

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE
FAKULTA ŽIVOTNÍHO PROSTŘEDÍ

KATEDRA APLIKOVANÉ GEOINFORMATIKY
A ÚZEMNÍHO PLÁNOVÁNÍ



Nástroj pro tvorbu hydrologicky korektních
digitálních modelů terénu

DIPLOMOVÁ PRÁCE

Vedoucí práce: Ing. Vojtěch Barták, Ph.D.

Autor: Bc. Daniel Röschel

2019

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Fakulta životního prostředí

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Daniel Röschel

Voda v krajině

Název práce

Nástroj pro tvorbu hydrologicky korektních digitálních modelů terénu

Název anglicky

Tool for hydrological correction of digital terrain models

Cíle práce

Vytvořit nástroj implementující vybrané algoritmy pro odstraňování hydrologicky problematických míst (depresí) z digitálních modelů terénu.

Metodika

Vybrané algoritmy pro odstraňování depresí budou implementovány pomocí programovacího jazyka Python jako nástroj integrovaný do programového prostředí ArcGIS 10.5. Nástroj bude opatřen uživatelským rozhraním a jeho funkčnost bude testována na reálných digitálních modelech terénu.

Doporučený rozsah práce

40 – 60 stran

Klíčová slova

Digitální analýza terénu, Hydrologicky korektní DTM, Python, ArcGIS

Doporučené zdroje informací

- Jenson S. K., Dominique J. O., 1988: Extracting Topographic Structure from Digital Elevation Data for Geographic Information System Analysis. *Photogrammetric Engineering And Remote Sensing*, 54(11). November 1988. S. 1593-1600.
- Martz L. W., De Jong E., 1988: CATCH: a FORTRAN program for measuring catchment area from digital elevation models. *Computers&Geosciences* 14(5). S. 627-640.
- Martz L. W., Garbrecht J., 1999: An outlet breaching algorithm for treatment of closed depressions in a raster DEM. *Computers&Geosciences* 25. S. 835-844.
- O'Callaghan J.F., Mark D.M., 1984: The extraction of drainage networks from digital elevation data. *Computer Vision, Graphics, and Image Processing* 28(3). S. 323-344.
- Soille P., Vogt J., Colombo R., 2003: Carving and adaptive drainage enforcement of grid digital elevation models. *Water Resources Research* 39(12). 1366. doi:10.1029/2002WR001879
- Soille P., 2004: Optimal removal of spurious pits in digital elevation models. *Water Resources Research* 40. W12509, doi:10.1029/2004WR003060.

Předběžný termín obhajoby

2018/19 LS – FŽP

Vedoucí práce

Ing. Vojtěch Barták, Ph.D.

Garantující pracoviště

Katedra aplikované geoinformatiky a územního plánování

Elektronicky schváleno dne 25. 3. 2019

doc. Ing. Petra Šímová, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 26. 3. 2019

prof. RNDr. Vladimír Bejček, CSc.

Děkan

V Praze dne 29. 03. 2019

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, pod vedením Ing. Vojtěcha Bartáka, Ph.D.. Další informace mi poskytl Ing. Petr Bašta. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal. Prohlašuji, že tištěná verze se shoduje s verzí odevzdanou přes Univerzitní informační systém.

V Praze dne 18.04.2019

.....

Poděkování

Touto cestou bych chtěl především moc poděkovat vedoucímu této diplomové práce Ing. Vojtěchu Bartákovi, Ph.D. za cenné rady, připomínky, velkou trpělivost a čas strávený při konzultacích. Dále bych chtěl poděkovat své rodině za velkou podporu.

V Praze dne 18.04.2019

.....

Abstrakt

Tato diplomová práce je zaměřena na problematiku uzavřených depresí, které v digitálních modelech terénu představují hydrologicky problémová místa, ze kterých nelze určit směr odtoku. Navazuje tím na diplomovou práci Ing. Petra Nováka, ve které vytvářel nástroj pro automatickou extrakci říční sítě z hydrologicky korektních digitálních modelů terénu.

První část této diplomové práce je převážně zaměřena na seznámení se s problematikou uzavřených depresí a plošin při simulaci povrchového odtoku modelovaného z rastrových digitálních modelů terénu. Jsou vysvětleny všechny pojmy vyskytující se v této práci a vznik uzavřených depresí v digitálních modelech terénu. Dále jsou představeny a vysvětleny jednotlivé algoritmy od různých autorů řešící tato hydrologicky problémová místa.

Praktická část je zaměřena na tvorbu dvou algoritmů odstraňujících uzavřené deprese v digitálním elevačním modelu pomocí programovacího jazyka Python, které jsou dále integrovány jako nástroje do prostředí ArcGIS. První algoritmus je vytvořen na principu plnění prohlubní (Fill) a druhý naopak využívá tzv. protržení hráze deprese vyrytím cesty ode dna deprese směrem ven z prohlubně (Carving).

Funkčnost nástrojů byla vyzkoušena na dvaceti digitálních elevačních modelech náhodně vybraných hydrologických povodích III. a IV. řádu a výstupy byly následně statisticky vyhodnoceny podle toho, jak moc se liší jejich výstupy od původního digitálního elevačního modelu z hlediska počtu a objemu modifikovaných buněk.

Klíčová slova:

Digitální analýza terénu, Hydrologicky korektní DTM, Python, ArcGIS, Simulace povrchového odtok

Abstract

This diploma thesis is focused on the issue of closed depressions, which in digital terrain models represent hydrologically problematic places from which cannot be determined the flow direction. This thesis is a continuation of the diploma thesis by Ing. Petr Novák, in which he created a tool for automatic river networks extraction from hydrologically correct digital terrain models.

The first part of this thesis is mainly focused on introduce with the issues of closed depressions and flat areas while surface outflow modeled from raster digital terrain models. Are explained all terms in this work and the origin of closed depressions in digital terrain models. Next are introduced and explained individual algorithms from various authors solving with these hydrologically problematic areas. are presented and explained.

The practical part is focused on creation of two algorithms removing closed depressions in digital elevation model in Python programming language. These are further integrated as tools into ArcGIS. The first algorithm is created on the principle of filling depressions (Fill) and second used outlet breaching of depression by engraving the path from the bottom of the depression to the outside of the depression (Carving).

The functionality of the tools was tested on twenty digital elevation models of randomly selected hydrological basins III. and IV. order and outputs were then statistically evaluated accordingly how much their outputs are differend from the original digital elevation model in terms of number and volume of modified cells.

Key words:

Digital terrain analysis, Hydrologically correct DTM, Python, ArcGIS, Surface outflow

Obsah

1. Úvod	1
2. Cíle práce	2
3. Literární rešerše	3
3.1. Základní pojmy	3
3.1.1. Digitální model terénu	3
3.1.2. Digitální elevační model	5
3.1.3. Uzavřené deprese.....	6
3.1.4. Plošiny.....	6
3.1.5. Povrchový odtok.....	7
3.2. Typy uzavřených depresí	8
3.2.1. Chyby vzniklé při měření terénu.....	8
3.2.2. Chyby vzniklé při interpolaci.....	9
3.3. Řešení uzavřených depresí	11
3.3.1. Metoda plnění depresí (Filling)	12
3.3.2. Metoda protržení hráze (outlet breaching)	18
3.3.3. Metoda vyrývání (carving).....	20
3.3.4. Metody s minimální cenou	21
3.3.5. Fyzikálně založené metody.....	23
3.4. Řešení plošin.....	25
3.4.1. Gradient ke spodnímu okraji plošiny.....	25
3.4.2. Metoda nejkratší cesty	26
3.4.3. Gradient od horního okraje plošiny	26
3.4.4. Metoda Wang a Liu	27
4. Metodika	28
4.1. Použitý software	28
4.1.1. ArcGIS 10.4.1	28
4.1.2. Programovací jazyk Python	28
4.1.3. ArcPy.....	29
4.1.4. NumPy	29
4.1.5. Programovací jazyk R.....	29
4.2. Načtení modulů	29

4.3.	Načtení a převod DEM.....	31
4.4.	Tvorba binární haldy	31
4.5.	Hledání a plnění depresí.....	32
4.6.	Vyrývání (Carving)	34
5.	Výsledky práce	36
5.1.	Popis nástrojů	37
5.2.	Popis skriptů	38
5.3.	Ukázka použití	39
5.3.1.	Spuštění nástroje	39
5.3.2.	Nalezení uzavřených depresí a plošin	39
5.3.3.	Vyplnění depresí	41
5.3.4.	Vyrytí cesty z deprese.....	42
5.4.	Porovnání metod	43
6.	Diskuze.....	48
7.	Závěr.....	49
8.	Přehled literatury a použitých zdrojů.....	50
9.	Seznam obrázků	55
10.	Seznam tabulek	57
11.	Přílohy.....	58
11.1.	border_of_raster.....	58
11.2.	neighbors.....	60
11.3.	depression_layer.....	62
11.4.	fill_DEM.....	67
11.5.	carving.....	70
11.6.	Obsah přiloženého CD	73

Seznam zkratk

DTM	Digitální model terénu
DEM	Digitální elevační model
DMR	Digitální model reliéfu
TIN	Triangulated Irregular Network
GNSS	Globální družicový polohový systém
GPS	Global Positioning System
ZABAGED	Základní báze geografických dat České republiky
DMÚ	Digitální model území
SFD8	Single Flow 8-Direction
PEM4PIT	Physical Erosion Model for PIT filling
GIS	Geographic Information System
IDLE	Python's Integrated DeveLopment Environment
NumPy	Numerical Python

1. Úvod

Hydrologické procesy patří k nejdůležitějším procesům v přírodě. K jejich pochopení a zjištění vzájemných závislostí nám v dnešní době pomáhají hydrologické modely, které lze najít v podobě nástrojů v geografických informačních systémech určených pro práci s prostorovými daty, jako je například ArcGIS. Tyto modely mohou sloužit např. k simulaci povrchového odtoku pro určité území, která má následně mnoho uplatnění.

Zpracovávané území bývá často prezentované v podobě digitálních modelů terénu. Ty většinou obsahují topografické útvary, které vytváří problém při zmíněné simulaci povrchového odtoku. Jedná se o uzavřené deprese a plošiny, tj. buňky digitálního modelu terénu, které nemají v žádném směru kladný sklon, a nejde pro ně tedy určit směr odtoku. V současné době existuje mnoho metod od různých autorů řešících tuto problematiku s rozdílnými výsledky.

Jednotlivé metody odstraňující tyto uzavřené deprese se mohou lišit ve složitosti vytvořeného algoritmu projevující se délkou výpočetního času nebo v počtu modifikovaných buněk oproti původnímu modelu terénu. V případě vzniku vyššího počtu změn ve výsledném modelu terénu na rozdíl od původního může následně dojít při simulaci povrchového odtoku k méně přesným výsledkům. Nejlepší je tedy vybrat metodu, která nebude při analýze většího území moc časově náročná a která ve výsledku prezentuje původní digitální model terénu s odstraněnými depresemi a plošinami co nejméně pozměněný.

2. Cíle práce

Cílem této diplomové práce je v první části seznámení s řešenou problematikou uzavřených depresí a plošin při simulaci povrchového odtoku. V další praktické části tvorba dvou nástrojů pro odstraňování hydrologicky problematických míst, implementujících metodu Fill a metodu Carving, integrovaných do programového prostředí ArcGIS. Dalším cílem je na reprezentativním vzorku dat porovnat výsledky obou algoritmů z hlediska počtu a objemu modifikovaných buněk.

3. Literární řešerše

Tato kapitola je především zaměřena na popis jednotlivých metod řešící problematiku uzavřených depresí. Do značné míry kopíruje strukturu diplomové práce Barták (2008), která zde byla použita jako základní zdroj informací.

3.1. Základní pojmy

3.1.1. Digitální model terénu

Jedním z nejpoužívanějších pojmů v této diplomové práci je Digitální model terénu (DTM, Digital Terrain Model) a jeho druh digitální elevační model (DEM, Digital Elevation Model). Ty jsou často používané v geografických informačních systémech. DTM jindy nazývaný jako DMR (Digitální model reliéfu) je zjednodušený model zemského povrchu v digitální podobě, který neobsahuje např. budovy, stromy a další objekty (Pacina, Brejcha, 2014) (Bašta, 2016).

Existují tři různé metody zobrazení reálného terénu v jeho digitální podobě, které se liší použitým algoritmem. Prvním příkladem vyjádření výškopisu jsou vrstevnice. Vrstevnice jsou křivky spojující výškové body se stejnou nadmožskou výškou, které tvoří tzv. nekompletní zobrazení povrchu. Jsou i vstupními daty pro tvorbu dalších zobrazení zemského povrchu (Rastr) (Pacina, Brejcha, 2014).

Další metodou je zobrazení DTM pomocí nepravidelných nepřekrývajících se sousedících trojúhelníkových ploch (TIN, Triangulated Irregular Network). Pro tvorbu TIN jsou potřeba vstupní data se souřadnicemi x , y , z , které následně tvoří ve výsledném TIN vrcholy nepravidelných trojúhelníků (Šimová, 2013). Jednotlivé vrcholy jsou propojeny tak, aby s nejbližším sousedem tvořily hranu trojúhelníku a následně s ostatními hranami trojúhelníku. Hrana má tedy dva vrcholy a z něj vycházející minimálně dvě hrany. Ze známých souřadnic vrcholů můžeme odvodit výšku, a i jiné vlastnosti (sklon, aspekt) mezi vrcholy a uvnitř trojúhelníku (Bašta, 2016) (El-Sheimy a kol., 2005).

Rastrový DTM je pravidelná pravouhlá síť (většinou) čtvercových buněk, v níž každá buňka nese informaci o nadmořské výšce. Buňka (Pixel) je zde chápána jako základní stavební prvek struktury rastru a jeho poloha v rastru je dána x a y souřadnicí. DTM je zde zobrazován jako elementární množina buněk v mřížce (rastru). Výhodou této metody je lepší implementace a rychlost prováděných analýz. Rychlost silně závisí na jeho rozlišení, což je velikost hrany buňky rastru. Čím je rozlišení menší, tím je rastrový model podrobnější (Brůha, 2016).

Zdroje dat pro tvorbu DTM se může získat z různých zdrojů za využití různých metod. Ty se rozdělují podle způsobu získávání dat na metody:

- Přímé
 - Kontaktní
 - Bezkontaktní
- Nepřímé metody

Přímou metodou získáváme data přímým měřením v terénu za pomoci geodetických měření (nivelace, tachymetrie) nebo GNSS (globální družicový polohový systém). Ten funguje na principu, kde výpočet polohy je určen pomocí družic a např. GPS přijímačem. U těchto metod se jedná o kontaktní měření a dokážou poskytnout velmi přesné určení polohy a výšky (Pacina, Brejcha, 2014).

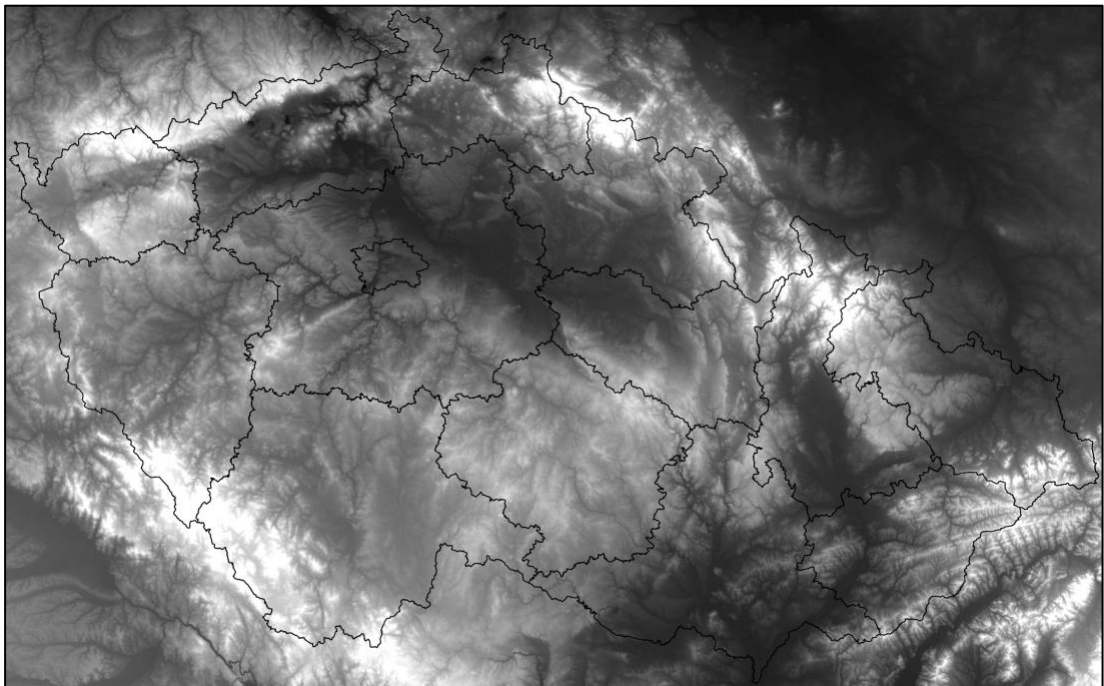
Naopak bezkontaktní metoda využívá dálkový průzkum Země. Ten zahrnuje např. fotogrammetrii zpracovávající stereo-dvojice leteckých snímků, laserové skenování a radarové snímání pomocí družic (Pacina, Brejcha, 2014).

Jestliže zpracováváme existující model terénu, jedná se o metodu nepřímou. Zpracováváný model nemusí být v digitální podobě. Např. se jedná o tvorbu DMR digitalizací vrstevnic analogových map, která pak můžou sloužit ke tvorbě DMR. Nevýhoda této metody je závislost měřítka a přesnost zdrojových map (Pacina, Brejcha, 2014).

Data ke tvorbě DTM lze stáhnout na příslušných stránkách, kde poskytují již existující digitální a analogové sady dat (ZABAGED, digitální model reliéfu, DMÚ 25). DEM pro Českou republiku v rozlišení 100 m je možné bezplatně stáhnout i na stránkách Gisat (Šimeček, 2017).

3.1.2. Digitální elevační model

DEM (viz obr. 1) je specifický případ DTM a je téměř vždy reprezentován v podobě rastru. Někdy jsou tyto dva termíny definovány stejně, ale ve skutečnosti je koncept DEM širší a univerzálnější. V souhrnu lze říci, že digitální elevační model je soubor digitálních dat popisující výškové hodnoty zemského povrchu (nebo jakéhokoli jiného povrchu), který může obsahovat i jiné informace o charakteru tohoto povrchu (např. lomy, vodní útvary) (Szypuła, 2017).



Obrázek 1: Digitální elevační model pro Českou republiku s rozlišením 100 m (GISAT, 2007).

DEM se používají k odvození informací o morfologii povrchu Země. Díky algoritmům, které jsou zahrnuty ve většině rastrových systémů, můžeme vypočítat např. sklon, aspekt a stínování reliéfu (Jenson, Domingue, 1988).

