tak nemůže být použita k pohonu turbíny. Pro kónické redukce však neexistuje obecně platný vztah pro výpočet těchto ztrát. Proto je nutné tlakové ztráty určit experimentem nebo numerickou simulací.

Tato práce se zabývá určením ztrátových součinitelů v kónických redukci vybraných geometrických parametrů prostřednictvím numerických simulací. Výsledky jsou dále analyzovány s důrazem na závislost ztrátového součinitele na průměru výstupní trubky, úhlu redukce a vstupním Reynoldsově čísle.

2 Součinitele místní ztráty

Tlakovou ztrátu v redukci vyjadřuje součinitel místní ztráty

$$\zeta = \frac{p_z}{p_{\text{dyn}1}} = \frac{p_{\text{st}1} - p_{\text{st}2} + p_{\text{dyn}1} - p_{\text{dyn}2}}{p_{\text{dyn}1}}, \quad (1)$$

kde $p_{\text{st}1,2}$ značí statický tlak před, resp. za redukci a $p_{\text{dyn}1,2}$ značí odpovídající dynamické tlaky.

Pro kónické redukce s malými vrcholovými úhly a pro náhlá zúžení lze nalézt příslušné vztahy např. v Noskiewič a kol. (1987), pro ostatní úhly však takovéto vztahy nejsou k dispozici.

3 Numerická simulace

Součinitele místní ztráty byly určeny pro následující parametry:

průměr výstupní trubky $d = 80, 100, 120$ mm,

vrcholový úhel $\alpha = 10, 20, 30, 60, 90, 120, 180^\circ$,

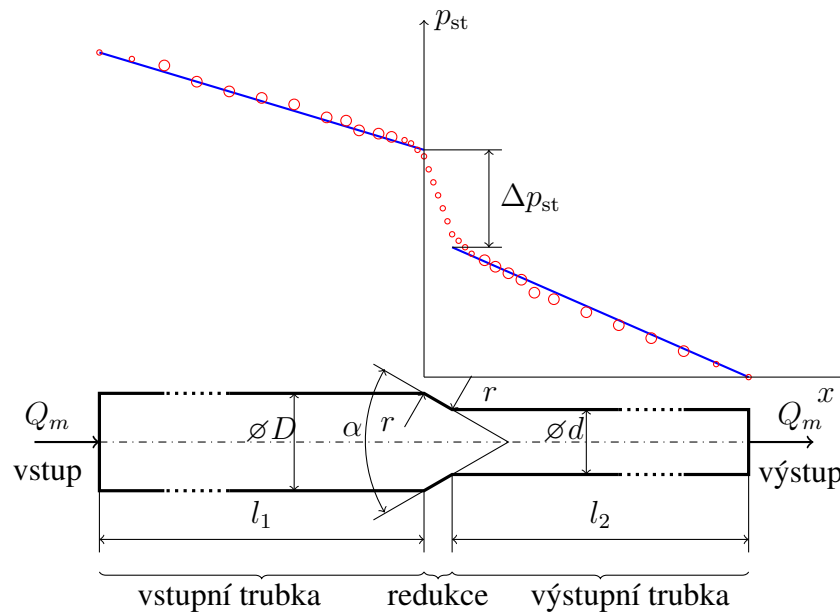
Reynoldsovo číslo $Re = 50\,000, 100\,000, 200\,000, 300\,000, 500\,000$,

celkem tedy pro 105 variant. Všechny simulace používaly výpočtový model proudění vycházející z dřívějšího výzkumu Katedry mechaniky, konkrétně z výzkumné zprávy Vimmr a kol. (2022), a byly provedeny v komerčním programu ANSYS Fluent.

Pro určení vhodné velikosti buněk výpočetní sítě byla provedena tzv. mesh resolution study, tj. analýza konvergence výsledků pro zjemňující se síť. Podobná metoda byla použita i pro určení vhodného zahuštění sítě v mezní vrstvě, zjednodušení výpočtové oblasti na základě osové symetrie úlohy a určení vhodné délky vstupní a výstupní trubky.

¹ student bakalářského studijního programu Počítačové modelování v mechanice, e-mail: eisentom@students.zcu.cz

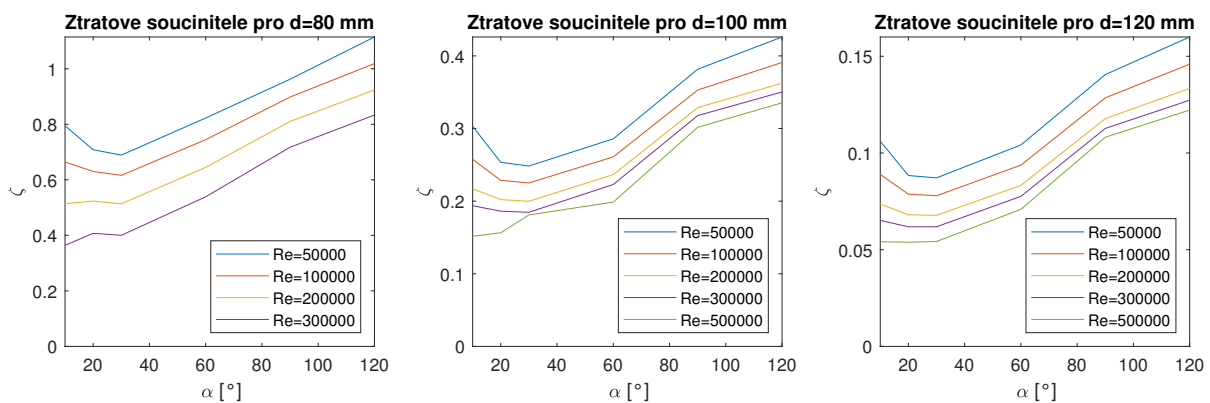
4 Určení ztrátových součinitelů z výsledků simulací



Obrázek 1: Schéma určení rozdílu statického tlaku v redukci

Rozdíl statických tlaků je určen extrapolací regresních přímek průměrných hodnot p_{st} ve vybraných průřezích vstupní, resp. výstupní trubky, viz obr. 1. Rozdíl dynamických tlaků je určen z průtoku, protože p_{dyn} po průběhu trubky značně kolísá a použitá metoda docílí větší přesnosti. Z těchto hodnot lze pak vypočítat ztrátový součinitel ζ pro danou variantu dle vztahu (1).

5 Výsledky

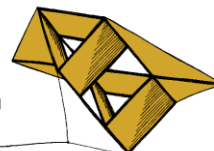


Obrázek 2: Výsledné ztrátové součinitele určené numerickými simulacemi

Literatura

Noskievič, J. a kol. (1986) *Mechanika tekutin*. Praha, SNTL – Státní nakladatelství technické literatury.

Vimr, J., Plánička, S., Jonášová, A. (2022) *Vývoj pokročilé metodiky pro stanovení průtočných charakteristik zpětné odběrové klapky u parního turbosoustrojí*. Souhrnná výzkumná zpráva: NTIS-VP3-010/2022, ZČU v Plzni.



Studium hydrofilních a hydrofobních povrchů

Jangir Kussainov¹

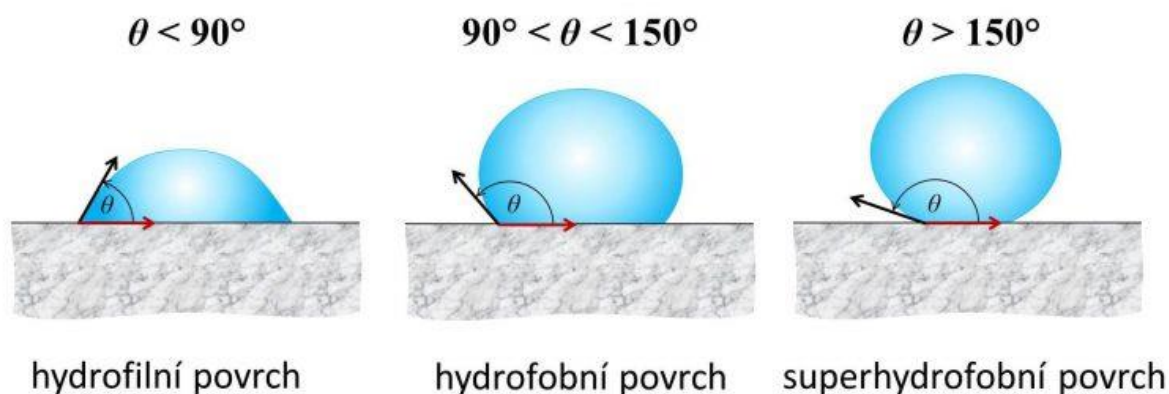
1 Úvod

Smáčivost povrchu je schopnost kapaliny udržovat kontakt s povrchem pevné látky prostřednictvím mezimolekulární interakce. Tento jev je výsledkem povrchového napětí látek a měří se pomocí veličiny zvané kontaktní úhel.