U modelů terénu můžeme tedy rozlišovat primární a sekundární vlastnosti odvoditelné z DEM. Vlastnosti, které se dokáží vypočítat přímo z hodnot elevací, lze považovat za primární vlastnosti (např. sklon, aspekt). Naopak sekundární vlastnosti je potřeba vypočítat z primárních vlastností povrchu terénu (např. povodí deprese). Primární vlastnosti mohou být zjištěny za pomoci hodnot osmi sousedů. Tedy z hodnot všech sousedů jedné buňky rastru (Moore a kol., 1991).

Dále jsou široce používány pro různé typy hydrologického modelování. Správným použitím příslušného DEM by bylo teoreticky možné odhadnout kritické parametry v hydrologických procesech povrchových vod, jako je právě směr proudění a akumulace toku. Tyto modely mohou být pro praktické použití kombinovány s jinými datovými soubory, jako jsou data o srážkách a infiltraci (Pilesjö, Zhou, 1997).

3.1.3. Uzavřené deprese

Skoro většina DEM obsahuje mnoho topografických uzavřených depresí, které jsou definovány jako oblasti bez odtoku povrchové vody a často označované jako propady nebo prohlubně (ang. sink) (Zandbergen, 2006). V DEM s pravidelnou sítí jsou deprese označovány jako oblast, která má jednu nebo více sousedících buněk, s nižší nadmořskou výškou než všechny její obklopující buňky. Tyto deprese pak vytvářejí problémy v simulaci povrchového odtoku a určování směru odtoku, neboť tok nemůže pokračovat „do kopce“ (Zhu a kol., 2013).

Postup při odstraňování uzavřené deprese zahrnuje někdy potřebu určení bodu přetečení (overflow point nebo pour point) a je definován jako bod, který má ze všech bodů ležících na okraji deprese nejnižší elevaci (O'Callaghan, Mark, 1984). Pokud bychom začali depresi postupně zalívat vodou, tak by voda právě začala nejprve přetékat v bodě přetečení (Barták, 2008).

3.1.4. Plošiny

Nevýhodou některých metod řešících uzavřené deprese, především u metod založené na principu plnění, je vznik plošin. Tyto plošiny však mohou prezentovat i skutečný odraz plochých míst. V DEM lze plošiny definovat jako skupinu sousedících buněk se stejnou elevací. Plošiny tedy tvoří oblast, ze které nelze také určit směr povrchového odtoku, neboť je obklopena buňkami se stejnou nadmořskou výškou a představují pro odtokový model podobný problém, jako samotné deprese (Jenson, Domingue, 1988).

3.1.5. Povrchový odtok

Odtokové sítě, související odtokové vztahy a odvodňovací povodí jsou základními pojmy ve vědě o Zemi. Odtokové cesty na povrchu terénu jsou linie, ve kterých říční procesy slouží k přepravě vody a nerostných surovin z povodí, což umožňuje, aby gravitační procesy na svazích pokračovaly do krajiny směrem po svahu. Topologie a geometrie sítí odvodnění jsou důležitými obory geomorfologie. Odtokové sítě jsou základem pro definování odtokových systémů, které jsou základním prvkem v hydrologických modelech a plánech řízení zdrojů (O'Callaghan, Mark, 1984) (Tarboton, 1997).

Povrchový odtok zde není brán jako model skutečného odtokového procesu, ale jako přiřazení směru odtoku (flow direction) z dané buňky do jedné nebo více sousedních buněk. Tento proces se provádí na všech buňkách DEM a tvoří uspořádaný řetězec buněk vznikající postupným určováním směru odtoku, které spojují danou buňku s uzávěrovým profilem a tvoří tzv. odtokovou trasu. Soustava těchto tras pak společně tvoří výše zmíněnou tzv. odtokovou síť. Povrchový odtok je tedy brán jako ekvivalent anglického „Overland flood“. Aby se jednalo o tento proces je zde brán přítok v podobě homogenní srážky dopadající na zcela nepropustný povrch bez vegetace. Jedná se tedy o idealizovaný odtok, kdy je pouze sklon terénu odpovědný za pohyb vody, který se neshoduje se skutečnými odtokovými procesy. I přesto se ale často používají k jejich reprezentaci (Barták, 2008) (Tarboton, 1997) (Arge a kol., 2003).

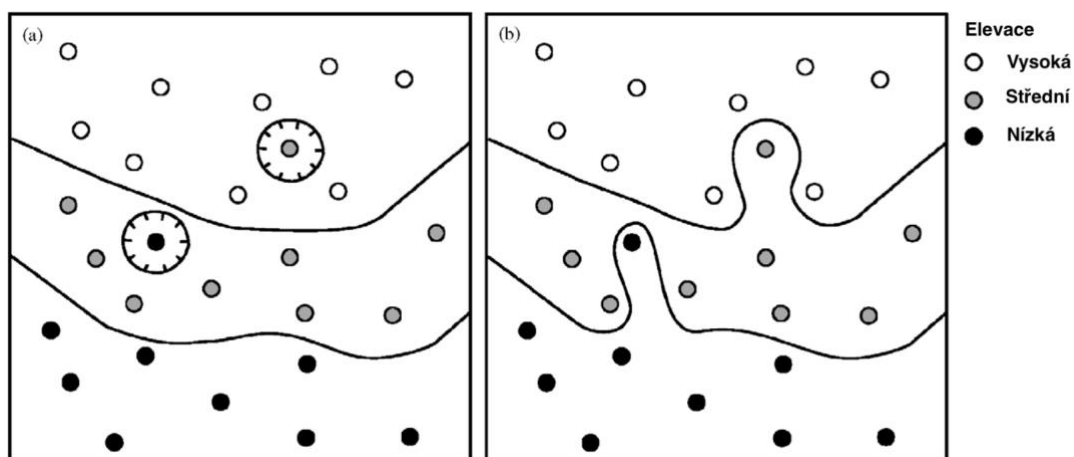
Pokud se jedná o algoritmus, který povolí z dané buňky pouze jeden směr odtoku, je označován odtok jako jednosměrný (single flow). Jestliže se ale jedná o vícesměrný odtok (multiple flow), je z dané buňky povoleno více směrů odtoku (Barták, 2008) (Arge a kol., 2003).

3.2. Typy uzavřených depresí

Buňky v DEM, které nemají přiřazený žádný směr odtoku mohou prezentovat prohlubně vyskytující se ve skutečné krajině, tedy tzv. skutečné deprese (actual depression), nebo jsou důsledkem chyb vzniklých při tvorbě digitálních modelů. V tom případě se jedná o deprese chybové (spurious depressions nebo artifact depressions). Ty jsou sice součástí DEM, ale nezobrazují skutečné prohlubně v krajině. Vznik chybových depresí má dvě příčiny. V první řadě se jedná o chyby v původních výškových datech vzniklé přímo při měření terénu a v druhé o chybové deprese, které mohou nastat při interpolaci původních výškových dat (Florinsky, 2002).

3.2.1. Chyby vzniklé při měření terénu

Tyto chybové deprese lze považovat za chyby způsobené ztrátou dat vznikající při měření nevhodných objektů (budovy) nebo chyby systematické. Systematické chyby vznikají např. při laserovém snímání nerovnoměrně rozložených bodů. Všechny tyto příčiny vytvářející chybové deprese, které lze ale v průběhu procesu minimalizovat a problémem jsou naopak vzniklé náhodné chyby při měření, které tvoří u kvalitně zpracovaných DEM většinu chybových depresí. Ty nelze nijak odstranit, neboť neexistují dokonalé měřicí nástroje nebo metody snímající zemský povrch (Lindsay, Creed, 2005a).



Obrázek 2: Dvě různé interpretace zobrazující vrstevnice (Lindsay, Creed, 2006).

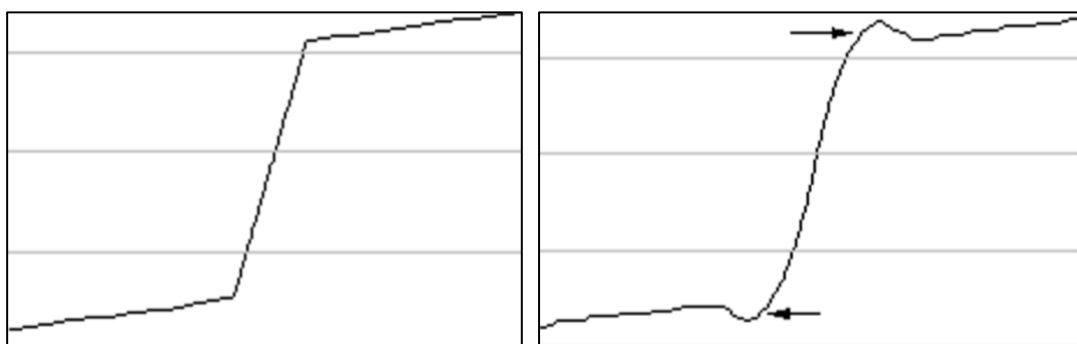
Pokud je DEM vytvořen z vrstevnic, kde jsou uzavřené deprese prezentovány pomocí uzavřených vrstevnic kolem bodu s nižší elevací, může nastat situace, kdy je osamocený bod s nižší elevací než body v jeho okolí označen jako deprese a uzavřen vrstevnicí. Ve skutečné krajině však nejde o skutečnou depresi. Podle kartografických pravidel jsou totiž vybrány ty varianty vrstevnic, které mají co nejmenší zakřivení a dojde v tomto případě ke vzniku uzavřené vrstevnice kolem tohoto bodu (viz obr. 2) (Lindsay, Creed, 2006).

3.2.2. Chyby vzniklé při interpolaci

Interpolace mohou být prováděny buď za účelem vzniku DEM nebo jen kvůli změně jeho rozlišení. Touto problematikou se zabýval Florinsky (2002), který rozděluje tyto chyby do tří základních typů:

- 1) Chyby v interpolaci DEM způsobené Gibbsovým jevem.
- 2) Chyby v interpolaci DEM způsobené příliš podrobnou interpolací.
- 3) Chyby v interpolaci DEM způsobené posunem a otočením mřížky.

Fenomén Gibbsův jev je specifickým chováním některých funkcí, které se projevují jako nadměrné výkyvy kolem diskontinuity skoku. Gibbsův jev je typický např. pro Fourierovy řady, ortogonální polynomy a některé další aproximační funkce. Projevuje se v mnoha vědeckých problémech a aplikacích zahrnujících digitální signál a zpracování obrazu (Jerri 1998). V našem případě se může objevit díky tomuto fenoménu nadhodnocení a podhodnocení elevací v blízkosti míst s prudkými sklony. Na horním okraji prudkého sklonu dojde ke zvýšení elevací buněk a u paty sklonu ke snížení. Tím může dojít i k vytvoření dvou uzavřených depresí u jednoho prudkého sklonu (viz obr. 3) (Florinsky, 2002).



Obrázek 3: Následek Gibbsova jevu v místě prudkého sklonu (Florinsky, 2002).

Ke vzniku chyb způsobených příliš podrobnou interpolací dochází v případě, je-li rozlišení vzniklého DEM příliš jemné vzhledem k hustotě původních dat. Zde nejsou informace o terénu zpřesněny, ale naopak dojde k nárůstu neurčitosti v hodnotách elevací, což může mít za následek nadhodnocení nebo podhodnocení buněk a vznik uzavřených depresí (Barták, 2008).

Chyby způsobené prodloužením a otočením pravidelné mřížky nejsou tak závažné, jak u předchozích dvou. Ze studie autorů Lindsay a Creed (2005a) dokonce vyplývá, že nemá žádný vliv na zvýšení počtu uzavřených depresí v DEM, ale dochází spíše k drobnému posunutí depresí. Počet depresí v DEM ovlivněný touto chybou bude tedy stejný jak u neovlivněného DEM.

3.3. Řešení uzavřených depresí

První způsob řešení problému uzavřených depresí v DEM byl zmíněn v článku od O'Callaghan a Mark (1984). Autoři popisují řešení depresí při automatické extrakci polohy říčních toků na základě simulace povrchového odtoku.

Autoři popisují dva principy odstraňování uzavřených depresí. První postup je založen na úpravě DEM před simulací povrchového odtoku. Dochází ke snížení počtu nebo nejlépe k úplnému odstranění depresí za pomoci tzv. vyhlazení terénu. Všechny buňky v DEM jsou lokálně shlazeny váženým průměrem nadmořských výšek z posuvného okna o rozlišení 3x3 buňky, kdy je hodnotě zpracovávané buňky přidělena váha 0.25 a nadmořským výškám jejich kardinálních sousedů váha 0.125. Okrajové body byly zpracovány tak, že sousedé každého okrajového bodu na hranici matice mají stejnou hodnotu jako okrajový bod. Tato technika byla účinně použita v předchozích studiích. Proces je možné opakovat vícekrát za sebou, dokud nejsou vyhlazeny všechny deprese. Je-li však procedura opakována vícekrát, její účinek se postupně snižuje, takže mělké deprese jsou vyhlazeny, zatímco hrubší mohou ještě přetrvávat. Se zvyšováním počtu opakování dochází i k nežádoucímu účinku způsobenému přílišným shlazováním terénu. Proto se doporučuje tento proces použít jen dvakrát (Barták, 2008).

Autoři článku popisují příklad účinnosti použití vyhlazování na DEM, kde bylo jedním vyhlazením odstraněno víc jak 90 % uzavřených depresí. Pozorovaný inverzní vztah jednoduše naznačuje, že pokud existuje mnoho uzavřených depresí, mají tendenci být malé, zatímco hluboké deprese se vyskytují jen málo (O'Callaghan, Mark, 1984).

Přes velký počet odstraněných uzavřených depresí, které autoři zaznamenali na svých vybraných územích, se tato metoda moc neuchytila a nerozšířila. Metoda má totiž dvě zásadní nevýhody. První je, že hlubší deprese zůstávají nevyřešeny, a tedy metoda není stoprocentní. Další nevýhodou je měnící se nadmořská výška i všech bodů, které neleží v uzavřené depresi, a dochází tedy k nežádoucí změně modelu terénu. Výhodou je však snadná implementace, která je nezávislá na počtu depresí v DEM, při omezeném počtu opakování (Barták, 2008).

Druhý přístup už popisuje případ, kdy jsou odstraněny deprese odolávající shlazením. U této metody je potřeba nejdříve najít bod přetečení ležící na okraji deprese s nejnižší elevací. Princip je založen na nalezení tohoto bodu, odkud se následně pomocí algoritmu SFD8 (tj. jednosměrného povrchového odtoku) najde cesta ke dnu deprese (dead-end element). Při implementaci dochází jednoduše k obrácení směru odtoku na této odtokové cestě a voda ve skutečnosti teče z deprese do kopce. Tím je dobře zachována kontinuita povrchového odtoku. U této metody nedochází na rozdíl od ostatních metod řešící uzavřené deprese k žádné úpravě elevací v modelu terénu. Povrchový odtok je zaznamenán pouze v nově vytvořeném n-rozměrném poli (Barták, 2008) (O'Callaghan, Mark, 1984).

Tato implementace neobsahovala žádný speciální způsob řešení plošin (deprese se stejnými hodnotami elevace) a kvůli nedostatku těchto plošin v datech nebyla ve výsledcích zřejmá zkreslení ve směru odtoku. Pro účely této studie nebylo nutné přezkoumat hodnoty v elevační matici tak, aby odrážely novou odtokovou cestu (O'Callaghan, Mark, 1984).

Nevýhoda této metody je vznik nereálně zakroucených odtokových cest ve větších depresích. Voda totiž nejdříve proudí ze všech míst deprese, kromě cesty spojující dno deprese s bodem přetečení, do dna deprese a následně ven z deprese (Barták, 2008).

3.3.1. Metoda plnění depresí (Filling)

Princip této metody je v první řadě založený na nalezení bodu přetečení uzavřené deprese. Následně je cílem vytvořit upravený DEM bez deprese, ve kterém jsou buňky obsažené v depresích zvednuty na nejnižší hodnotu elevace na okraji deprese, tedy právě na bod přetečení. Po vyřešení vzniklé plošiny bude každá buňka v DEM součástí alespoň jedné monotonicky klesající cesty buněk vedoucí k okraji sady dat (rastru). Povrchový odtok je pak tvořen buňkami, které jsou ve vodorovném, vertikálním nebo diagonálním sousedství v rastru a u kterých se stále snižuje hodnota. V případě, kdyby byly použity původní hodnoty DEM spíše než hodnoty

DEM bez uzavřených depresí, povrchový odtok by skončil ve spodní části uvnitř deprese než na hranici datové sady (Jenson, Domingue, 1988).

Tímto způsobem řešení uzavřených depresí se zabýval Martz a Jong (1988), který jej použili ve svém programu „CATCH“. Program nejdříve pomocí SFD8 algoritmu najde všechny buňky bez směru odtoku (dna depresí) a následně jejich povodí (dead-end zone) s jeho bodem přetečení. Postup je následující:

Po nalezení buňky reprezentující dno deprese je pro tuto buňku identifikován rozsah deprese. Ten je stanoven tak, že buňka dna deprese je skenována skenovacím oknem, které je zpočátku o velikost 10 řádků a 10 sloupců daleko od buňky dna deprese, které je zrovna skenováno. Pro případ, kdy skenovací okno překročí hranice elevační matice, jsou provedeny kontroly, aby se zajistilo, že budou zkoumány pouze prvky v elevační matici. Pokud se objeví buňka, která bude dnem deprese, budou zkoumány všechny přilehlé prvky a ty, které leží ve stejné nebo vyšší výšce budou označeny. Okno se opakovaně skenuje, dokud už nejsou označovány žádné další buňky. Tento proces identifikuje všechny buňky v okně, ze kterých by mohla proudit voda do dna deprese. Tato skupina buněk pak tvoří povodí deprese (Martz, Jong, 1988).

Jakmile je definováno povodí deprese, identifikují se všechny potenciální body přetečení. Potenciální bod přetečení je buňka, umístěná na okraji povodí deprese a která je přilehlá k buňce, která není součástí povodí deprese a existuje k němu pozitivní sklon. Bod přetečení je pak ten, který má ze všech těchto potenciálních bodů nejnižší elevaci (Martz, Jong, 1988).

Pokud existují dva nebo více potenciálních bodů přetečení se stejnou nejnižší elevací, je vybrán ten, který má nejstrmější svah z povodí deprese. V případě, kdy má více než jeden nejnižší elevaci a stejný sklon z povodí, zvolí se jako bod přetečení ten, který byl vybrán nejdříve (Martz, Jong, 1988).