Cílem tohoto projektu je měřit změny kontaktního úhlu různých hydrofobních a hydrofilních materiálů před a po vysokofrekvenčním výboji různých časových intervalech. Zjistíme, které materiály po vysokofrekvenčním výboji výrazněji mění svou smáčivost a jak dlouho u různých materiálů efekt vydrží.

2 Kontaktní úhel

Když kapalina a pevná látka interagují (například kápnete trochu vody na pevnou látku), vytvoří se mezi nimi úhel, který se nazývá kontaktní úhel. Je to veličina, která měří smáčení pevné látky a velikost tohoto úhlu je indikátorem hydrofobnosti a hydrofilnosti látky. Pokud je kontaktní úhel menší než 90 stupňů, pak je těleso považováno za hydrofilní. A pokud je kontaktní úhel větší než 90 stupňů, pak je tělo považováno za hydrofobní. Čím větší je kontaktní úhel, tím vyšší je hydrofobnost látky.



Obrázek 1: Kontaktní úhel

Youngova rovnice pro zjištění kontaktního úhlu pomocí koeficientu povrchového napětí (1):

$$\cos(\theta) = \frac{\gamma_{23} - \gamma_{13}}{\gamma_{12}} \quad (1)$$

¹ student bakalářského studijního programu Aplikovaná fyzika a fyzikální inženýrství, obor Fyzika, e-mail: jangir@students.zcu.cz

3 Měření

Nejprve se změří kontaktní úhel samotného tělesa, poté se provede vysokofrekvenční výboj. Těleso bude poté znovu změřeno, aby se zjistily změny kontaktního úhlu. Měření kontaktního úhlu se bude provádět každý týden. První měření bude před vysokofrekvenčním výbojem, druhé bezprostředně po a další v týdenních intervalech, dokud se vlastnosti povrchu nevrátí do původního stavu.

V této studii jsem použil celkem čtyři materiály. Jednalo se o: plast, karton, gumu a pěnovou pryž.

Vyberu si také ten nejhydrofobnější ze svých materiálů (gumu) a změřím kontaktní úhel v mnohem kratším časovém intervalu (nejprve každou půlhodinu, pak každou hodinu a pak každý den). Pro nalezení přesnější rychlosti návratu k předchozímu kontaktnímu úhlu po vysokofrekvenčním výboji

4 Výsledky

Tabulka 1. Výsledky měření kontaktního úhlu

Týden	Plast		Karton		Guma		Pěnová pryž	
	Kontaktní úhel [°]	Chyba měření [°]	Kontaktní úhel [°]	Chyba měření [°]	Kontaktní úhel [°]	Chyba měření [°]	Kontaktní úhel [°]	Chyba měření [°]
0.(před výbojem)	63,36	±4,52	67,87	±10,5	90,81	±7,65	107,37	±14,08
0.(po výboji)	38,85	±3,36	27,09	±8,45	27,02	±5,16	70,34	±5,74
1.	64,87	±11,4	47,03	±10,69	96,58	±6,4	111,02	±14,87
2.	75,87	±6,4	61,06	±13,95	96,73	±6,94	118,63	±18,46

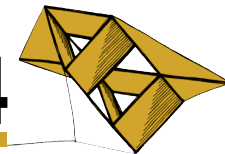
Vysokofrekvenční výboj dodal látkám náboj a tím zvýšil jejich hydrofilní vlastnosti. Přesvědčili jsme se o tom, jak moc se změnil kontaktní úhel po výboji. Bylo také potvrzeno, že látka nemůže udržet náboj, který je jí dán po dlouhou dobu. Většina materiálů se vrátila ke svým předchozím vlastnostem a kontaktnímu úhlu po jednom týdnu (plast, guma a pěnová pryž). Jediným materiálem, kterému trvalo déle, než se vrátilo ke svým předchozím vlastnostem a kontaktnímu úhlu, byl karton, což trvalo dva týdny. O měsíc později jsem také provedl kontrolní měření kontaktního úhlu, která potvrdila dříve získané výsledky. Protože experimentální podmínky byly pro všechny materiály stejné, můžeme dojít k závěru, že rychlost ztráty náboje a návrat látky k jejím předchozím smáčivým vlastnostem závisí na vlastnostech samotného materiálu.

Poděkování

Děkuji Zbyňku Soukupovi za supervizi mého projektu.

Literatura

- [1] Arnold B.J., Aplan F.F.: The hydrophobicity of coal macerals, Fuel, Vol. 68, Issue 5, May 1989, 651-658, dostupné z: [https://doi.org/10.1016/0016-2361\(89\)90168-3](https://doi.org/10.1016/0016-2361(89)90168-3)
- [2] Atkins P., De Paula J., Keeler J.: Atkins' Physical Chemistry, 4th ed., Oxford: Oxford University Press, 1992, xii, 995 s. ISBN 01 985-5284-X.
- [3] Dey S.: Enhancement in hydrophobicity of low rank coal by surfactants — A critical overview, Fuel Processing Technology, Vol. 94, Issue 1, February 2012, 151-158, dostupné z: <https://doi.org/10.1016/j.fuproc.2011.10.021>



Neautonomní dynamické systémy a jejich aplikace

Jaroslav Mužík¹

1 Úvod

Tato práce je zaměřena na analýzu neautonomních růstových modelů ve spojitém čase. V práci je zkoumán logistický model růstu, jenž byl poprvé popsán v devatenáctém století P. F. Verhulstem (viz Allen (2007)). V logistickém modelu figurují dva autonomní parametry. Růstový parametr, symbolizující rychlost růstu sledované veličiny a parametr symbolizující nosnou kapacitu prostředí. V práci jsou zavedeny a analyzovány dvě neautonomní období tohoto standardního modelu, které mají každá právě jeden z výše zmíněných parametrů nahrazen jeho obdobou, která je nekonzstantní v čase. Porovnávají jsou kvalitativní a kvantitativní vlastnosti jednotlivých modelů. V jednotlivých sekcích jsou ilustrovány a diskutovány rozdíly mezi autonomními řešeními a jim odpovídajícími neautonomními variantami.

Příkladem systému s parametry závislými na čase může být např. populace krysy obecné v severovýchodní Indii, jejíž reprodukce je úzce navázána na životní cyklus bambusu *melocanna baccifera*. Tento druh bambusu se dle Jeeva et al. (2009) vysemeňuje pouze jednou za několik desítek let a živočichům živícím se jeho plody, mezi nimiž je i krysa, náhle prudce naroste kapacita prostředí a s ní i velikost jejich populace. Krysy pak pravidelně po úbytku semen způsobují škody na úrodě místním obyvatelům. Přesnějším modelováním těchto jevů by se dalo lépe a s předstihem připravit na budoucí vlny škůdců. Jiným příkladem kapacity prostředí závislé na čase může být trh se sezónním zbožím. Kapacitu lze interpretovat jako poptávku po zboží, která v průběhu roku periodicky kolísá. Příkladem skokového růstu kapacity pak může být rozšíření výrobního závodu na nový trh nebo rozšíření repertoáru výrobků. Nabízí se pak otázka jak rychle dokáže podnik tuto kapacitu zaplnit.