Aby nedošlo k selhání skenovacího okna, je potřeba se držet základních podmínek:

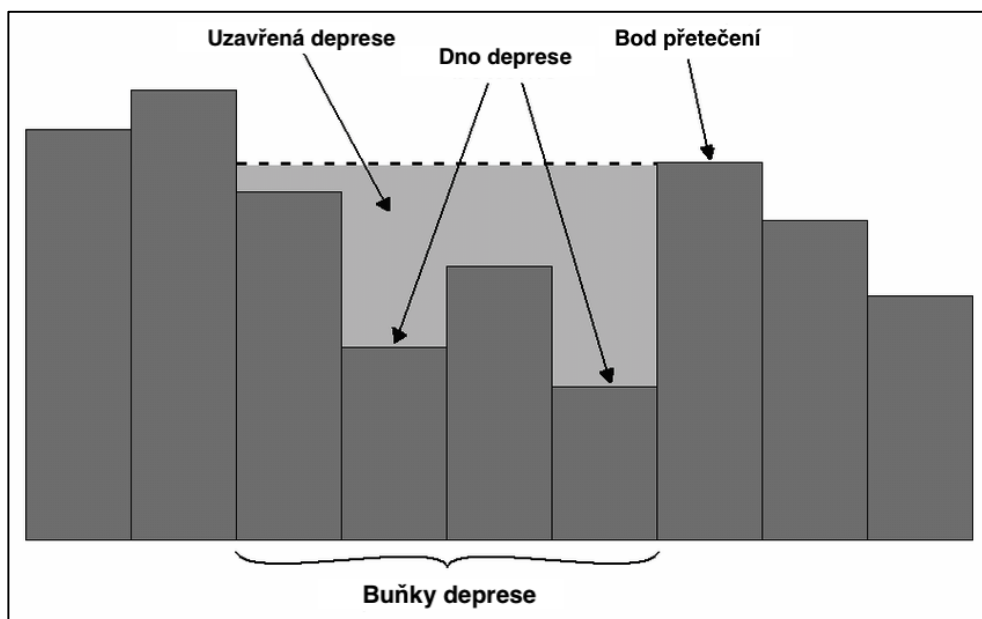
- 1) Žádný potenciální bod přetečení nebude umístěn, pokud bude okno zcela obsazeno v depresi. Všechny buňky by byly v depresi.
- 2) Pokud okno pouze částečně zahrnuje depresi, mohou mít buňky ležící po obvodě okna a označený jako deprese nižší elevaci než vybraný bod přetečení a dojde ke zvětšení skenovacího okna.

Tyto buňky po obvodě okna nejsou označeny jako potenciální body přetečení, protože nemají klesající cestu k sousednímu prvku a ukazují, že nižší bod přetečení může být umístěn mimo okno. Obvod okna se tedy skenuje, aby se zjistilo, zda tento stav existuje. Pokud tento stav existuje, nebo není umístěn žádný potenciální bod přetečení, okno se zvětší a povodí deprese se znovu definuje (Martz, Jong, 1988).

Poté, co je správně definováno povodí deprese a jeho bod přetečení, je určena oblast prohlubně. Uzavřená deprese je ta část povodí deprese, která leží pod výškou bodu přetečení a veškeré buňky v oblasti deprese se pak upravují na hodnotu elevace bodu přetečení (Martz, Jong, 1988).

Tato metoda bohužel není moc efektivní pro příliš velké a složité DEM, neboť je potřeba jedna úroveň opakování k prohledávání okna a druhá úroveň opakování k jeho případnému zvětšování. Může nastat situace tzv. složitých depresí, kdy je potřeba vyřešit dvě deprese ležící vedle sebe sdílející bod přetečení. Za pomoci této metody jsou ale tyto deprese vyřešeny jen někdy. Může dojít pouze k vyplnění deprese, která tzv. leží ve větší depresi. Tento problém se dá vyřešit opakováním celého procesu, dokud nebudou vyřešeny všechny deprese. Tím se ale zvyšuje náročnost této metody (Barták, 2008).

Další metodu, jak vyplňovat deprese v DEM, popisují Jenson a Domingue (1988). Autoři přišli na jednodušší a efektivnější způsob, který řeší i případy složitých depresí (viz obr. 4).



Obrázek 4: Příklad vyplnění složité deprese (Barták, 2009).

Metoda je dobrým příkladem filozofie vývoje nástrojů, které lze kombinovat různými způsoby, protože několik kroků jako je vymezení povodí, je nezávisle užitečné. Zatímco výsledkem postupu popsáním v tabulce č. 1 je DEM bez uzavřené deprese, postup zahrnuje i další postupy pro generování potřebných datových sad. (Jenson, Domingue, 1988).

Tabulka 1: Plnění depresí v DEM podle autorů Jenson a Domingue (1988) (Jenson, Domingue, 1988).

Krok	Procedura
1	Odstaňte jednobuněčné deprese tím, že zvýšíte hodnoty elevace každé buňky na výšku jejího nejnižšího souseda. Pokud je tento soused vyšší ve výšce než zpracovávaná buňka, jedná se o jednoduchý případ a jejím vyplněním snižujeme počet depresí, které je třeba řešit.
2	Vypočtete směry odtoku (SFD8 algoritmus)
3	Pro každou prostorově propojenou skupinu buněk, která má nedefinované směry toku (dno deprese), protože by vyžadovala tok do kopce, najděte jednoznačně všechny povodí deprese pomocí směru odtoku
4	Zjistěte všechny potenciaální body přetečení dělící povodí deprese (tabulka č. 2).
5	Pro každé povodí deprese vyberte z potenciaálních bodů přetečení ten, který má nejnižší elevaci a označte ho jako bod přetečení.

Krok	Procedura
6	Od každého bodu přetečení sleduj směr odtoku, dokud nenarazí na okraj datové sady (v tomto případě pokračuj krokem č. 7). Pokud nenarazí na okraj datové sady, je nalezena složitá deprese. Spoj je do jednoho povodí deprese, urči pro ně nový bod přetečení a pokračuj krokem č. 6.
7	V každém povodí deprese nalezni jejich nejvyšší bod přetečení a všem buňkám ležící v tomto povodí deprese, které mají menší elevaci než vybraný bod přetečení, změň elevaci na elevaci vybraného bodu přetečení.

Krok č. 4. v tabulce uvedené výše, kde hledáme všechny potenciální body přetečení je uskutečněný v následující tabulce (viz tab. 2):

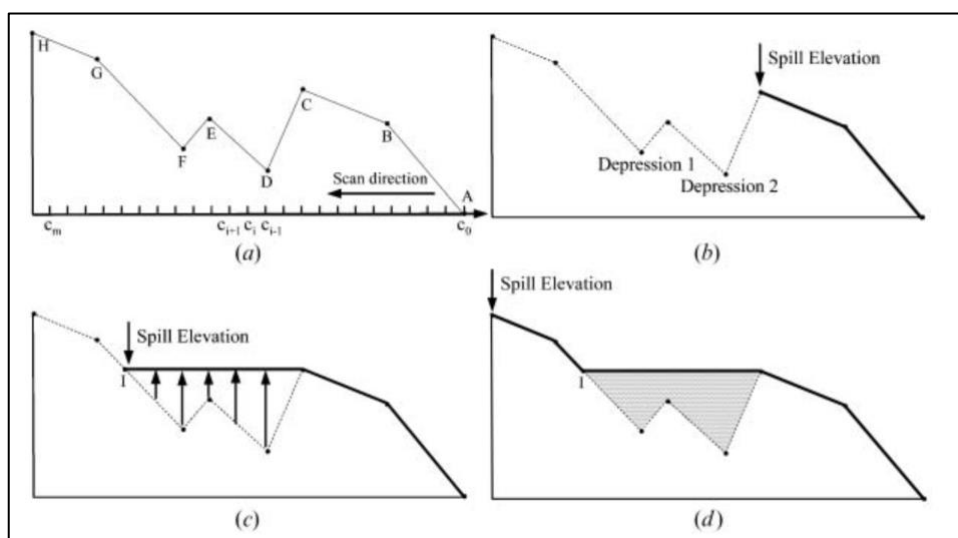
Tabulka 2: Postup hledání bodu přetečení od autorů Jenson a Domingue (1988) (Barták, 2008).

Krok	Procedura
1	Postupně u každé buňky ležící v povodí deprese zjisti, jestli jeden z jejich sousedů neleží v jiném povodí deprese. Pokud ano pokračuj krokem č. 2.
2	Zjisti, jestli má větší elevaci zkoumaná buňka nebo její soused. Ten s větší elevací označ jako potenciální bod přetečení.
3	Vytvoř novou položku tabulky zobrazující tyto dvě povodí deprese a zaznamenej do této tabulky nalezený potenciální bod přetečení. Jestli tabulka obsahuje tento pár povodí deprese, porovnej před tím zaznamenaný bod přetečení s nově zaznamenaným a v tabulce nech ten, který má nižší elevaci.
4	Opakuj pro všechny buňky ležící v povodí deprese

Metoda plnění depresí dostala větší efektivitu až při využívání lepší datové struktury, jako např. prioritní fronta (binární halda), která je více popsána v metodice této práce. Tuto metodu představili jako první autoři Soille a Gratin (1994), ale objevila se i později v článku od autorů Wang a Lui (2006).

Postup (viz obr. 5) je jednoduchý a její podrobnější implementace je popsána v metodice této práce. V prvním kroku je potřeba najít buňky tvořící okraj DEM a tyto buňky s hodnotami elevací vložit do datové struktury, která bude seřazena tak, aby

první prvek v datové sadě představoval buňku s nejnižší elevací. Tato buňka se následně nastaví jako zpracovaná a zkontrolují se její sousedé. Pokud má soused nižší elevaci než zpracovávající buňka, je dekováná deprese a přepíše se její nadmořská výška na hodnotu elevace zpracovávající buňky. Poté jsou všichni sousedé přidáni do datové struktury, pokud v ní už nejsou. Zpracovávající buňka se odstraní z datové struktury, která se následně znova seskupí a najde další buňku s nejnižší elevací. Proces se opakuje, dokud datová struktura není prázdná (Soille, Gratin, 1994).

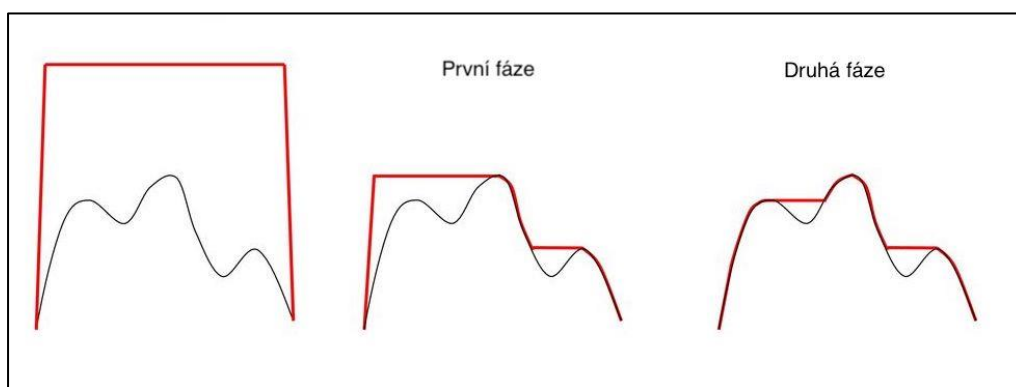


Obrázek 5: Postup vyplnění depresí metodou Wang a Lui (2006) (Wang, Lui, 2006).

Výhodou tohoto algoritmu je, že deprese jsou plněny všechny naráz, a ne každá zvlášť. Metodě tedy nevádí velký počet a složitost depresí. Neřeší ale problém vzniklých plošin, které jsou následně potřeba řešit (Barták, 2008).

Metody řešící uzavřené deprese jejím vyplnění jsou složité a mají dlouhý výpočetní čas, obzvlášť když DEM obsahuje vysoký počet depresí. Tento problém se snažili vyřešit autoři Planchon a Frederic (2001). Ty představili jednoduchý algoritmus na pochopení a implementaci. U této metody nedochází k postupnému plnění depresí, ale k tzv. zaplavení celého povrchu vodou (viz obr. 6). Následně je pak tato přebytečná voda odstraněna. Metoda je mnohem rychlejší než obvyklé algoritmy a pro jejich vytvoření stačí několik řádků zdrojového kódu (Planchon a Darboux, 2001).

Výhodou této metody je, že je všestranný a prohloubení může být nahrazeno plošinou nebo plochou mírně šikmou. V případě plošiny je následně možnost zjistit přesný objem uzavřené deprese a při použití mírně šikmé plochy jsou v jednom kroku vyřešeny uzavřené deprese i plošiny. To poskytuje možnost hned z modifikovaného DEM určit povrchový odtok (Planchon a Darboux, 2001).



Obrázek 6: Postup vyplnění depresí metodou Planchon a Darboux (2001) (Tarboton, 2018).

3.3.2. Metoda protržení hráze (outlet breaching)

Autoři Martz a Garbrecht (1999) představili nový algoritmus pro odstranění uzavřených depresí, který je založen na realističtějších předpokladech povahy těchto míst v rastrových DEM. Upozorňují na možnost, že některé uzavřené deprese mohli vzniknout nadhodnocením elevací, a ne jejich podhodnocením. V tomto případě předkládají jako vhodnější způsob odstranění deprese protržení hráze, a ne její vyplnění.

Metoda protržení hráze zahrnuje třístupňovou analýzu, která se aplikuje na každou depresi. V prvním kroku dochází k určení rozsahu deprese. V druhém kroku je vytvořen (tzv. protržen) odtok z deprese, pokud jsou splněna určitá kritéria. A za třetí je deprese vyplněna na hodnotu elevace vytvořeného odtoku metodou vyplnění (Martz, Garbrecht, 1999).

V prvním kroku se skenuje DEM, kde se lokalizuje pomocí algoritmu SFD8 všechny dna depresí. Pro každé dno deprese se následně definuje povodí deprese. To je zjištěno pomocí okna nastaveného zpočátku na 5 x 5 buněk a je soustředěné

na dno deprese, které je označeno. Okno pak skenuje ode dna deprese sousední buňky, a ty, které mají vyšší nebo stejnou elevaci jsou označeny a skenovány. Tento proces probíhá, dokud nejsou už žádné označené buňky (Martz, Garbrecht, 1999).

Po nalezení povodí deprese je potřeba zjistit všechny buňky představující potenciaální bod přetečení. Jestli existuje víc jak jeden potenciaální bod přetečení se stejnou nejnižší elevací je vybrán ten, který má nejprudší sklon z deprese. Pokud je v povodí deprese alespoň jedna buňka s nižší elevací než bod přetečení, jedná o uzavřenou depresi, a ne o plošinu a je možné pokračovat druhou fází (Martz, Garbrecht, 1999).

Jestli je detekována deprese uvnitř povodí deprese, je zahájen proces protržení hráze (viz tab. 3). Tento krok zabraňuje tvorbu plošin, a tedy i narušení simulace povrchového odtoku. Protržení je ale pouze povoleno na okraji uzavřené deprese. Takové simulované protržení topografické překážky nemá totiž podstatný vliv na změny povrchového odtoku, neboť protržení probíhá po stejné cestě jako voda, která by se vylila z uzavřené deprese. Buňky, které jsou sníženy při simulaci protržení, tvoří společně tzv. délku protržení. Tato délka protržení je ale omezena maximálně na dvě buňky. Pokud se jedná o větší počet buněk, tak můžeme předpokládat, že se jedná o skutečný topografický rys vzniklý podhodnocením elevací a neměl by být porušován (Martz, Garbrecht, 1999).

Tabulka 3: Postup protržení hráze uzavřené deprese (Martz, Garbrecht, 1999).

Krok	Procedura
1	Všechny buňky, které jsou uvnitř povodí deprese a ve stejné výšce jako pod přetečení, jsou zkoumány, jestli se nenacházejí v blízkosti buňky, která je mimo povodí deprese a má nižší elevaci, než je bod přetečení. Tyto buňky se pak zkoumají, jestli v jejich blízkosti se nenachází do vzdálenosti délky protržení buňka ležící v povodí deprese a s nižší elevací. Pokud ano, označ buňku jako potenciaální bod protržení. Pokud žádné takové buňky neexistují, není možné protržení hráze a deprese je jednoduše vyplněna na výšku bodu přetečení.
2	Pokud existuje víc jak jeden potenciaální bod protržení, je vybrán ten s nejstrmějším sklonem k buňce ležící mimo povodí deprese (primární kritérium) a nejprudším sklonem k buňce uvnitř povodí deprese (sekundární kritérium). Pokud jsou tato kritéria splněna na více místech je vybráno to, které se objevilo jako první.

Krok	Procedura
3	Výška bodu protržení se sníží na elevaci jejího souseda, který leží mimo povodí deprese a má z nich nejnižší elevaci. Na tuto elevaci sníží i jeho souseda s nejstrmějším sklonem a ležící v povodí deprese, který tvoří s bodem protržení délku protržení. Tím se změní výška bodu přetečení a účinně se poruší překážka, která je odpovědná za uzavřenou depresi.

Z toho ale nevyplývá, že všechny uzavřené deprese budou odstraněny, když dojde k protržení hráze. Deprese, u nichž došlo k tomuto procesu, mohou být prostorově omezeny nebo dokonce nezměněny. Proto je proces vyplnění prováděn bez ohledu na tom, zda došlo k protržení. Tato fáze zahrnuje už zmíněnou metodu plnění, kde jsou všechny buňky uvnitř povodí deprese, které mají nižší elevaci než bod přetečení, přepsány na hodnotu elevace bodu přetečení (Martz, Garbrecht, 1999).

Tento algoritmus byl rozsáhle testován a podával uspokojivé výsledky i při řešení složitých depresí (Martz, Garbrecht, 1999).

3.3.3. Metoda vyrývání (carving)

Tato metoda je podobná té předchozí, ale s tím rozdílem, že délka protržení zde není omezena maximálně na dvě buňky, ale může být libovolně dlouhá, jak potřebuje. I když se tato metoda může zdát, jako nevhodná pro řešení větších depresí, je rozhodně nejšetrnější, co se týče počtu modifikovaných buněk v DEM. Nedochozí totiž k žádnému plnění, ale pouze k tzv. vyrytí nejkratší odtokové cesty spojující dno deprese s bodem přetečení a s buňkou ležící mimo depresi, která má nižší elevaci než dno deprese. Této vyryté cestě se pak přiřadí rovnoměrný sklon, vypočítaný lineární interpolací z hodnot elevace dna deprese a elevace buňky ležící mimo depresi, aby byl vyřešen problém vznikajících plošin (Rieger, 1998).

Metoda je představena v článku Soille a kol. (2003), ve kterém autoři představili efektivní a jednoduchou implementaci (viz tab. 4), která je založená na prohledávání celého DEM pomocí prioritní fronty.

Tabulka 4: Implementace metody carving (Soille a kol., 2003).

Krok	Procedura
1	Všechny buňky tvořící okraj DEM se přidají do prioritní fronty
2	Z prioritní fronty se vybere první prvek (buňka s nejnižší elevací) a označí se jako zpracovaná
3	Do prioritní fronty se přidají všichni sousedi právě zpracovávané buňky, které nejsou ještě součástí fronty a zaznamená se odkaz na aktuálně zpracovávanou buňku
4	Pokud zrovna zpracovávaná buňka nemá žádného souseda s nižší elevací, jedná se o dno deprese a bude snížena hodnota elevace na hodnotu elevace dna deprese všem buňkám tvořící cestu ke dnu depresi, které mají vyšší elevaci než právě dno deprese
5	Opakuj tento postup od kroku č. 2, dokud není prioritní fronta prázdná.

3.3.4. Metody s minimální cenou

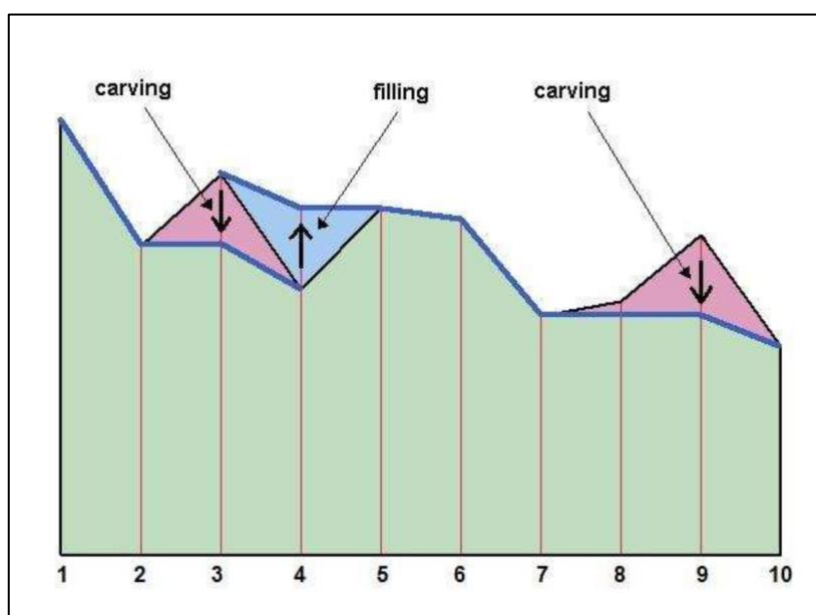
Metody pro odstraňování depresí by měly prosazovat kontinuální průtokové cesty způsobem, který vyžaduje nejmenší modifikaci buněk v DEM. Proto byla vyvinuta metoda IRA, která kombinuje metodu plnění a protržení hráze v závislosti na tom, která metoda má nejmenší vliv, pokud jde o počet modifikovaných buněk a střední absolutní odchylky původních a nových elevací v DEM (Lindsay, Creed, 2005).

Algoritmus IRA vytvoří dvě kopie DEM, jeden s vyplněnými depresemi a druhý DEM upravený metodou protržení hráze. Následně měří počet modifikovaných buněk (NMC) a střední absolutní odchylku (MAD) pro každou metodu zvlášť. Rozhodnutí, jestli provést vyplnění nebo protržení deprese je založeno na impaktním faktoru IF ,

$$IF = \frac{1}{2} \left(\frac{NMC_f}{NMC_b} + \frac{MAD_f}{MAD_b} \right) \quad (1)$$

kde index f odkazuje na metodu plnění a b na protržení hráze. Pokud faktor IF je větší nebo rovno jedné, bude provedena v původním DEM metoda protržení, pokud ne, dojde k vyplnění deprese (Lindsay, Creed, 2005).

V ideálním případě může být metoda odstranění deprese, která vede k minimálnímu dopadu, použita k odstranění každé individuální deprese v DEM. V praxi však vzájemná závislost mezi depresemi neumožňuje toto optimální řešení. Autoři zmiňují jako ukázkou řadu kaskádovitých depresí (viz obr. 7), které se často vyskytují v úzkých údolích v DEM. Pokud by došlo k odstranění každé deprese zvlášť, pak by došlo v tomto případě k protržení horní deprese a k vyplnění spodní. To by mělo za následek přerušení povrchového odtoku, neboť by vznikla vyplněním dolní deprese nová překážka, a tedy i deprese (Lindsay, Creed, 2005).



Obrázek 7: Příklad problému kaskádovité deprese (Barták, 2008).

Nejllepší metodu, co se týče minimalizace ceny při odstraňování uzavřených depresí představil Soille (2004). Metoda je opět kombinací plnění a protržení hráze a nese název optimální hybridní metoda (optimal hybrid approach). Rozdíl od předešlé je v tom, že v této metodě se řeší odstranění depresí individuálně, a dokonce lze danou depresi odstranit kombinací, a nejen jednou z metod. Např. může být daná deprese z jedné části řešena vytvořením odvodňovacího kanálu podle metody vyrývání a zbytek může být vyplněn. Opět je tu ale problém s depresemi kaskádovitého charakteru, které zde také nejsou vyřešeny a nastává situace, kdy vyplnění vede ke vzniku další překážky, která se řeší vyrýváním. Tento problém však

nemusí být tak závažný, když se vezme v úvahu malý výskyt těchto depresí, které jsou stejně vyřešeny, i když ne za nejlevnější cenu (Soille, 2004).

3.3.5. Fyzikálně založené metody

Nevýhodou některých předchozích metod, především metod založených na principu plnění (krom metody od autorů Planchon a Frederic), je vznik plošin. První fyzikálně založená metoda je popsána v článku Tianqi a kol. (2003). V ní je představena nová automatická metoda vyplňující uzavřené deprese, u kterých je problém vzniklých plošin vyřešen tím, že jsou zde počítány prostorově rozložené výškové přírůstky depresí. Následné překrytí simulovaného povrchového odtoku do digitalizovaných sítí skutečného odtoku ukázalo dobré vizuální konzistence. Výsledky z této metody navíc ukazují, že navrhovaný algoritmus je snadno použitelný pro DEM s širokým rozsahem reliéfu terénu a rozlišením.

Prostorově výškový přírůstek (2) se počítají pro všechny buňky deprese a při plnění na elevaci přelivné hrany se připočítají k daným buňkám, tak aby vznikl pozitivní sklon. Výškový přírůstek je nastaven tak, aby odrazil celkový sklon řešeného DEM. To je provedeno tak, že se ze začátku rozdělí DEM na čtyři části, kde každá část je přilehlá k jednomu rohu. Následně je spočítána průměrná hodnota elevací pro každý roh DEM. Ten roh s nejvyšší průměrnou elevací je považován za počátek souřadnic. Buňky uzavřené deprese, které leží blíže k počátku souřadnic, budou mít tedy vyšší výškový přírůstek než buňky, které jsou od něj dál (Barták, 2010).

Podle následující rovnice se spočítá výškový přírůstek:

$$dh(i, j) = h_c [1 - (\alpha i + \beta j) / (N_r + N_c)] \quad (2)$$

- dh ... výškový přírůstek
- i, j ... souřadnice buňky
- h_c ... volitelný maximální přírůstek ($h_c = 0,1$)
- α, β ... koeficienty vyjadřující svažitost terénu
- N_r ... počet řádků v DEM
- N_c ... počet sloupců v DEM

U koeficientů vyjadřující stupeň svažitosti terénu platí, že jejich součet je roven 2 a vzhledem k jejich malému vlivu se přiřazuje oběma koeficientům hodnota 1 (Tianqi a kol., 2003).

Druhou fyzikálně založenou metodou je PEM4PIT (Physical Erosion Model for PIT filling). Ta simuluje vyplnění depresí vzniklé erozí, jejíž intenzita odpovídá okolní topografii. Zvýšení buněk v depresích se mění tak, že nová topografie splňuje rovnici hmotnostní kontinuity (3) pro topografii v ustáleném stavu (Grimaldi a kol., 2007).

$$O = U - \beta A^\theta S + D\Delta z \tag{3}$$

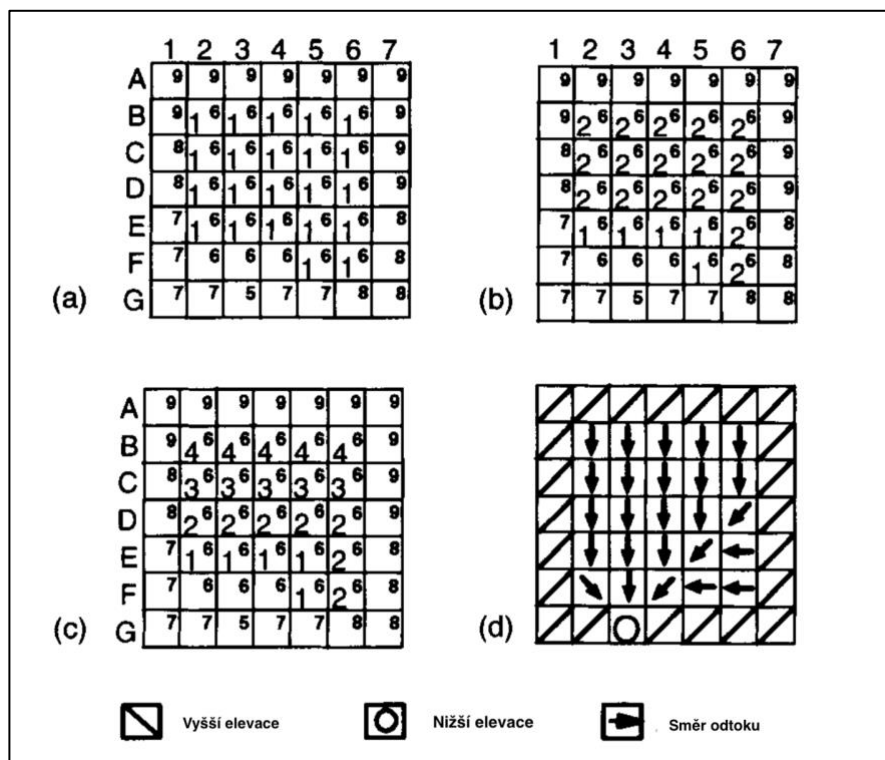
- z ... nadmořská výška [L]
- U ... rychlost tektonického zdvihu [L/T]
- $\beta A^\theta S$... člen vyjadřující fluvialní zařezávání terénu [L/T]
- $D\Delta z$... člen vyjadřující rychlost eroze [L/T]
- S ... nejstrmější kladný sklon [L/L]
- A ... přispívající plocha [L²]
- β ... erodovatelnost povrchu [L¹⁻²⁰/T]
- θ ... parametr vyjadřující vztah mezi přispívající plochou a sklonem [-]
- Δz ... křivost svahu [L⁻¹]
- D ... koeficient difuzivity svahu [L²/T]

Ačkoli tyto dvě metody generují realističtější terény, mají nešikovný charakter, neboť při zvyšování elevací mohou vzniknout nové (většinou malé a mělké) deprese. To vede k relativně vysokým cenám, obsahující změny elevace buněk, které původně nebyly v žádné depresi. Na druhé straně je výhodou těchto způsobů, že plošiny a deprese mohou být zpracovány současně stejným postupem (Barták, 2010).

3.4. Řešení plošin

3.4.1. Gradient ke spodnímu okraji plošiny

První algoritmus řešící problém plošin je od autorů Jenson a Domingue (1988). Ten je založen na nalezení všech buněk plošiny, které sousedí s buňkou, která má už směr odtoku přiřazen. Tím je nalezen tzv. spodní okraj plošiny, který je zde definován jako skupina buněk plošiny ležící na okraji a sousedící s buňkou s nižší elevací. Těmto buňkám je přiřazen směr odtoku, který se následně přiřazuje i zbylým buňkám plošiny směrem od spodního k hornímu okraji plošiny (viz obr. 8). Ten je naopak tvořen všemi buňkami, které leží na okraji plošiny a sousedí s buňkou s větší elevací. Nevýhodou této metody je ale absence konvergence a vznik paralelních linií (Barták, 2008).



Obrázek 8: Postup metody založené na gradientu ke spodnímu okraji (Garbrecht, Martz, 1997).

3.4.2. Metoda nejkratší cesty

Algoritmus představila Tribeová (1992), který najde dvě buňky představující pro danou plošinu její přítok a odtok. Tyto dvě buňky jsou pak spojeny přímkou, která zobrazuje směr odtoku směřující od přítoku do výtoku z plošiny. Aby byl vyřešen i problém paralelního toku a zajištěna konvergence, je následně přiřazen směr odtoku i ostatním buňkám plošiny, který je veden od přímky spojující přítok a odtok směrem ven z plošiny.

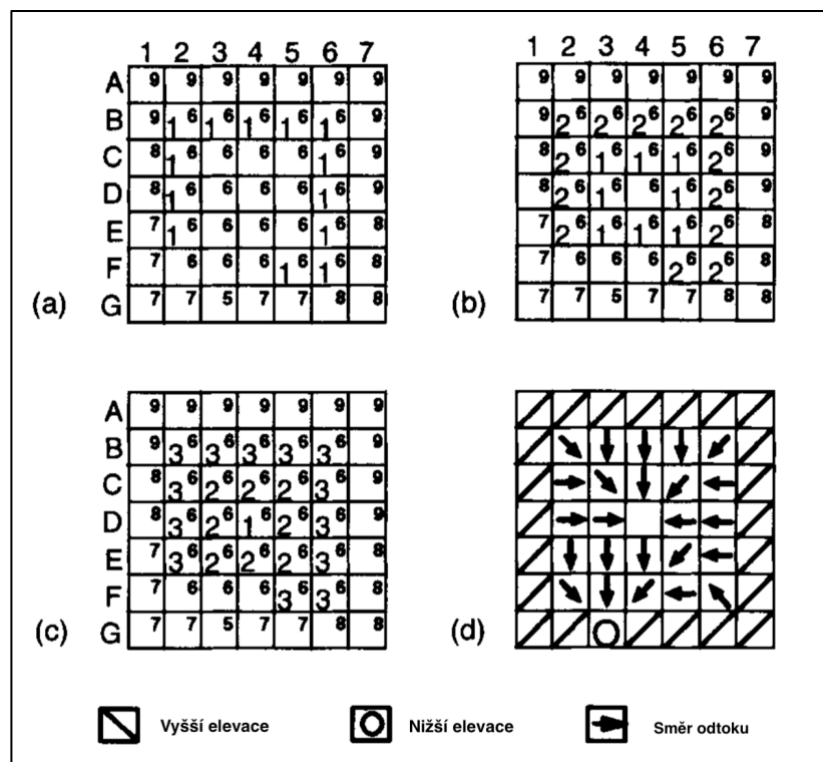
Nevýhodou této metody je změna topografie terénu, při řešení složitých plošin. Může dojít k situaci, kdy je přímka spojující přítok a odtok vedena i mimo plošinu. V tomto případě je buňkám, které leží na spojnici a mimo plošinu, snížena elevace na hodnotu plošiny (Tribe, 1992).

3.4.3. Gradient od horního okraje plošiny

Zde je prezentována metoda od autorů Garbrecht a Martz (1997), která řeší nevýhody předchozích metod a výsledkem je topograficky konzistentní povrch vykazující vlastnosti konvergence toku (viz obr. 9). Postup metody je znázorněn v tabulce č. 5.

Tabulka 5: Postup metody založené na gradientu od horního okraje (Garbrecht, Martz, 1997).

Krok	Procedura
1	Jsou identifikovány všechny buňky plošiny, které nesousedí s žádnou buňkou s nižší elevací. Tyto buňky jsou opakovaně navýšeny o tzv. elevační přírůstek, dokud nemají všechny buňky kladný sklon. Tímto krokem je zjištěna plocha reprezentující gradient ke spodnímu okraji.
2	Těm buňkám plošiny, které sousedí s některou buňkou, která má vyšší elevaci a zároveň nesousedí s žádnou, která má nižší elevaci, se zvýší elevace.
3	Krok č. 2 se opakuje, dokud už nejsou žádné buňky, které by nesousedili s buňkou s nižší elevací. Tím nám vznikne plocha reprezentující gradient k hornímu okraji.
4	U těchto dvou vzniklých ploch reprezentující oba gradienty dojde k jejich kombinaci, sečtením hodnot, o které se jednotlivé buňky navýšily v předchozích krocích.



Obrázek 9: Postup metody založené na gradientu od horního okraje (Garbrecht, Martz, 1997).

3.4.4. Metoda Wang a Liu

Metoda je založena na kontrole všech buněk v DEM pomocí prioritní fronty. Každé zpracovávané buňce jsou kontrolováni její sousedé a pokud algoritmus narazí na plošinu, přiřazování směru odtoku probíhá tak, že k právě zpracovávané buňce, která je už součástí plošiny, se přiřadí směr odtoku z těch jejich sousedů, kteří leží ve stejné plošině. Následně jsou stejně zpracovávány i její sousedé (Wang, Liu, 2006).

4. Metodika

Pro zpracování praktické části této diplomové práce bylo potřeba mít základní znalosti v geografickém informačním systému ArcGIS, v programovacím jazyce Python a následně si osvojit využití Pythonu při práci s ArcGISem.

4.1. Použitý software

4.1.1. ArcGIS 10.4.1

ArcGIS je geografický informační systém (GIS) od společnosti Esri určený pro práci s prostorovými daty. Může data vytvářet a spravovat, ale především je dokáže analyzovat, najít v nich nové vztahy a vše přehledně vizualizovat. Jeho součástí je i aplikační prostředí pro práci s Pythonem, který dokáže využívat vestavěné funkce ArcGISu. V programu lze vytvářet uživatelská rozhraní, ze kterých lze spouštět vytvořené pythonovské skripty. Ta mají stejnou podobu, jako ostatní okna nástrojů v tomto programu (Novák, 2015).

ArcGIS byl vybrán z důvodu jeho snadné integrace programovacího jazyka Python, ale i z důvodu využití jeho mnoha funkcí pracujících s rastrovými a vektorovými daty i přímo z programovacího prostředí Pythonu.

4.1.2. Programovací jazyk Python

Tento jazyk je nezávislý na operačním systému a volně stažitelný. Je výkonný, vyspělý a objektově orientovaný jazyk, a přesto velmi jednoduchý. Navíc pokud je nainstalovaný ArcGIS, je nainstalovaný i program IDLE (Python's Integrated DeveLopment Environment). Program IDLE je jeden z mnoha programovacích prostředí nebo interpretů pro Python, ve kterém je možné psát a spouštět libovolné části pythonovských kódů, které jsou okamžitě vykonány (Barták a kol., 2012).

Program IDLE nabízí vytváření nástrojů a funkcí v tzv. vlastním skriptu. Skripty jsou nyní chápány jako textový soubor s příponou .py, do kterého můžeme zadávat posloupnost složitých příkazů, které lze spustit všechny najednou.

4.1.3. ArcPy

ArcPy je knihovna jazyka Python, která zahrnuje všechny funkce využívající program ArcGIS. Funkce lze po načtení příslušného modulu použít přímo ve skriptu, kde byl modul importován.

4.1.4. NumPy

NumPy (Numerical Python) je základní knihovna pro vědecké výpočty s Pythonem, která umožňuje práci s numerickými daty, a to především s n-rozměrnými poli. Pole jsou podobná klasickým seznamům v Pythonu, ale na rozdíl od nich jsou tyto pole mnohem méně náročné na operační paměť. Navíc není tato knihovna napsaná v jazyce Python, ale v programovacím jazyce C, což umožňuje rychlejší výpočty s daty (SciPy, 2019).