2 Výsledky analýz

V práci jsou srovnávány tři úlohy. Jsou jimi úloha autonomní:

$$\begin{cases} x'(t) = \alpha x(t) \left(1 - \frac{x(t)}{\varkappa}\right), & t \geq 0, \\ x(0) = x_0, \end{cases} \quad (\text{A})$$

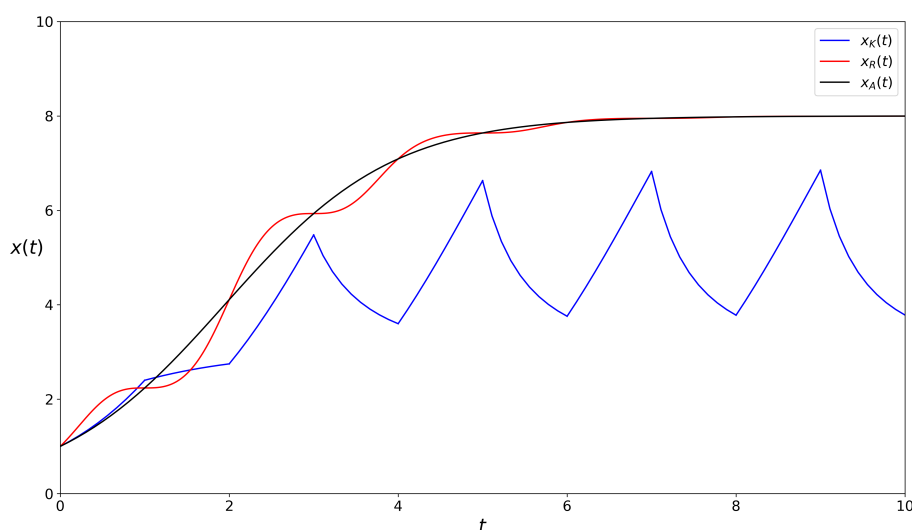
dále úloha neautonomní v růstovém parametru:

$$\begin{cases} x'(t) = a(t)x(t) \left(1 - \frac{x(t)}{\varkappa}\right), & t \geq 0, \\ x(0) = x_0 \end{cases} \quad (\text{R})$$

a nakonec úloha neautonomní v parametru kapacity prostředí:

$$\begin{cases} x'(t) = \alpha x(t) \left(1 - \frac{x(t)}{k(t)}\right), & t \geq 0, \\ x(0) = x_0. \end{cases} \quad (\text{K})$$

¹ student bakalářského studijního programu Matematika a finanční studia, e-mail: jamuzik@students.zcu.cz



Obrázek 1: Srovnání partikulárních řešení úloh (A), (R) a (K).

Na obrázku 1 jsou srovnány průběhy partikulárních řešení těchto úloh. Prvním je řešení $x_A(t)$ úlohy (A), dále řešení $x_R(t)$ úlohy (R) a řešení $x_K(t)$ úlohy (K). Zvolené hodnoty parametrů pro všechna řešení jsou $\alpha = 0.9$, $\varkappa = 8$, $x_0 = 1$ a pro řešení $x_K(t)$ je navíc zvolen parametr $\varepsilon = 5$.

Chování řešení úlohy (R) je v práci popsáno následujícími lemmaty:

Lemma 1. *Pokud je $x_0 > 0$, pak pro trajektorie úloh (A) a (R) platí:*

- I. *existuje-li takové $T > 0$, že $A(T) = \alpha T$, pak $x_R(T) = x_A(T)$.*
- II. *Je-li $x_0 < \varkappa$ a existuje-li takové $T > 0$, že $A(T) > \alpha T$, pak $x_R(T) > x_A(T)$.*
- III. *Je-li $x_0 > \varkappa$ a existuje-li takové $T > 0$, že $A(T) > \alpha T$, pak $x_R(T) < x_A(T)$.*
- IV. *Je-li $x_0 < \varkappa$ a existuje-li takové $T > 0$, že $A(T) < \alpha T$, pak $x_R(T) < x_A(T)$.*
- V. *Je-li $x_0 > \varkappa$ a existuje-li takové $T > 0$, že $A(T) < \alpha T$, pak $x_R(T) > x_A(T)$.*

Lemma 2. *Pokud je $0 < x_0 < \varkappa$, pak bude pro všechna $t \geq 0$ platit $x_R(t) < \varkappa$.*

Pro úlohu (K) bylo analyzováno pouze řešení $x_K(t)$ se specifickou po částech konstantní funkcí $k(t)$. Pro toto řešení mimo jiné platí:

Lemma 3. *Interval (t_1, t_2) , na kterém platí $x_K(t) > \varkappa$ existuje, pokud je splněna nerovnost*

$$\frac{\varkappa}{\varepsilon} > \frac{e^\alpha + 1}{e^\alpha - 1}.$$

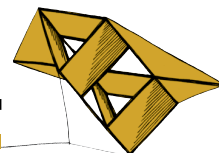
Literatura

Allen, L. J. S. (2007) *An Introduction to Mathematical Biology*. Pearson/Prentice Hall, Upper Saddle River, NJ.

Jeeva, S. a Kiruba, S. a Lalhruaitluanga, H. a Mnv., Prasad a Rao, R.R. (2009) Flowering of *Melocanna baccifera* (Bambusaceae) in northeastern India. *Curr. Sci.*.

Sekce

Informatika, kybernetika



Vizualizace alokací paměti a jejich duplicit

Štěpán Faragula¹

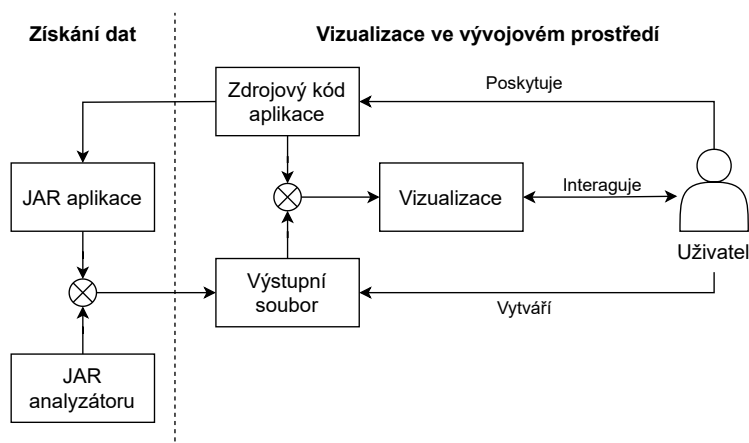
1 Úvod

Programátoři mohou během vývoje softwaru používat dynamické analytické nástroje, tzv. *profilery*, které poskytují různé informace o běhu systému. Charakter získaných metrik dále určuje, jak by mohly být data prezentována. Typicky se jedná o souhrn, který poskytuje statický pohled na získaná dynamická data za určité období. Ten může být reprezentován ve formě textového souboru nebo vizualizace s interaktivním ovládáním.

Tato práce se zabývá vizualizací dat alokací paměti a duplicit, které byly detekovány za běhu Java programů. Za tímto účelem vzniklo rozšíření textového editoru Visual Studio Code, které umí zobrazit podrobnosti o přidělené paměti přímo ve zdrojovém textu vyvíjeného softwaru. Metriky jsou poskytovány již existujícím analyzátozem, který vznikl jako součást bakalářské práce Velek (2023) na Západočeské univerzitě v Plzni. Instalační soubor pluginu je zveřejněn online na školním repozitáři <https://gitlab.kiv.zcu.cz/lipka/visualisation-of-allocations>.

2 Navržený systém

Plugin je navržen tak, aby jeho vstup byl závislý pouze na formátu výstupního souboru analyzátozem, a ne na konkrétní implementaci nástroje. Dále potřebuje přístup ke zdrojovému textu pozorované aplikace, což je zajištěno otevřením projektu v textovém editoru. Aby byla prezentace metrik smysluplná, verze poskytnutého textu se musí shodovat s tou, nad kterou byl spuštěn analyzátor. Celý proces analýzy je tak možné rozdělit na dvě části, a to získání dat a jejich následnou vizualizaci. Schéma navrženého systému je vidět na obrázku 1.



Obrázek 1: Postup získání dat spolu s jejich následnou vizualizací

¹ student bakalářského studijního programu Informatika a výpočetní technika, obor Informatika, e-mail: farag844@students.zcu.cz

3 Popis vizualizace

Záměrem bylo implementovat minimalistickou vizualizační metodu, která bude upozorňovat na alokace paměti přímo v textovém editoru. Výslednou podobu rozšíření je možné pozorovat na obrázku 2. Vizualizace se zakládá na několika principech lidského vnímání, které jsou popsány ve Ward et al. (2010).