4.1.5. Programovací jazyk R

Všechny statistický vyhodnocení a grafy byly dělaný v programu R (<https://www.R-project.org/>, 2016)

4.2. Načtení modulů

Modul je textový soubor s příponou .py obsahující posloupnost příkazů definujících různé funkce, které po načtení příslušného modulu lze použít ve skriptu a není třeba potřebnou funkci složitě vytvářet. Modul lze vytvořit založením nového skriptu, sepsáním potřebné funkce a uložením skriptu do složky, ve které program IDLE moduly hledá nebo jej lze jednoduše uložit do složky, kde je i uložený skript, ze kterého bude modul načítat (Barták a kol., 2012). V tabulce č. 6 jsou zobrazeny všechny moduly, které jsou používány ve zde vytvořených nástrojích.

Tabulka 6: Seznam modulů používaných v nástrojích Depression layer, Fill DEM a Carving.

Modul	Popis modulu
arcpy	Zpřístupní všechny nástroje pro geoprocessing nalezené v standartních sadách nástrojů nainstalovaných s ArcGIS. Je součástí základní výbavy programu IDLE.
numpy	Základním balíček pro vědecké výpočty s Pythonem díky, kterému lze pracovat s n-rozměrným polem. Je součástí základní výbavy programu IDLE.
sys	Poskytuje přístup k některým proměnným používaným interpretem (např. program IDLE), které komunikují s tlumočnickem (např. ArcGIS). Je součástí základní výbavy programu IDLE.
os	Slouží především pro správu adresářů a procesů, např. když je potřeba odstranit určitou složku nebo soubor. Je součástí základní výbavy programu IDLE.
rta	Obsahuje funkci určenou k načtení rastru, převedí ho na „numpy.array“ a získání všech informací potřebných k zpětnému převodu na rastr. Modul byl vytvořen Ing. Petrem Novákem v rámci jeho diplomové práce.
heap	Zpřístupní funkce určené k práci s binární haldou (přidání nového prvku do haldy, odebrání prvního prvku v haldě a jeho nahrazení). Modul byl vytvořen Ing. Petrem Novákem v rámci jeho diplomové práce. Funkce v modulu byli upraveny tak, aby vraceli binární haldu s prvkem obsahující nejnižší elevaci na prvním místě.
neighbors	Obsahuje funkci, která nalezne všechny sousedy zpracovávané buňky a zaznamená jejich nadmořské výšky se souřadnicemi do seznamu. Modul byl vytvořen autorem této práce.
border_of_raster	Obsahuje funkci určenou k nalezení všech buněk představující okraj rastru, ze kterých vytvoří binární haldu obsahující prvek s nejnižší elevací na prvním místě. Modul byl vytvořen autorem této práce a využívá funkci z modulu „heap“.

4.3. Načtení a převod DEM

Pokud chceme nějak manipulovat s hodnotami jednotlivých buněk rastru, např. s výškovými daty DEM, je potřeba nejprve daný rastr načíst a převést ho na tzv. „numpy.array“. Pro načtení rastru se používá funkce „arcpy.Raster“ s jedním vstupním parametrem (rastrem) a pro převedení je využívána funkce „arcpy.RasterToNumPyArray“. Obě funkce jsou dostupné ihned po načtení balíčku „arcpy“. Rastr často obsahuje buňky s tzv. „NoData“, které neobsahují žádné hodnoty. Pokud je ale potřeba těmto buňkám při převodu přiřadit určitou hodnotu, lze to pomocí parametru „nodata_to_value“.

Při načítání rastru je důležité zjistit a uložit základní informace o rastru (souřadnice levého dolního rohu, rozlišení buňky, počet sloupců a řádku). Tyto informace jsou později využívány jako vstupní parametr ve funkci „arcpy.NumPyArrayToRaster“ určenou zde k převodu modifikovaného n-rozměrného pole zpět na rastr.

4.4. Tvorba binární haldy

Binární halda (binary heap) je jednou z nejjednodušších datových struktur pro rychlé řazení hodnot. Dokáže z daného souboru získat nejnižší nebo nejvyšší hodnotu, kde nám nezáleží na pořadí ostatních hodnot. Binární haldu lze představit jako binární strom (Barták, 2008).

Obvykle podporuje více operací. Jednou z nich je přidání nového prvku, který nechá tzv. proskákat haldou, a pokud obsahuje nejmenší hodnotu ze všech hodnot haldy, umístí se prvek na začátek. Další operací je odebrání prvku umístěného na začátku haldy a nahrazení dalšího prvku s nejmenší hodnotou (Král a kol., 2013).

Přidání prvku do haldy se provádí tak, že prvek s danou hodnotou je přidán na konec některé větve (za poslední prvek v haldě) a následně postupným porovnáváním (proskákáním) se umístí přidaný prvek na příslušné místo v dané větvi (Barták, 2008).

Po odebrání prvku s nejnižší hodnotou v haldě je potřeba haldu přeskupit, aby se znova našel prvek s nejnižší hodnotou a zachovat tedy její klíčovou vlastnost. Toho docílíme tím, že na místo prvního prvku haldy je vložen prvek z konce jakékoli větve haldy. Nyní dojde k porovnávání podobnému jako to bylo v případě vložení nového prvku do haldy a prvek je přesunut na příslušné místo. Pokud je prvek vyšší než potomek, přesune se na jeho pozici (Barták, 2008).

Pro vytvoření funkce řešící operace binární haldy v programovacím jazyce Python postačí základní dovednosti v něm. Rychlejší a pohodlnější způsob je však možnost přímo v Pythonu načíst modul „heapq“, který ihned poskytne implementaci algoritmů binární haldy. V případě této práce byla využita binární halda vytvořená Ing. Petrem Novákem, který ji využil ve své diplomové práci (více viz. kapitola „Načtení modulů“).

4.5. Hledání a plnění depresí

Plnění uzavřených depresí v DEM si můžeme přestavit pomocí např. obdélníkového akvária, ve kterém je umístěn zmenšený model terénu. Dojde-li k postupnému plnění akvária vodou, bude se postupně zvedat hladina vody, která pomalu zakryje celý model terénu. Pokud se v modelu terénu při plnění vyskytne uzavřená deprese, nedojde k vyplnění prohlubně vodou, ale elevace všech buněk ležících v depresi budou navýšeny na elevaci přelivné hrany deprese.

V první řadě je potřeba vytvořit binární haldu z buněk umístěných na okraji rastru, která by na první pozici měla buňku s nejnižší hodnotou elevace. Okraj rastru je zde definován jako množina buněk, které obsahují informaci o nadmořské výšce a sousedí buď s aspoň jednou buňkou, která nemá informaci o elevaci (obsahuje tzv. NoData) nebo leží přímo u hrany rastru. Při tvorbě haldy jsou zaznamenávány i k přidávajícím prvkům jejich x a y souřadnice. Vznikne tedy tzv. seznam v seznamu, kde index seznamu bude obsahovat na prvním místě x souřadnici, na druhém y a na třetím nadmořskou výšku buňky.

Pokud by halda nebyla vytvořena nejprve z okraje rastru, ale sestavená ze všech hodnot rastru, mohlo by docházet v určitém případě k vyplnění buněk

ležících u okrajů rastru, i když by v tomto případě nešlo o uzavřenou depresi. Tento problém by nastal, pokud by z některého bodu byla detekována deprese, která by zasahovala až okraji rastru, a tedy by nešlo o depresi, ale pouze o klesající svah.

Okraj rastru neobsahující „NoData“ není těžké najít. Ten se najde pomocí for cyklu, který projede všechny buňky u hrany rastru a přidá všechny buňky do haldy. U rastru obsahující „NoData“ je složitější určit jeho okraj. Ten nebude mít tvar obdélníku, ale bude mít jakýkoli libovolný tvar. Okraj rastru bude tedy hledán způsobem, který projede všechny buňky rastru a přidá do haldy ty, které sousedí aspoň s jednou buňkou obsahující „NoData“. Následně se zkontrolují pouze buňky ležící u hrany rastru. Pokud tato buňka neobsahuje „NoData“ a není součástí haldy, je do ní ihned přidána.

Buňka umístěná na první pozici v haldě je nastavena jako zpracovaná a zkontrolují se všechny její sousední buňky. Před tím, než se zkontroluje, jestli má soused nižší elevaci nebo ne, se prověří, jestli soused není už zpracovaný nebo jestli neobsahuje „NoData“. Pokud ano, soused se nebude dál řešit. V opačném případě se u každého souseda zkontroluje, jestli má nižší nadmořskou výšku nebo ne. Soused s vyšší elevací se dál řešit nebude, neboť zde není detekovaná deprese a hodnota elevace souseda se pouze přidá se souřadnicemi do haldy, pokud v ní nejsou. V případě, kdy je detekovaná deprese, se soused nastaví jako zpracovaný a jeho hodnota se přepíše na hodnotu elevace její přelivné hrany, tedy buňky, ze které jsme hledali sousedy (první položka v haldě). Následně se soused přidá se změněnou elevací do haldy. Na konci tohoto procesu se odstraní první prvek v haldě a nahradí se dalším s nejnižší hodnotou elevace. Tento proces je opakován, dokud halda není prázdná.

Implementace v této práci je řešena tak, že lze některé deprese např. s určitou hloubkou nevyplnit a nechat je prohlubněmi. Do funkce je možné zadat jako nepovinný argument polygonovou vrstvu depresí. Vrstva je převedena na rastr funkcí „arcpy.PolygonToRaster_conversion“, který je následně skenován. Jestli je buňka označena jako deprese, vezme se hodnota elevace z vyplněného DEM a nastaví se na stejných souřadnicích v nově vytvořeném rastru. Pokud není označena jako

deprese, bere se hodnota z původního DEM. Před spuštěním funkce je možné z atributové tabulky depresí odstranit ty deprese, které nebudou předmětem plnění.

Pro tvorbu polygonové vrstvy byl vytvořen samostatný skript s funkcí určenou pouze ke hledání depresí. Ten je na stejném principu, jako jejich plnění. Změna nastává v okamžiku, kdy je detekována deprese. Zde je buňka souseda označena v nově vytvořeném rastru nul číslem reprezentujícím pořadí dané depresí. Rastr je následně převeden na polygon funkcí „arcpy.RasterToPolygon_conversion“. Postupně jsou zaznamenávány i informace (číslo depresí, maximální a minimální nadmořská výška depresí, hloubka, souřadnice dna depresí), které jsou ukládány ve vytvořeném seznamu. Z nich je vytvořena tabulka, která se funkcí „arcpy.JoinField_management“ propojí s atributovou tabulkou polygonové vrstvy.

Předmětem hledání jsou i plošiny, které jsou následně zobrazeny pomocí vlastní polygonové vrstvy. Každý polygon představující plošinu pak nese informaci o čísle a nadmořské výšce plošiny.

4.6. Vyrývání (Carving)

Implementace této metody je z počátku podobná té předešlé. Je vytvořena binární halda z okraje rastru a postupně jsou kontrolovány sousedi buňky, která je se svojí hodnotou elevace v danou chvíli na začátku binární haldy. Rozdílem je, že se neporovnává elevace souseda se středovou buňkou, ale jen sousedovi přiřadíme informaci odkud tzv. přišel. Jinými slovy sousedovi přiřadíme souřadnice zrovna zpracovávané buňky, pokud již nebyly sousedovy dříve přidány a je označen jako zpracovaný. Tento proces je opět opakován, dokud halda není prázdná. Tím je vytvořen tzv. rastr „from“.

Následně jsou zjištěny souřadnice všech buněk zobrazující v DEM dna depresí. Dna depresí jsou zjištěna jedním skenováním rastru, kde jsou všem buňkám kontrolováni jejich sousedé. Jestli jsou všichni sousedé vyšší než zpracovávaná buňka a zároveň tato buňka neleží na okraji rastru, jedná se o buňku reprezentující dno depresí a jsou její souřadnice přidány do předpřipraveného seznamu. Pokud je alespoň jeden soused vyšší a žádný není nižší než zpracovávaná buňka, která opět

neleží na okraji rastru, jedná se o možné dno deprese, které by mohlo být tvořeno buňkami se stejnou elevací a její souřadnice jsou též přidány do seznamu.

Z tohoto seznamu jsou postupně brány souřadnice, od kterých bude docházet k vyrývání cesty směřující ven z uzavřené deprese. Ta vznikne tak, že od buňky reprezentující dno deprese jsou následně po směru, který byl předtím určen pomocí rastru „from“, sníženy hodnoty elevace na hodnotu elevace dna deprese všem buňkám, dokud se nenarazí na buňku, která má nižší elevaci než dno deprese nebo na buňku ležící na okraji rastru.

5. Výsledky práce

Výsledkem této diplomové práce je sada vytvořených nástrojů:

- Depression layer
- Fill DEM
- Carving

Nástroje řeší problematiku uzavřených depresí v DEM a jsou implementovány do prostředí ArcGIS. Vyzkoušení nástrojů bylo provedeno na dvaceti náhodně vybraných hydrologických povodích III. a IV. řádu. Výsledky pro jednotlivá povodí následně sloužily ke statistickému vyhodnocení, sloužící k určení, která z těchto dvou metod je šetrnější, z hlediska počtu a objemu modifikovaných buněk.

Základem nástrojů jsou tři skripty:

- depression_layer
- fill_DEM
- carving

Ty obsahují zdrojový kód nástrojů vytvořený autorem této práce, které jsou propojeny s dalšími vytvořenými skripty (moduly):

- border_of_raster
- neighbors

Svémi funkcemi doplňují jednotlivé nástroje a též patří do výsledků práce. Nástroje využívají další moduly, které však nebyly vytvořeny autorem této práce a jejich funkce i s autorem jsou zmíněny v metodice.

Pro ukázkou nástrojů a statistické vyhodnocení byla použita data SRTM DEM (2007) volně dostupná na adrese: <http://www.gisat.cz/content/cz/produkty/data-ke-stazeni>. DEM představuje jedinečný produkt z oblasti digitálních modelů terénu, který byl vytvořen v rámci amerického projektu vedeného organizacemi NASA a NGA s přispěním německé a italské kosmické agentury. Jedná se o interferometrická radarová data prakticky pokrývající celé území světa.

5.1. Popis nástrojů

Všechny nástroje jsou umístěny v jedné nástrojové sadě „HyCoTer“ (zkratka Hydrological Correct Terrain). Popis jednotlivých nástrojů je znázorněn v tabulce č. 7.

Tabulka 7: Popis vytvořených nástrojů.

Nástroj	Popis
Depression layer	<p>V DEM najde všechny uzavřené deprese a plošiny, u kterých zjistí jejich základní údaje (maximální a minimální nadmořská výška deprese, hloubka deprese, souřadnice dna deprese).</p> <p>Povinným vstupním parametrem je rastr DEM. Výstupem je polygonová vrstva depresí a plošin. Atributová tabulka této vrstvy obsahuje u jednotlivých depresí a plošin jejich získané informace.</p> <p>Zdrojový kód nástroje je umístěn ve skriptu „depression_layer“.</p>
Fill DEM	<p>Vyplní všechny uzavřené deprese v DEM na hodnotu elevace jejich bodu přetečení.</p> <p>Povinným vstupním parametrem je rastr DEM a výstupem je rastr DEM s vyplněnými depresemi.</p> <p>Je možné jako vstupní parametr zadat polygonovou vrstvu depresí vytvořenou nástrojem Depression layer. Před použitím nástroje lze ve vrstvě depresí ručně vybrat a odstranit deprese, které nemají být předmětem plnění.</p> <p>Zdrojový kód nástroje je umístěn ve skriptu „fill_DEM“.</p>
Carving	<p>Vytvoří od nejnižšího bodu deprese (dna deprese) cestu ven z prohlubně tzv. vyrýváním. Dojde tedy k probourání hráze deprese. Cesta je vytvořena postupným snižováním elevace buněk rastru na hodnotu dna deprese, dokud nejsou po cestě už žádné buňky s větší elevací než dno deprese.</p> <p>Povinným vstupním parametrem je rastr DEM a výstupem je rastr DEM s odstraněnými uzavřenými depresemi.</p> <p>Zdrojový kód nástroje je umístěn ve skriptu „carving“.</p>

5.2. Popis skriptů

Zkrácený popis vytvořených skriptů je znázorněn v tabulce č. 8. Celý zdrojový kód vytvořených skriptů je k dispozici v příloze práce.

Tabulka 8: Popis vytvořených skriptů.

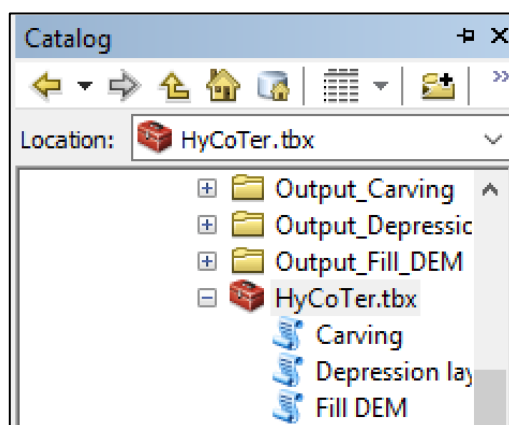
Skript	Popis
depression_layer	Skript obsahuje funkci „Dep_layer“, která nejdříve načte pomocí modulu „rta“ rastr DEM a převede ho na „numpy.array“. Po převedení nalezne modulem „border_of_raster“ všechny buňky rastru tvořící jeho okraj, které prostřednictvím modulu „heap“ přidá do binární haldy. Následně zkontroluje všechny sousedy buňky, která představuje v danou chvíli první prvek v haldě, jestli nepředstavují v DEM uzavřenou depresi nebo plošinu. Pokud je detekována deprese nebo plošina označí souseda v nově vytvořeném n-rozměrném poli nul jedničkou. V případě deprese zvýší i hodnotu elevace souseda na hodnotu elevace středové (zpracovávané) buňky. V tomto kroku získává i základní informace o depresi nebo plošině. Funkce závěrem všechny buňky, které byly označeny jako deprese nebo plošiny, převede z pole na rastr, následně z rastru na polygon a získané údaje doplní k daným polygonům.
fill_DEM	Skript obsahuje funkci „fill_DEM“, která stejným způsobem jako předchozí funkce detekuje a vyplní všechny uzavřené deprese, ale místo polygonové vrstvy depresí a plošin je jejím výstupem DEM s vyplněnými depresemi.
carving	Skript obsahuje funkci „Carving“, která nejprve stejným způsobem načte rastr DEM a vytvoří binární haldu z okraje rastru. Přiřadí všem nezpracovaným sousedům zpracovávané buňky, která představuje v danou chvíli první prvek v haldě, souřadnice zpracovávané buňky. Ve vstupním rastru DEM najde všechny buňky představující dna depresí a následně od těchto buněk po směru, který byl vytvořen v předchozím kroku, vytvoří cestu ven z deprese postupným snižováním elevací buněk na hodnotu elevace dna deprese, dokud po cestě nenarazí na buňku s nižší elevací, než je dno deprese nebo na okraj rastru. Závěrem převede upravený DEM na rastr, který neobsahuje žádnou uzavřenou depresi.

Skript	Popis
border_of_raster	Skript obsahuje funkci „border“, která každou buňku rastru sousedící s alespoň jednou buňkou obsahující „NoData“ označí jako okraj rastru a pomocí modulu „heap“ ji přidá do binární haldy. Následně projede pouze okraj datové sady a přidá do binární haldy ty buňky, které neobsahují „NoData“ a nejsou označené jako okraj rastru.
neighbors	Modul obsahuje funkci „neighbors_z“, která najde elevaci všech sousedních buněk zpracovávané buňky, zjistí jejich souřadnice a přidá tyto informace do seznamu.

5.3. Ukázka použití

5.3.1. Spuštění nástroje

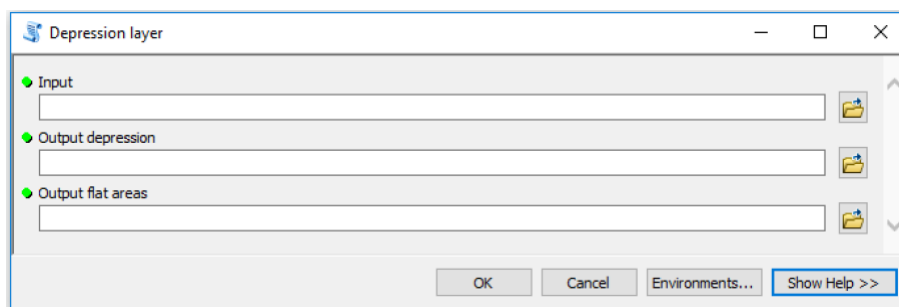
Po spuštění programu ArcMap je možné vytvořenou nástrojovou sadu najít v katalogu programu (viz obr. 10).



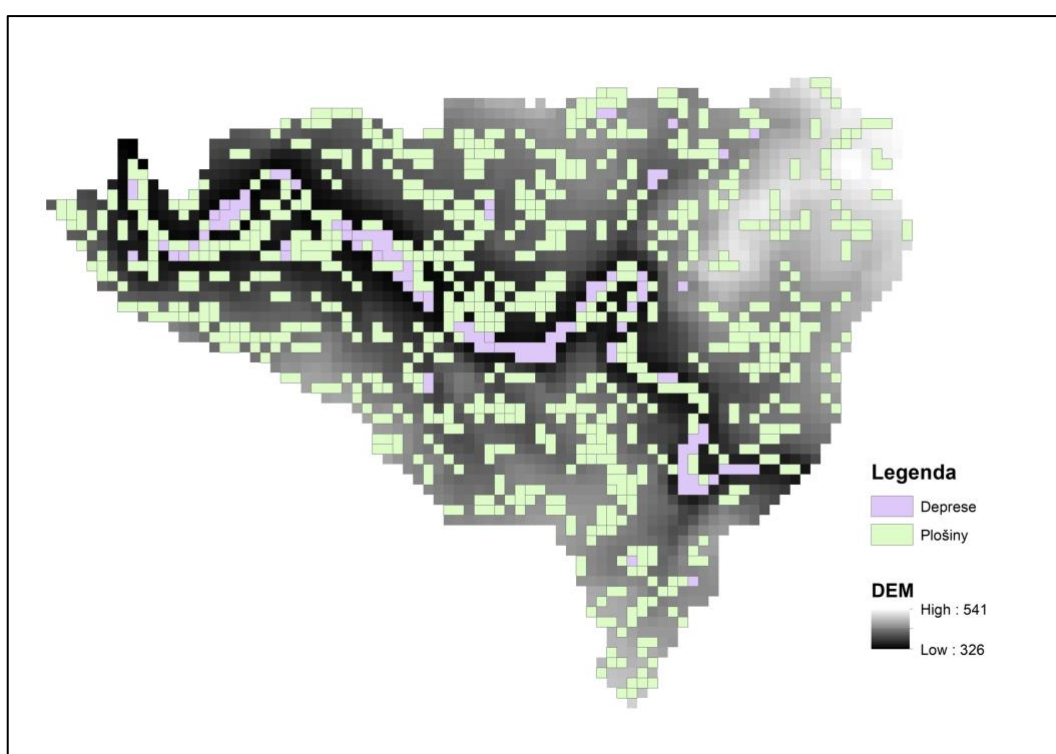
Obrázek 10: Nástrojová sada v katalogu ArcMap.

5.3.2. Nalezení uzavřených depresí a plošin

Pro nalezení uzavřených depresí a plošin slouží nástroj „Depression layer“ (viz obr. 11). Jako vstupní parametr je zde zadáván rastr DEM. Výsledkem nástroje je polygonová vrstva depresí a plošin (viz obr. 12).



Obrázek 11: Okno nástroje Depression layer.

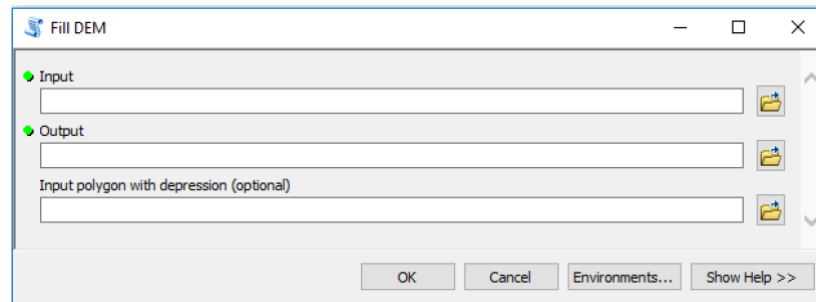


Obrázek 12: Vytvořený polygonový vrstvy depresí a plošin.

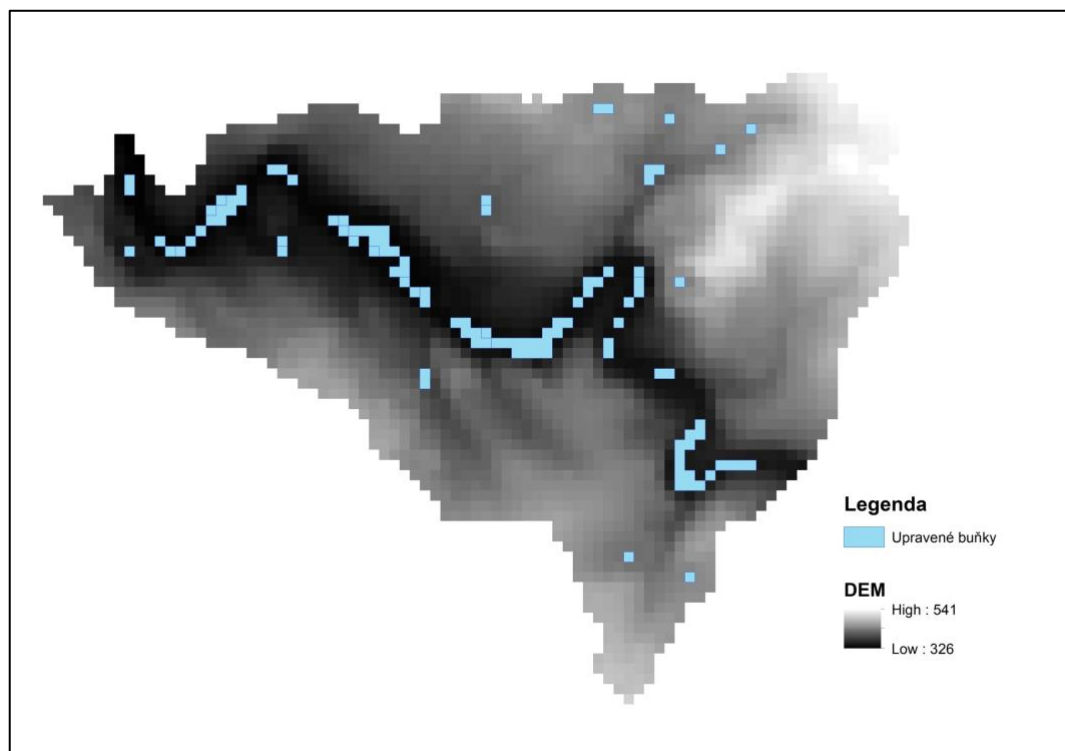
Pro ukázkou bylo DEM oříznuto polygonovou vrstvou reprezentující hydrologické povodí IV. řádu. Číslo hydrologického pořadí vybraného povodí je 1-09-01-1330.

5.3.3. Vyplnění depresí

K vyplnění všech depresí je potřeba spustit nástroj „Fill DEM“ (viz obr. 13). Povinným vstupním parametrem je rastr DEM a nepovinným polygonová vrstva depresí. Výstupem je rastr DEM s vyplněnými depresemi (viz obr. 14).



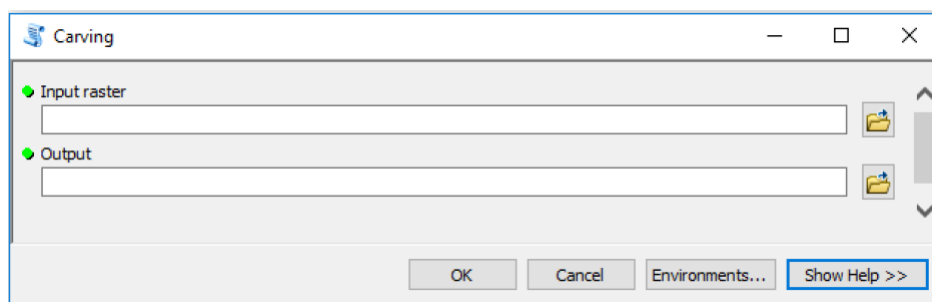
Obrázek 13: Okno nástroje Fill DEM.



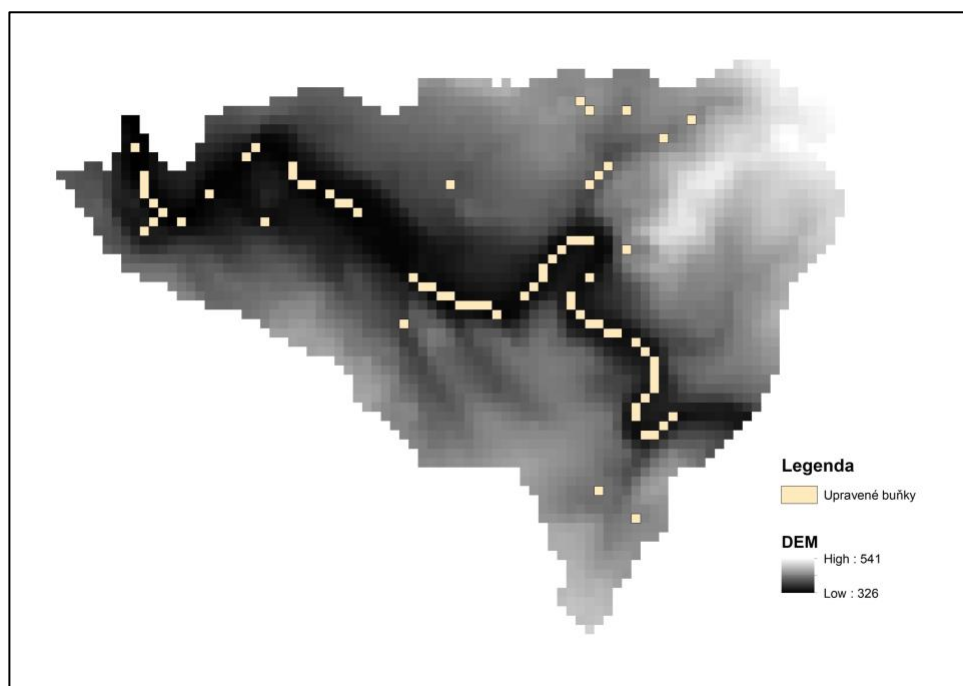
Obrázek 14: Buňky modifikované nástrojem Fill DEM.

5.3.4. Vyrytí cesty z deprese

K vyrytí cesty z uzavřené deprese byl vytvořen nástroj „Carving“ (viz obr. 15). Vstupním parametrem je rastr DEM a výstupem je upravený rastr DEM neobsahující uzavřené deprese (viz obr. 16).



Obrázek 15: Okno nástroje Carving.



Obrázek 16: Buňky modifikované nástrojem Carving.

5.4. Porovnání metod

Pro porovnání byly vytvořeny dvě tabulky, kde je pro každé hydrologické povodí znázorněn absolutní i relativní počet modifikovaných buněk. Pro nástroj „Fill DEM“ byl vypočítán objem „terénu“ potřebný k vyplnění všech uzavřených depresí a pro „Carving“ objem sníženého terénu nezbytný k vytvoření cesty ven z deprese. Objemy byly vypočítány, jako násobek celkové výšky zvednutého nebo sníženého „terénu“ v povodí a plochy buňky (100 x 100 m).

Tabulka 9: Porovnání nástrojů Fill DEM a Carving na dvaceti hydrologických povodí III. řádu.

ID	Počet buněk v DEM	Počet depresí v DEM	Počet buněk v depresi	Počet změných buněk		Počet změných buněk [%]		Objem zvýšeného terénu [m3]		Objem sníženého terénu [m3]	
				Fill	Carving	Fill	Carving	Fill	Carving		
1	71166	726	2546	2546	1182	3,58	1,66	9,20E+07	3,53E+07		
2	73679	1166	2735	2735	1614	3,71	2,19	5,65E+07	3,64E+07		
3	64547	1166	3463	3463	1443	5,37	2,24	1,27E+08	3,04E+07		
4	78204	549	2093	2093	1050	2,68	1,34	1,33E+08	4,63E+07		
5	116637	1734	9199	9199	3176	7,89	2,72	4,37E+08	1,20E+08		
6	75019	1377	7730	7730	2359	10,30	3,14	3,27E+08	1,02E+08		
7	100069	2417	11204	11204	3858	11,20	3,86	4,32E+08	1,40E+08		
8	150805	2229	9103	9103	3827	6,04	2,54	4,34E+08	1,32E+08		
9	118835	1507	6976	6976	2772	5,87	2,33	3,77E+08	1,33E+08		
10	165343	1768	6732	6732	3336	4,07	2,02	2,83E+08	1,19E+08		
11	126830	1326	5774	5774	2036	4,55	1,61	2,05E+08	5,48E+07		
12	75562	700	2459	2459	1124	3,25	1,49	7,16E+07	3,20E+07		
13	151958	1349	5110	5110	2601	3,36	1,71	2,65E+08	1,08E+08		
14	114749	1143	4126	4126	2038	3,60	1,78	1,83E+08	8,22E+07		
15	126198	1114	3608	3608	1429	2,86	1,13	3,89E+08	4,81E+07		
16	132216	1347	4732	4732	2054	3,58	1,55	2,34E+08	6,98E+07		
17	80968	1473	3013	3013	1888	3,72	2,33	5,35E+07	3,45E+07		
18	114950	1294	5037	5037	2164	4,38	1,88	2,59E+08	8,21E+07		
19	86926	1312	4779	4779	2232	5,50	2,57	1,94E+08	7,49E+07		
20	58432	667	2544	2544	1152	4,35	1,97	9,49E+07	4,08E+07		
Suma	2083093	26364	102963	102963	43335	-	-	4,65E+09	1,52E+09		
Průměr	104154,65	1318,20	5148,15	5148,15	2166,75	4,99	2,10	2,32E+08	7,61E+07		
Směrodatná odchylka	31193,27	466,10	2536,84	2536,84	849,42	2,28	0,63	1,29E+08	3,77E+07		

Tabulka 10: Porovnání metod Fill DEM a Carving na dvaceti hydrologických povodí IV. řádu.

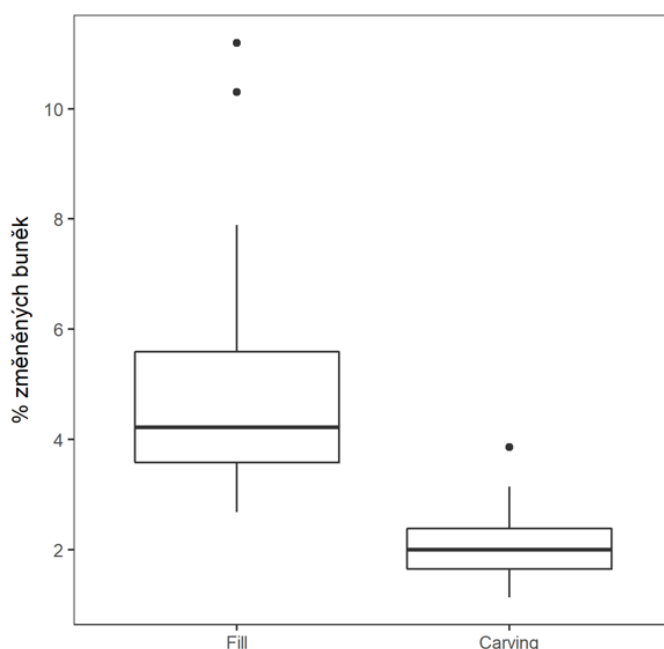
ID	Počet buněk v DEM	Počet depresí v DEM	Počet buněk v depresi	Počet změněných buněk		Počet změněných buněk [%]		Objem zvedlého terénu [m3]	Objem sníženého terénu [m3]
				Fill	Carving	Fill	Carving	Fill	Carving
1	2615	24	38	38	30	1,45	1,15	1,06E+06	8,70E+05
2	3770	61	176	176	84	4,67	2,23	2,59E+06	1,62E+06
3	2662	26	56	56	28	2,10	1,05	7,10E+05	3,70E+05
4	3673	44	74	74	59	2,01	1,61	8,40E+05	7,80E+05
5	3927	51	81	81	55	2,06	1,40	1,42E+06	9,50E+05
6	1837	18	35	35	19	1,91	1,03	4,00E+05	2,40E+05
7	3564	26	123	123	87	3,45	2,44	5,55E+06	3,17E+06
8	3657	112	359	359	186	9,82	5,09	1,33E+07	6,72E+06
9	2133	3	8	8	8	0,38	0,38	2,20E+05	3,00E+05
10	4863	37	97	97	58	1,99	1,19	2,92E+06	1,48E+06
11	2936	9	15	15	13	0,51	0,44	3,10E+05	2,90E+05
12	3129	62	146	146	90	4,67	2,88	2,85E+06	1,68E+06
13	2530	6	9	9	6	0,36	0,24	1,00E+05	7,00E+04
14	2269	15	24	24	17	1,06	0,75	3,50E+05	2,50E+05
15	2750	49	95	95	61	3,45	2,22	1,98E+06	1,54E+06
16	2082	39	115	115	47	5,52	2,26	2,81E+06	1,17E+06
17	2233	22	86	86	52	3,85	2,33	4,96E+06	2,87E+06
18	2289	26	124	124	44	5,42	1,92	4,06E+06	1,21E+06
19	2769	14	37	37	17	1,34	0,61	1,12E+06	5,80E+05
20	2485	40	133	133	63	5,35	2,54	1,96E+06	1,02E+06
Suma	58173	684	1831	1831	1024	-	-	4,95E+07	2,72E+07
Průměr	2908,65	34,20	91,55	91,55	51,20	3,07	1,69	2,47E+06	1,36E+06
Směrodatná odchylka	755,62	24,60	77,77	77,77	40,12	2,29	1,11	2,92E+06	1,47E+06

Z celkového počtu buněk v DEM pro vybraná povodí III. řádu tvoří průměrně 5 % buňky představující uzavřenou depresi. Pro vybraná povodí IV. řádu jsou to 3,07 %. Nástroj „Fill DEM“ modifikoval průměrně stejné procento buněk, neboť u každého povodí je počet upravených buněk shodný s počtem buněk představující depresi. Nástroj „Carving“ upravil u vybraných povodí III. řádu průměrně 2,10 % buněk z celkového počtu buněk. Toto procento u povodí IV. řádu činilo 1,69 %.