Nejvýznamnějším z využitých poznatků je zvýraznění prvku odlišným barevným tónem. Jednotlivé řádky, na kterých byla alokována paměť, jsou vůči ostatním jinak podbarveny. Jejich součástí je také dekorativní text, který ukazuje velikost přidělené paměti v bytech a počet nalezených duplicit. Tato kombinace umožňuje programátorovi dohledat a vyčíst potřebné informace pouhým prohlížením zdrojového kódu. Velikosti alokací na řádcích jsou také agregovány do metod a tříd. Tímto způsobem je možné zjistit velikost přidělené paměti ve třech úrovních (řádek, metoda, třída). Každá z těchto skupin je vyznačena jinou barvou.

Veškeré zobrazované údaje, ať už individuální řádky či agregované záznamy, lze detailněji zkoumat ve dvou tabulkách. V horní jsou vypsány informace o přidělené paměti, v dolní podrobnosti o duplicitách. V obou tabulkách jsou obsaženy odkazy na jednotlivé řádky, které přidělily paměť vybranému záznamu. Tím je zejména možné dohledat místo vzniku stejných instancí napříč celým zdrojovým textem. Přestože analyzátor neumí poskytnout konkrétní hodnotu stejných instancí, kopie jsou v tabulce vizuálně seskupeny tučnou čarou a mezerou.

Name	Size [B]	Count	Source
TestingObject	16	1	main.Testing:8
TestingObject	16	1	main.Testing:5
TestingObject	16	1	main.Testing:9
BiggerTestingObject	24	1	main.Testing:6
TestingObject	16	10	main.Testing:12

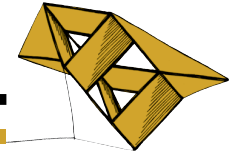
Name	Size [B]	Count	Source
TestingObject	16	1	main.Testing:9
TestingObject	16	1	main.Testing:8
TestingObject	16	10	main.Testing:12

Obrázek 2: Ukázka vytvořeného rozšíření pro Visual Studio Code

Literatura

Velek, J. (2023) *Monitorování alokací paměti za běhu Java aplikací*. Bakalářská práce. Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Katedra informatiky a výpočetní techniky. Dostupné také z: <http://hdl.handle.net/11025/53749>.

Ward, M., Grinstein, G., Keim, D. (2010) *Interactive Data Visualization: Foundations, Techniques, and Applications*. Natick, MA: A. K. Peters, Ltd., ISBN 978-1-56881-473-5.



Kompresa konektivity trojúhelníkových sítí se známou geometrií s využitím neuronových sítí

Viktor Havlík¹

1 Úvod

Při práci s 3D objekty jsou jako struktury, používané pro jejich povrchovou reprezentaci, běžně voleny trojúhelníkové sítě. U nich jsou nejprve pomocí x , y a z souřadnic na povrch objektu namapovány body, které pak společně tvoří takzvanou geometrii sítě. Následně jsou body členěny do trojic, kdy každá z nich reprezentuje jeden trojúhelník. Tento způsob propojení vrcholů je označován jako konektivita sítě. Při vytvoření dostatečného množství těchto trojúhelníků vzniká celistvá trojúhelníková síť obalující celý objekt a popisující tak jeho povrch.

Tato práce se zabývá kompresí zmíněné konektivity trojúhelníkové sítě a navazuje na již existující, velice efektivní metodu Dvořák et al. (2023) založenou na principu kandidátních vrcholů a dává si za cíl její další zefektivnění. Toho se snaží dosáhnout za pomoci strojového učení. Práce se tak zaměřuje na několik klíčových částí původní metody a ty nahrazuje modelem umělé neuronové sítě. Cílem je zjistit, jak efektivně je model schopen, oproti statickým vzorcům výchozí metody, predikovat konektivitu trojúhelníkové sítě a zda-li tento přístup může vést k požadovanému zlepšení. Jak získané výsledky naznačují, je tomu skutečně tak a metoda je s využitím neuronové sítě schopna dosáhnout signifikantního zlepšení.

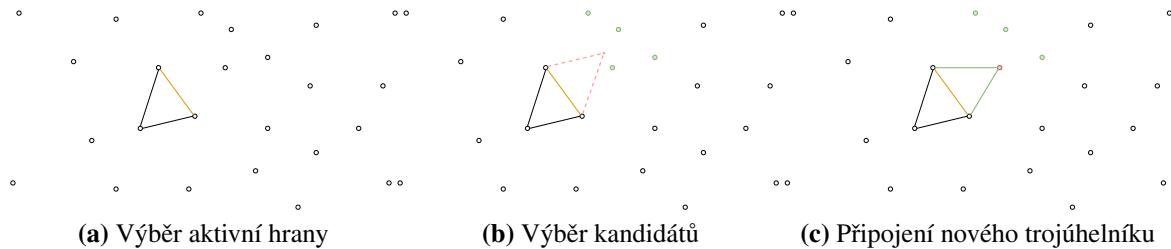
2 Princip fungování a popis algoritmu

Metoda vychází z předpokladu známé geometrie sítě (zná tedy rozložení bodů v prostoru) a jejím cílem je s využitím této znalosti snížit objem potřebných dat pro popis zbylé konektivity. Během procesu bude totiž spoléhat na hypotézu, že existuje korelace mezi pozicemi bodů a trojúhelníkovou sítí, kterou tvoří. Ta říká, že je výrazně vyšší pravděpodobnost, že body tvořící trojúhelník budou ležet ve svém relativně blízkém okolí, než daleko od sebe. Byl proto vytvořen model neuronové sítě, který je schopen tohoto faktu využít a z předaných bodů danou konektivitu predikovat.

Samotný algoritmus funguje na principu postupného průchodu sítí, který provádí jak kodér tak dekodér. Během něj zasílá kodér informace o topologii sítě dekodéru, který ji tak zpětně rekonstruuje. Postup je rozdělen do několika kroků: Vybrání *aktivní hrany*, ke které je nový trojúhelník připojen, vytvoření seznamu *kandidátních vrcholů* pro připojení a ohodnocení každého kandidáta pravděpodobností, se kterou tvoří vrchol připojovaného trojúhelníku. Na začátku, během inicializace, proběhne krátká synchronizace mezi kodérem a dekodérem zasláním polohy prvního trojúhelníku sítě, od kterého algoritmus započne. Následně jsou z prioritní fronty vybírány aktivní hrany, přes které je průchod sítí prováděn (Priorita hran v prioritní frontě je určena na základě jistoty, se kterou algoritmus ví jaký trojúhelník k hraně připojit.). Dekodér ke každé hraně vybere k -nejbližších vrcholů a pomocí neuronové sítě určí pravděpodobnost,

¹ student bakalářského studijního programu Informatika a výpočetní technika, obor Informatika, e-mail: vha-
vlik@students.zcu.cz

se kterou každý z nich tvoří připojovaný trojúhelník. Seznam takto ohodnocených kandidátů poté seřadí. Kodér postup dekodéru po celou dobu emuluje a na konci získává stejně seřazený seznam jako dekodér. Zároveň také ví s jakým vrcholem v síti je připojovaný trojúhelník skutečně tvořen. Dekodéru tedy zasílá jeho index v seřazeném seznamu. Takto je postupováno dokud není zpracována celá síť. Celý postup je zobrazen na obrázku 1.



Obrázek 1: Princip připojení nového trojúhelníku

Klíčovou vlastností tohoto kódování je, že informace zasílané mezi kodérem a dekodérem (tj. indexy vrcholů v seznamu) jsou vždy celá, nezáporná čísla a při správné predikci pravděpodobnosti pomocí modelu neuronové sítě také silně vychýlená k nule. Nad touto sekvencí je tak možné provést další kódování pomocí aritmetického kodéru, který dokáže často se opakující symboly v sekvenci kódovat značně efektivněji na úkor těch méně častých. výsledná sekvence je uložena do souboru a obsahuje komprimovanou konektivitu celé sítě. V případě potřeby pak dekodér nejprve tuto sekvenci pomocí aritmetického kodéru dekomprimuje a následně výše zmíněným postupem konektivitu sítě zpětně rekonstruuje.

3 Dosažené výsledky

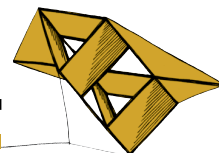
Během vývoje byla vytvořena řada modelů využívaných pro predikci konektivity trojúhelníkové sítě. Stejně tak bylo experimentováno s různými technikami výpočtu priority hran v prioritní třídě. Ve výsledku se podařilo navrhnout model a zvolit takovou strategii výpočtu, umožňující téměř **32% zlepšení**. Výsledky jsou porovnány v tabulce 1.

Tabulka 1: Výsledky komprese navržené metody. Sloupec *bpf* udává průměrný počet bitů potřebný na zakódování konektivity jednoho trojúhelníku sítě.

Konfigurace	bpf	Zlepšení
Referenční metoda	0,16063	
Navržená metoda	0,10925	31,99%

Literatura

Dvořák, J., Káčereková, Z., Vaněček, P., a Váša, L. (2023) *Priority-based encoding of triangle mesh connectivity for a known geometry*. Computer Graphics Forum, 42 (1), s. 60-71. Dostupné z DOI: <https://doi.org/10.1111/cgf.14719>.



Rozšíření možností Czech Salivary Gland Database pro klinickou praxi a pro analýzu dat

Vojtěch Jelínek¹

1 Úvod

Rakovina slinných žláz je velmi závažné a nebezpečné onemocnění, které má zásadní vliv na kvalitu života postižených pacientů. Vzhledem ke komplexnosti léčby je nezbytné, pro lékaře zabývající se touto problematikou, mít k dispozici efektivní nástroj pro shromažďování, sledování a následnou analýzu dat pacientů s tímto onemocněním. Jedním z těchto nástrojů je Czech Salivary Gland Database (CSGDB), který je vyvíjen na Západočeské univerzitě v Plzni.

Cílem práce bylo rozšířit možnosti využití aplikace Czech Salivary Gland Database v klinické praxi a pro analýzu nashromážděných dat. Prvním krokem při rozvoji aplikace bylo identifikování veškerých nedostatků předchozí verze a následné navržení řešení. Součástí rozšíření byla také implementace nástroje umožňujícího analýzu získaných dat pomocí Kaplan-Meierovy metody. Závěrem byly provedeny navržené úpravy a aplikace byla nasazena. Finální aplikace je dostupná na https://github.com/vjelinekk/CzechSalivaryGlandDB_v2.

2 Popis původní aplikace

Původní aplikace vznikla ve spolupráci s panem doktorem MUDr. Davidem Kalfeřtem, Ph.D., v rámci semestrální práce z KIV/ZSWI. Byla vytvořena studenty Mikulášem Machem, Viktorem Havlíkem a Vojtěchem Jelínkem.

The screenshot shows the user interface of the Czech Salivary Gland Database. On the left, there is a sidebar with a menu containing: 'Seznam pacientů', 'Přidat pacienta', 'Studie', 'Přidat studii', and 'Exportovat data'. The central part of the interface features a search bar labeled 'Vyhledat...' with a dropdown menu showing search filters: 'Test (podělelstní)', 'Name Surname (podělelstní)', 'Test (podjazykově)', 'Pacient (přiušní)', 'Pac Priusni (přiušní)', and 'Test (přiušní)'. The right panel is titled 'ANAMNESTICKÁ/PERSONÁLNÍ DATA' and contains a form for entering patient information. The form is divided into two sections: 'Základní informace' and 'Pohlaví pacienta'. The 'Základní informace' section includes fields for 'Jméno:', 'Příjmení:', 'Identifikační kód pacienta:', 'RČ:', and 'Věk pacienta v době diagnózy:'. The 'Pohlaví pacienta' section includes radio buttons for 'Žena' and 'Muž'. At the top right of the right panel, there are two buttons: 'Smazat pacienta' (red) and 'Edlovat' (blue).

Obrázek 1: Uživatelské rozhraní původní aplikace

¹ student bakalářského studijního programu Informatika a výpočetní technika, obor Informatika, e-mail: vjelinek@students.zcu.cz

Uživatelské rozhraní původní aplikace je vidět na obr. 1. V levé části okna se nachází hlavní menu, které poskytuje základní funkce, jako je přidávání studií, pacientů, zobrazování seznamů a export dat. Centrální část okna zobrazuje konkrétní prvky uživatelského rozhraní, které patří ke zvolené části v menu.

3 Nalezené nedostatky

Nedostatky aplikace byly rozděleny do dvou hlavních skupin. První skupina obsahuje nedostatky týkající se funkcionality aplikace a uživatelského rozhraní, druhá skupina se zaměřuje na nedostatky v implementaci a struktuře kódu.

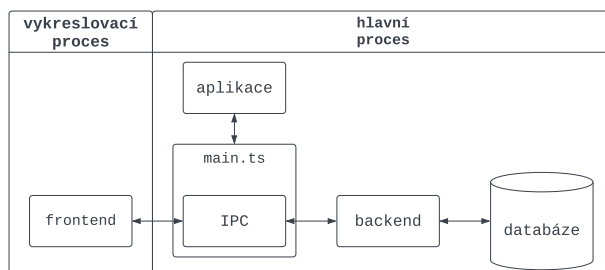
Hlavním nedostatkem funkcionality aplikace bylo nedostatečné zabezpečení, nemožnost anonymizovaného exportu dat a absence filtračního menu v seznamu pacientů. Vzhledem k tomu, že aplikace obsahuje citlivá osobní data pacientů, bylo nezbytné přidat ochranné mechanismy, které poskytnou potřebné zabezpečení.

Struktura kódu původní verze aplikace byla špatně rozdělena do modulů a kód porušoval několik základních principů čistého kódu, což limitovalo dlouhodobou udržitelnost aplikace.

4 Implementace a nasazení

Nově vytvořená aplikace zachovala základní strukturu uživatelského rozhraní a hlavní technologii ElectronJS. Zásadní změnou bylo nahrazení čistého (Vanilla) JavaScriptu za typovanou nadstavbu (TypeScript) spolu s knihovnou ReactJS.

Kód byl rozdělen do modulů frontend a backend, jak je vidět na obr. 2. Komunikace mezi nimi je zajištěna pomocí modulu IPC (Inter Process Communication), který je součástí ElectronJS.



Obrázek 2: Struktura nové verze aplikace

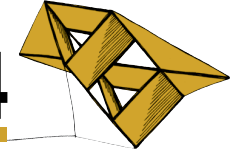
Při procesu přepisování aplikace do nových technologií bylo využito CI/CD akcí na platformě GitHub, které umožnily kontinuální testování nově přidaných částí kódu. Tyto akce také automatizovaly kontrolu sestavení aplikace do spustitelného souboru, což zajišťovalo hladké nasazení finální verze programu.

5 Závěr

Výsledná aplikace byla podrobena testování na skupině dvaceti pacientů. Na základě výsledků testování a pozitivní zpětné vazby bylo vyhodnoceno, že aplikace poskytuje všechny požadované funkce a byla nasazena na pracovišti Fakultní nemocnice v Motole.

Poděkování

Děkuji panu doktoru MUDr. Davidu Kalfeřtovi, Ph.D., za spolupráci při tvorbě aplikace.



Estimating optimal encoder parameters for a priority-driven connectivity compression algorithm for triangle meshes with known geometry

Stanislav Kafara¹

1 Introduction

The algorithm proposed in Dvořák et al. (2022) is a state-of-the-art triangle mesh compression algorithm that exploits the knowledge of mesh geometry, i.e., the vertex positions, at both the encoder and decoder to efficiently encode its connectivity, i.e., the triangle faces.