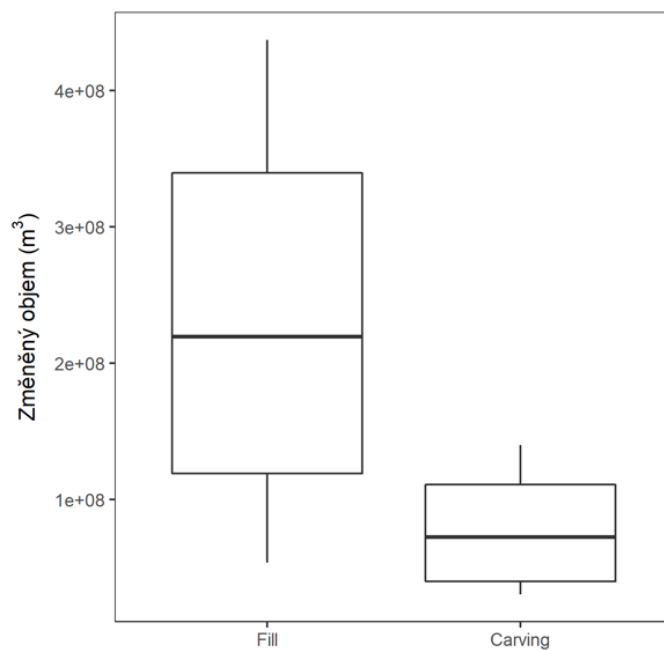
Ve všech případech krom jednoho nástroj „Carving“ modifikoval menší počet buněk než nástroj „Fill DEM“ a bylo i odstraněno menší množství objemu potřebného k vytvoření cesty ven z deprese než množství objemu použitého k vyplnění deprese.

V tomto jednom případě oba nástroje modifikovali stejný počet buněk a odstraněného objemu bylo o 26 % více než vyplněného. Celkově však vypočítaného objemu u nástroje „Fill DEM“ u vybraných povodí III. řadu bylo o 67,25 % více než u nástroje „Carving“. U menších povodích byl tento objem větší o 45 %.

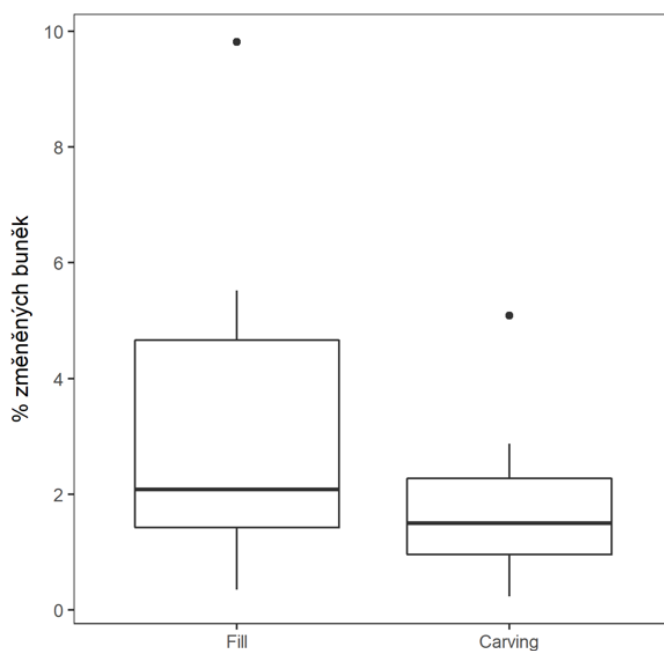
Rozdíl v ovlivnění DEM z hlediska relativního počtu změněných buněk a změněného objemu byl rovněž posouzen graficky pomocí krabicových grafů (viz obr. 17 až 20) a statisticky pomocí párového Wilcoxonova testu (viz tab. 11). Ve všech případech bylo potvrzeno významně vyšší ovlivnění DEM metodou Fill v porovnání s metodou Carving, ačkoli u povodí III řádu byl tento efekt výraznější.



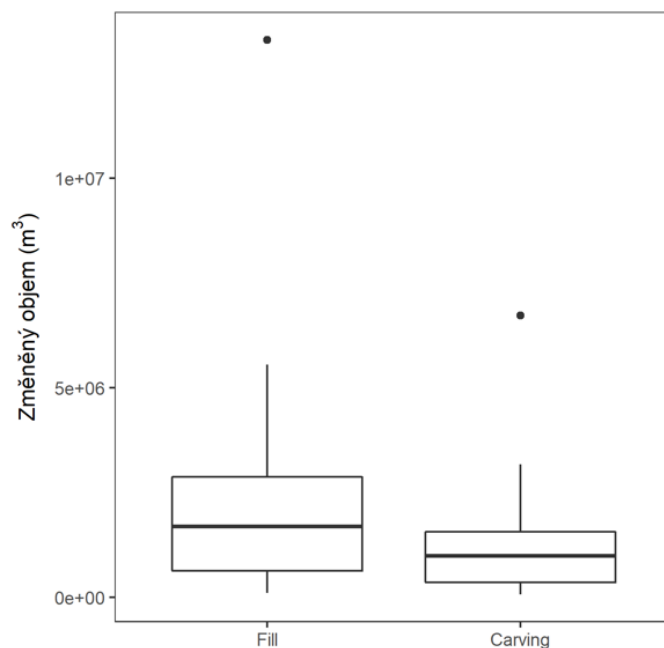
Obrázek 17: Rozdíl mezi procentem ovlivněných buněk metodami Fill a Carving na vzorku dvaceti náhodně vybraných povodí ČR III. řádu.



Obrázek 18: Rozdíl mezi změněným objemem DEM metodami Fill a Carving na vzorku dvaceti náhodně vybraných povodí ČR III. řádu.



Obrázek 19: Rozdíl mezi procentem ovlivněných buněk metodami Fill a Carving na vzorku dvaceti náhodně vybraných povodí ČR IV. řádu.



Obrázek 20: Rozdíl mezi změněným objemem DEM metodami Fill a Carving na vzorku dvaceti náhodně vybraných povodí ČR IV. řádu.

Tabulka 11: Porovnání ovlivnění DEM metodami Fill a Carving na vzorku náhodně vybraných povodí III. a IV. řádu párovým Wilcoxonovým testem.

Řád povodí	Hypotéza	Statistika	Počet měření	p-hodnota
III	Medián rozdílu počtu ovlivněných buněk mezi metodami Fill a Carving je roven nule	210	20	1.91E-06
III	Medián rozdílu změněného objemu DEM mezi metodami Fill a Carving je roven nule	210	20	1.91E-06
IV	Medián rozdílu počtu ovlivněných buněk mezi metodami Fill a Carving je roven nule	190	20	0.00014
IV	Medián rozdílu změněného objemu DEM mezi metodami Fill a Carving je roven nule	206	20	1.34E-05

Z výsledků vyplývá, že zde implementovaná metoda založená na vyrývání cesty ven z deprese je z hlediska počtu modifikovaných buněk a upraveného objemu šetrnější než metoda vyplňující deprese. Vyplnění depresí bylo z tohoto hlediska šetrnější jen v případě, když se v DEM představující povodí vyskytovaly pouze malé a mělké deprese.

6. Diskuze

Výstupy z vytvořených nástrojů ukázaly, že dokáží přesně nalézt všechny uzavřené deprese v DEM a následně je odstranit pro další hydrologické zpracování. Výhodou oproti dosavadním nástrojům je možnost před vyplněním všech prohlubní zobrazit všechny buňky umístěné v depresi v podobě polygonové vrstvy a informace o každé depresi uložené v atributové tabulce polygonové vrstvy.

Nevýhodou je však rychlost procesu. Ta je omezena možnostem programovacího jazyka Python, který zde byl vybrán díky snadnému pochopení a jeho jednoduché implementaci v ArcGISu. Alternativně mohl být použit i jiný podporovaný programovací jazyk např. Visual C++, který je složitější, ale výsledný nástroj by mohl být rychlejší.

V průběhu práce se vyskytla řada drobných problémů vzniklých převážně chybou v pythonovských skriptech. Ve většině případů se jednalo o chyby způsobené v začátcích mého programování. Tyto chyby však byly rychle vyřešeny po konzultaci s vedoucím této diplomové práce nebo po doplnění informací z poskytnuté literatury.

Asi největší problém byl u nástroje „Depression layer“ přidělit v průběhu procesu získané informace o depresích příslušným polygonům představující uzavřenou depresi. Jedno projetí všech řádků atributové tabulky a propojení polygonu s patřičnými údaji hlásilo chybu v SQL dotazu, která nedokázala být vyřešena. Problém byl vyřešen až vytvořením nové tabulky, do které byly nahrány všechny informace a následně byla tabulka propojena s polygonovou vrstvou depresí vhodným klíčem.

7. Závěr

Všechny nástroje byly vytvořeny v programovacím jazyce Python a implementovány do prostředí ArcGIS. Použité algoritmy v nástrojích umožňují nalézt všechny uzavřené deprese v DEM a následně je odstranit pro další možné hydrologické zpracování. Pro odstranění depresí byl vybrán algoritmus pracující na principu plnění deprese a na tzv. vyrytí cesty ode dna deprese směrem ven z prohlubně.

Přínosem nástroje určeného k odstranění deprese jejím vyplněním je možnost před samotným zahájením procesu zobrazit v podobě polygonové vrstvy všechny buňky umístěné v prohlubni a v atributové tabulce informace o každé uzavřené depresi. Na základě těchto informací se následně může rozhodnout, které prohlubně budou v procesu odstraněny.

Ze statistického vyhodnocení bylo zjištěno, že odstranění uzavřených depresí založené na principu vyrývání je z hlediska počtu modifikovaných buněk a objemu ve většině případů šetrnější než odstranění založené na principu plnění. Plnění depresí bylo šetrnější jen v případě, kdy se v DEM vyskytovaly jen malé a mělké deprese.

Vytvořené nástroje sice odstraňují z DEM uzavřené deprese, ale problém vzniklých a původních plošin, ze kterých též nelze určit směr odtoku nedokáží řešit. S tím souvisí i výhled dalšího rozvoje, který by měl být směřován k doplnění dalších algoritmů určených právě k ošetření těchto plošin a vytvoření hydrologicky korektního modelu terénu.

8. Přehled literatury a použitých zdrojů

Arge L., Chase J. S., Halpin P., Toma L., Vitter J. S., Urban D., Wickremesinghe R., 2003: Efficient Flow Computation on Massive Grid Terrain Datasets. *Geoinformatica*, Volume 7, 4. S. 283 - 313.

Barták V., 2008: Algoritmy pro zpracování digitálních modelů terénu s aplikacemi v hydrologickém modelování. Česká zemědělská univerzita v Praze, Fakulta životního prostředí, Praha. 202 s. (diplomová práce). „nepublikováno“. Dep. SIC ČZU v Praze.

Barták V., 2010: How to extract river networks and catchment boundaries from DEM: a review of digital terrain analysis techniques. *Journal of Landscape Studies* 2. S. 57 - 68.

Barták V., Wild J., Gdulová K., 2012: Geoinformatické aplikace I - Úvod do jazyka Python, FŽP ČZU, Praha, 79 s.

Bašta P., 2016: Prostorové interpolační modely ve vybraných hydrologických aplikacích. Česká zemědělská univerzita v Praze, Fakulta životního prostředí, v Praze. 63 s. (Teze dizertační práce). „nepublikováno“ Dep. SIC ČZU v Praze.

Brůha L., 2016: Digitální modely terénu. Přírodovědecká fakulta Univerzita Karlova, Praha, 13 s.

El-Sheimy N., Valeo C., Habib A., 2005: Digital terrain modeling: acquisition, manipulation, and application. Artech House remote sensing library, Artech House, Inc., 257 s.

Florinsky I. V., 2002: Errors of signal processing in digital terrain modelling. *International Journal of Geographical Information Science* 16. S. 475 - 501.

Garbrecht J., Martz L. W., 1997: The assignment of drainage direction over flat surfaces in raster digital elevation models. *Journal of Hydrology* 193. S. 204 - 213.

Grimaldi S., Nardi F., Di Benedetto F., Istanbuloglu E., Bras R.L., 2007: A physically-based method for removing pits in digital elevation models. *Advances in Water Resources* 30. S. 2151 - 2158.

Jenson S. K., Domingue J. O., 1988: Extracting Topographic Structure from Digital Elevation Data for Geographic Information System Analysis. *Photogrammetric Engineering And Remote Sensing*, 54(11). November 1988. S. 1593 - 1600.

Jerri A. J., 1998: A. J., 1998, *The Gibbs Phenomenon in Fourier Analysis, Splines and Wavelet Approximation*, Kluwer Academic Publishers, Boston, 340 s.

Král D., Mareš M., Škoda P., 2013: Recepty z programátorské kuchařky Haldy, heapsort a Dijkstrův algoritmus (online) [cit. 2019.04.03], dostupné z <<https://ksp.mff.cuni.cz/kucharky/halda-a-cesty/>>.

Lindsay J. B., Creed I. F., 2005a: Sensitivity of digital landscapes to artifact depressions in remotely-sensed DEMs. *Photogrammetric Engineering and Remote Sensing* 71. S. 1029 - 1036.

Lindsay J. B., Creed I. F., 2005b: Removal of artifact depressions from digital elevation models: towards a minimum impact approach. *Hydrological Processes* 19. S. 3113 - 3126.

Lindsay J. B., Creed I. F., 2006: Distinguishing actual and artefact depressions in digital elevation data. *Computers&Geosciences* 32. S. 1192 - 1204.

Martz L. W., De Jong E., 1988: CATCH: a FORTRAN program for measuring catchment area from digital elevation models. *Computers&Geosciences* 14(5). S. 627 - 640.

Martz L. W., Garbrecht J., 1999: An outlet breaching algorithm for treatment of closed depressions in a raster DEM. *Computers&Geosciences* 25. S. 835 - 844.

Moore I. D., Grayson R. B., Ladson A. R., 1991: Digital terrain modelling: a review of hydrological, geomorphological, and biological applications. *Hydrological Processes* 5(1). S. 7 - 34.

Novák P., 2015: Tvorba nástroje pro digitální analýzu terénu. Česká zemědělská univerzita v Praze, Fakulta životního prostředí, Katedra aplikované geoinformatiky a územního plánování, Praha, 54 s. (diplomová práce). „nepublikováno“. Dep. SIC ČZU v Praze.

O'Callaghan J.F., Mark D.M., 1984: The extraction of drainage networks from digital elevation data. *Computer Vision, Graphics, and Image Processing* 28(3). S. 323 - 344.

Pacina J., Brejcha M., 2014: Digitální modely terénu. Univerzita J. E. Purkyně, Fakulta životního prostředí, Ústí nad Labem, 73 s.

Pilesjö P., Zhou Q., 1997: Theoretical Estimation Of Flow Accumulation From A Grid-Based Digital Elevation Model. in *Proceedings of GIS AM/FM ASIA'97 and Geoinformatics'97 Conference*, 26-29 May. S. 447 - 457.

Planchon O., Darboux F., 2001: A fast, simple and versatile algorithm to fill the depressions of digital elevation models. *Catena* 46. S. 159 - 176.

R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.)

Rieger W., 1998: A phenomenon-based approach to upslope contributing area and depressions in DEMs. *Hydrological Processes* 12. S. 857 - 872.

SciPy, 2019: What is NumPy? (online) [cit. 2019.20.03], dostupné z <<https://docs.scipy.org/doc/numpy/user/whatisnumpy.html>>.

Soille P., Vogt J., Colombo R., 2003: Carving and adaptive drainage enforcement of grid digital elevation models. *Water Resources Research* 39(12). 1366. doi:10.1029/2002WR001879

Soille P., 2004: Optimal removal of spurious pits in digital elevation models. *Water Resources Research* 40. W12509, doi:10.1029/2004WR003060.

Soille P., Gratin Ch., 1994: An efficient algorithm for drainage network extraction on DEMs. *Journal of Visual Communication and Image Representation* 5. S. 181 - 189.

Szypuła B., 2017: Digital Elevation Models in Geomorphology (online) [cit. 2019.25.02], dostupné z <https://www.researchgate.net/publication/320472815_Digital_Elevation_Models_in_Geomorphology>.

Šimeček S., 2017: Analýza odtokových cest v povodí horního toku Úpy užitím nástrojů GIS. Česká zemědělská univerzita v Praze, Fakulta životního prostředí, Praha. 51 s. (bakalářská práce). „nepublikováno“. Dep. SIC ČZU v Praze.

Šímová P., 2013: Návody ke cvičení pro ArcGIS 10.2, FŽP ČZU, Praha, 87 s.

Tarboton D. G., 1997: A new method for the determination of flow directions and upslope areas in grid digital elevation models. *Water Resources Research* 33. S. 309 - 319.

Tianqi A., Takeuchi K., Ishidaira H., Yoshitani J., Fukami K., 2003: Development and application of a new algorithm for automated pit removal for grid DEMs. *Hydrological Sciences Journal* 48. S. 985 - 997.

Tribe A., 1992: Automated recognition of valley lines and drainage networks from grid digital elevation models: a review and a new method. *Journal of Hydrology* 139. S. 263 - 293.

Zandbergen P. A., 2006: The effect of cell resolution on depressions in digital elevation models. *Applied GIS* 2. S. 04.1 - 04.35.

Zhu D., Ren Q., Xuan Y., Chen Y., Cluckie I. D., 2013: An effective depression filling algorithm for DEM-based 2-D surface flow modelling. *Hydrol. Earth Syst. Sci.* 17. S. 495 - 505.

Wang L., Liu H., 2006: An efficient method for identifying and filling surface depressions in digital elevation models for hydrologic analysis and modelling. *International Journal of Geographical Information Science* 20(2). S. 193 – 213.

9. Seznam obrázků

Obrázek 1: Digitální elevační model pro Českou republiku s rozlišením 100 m (GISAT, 2007): (online) [cit. 2019.03.26], dostupné z <
<http://www.gisat.cz/content/cz/produkty/data-ke-stazeni>>.

Obrázek 2: Dvě různé interpretace zobrazující vrstevnice (Lindsay, Creed, 2006).

Obrázek 3: Následek Gibbsova jevu v místě prudkého sklonu (Florinsky, 2002).

Obrázek 4: Příklad vyplnění složité deprese (Barták, 2009).

Obrázek 5: Postup vyplnění depresí metodou Wang a Lui (2006) (Wang, Lui, 2006).