The encoder and decoder process the mesh one triangle at a time. In every iteration, the algorithm extends the processed part with a triangle from the unprocessed part. The processed and unprocessed parts are separated by gates, i.e., the common bordering edges. The selection of the gate to extend the processed part is driven by priority. The priority of a gate is derived from the quality of its individual candidate vertices and represents the certainty of the decoder to identify the actual tip vertex that forms the extension triangle. The quality q_c of the candidate vertex c is a linear combination of geometric properties, illustrated in Fig. 1, and is defined as

$$q_c = \theta_{Cc} - w_1 \cdot d_{cp} + w_2 \cdot \phi_{BC} + w_3 \cdot S_{BC}, \quad (1)$$

weighted by the coefficients w_1 , w_2 and w_3 . The symbol θ_{Cc} is the inner angle, d_{cp} is the distance to the parallelogram prediction, ϕ_{BC} is the dihedral angle, and S_{BC} is the triangle similarity.

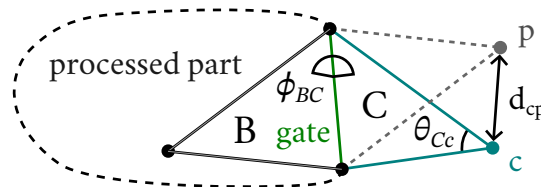


Figure 1: Candidate vertex quality

There is no general means to determine the weights w_1 , w_2 and w_3 of the candidate quality function defined by Eq. (1) for an individual mesh. Since these encoder parameters directly influence the performance of the algorithm, it is promising to determine them per mesh based on its general properties and transmit them to the decoder alongside the compressed data.

2 Proposed Method

The proposed method approaches the problem by learning the relation between meshes and the respective optimal encoder parameters. It utilises a set of descriptive statistics to capture the surface of a mesh and its properties and pairs it with the RBF approximant of the data rate for the corresponding mesh. The method involves building a dataset of many sample pairs of global

¹ student of the bachelor degree program Computer Science and Engineering, field of study Informatics, e-mail: skafara@students.zcu.cz

surface statistics of meshes and respective data rate approximants. This dataset is then used to train an artificial neural network, which models the relation between the surface statistics and the optimal encoder parameters. The model is trained to predict the encoder parameters that minimise the data rate, utilising the approximate data rate provided by the approximants.

The selected global surface statistics consider the geometric properties of a mesh that directly relate to the candidate vertex quality from Eq. (1) and a few others to provide additional information about the mesh. The considered properties are triangle inner angles, distances of triangle vertices to their parallelogram predictions, dihedral angles between adjacent triangles, the similarity of adjacent triangles, triangle equilaterality, edge lengths, and vertex degrees.

The data rate for a mesh is a function of the weights w_1 , w_2 and w_3 from Eq. (1). The function is sampled alongside searching for the optimal parameters over the range of feasible values, as visualised in Fig. 2. The data rate is subject to approximation to enable the estimation of the data rate for parameters for which the function was not sampled, as visualised in Fig. 3.

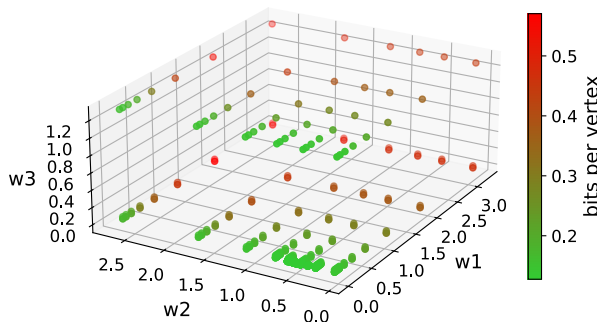


Figure 2: Sampled data rate function

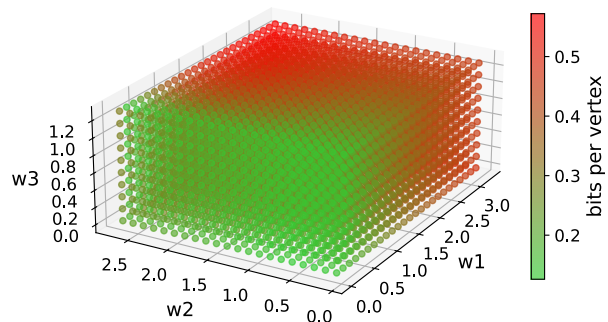


Figure 3: Data rate function approximant

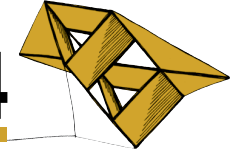
3 Experimental Results

The datasets used in Dvořák et al. (2022) were used to build an experimental dataset comprising 29958 meshes of varying properties. A subset of 26963 meshes was used to train two models, the first model_{total} minimising the total sum of the data rate for the meshes of the dataset and the other model_{mesh} minimising the data rate per mesh. The algorithm performance was evaluated on the other subset of 2995 meshes with the estimated mesh-optimal encoder parameters and compared to the default, dataset-optimised, and mesh-optimal encoder parameters. The default and dataset-optimised parameters were determined by Dvořák et al. (2022) to suit all the meshes, respectively, the individual datasets of the experimental dataset well.

The estimator model_{total} improved the total sum of the data rate by 16.69, respectively, 16.51 per cent over the default and dataset-optimised parameters. The other estimator model_{mesh} improved the data rate by 17.98, respectively, 12.86 per cent per mesh on average over the default and dataset-optimised parameters. The proposed method improved the data rate to be 22.34 per cent in total, respectively, 18.85 per cent per mesh on average worse than the data rate achieved with the mesh-optimal parameters, compared to 46.85 and 53.56 per cent achieved with the default, respectively, 46.55 and 45.56 per cent with the dataset-optimised parameters.

References

Dvořák, J., Káčereková, Z., Vaněček, P. and Váša, L. (2022) Priority-based encoding of triangle mesh connectivity for a known geometry. *Computer graphics forum*, 42(1), pp. 60–71. Available from: <https://doi.org/10.1111/cgf.14719>.



Porovnání paralelní implementace B algoritmu v Javě a C/C++

Petr Pernička¹

1 Úvod

Předpověď dopravy je pokus o odhadnutí počtu vozidel či cestujících, kteří chtějí použít dopravní síť k přepravě mezi různými místy zájmu. Tato oblast dopravního plánování je čím dál tím důležitější pro budoucí vylepšování dopravních sítí, zvláště se stále rostoucím počtem účastníků provozu. Jedním z modelů určující metodiku pro předpověď dopravy je takzvaný čtyřkrokový model, jehož posledním krokem je problém přiřazení dopravy.

Zde je představen problém přiřazení dopravy a Algoritmus B jakožto jeden z algoritmů řešící statickou variantu tohoto problému. Následně jsou zde uvedeny výsledky porovnání implementace paralelního Algoritmu B v Javě a C++.

2 Problém přiřazení dopravy

Ve studii Mitchell a Rapkin (1954) pokládající základy dopravní analýzy byl představen čtyřkrokový model, který je používán pro předpověď rozložení dopravních požadavků cestujících. Vstupem modelu je graf reprezentující dopravní síť, kde hrana reprezentuje silnici a vrchol pak křižovatku nebo zónu, což jsou agregovaná prostorová data, např. obec či sousedství. Zóny v modelu slouží jako zdroje a stoky dopravního toku.

V prvních třech krocích modelu se identifikuje velikost toků, které začínají a končí v každé ze zón. Tento výsledek je zaznamenán ve formě matice počátků a cílů, kde prvek na pozici ij reprezentuje tok ze zóny i do zóny j . Graf s maticí jsou vstupem posledního kroku — přiřazení dopravy. Ve statickém přiřazení dopravy jsou prvky matice konstanty a výsledek reprezentuje průměr přes modelovaný časový úsek.

Cílem přiřazení dopravy je rozdělit toky z matice mezi cesty v dopravní síti tak, aby platila uživatelská rovnováha formulovaná ve Wardrop (1952). Ta nastává právě tehdy, když žádný cestující nemůže snížit dobu své jízdy volbou alternativní trasy. Jelo-li by z i do j nějaké vozidlo samotné, zvolilo by nejkratší cestu, protože je nejrychlejší. Jakmile se však dopravní tok začne zvyšovat, nemusí být nejkratší cesta ta nejrychlejší. Tento dopravní odpor je charakterizován tzv. cenovými funkcemi. Nejpoužívanější z nich je BPR funkce představena v U.S. Bureau of Public Roads (1964).

3 Algoritmus B

Algoritmus B z Dial (2006) je numerický grafový algoritmus pro statické přiřazení dopravy a dosažení uživatelské rovnováhy. Stejně jako ostatní algoritmy pro statické přiřazení dopravy je B algoritmus iterativní — s každou iterací se více a více blíží uživatelské rovnováze. Efektivita B vychází z rozdělení sítě na keře, což jsou acyklické podgrafy s jediným kořenem.

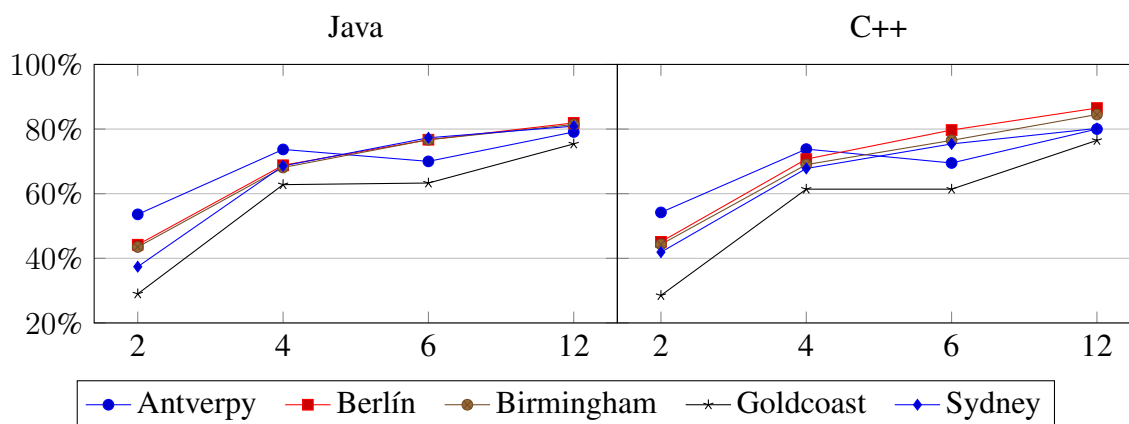
¹ student bakalářského studijního programu Informatika a výpočetní technika, specializace Informatika, e-mail: pperncka@students.zcu.cz

Algoritmus začíná tím, že si pro každou zónu v síti vytvoří keř v ní zakořeněný. V keři jsou pak jen hrany začínající ve vrcholu, který je blíže kořeni než vrchol, ve kterém hrana končí. Všechny tok z i do j je na začátku přiřazen nejkratší cestě mezi těmito zónami.

Následuje iterační fáze, kdy se v každé iteraci zpracují všechny keře a to povede ke konvergenci k uživatelské rovnováze. Každému keři se nejdříve zlepší topologie — jsou přidány některé hrany za zachování acyklicity. Následně se naleznou stromy minimálních a maximálních cest použitím standardního algoritmu pro nalezení těchto cest v acyklickém grafu. Poté se pomocí Newtonova algoritmu vyváží toky mezi minimální a maximální cestou ke každému vrcholu tak, aby mezi cestami nastala uživatelská rovnováha. Zpracování keře končí odstraněním hran, na kterých neteče žádný tok.

Samotný algoritmus končí poté, co dosáhl požadované přesnosti.

4 Výsledky porovnání

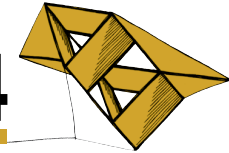


Obrázek 1: Urychlení paralelní implementace oproti sekvenční na počítači s AMD Ryzen 5 1600X. Horizontální osa značí počet vláken.

Algoritmus byl porovnáván na pěti reálných sítích. Bylo zjištěno, že implementace v C++ je o $\sim 8\%$ rychlejší a potřebuje o $\sim 4\%$ méně paměti. Nastaly však případy, kdy byla Java neznatelně rychlejší. Oba jazyky paralelizací urychlují algoritmus skoro stejně, viz Obrázek 1. S 12 vlákny může být algoritmus rychlejší až o 80%. Někdy však zvýšení počtu vláken dokáže zvýšit počet iterací potřebných ke konvergenci natolik, že výsledně algoritmus potřebuje více času.

Literatura

- Dial, R. B. (2006) A path-based user-equilibrium traffic assignment algorithm that obviates path storage and enumeration. *Transportation Research Part B*. Amsterdam, Elsevier, Volume 40, pp 917-936.
- Mitchell, R. B., Rapkin, C. (1954) *Urban Traffic: A Function of Land Use*. New York, Columbia University Press.
- United States. Bureau of Public Roads (1964) *Traffic Assignment Manual: For Application with a Large High Speed Computer*. U.S. Department of Commerce, Bureau of Public Roads, Office of Planning, Urban Planning Division.
- Wardrop, J. G. (1952) Some Theoretical Aspects of Road Traffic Research. *Proceedings of the Institution of Civil Engineers*. ICE Publishing, Volume 1, pp. 325-362.



Porovnání algoritmů pro dělení silniční sítě

Lucie Roy¹

1 Úvod

V dnešní době, kdy je ve velkoměstech velká hustota obyvatel a tím pádem i hodně dopravních prostředků, není jednoduché zachovat plynulost dopravy. Jedním z nástrojů, který tomu může pomoci, je distribuovaná simulace dopravy umožňující mj. předvídat, jak bude doprava vypadat při uzavírce pruhu nebo změně světelných plánů křižovatek. Distribuovaná simulace se používá hlavně pro simulace velkých sítí (např. celého velkoměsta). Tato simulace využívá více propojených počítačů a každý počítač simuluje pouze část silniční sítě. Proto je nutné silniční síť před samotnou simulací rozdělit. Jelikož kvalita dělení může hodně ovlivnit rychlost simulace, bylo již vytvořeno několik algoritmů řešících tento problém.

Kvalita algoritmů je často testována na různých sítích, jsou implementovány v různých jazycích, a proto je vzájemné porovnání na základě testů popsaných ve vědeckých publikacích obtížné. Cílem této práce bylo vytvořit nástroj umožňující porovnání různorodých algoritmů v jednotném prostředí. Díky tomuto nástroji budou algoritmy implementovány v jednom jazyce a testovány stejnou sadou testů. Kromě samotného nástroje byly implementovány a porovnány tři vybrané algoritmy.

2 Vytvořený nástroj a vybrané dělicí algoritmy

Byla vytvořena desktopová aplikace v jazyce Java s uživatelským rozhraním. Tato aplikace umožňuje přidávat nové dělicí algoritmy pomocí reflexe. Uživatel si do ní může nahrát graf, který má v GeoJSON souboru, ten si zobrazit a rozdělit podle libovolného algoritmu dostupného v aplikaci. Následně je i v aplikaci možné toto dělení graficky znázornit. Také je zde možné otestovat všechny dělicí algoritmy najednou se stejnými testovacími daty několikrát za sebou.

Byly implementovány tři algoritmy, a to METIS, SPaRTSim a Inertial Flow. Algoritmy byly vybrány na základě kvality jejich popisu, který musel být dostatečný pro následnou implementaci pro účely této práce, požadovaných dat pro běh algoritmu a vhodnosti pro homogenní statické dělení silniční sítě. Zároveň byly vybrány algoritmy, které si nejsou navzájem moc podobné.

Algoritmus METIS je jedním z pokročilejších algoritmů pro dělení grafu. Tento algoritmus používá víceúrovňové schéma, kde se provede nejdříve zhrubnutí grafu, poté se zhrubnutý graf rozdělí a nakonec se provede zjemnění grafu [Karypis a Kumar (1999)].

Dalším algoritmem je SPaRTSim, který patří do kategorie dělicích algoritmů, které byly vytvořeny speciálně pro dělení dopravní sítě. Tento algoritmus má čtyři hlavní části: inicializaci, růst, vyvážení a propojení. Při inicializaci se vyberou nejvhodnější počáteční vrcholy, kterých je tolik, jako je požadovaných částí. Z vybraných vrcholů se jednotlivé části dále rozrůstají,

¹ studentka bakalářského studijního programu Informatika a výpočetní technika, obor Informatika, e-mail: koudelko@students.zcu.cz

pokud už ani jedna část nemá kam růst, pak se provede vyvážení jednotlivých částí a jelikož je možné, že se během vyvážení některé části rozpojily, spojí se některé sousední části do jedné tak, aby výsledné dělení mělo požadovaný počet částí a bylo co nejvyváženější [Ventresque et al. (2012)].

Posledním implementovaným algoritmem je Inertial Flow. Tento algoritmus funguje na bázi řazení vrcholů grafu podle výskytu jejich ortogonálního obrazu na zvolené přímce. Z prvních několika vrcholů se vytvoří jeden hromadný vrchol a z posledních několika vrcholů také. Mezi těmito hromadnými vrcholy se poté najde maximální tok a v hranách náležících maximálnímu toku se provede řez grafem, tím vznikne minimální řez [Schild a Sommer (2015)]. Jelikož v grafu může existovat více možností minimálního řezu a v článku nebyla popsána metoda, jak vybrat tu nejlepší, byla implementována vlastní metoda pro účely této práce. Metoda vybírá minimální řez, který vytvoří nejvyváženější dělení.

Funkčnost vytvořené aplikace byla demonstrována na všech implementovaných algoritmech, které byly pomocí aplikace otestovány a porovnány. Hodnoceny byly tyto vlastnosti: čas dělení, relativní odchylka od ideálního vyvážení, počet rozdělených silnic/hran, minimální počet sousedů, maximální počet sousedů a průměrný počet sousedů. Ovšem nedá se říci, že zjištěné hodnoty platí pro zmíněné algoritmy obecně, jelikož jejich implementace nemusí být optimální.

Jako nejstabilnější z implementovaných algoritmů s dobrým kompromisem mezi vyvážením a malým počtem rozdělených hran a malým počtem sousedů mezi částmi se ukázal SParTSim. Pro minimalizaci počtu rozdělených hran a počtu sousedů mezi částmi, s čímž právě souvisí množství komunikace mezi částmi, by se nejvíce hodil Inertial Flow, ovšem ten má často značně nevyvážené dělení. Závěrem lze tedy říci, že při volbě algoritmu, je dobré si ujasnit požadavky dělení a podle nich zvolit dělicí algoritmus.

3 Závěr

V rámci této práce byl vytvořen nástroj pro porovnání algoritmů pro dělení silniční sítě v jednotném prostředí a jednotnou sadou testů. Vytvořený nástroj je plně funkční a umožňuje opakované spuštění testů vybraného algoritmu pro dělení silniční sítě s vybranou dopravní sítí a dalšími parametry. Veškerý zdrojový kód, testovací data a spustitelná aplikace jsou k dispozici na Github.com¹, kde je možné si to stáhnout a otestovat vlastní dělicí algoritmy.

Literatura

- Karypis, George; Kumar, Vipin. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. *Siam Journal on Scientific Computing*. 1999, roč. 20, č. 1, s. 359–392. Dostupné z doi: 10.1137/S1064827595287997.
- Ventresque, Anthony et al. SParTSim: A Space Partitioning Guided by Road Network for Distributed Traffic Simulations. In: *2012 IEEE/ACM 16th International Symposium on Distributed Simulation and Real Time Applications*. 2012, s. 202–209. Dostupné z doi: 10.1109/DS-RT.2012.37.
- Schild, Aaron; Sommer, Christian. On Balanced Separators in Road Networks. In: *BAMPIS, Evripidis (ed.). Experimental Algorithms*. Cham: Springer International Publishing, 2015, s. 286–297. isbn 978-3-319-20086-6.

¹<https://github.com/Lulu1234/RoadNetworkPartitioning>

Automatic Creation of Summaries of Historical Documents

Václav Tran¹

1 Introduction

In the domain of automatic text summarization, neural networks show promising performances. This work probes into the task of automatic summarization of Czech historical documents, a largely unexplored niche area with a scant amount of datasets available. To evaluate and improve the performance of our methods, we created our own dataset constructed from a corpus of historical documents. Then we fine-tuned and utilized Transformer-based models Mistral 7B and mT5. We also implemented and evaluated a method, which we refer to as Translation-Summarization-Translation, where we utilize state-of-the-art machine translation and English summarization methods to generate Czech summaries. The performance of these methods set a new baseline for the task of summarizing Czech historical documents.

2 Methods

We fine-tuned a 512-million parameter variant of mT5 from Xue et al. (2021) and the Mistral 7B model from Jiang et al. (2023) on SumeCzech dataset from Straka et al. (2018), which contains collection of articles from Czech news sites alongside their summaries. Additionally, we curated our own dataset, which we abbreviated as POC, comprising of historical documents from *Posel od Čerchova* journals and their respective summaries. Mistral 7B was additionally fine-tuned on the POC dataset. By using various training optimization methods, we managed to fit the entire process of fine-tuning Mistral 7B on a single NVIDIA A40 45 GB GPU.

For Translation-Summarization-Translation (TST) method, we used a machine translation model ALMA from Xu et al. (2024) and instruct fine-tuned Mistral 7B for text summarization. TST translates the given Czech text to English, summarizes it using the preferred English summarization model and translates the English summary back to Czech.

3 Results

Mistral 7B additionally fine-tuned on POC is abbreviated as M7B-POC. The mT5 model fine-tuned on SumeCzech dataset is abbreviated as mT5-SC. Using POC dataset, we evaluated these three methods on POC-I and POC-P using ROUGE_{RAW} metric by Straka et al. (2018), where M7B-POC achieved the highest overall performance. POC-I is a subset of POC that contains summaries of issues and POC-P is a subset of POC which contains summaries of pages of individual issues. However, our limited observations of the generated summaries suggest that higher performance on the chosen evaluation metric does not necessarily indicate superior summarization quality, particularly with regard to factuality.

¹ student of the bachelor degree program Applied Sciences, field of study Computer Science, e-mail: nuva@students.zcu.cz

Table 1: Results of implemented methods on POC-P. M7B-POC was evaluated only on summaries it has not been trained on.

Method	ROUGE _{raw} -1			ROUGE _{raw} -2			ROUGE _{raw} -L		
	P	R	F	P	R	F	P	R	F
M7B-POC	23.5	17.4	19.6	4.8	3.5	4.0	16.6	12.2	13.8
TST	17.2	25.1	19.9	2.5	3.8	2.9	11.3	16.4	13.0
mT5-SC	20.2	8.2	11.1	1.4	0.5	0.7	14.9	6.1	8.2

Table 2: Results of implemented methods on POC-I. M7B-POC was evaluated only on summaries it has not been trained on.

Method	ROUGE _{raw} -1			ROUGE _{raw} -2			ROUGE _{raw} -L		
	P	R	F	P	R	F	P	R	F
M7B-POC	19.3	17.6	18.0	3.2	2.8	2.9	13.7	12.4	12.8
TST	14.0	24.8	17.5	1.7	3.1	2.1	9.1	16.3	11.4
mT5-SC	18.2	5.9	8.6	1.0	0.3	0.4	14.0	4.5	6.5

Acknowledgement

I would like to thank my thesis advisor Doc. Ing. Pavel Král, Ph.D. for their constant support and help. I would also like to thank everyone, who supported me throughout the creation of this Bachelor thesis, including Zach for checking my work for grammatical errors. Computational resources were provided by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth and Sports of the Czech Republic.

References

- Straka, M., Mediankin, N., Kocmi, T., Žabokrtský, Z., Hudeček, V., and Hajič, J. (2018). SumeCzech: Large Czech News-Based Summarization Dataset. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA), Miyazaki, Japan.
- Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. (2021). mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou (Eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 483–498). Association for Computational Linguistics, Online. <https://doi.org/10.18653/v1/2021.naacl-main.41>
- Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Singh Chaitan, D., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Renard Lavaud, L., Lachaux, M.-A., Stock, P., Le Scao, T., Lavril, T., Wang, T., Lacroix, T., and El Sayed, W. (2023). Mistral 7B. arXiv preprint arXiv:2310.06825.
- Xu, H., Sharaf, A., Chen, Y., Tan, W., Shen, L., Van Durme, B., Murray, K., and Kim, Y. J. (2024). Contrastive Preference Optimization: Pushing the Boundaries of LLM Performance in Machine Translation. arXiv preprint arXiv:2401.08417.