Obrázek 6: Postup vyplnění depresí metodou Planchon a Darbaux (2001) (Tarboton, 2018): (online) [cit. 2019.03.26], dostupné z
<<https://slideplayer.com/slide/12137887/>>.

Obrázek 7: Příklad problému kaskádovité deprese (Barták, 2008).

Obrázek 8: Postup metody založené na gradientu ke spodnímu okraji (Garbrecht, Martz, 1997).

Obrázek 9: Postup metody založené na gradientu od horního okraje (Garbrecht, Martz, 1997).

Obrázek 10: Nástrojová sada v katalogu ArcMap.

Obrázek 11: Okno nástroje Depression layer.

Obrázek 12: Vytvořený polygonový vrstvy depresí a plošin.

Obrázek 13: Okno nástroje Fill DEM.

Obrázek 14: Buňky modifikované nástrojem Fill DEM.

Obrázek 15: Okno nástroje Carving.

Obrázek 16: Buňky modifikované nástrojem Carving.

Obrázek 17: Rozdíl mezi procentem ovlivněných buněk metodami Fill a Carving na vzorku dvaceti náhodně vybraných povodí ČR III. řádu.

Obrázek 18: Rozdíl mezi změněným objemem DEM metodami Fill a Carving na vzorku dvaceti náhodně vybraných povodí ČR III. řádu.

Obrázek 19: Rozdíl mezi procentem ovlivněných buněk metodami Fill a Carving na vzorku dvaceti náhodně vybraných povodí ČR IV. řádu.

Obrázek 20: Rozdíl mezi změněným objemem DEM metodami Fill a Carving na vzorku dvaceti náhodně vybraných povodí ČR IV. řádu.

10. Seznam tabulek

Tabulka 1: Plnění depresí v DEM podle autorů Jenson a Domingue (1988) (Jenson, Domingue, 1988).

Tabulka 2: Postup hledání bodu přetečení od autorů Jenson a Domingue (1988) (Barták, 2008).

Tabulka 3: Postup protržení hráze uzavřené deprese (Martz, Garbrecht, 1999).

Tabulka 4: Implementace metody carving (Soille a kol.,2003).

Tabulka 5: Postup metody založené na gradientu od horního okraje (Garbrecht, Martz, 1997).

Tabulka 6: Seznam modulů používaných v nástrojích Depression layer, Fill DEM a Carving.

Tabulka 7: Popis vytvořených nástrojů.

Tabulka 8: Popis vytvořených skript.

Tabulka 9: Porovnání nástrojů Fill DEM a Carving na dvaceti hydrologických povodí III. řádu.

Tabulka 10: Porovnání metod Fill DEM a Carving na dvaceti hydrologických povodí IV. řádu.

Tabulka 11: Porovnání ovlivnění DEM metodami Fill a Carving na vzorku náhodně vybraných povodí III. a IV. řádu párovým Wilcoxonovým testem.

11. Přílohy

11.1. border_of_raster

```
# -*- coding: cp1250 -*-
"""
Module "border_of_raster" contains the function "border" that finds
the edge of the raster and adds it to the heap.

Modul "border_of_raster" obsahuje funkci "border", která najde okraj
rastru a přidá jej do haldy.
"""
import numpy, heap

def border(array, ha, Xmax, Ymax):
    rc = numpy.zeros((Xmax, Ymax))
    for x in range(Xmax):
        for y in range(Ymax):
            if array[x, y] != -999:
                if x == 0:
                    if y == 0:
                        if (array[x+1, y] == -999) or (array[x+1, y+1]
                            == -999) or (array[x, y+1] == -999):
                            heap.ins_el (ha, [x, y, array[x, y]],
                                index=2)
                            rc[x, y] = 1
                    elif y == Ymax-1:
                        if (array[x, y-1] == -999) or (array[x+1, y]
                            == -999) or (array[x+1, y-1] == -999):
                            heap.ins_el (ha, [x, y, array[x, y]],
                                index=2)
                            rc[x, y] = 1
                    elif y != 0 and y != Ymax-1:
                        if (array[x, y+1] == -999) or (array[x+1, y+1]
                            == -999) or (array[x+1, y] == -999 or
                                (array[x+1, y-1] == -999) or (array[x, y-1]
                                    == -999)):
                            heap.ins_el (ha, [x, y, array[x, y]],
                                index=2)
                            rc[x, y] = 1
                elif x == Xmax-1:
                    if y == 0:
                        if (array[x, y+1] == -999) or (array[x-1, y]
                            == -999) or (array[x-1, y+1] == -999):
                            heap.ins_el (ha, [x, y, array[x, y]],
                                index=2)
                            rc[x, y] = 1
                    elif y == Ymax-1:
                        if (array[x, y-1] == -999) or (array[x-1, y-1]
                            == -999) or (array[x-1, y] == -999):
                            heap.ins_el (ha, [x, y, array[x, y]],
                                index=2)
                            rc[x, y] = 1
```

```

    elif y != 0 and y != Ymax-1:
        if (array[x,y+1] == -999) or (array[x,y-1]
            == -999) or (array[x-1,y] == -999) or
            (array[x-1,y+1] == -999) or (array[x-1,y-1]
            == -999):
            heap.ins_el (ha, [x,y,array[x,y]],
                index=2)
            rc[x,y] = 1
elif x != 0 and x != Xmax-1:
    if y == 0:
        if (array[x,y+1] == -999) or (array[x+1,y+1]
            == -999) or (array[x+1,y] == -999) or
            (array[x-1,y] == -999) or (array[x-1,y+1]
            == -999):
            heap.ins_el (ha, [x,y,array[x,y]],
                index=2)
            rc[x,y] = 1
    elif y == Ymax-1:
        if (array[x+1,y] == -999) or (array[x-1,y]
            == -999) or (array[x+1,y-1] == -999) or
            (array[x-1,y-1] == -999) or (array[x,y-1]
            == -999):
            heap.ins_el (ha, [x,y,array[x,y]],
                index=2)
            rc[x,y] = 1
    elif y != 0 and y != Ymax-1:
        if (array[x,y+1] == -999) or (array[x+1,y+1]
            == -999) or (array[x+1,y] == -999) or
            (array[x+1,y-1] == -999) or (array[x,y-1]
            == -999) or (array[x-1,y-1] == -999) or
            (array[x-1,y] == -999) or (array[x-1,y+1]
            == -999):
            heap.ins_el (ha, [x,y,array[x,y]],
                index=2)
            rc[x,y] = 1

for x in range(Xmax):
    if x == 0 or x == Xmax-1:
        for y in range (Ymax):
            if (array[x,y] != -999) and (rc[x,y] != 1):
                heap.ins_el (ha, [x, y, array[x,y]], index=2)
    else:
        if (array[x,0] != -999) and (rc[x,0] != 1):
            heap.ins_el (ha, [x, 0, array[x,0]], index=2)
        if (array[x,Ymax-1] != -999) and (rc[x,Ymax-1] != 1):
            heap.ins_el (ha, [x, Ymax-1, array[x,Ymax-1]],
                index=2)

return ha

```

11.2. neighbors

```
# -*- coding: cp1250 -*-  
"""
```

Module "neighbors" contains one function that finds the elevation of neighboring cells, detects its coordinates and adds this informations to the list.

Modul "neighbors" obsahuje jednu funkci "neighbors_z", která najde elevaci sousedních buněk, zjistí jejich souřadnice a přidá tyto informace do seznamu.

```
"""
```

```
def neighbors_z(array,x,y,Xmax,Ymax):  
    sez = []  
    if x == 0:  
        if y == 0:  
            sez.extend([[x,y+1,array[x,y+1]])]  
            sez.extend([[x+1,y+1,array[x+1,y+1]])]  
            sez.extend([[x+1,y,array[x+1,y]])]  
        elif y == Ymax-1:  
            sez.extend([[x,y-1,array[x,y-1]])]  
            sez.extend([[x+1,y,array[x+1,y]])]  
            sez.extend([[x+1,y-1,array[x+1,y-1]])]  
        elif y != 0 and y != Ymax-1:  
            sez.extend([[x,y+1,array[x,y+1]])]  
            sez.extend([[x+1,y+1,array[x+1,y+1]])]  
            sez.extend([[x+1,y,array[x+1,y]])]  
            sez.extend([[x+1,y-1,array[x+1,y-1]])]  
            sez.extend([[x,y-1,array[x,y-1]])]  
    elif x == Xmax-1:  
        if y == 0:  
            sez.extend([[x,y+1,array[x,y+1]])]  
            sez.extend([[x-1,y,array[x-1,y]])]  
            sez.extend([[x-1,y+1,array[x-1,y+1]])]  
        elif y == Ymax-1:  
            sez.extend([[x,y-1,array[x,y-1]])]  
            sez.extend([[x-1,y-1,array[x-1,y-1]])]  
            sez.extend([[x-1,y,array[x-1,y]])]  
        elif y != 0 and y != Ymax-1:  
            sez.extend([[x,y+1,array[x,y+1]])]  
            sez.extend([[x,y-1,array[x,y-1]])]  
            sez.extend([[x-1,y,array[x-1,y]])]  
            sez.extend([[x-1,y+1,array[x-1,y+1]])]  
            sez.extend([[x-1,y-1,array[x-1,y-1]])]  
    elif x != 0 and x != Xmax-1:  
        if y ==0:  
            sez.extend([[x,y+1,array[x,y+1]])]  
            sez.extend([[x+1,y+1,array[x+1,y+1]])]  
            sez.extend([[x+1,y,array[x+1,y]])]  
            sez.extend([[x-1,y,array[x-1,y]])]  
            sez.extend([[x-1,y+1,array[x-1,y+1]])]
```

```

elif y == Ymax-1:
    sez.extend([[x+1,y,array[x+1,y]]])
    sez.extend([[x-1,y,array[x-1,y]]])
    sez.extend([[x+1,y-1,array[x+1,y-1]]])
    sez.extend([[x-1,y-1,array[x-1,y-1]]])
    sez.extend([[x,y-1,array[x,y-1]]])
elif y != 0 and y != Ymax-1:
    sez.extend([[x,y+1,array[x,y+1]]])
    sez.extend([[x+1,y+1,array[x+1,y+1]]])
    sez.extend([[x+1,y,array[x+1,y]]])
    sez.extend([[x+1,y-1,array[x+1,y-1]]])
    sez.extend([[x,y-1,array[x,y-1]]])
    sez.extend([[x-1,y-1,array[x-1,y-1]]])
    sez.extend([[x-1,y,array[x-1,y]]])
    sez.extend([[x-1,y+1,array[x-1,y+1]]])

return sez

```

11.3. depression_layer

```
#!/usr/bin/env python
#-*- coding: cp1250 -*-
"""
This script contains a "Dep_layer" function that loads the DEM raster
and creates a polygon layer of depressions and flat areas. In the
attribute table of the newly created layer fills in information
(depression number, maximum elevation, minimum elevation, maximum
depression depth, x and y depression coordinates) about each
depression.

Tento skript obsahuje funkci "Dep_layer", která načte rastr DEM
a vytvoří polygonovou vrstvu depresí a plošin. V atributové tabulce
nově vytvořené vrstvy vyplní informace (číslo deprese, maximální
elevaci, minimální elevaci, maximální hloubku deprese, x a y
souřadnici deprese) o každé depresi.
"""

# Load modules.
# Načtení modulů.
import arcpy, numpy, sys, os, rta, heap, neighbors, border_of_raster
from arcpy.sa import *

arcpy.env.overwriteOutput = True

# Linking the parameters specified in the window toolbox to the script.
# Propojení parametrů zadaných v okně nástroje se skriptem.
raster = sys.argv[1]
output = sys.argv[2]
output_flat = sys.argv[3]

def Dep_layer(raster, output, output_flat):

    # Finding the path to the input raster.
    # Zjištění cesty ke vstupnímu rastru.
    location = arcpy.Describe(raster)
    path = location.path

    # Loads and convert raster to numpy.array.
    # Načtení a převedení rastru do numpy.array.
    aar_dem = rta.raster_to_numpy(raster)

    DEM = aar_dem[0]
    Xmax = aar_dem[1]
    Ymax = aar_dem[2]
    vCell = aar_dem[3]
    LeftX = aar_dem[4]
    LeftY = aar_dem[5]

    # Creates a zero array of the same size as the converted raster.
    # Vytvoření pole nul o stejné velikosti jako převedený rastr.
    ra = numpy.zeros((Xmax, Ymax))
    rb = numpy.zeros((Xmax, Ymax))
```



```

rc = numpy.zeros((Xmax,Ymax))

# Creates a heap (x,y,z) sorted by the "z" value from values on
# the border of the input raster.
# Vytvoření haldy (x,y,z) řazenou podle hodnoty "z" z hodnot na
# okraji vstupního rastru.
ha = []
border_of_raster.border(DEM,ha,Xmax,Ymax)

# Checks the neighbors of the first element in the heap.
# Kontrola sousedů prvního prvku v haldě.
List1 = []
List2 = []
y = 1
x = 1
while len(ha) != 0:
    # Sets the first element in the heap as processed.
    # Nastavení prvního prvku v halde jako zpracovaný.
    ra[ha[0][0],ha[0][1]] = 1

    # Detects coordinates and elevation of neighboring cells.
    # Zjištění souřadnic a elevací sousedních buněk.
    sou = neighbors.neighbors_z(DEM,ha[0][0],ha[0][1],Xmax,Ymax)

    for i in sou:
        # Checks if the neighbor is not already processed.
        # Kontrola souseda, jestli není už zpracovaný.
        if ra[i[0],i[1]] != 1:
            # Check if the neighbor contains NoData.
            # Kontrola, jestli soused neobsahuje NoData.
            if i[2] != -999:
                if i[2] < ha[0][2]:
                    ra[i[0],i[1]] = 1
                    # Getting basic data about depression.
                    # Získání základních údajů o depresi.
                    if rb[ha[0][0],ha[0][1]] == 0:
                        rb[i[0],i[1]] = x
                        list1 = [x,ha[0][2],i[2],float(ha[0][2])
                                - float(i[2]), i[0], i[1]]
                        List1.append(list1)
                        del(list1)
                        x = x+1
                    else:
                        rb[i[0],i[1]] = rb[ha[0][0],ha[0][1]]
                        if i[2]< List1[int(rb[i[0],i[1]))-1][2]:
                            List1[int(rb[i[0],i[1]))-1][2]= i[2]
                            List1[int(rb[i[0],i[1]))-1][3]=
                                float(ha[0][2])-float(i[2])
                            List1[int(rb[i[0],i[1]))-1][4]= i[0]
                            List1[int(rb[i[0],i[1]))-1][5]= i[1]
                        # Override neighbor elevation.
                        # Přepsání elevace souseda.
                        DEM[i[0],i[1]] = ha[0][2]

```

```

if i[2] == ha[0][2]:
    ra[i[0],i[1]] = 1
    # Getting basic data about flat areas.
    # Získání základních údajů o plošině.
    if rc[ha[0][0],ha[0][1]] == 0:
        rc[ha[0][0],ha[0][1]] = y
        rc[i[0],i[1]] = y
        list2 = [y, ha[0][2]]
        List2.append(list2)
        del(list2)
        y = y+1
    else:
        rc[i[0],i[1]] = rc[ha[0][0],ha[0][1]]
if i not in ha:
    # If neighbor is not in the heap, adds
    # Pokud není soused v haldě, přidá se.
    heap.ins_el(ha, [i[0], i[1], DEM[i[0],i[1]]],
                index=2)

    # Removes the first element in the heap and replaces the next
    # with the smallest elevation value.
    # Odstranění prvního prvku v haldě a nahrazení dalším s
    # nejmenší hodnotou elevace.
    heap.ret_min(ha)

# Turn on extension to the tools "Spatial".
# Zapne extenzi k nástrojům "Spatial".
arcpy.CheckOutExtension("Spatial")

if 1 in rb:
    # Convert numpy.array to raster and save.
    # Převedení numpy.array na rastr a jeho uložení.
    depression_raster = arcpy.NumPyArrayToRaster(rb,arcpy.Point(
        LeftX,LeftY),vCell,value_to_nodata = 0)

    # Convert raster to integer format.
    # Převedení rastru na formát integer.
    outInt = Int(depression_raster)

    # Convert raster to polygon layer.
    # Převedení rastru na polygonovou vrstvu.
    arcpy.RasterToPolygon_conversion(outInt,output,
        "NO_SIMPLIFY")

    # Create a table and save it.
    # Vytvoření tabulky a její uložení.
    arcpy.CreateTable_management(path+"\\", "Tab.dbf")
    path_tab = path+"\\"+"Tab.dbf"

    # Create new fields in the newly created table.
    # Vytvoření nových polí v nově vytvořené tabulce.
    arcpy.AddField_management( path_tab,"Num_dep","DOUBLE")
    arcpy.AddField_management( path_tab,"Max_Heigh","DOUBLE")

```

```

arcpy.AddField_management( path_tab,"Min_Heigh","DOUBLE")
arcpy.AddField_management( path_tab,"Depth","DOUBLE")
arcpy.AddField_management( path_tab,"x","DOUBLE")
arcpy.AddField_management( path_tab,"y","DOUBLE")

# Adding information about each depression in created table.
# Doplnění údajů o každé depresi v nově vytvořené tabulce.
with arcpy.da.InsertCursor(path_tab, ["Num_dep","Max_Heigh",
"Min_Heigh","Depth","x","y"]) as cursor:
    for i in List1:
        cursor.insertRow([int(i[0]),float(i[1]),float(i[2])
, float(i[3]),i[4],i[5]])

# Linking the table to the polygon layer of depression.
# Propojení tabulky s polygonovou vrstvou depresi.
arcpy.JoinField_management(output, "GRIDCODE", path_tab,
"Num_dep")

# Deleting unnecessary fields in the table.
# Smazání nepotřebných polí v tabulce.
arcpy.DeleteField_management(output_flat,["GRIDCODE",
"Field1"])

# Deleting a created table.
# Smazání vytvořené tabulky.
os.remove(path_tab)
os.remove(path_tab+".xml")
os.remove(path_tab[:len(path_tab)-4]+".cpg")

else:
    arcpy.AddMessage("-----")
    arcpy.AddMessage("DEM does not contain any depression")
    arcpy.AddMessage("DEM neobsahuje žádné uzavřené deprese")

if 1 in rc:
    # Convert numpy.array to raster and save.
    # Převedení numpy.array na rastr a jeho uložení.
    flat_area_raster = arcpy.NumPyArrayToRaster(rc,arcpy.Point(
LeftX,LeftY),vCell,value_to_nodata = 0)

    # Convert raster to integer format.
    # Převedení rastru na formát integer.
    outInt2 = Int(flat_area_raster)

    # Convert raster to polygon layer.
    # Převedení rastru na polygonovou vrstvu.
    arcpy.RasterToPolygon_conversion(outInt2, output_flat,
"NO_SIMPLIFY")

    # Create a table and save it.
    # Vytvoření tabulky a její uložení.
    arcpy.CreateTable_management(path+"\\", "Tab2.dbf")
    path_tab2 = path+"\\\\"+"Tab2.dbf"

```

```

# Create new fields in the newly created table.
# Vytvoření nových polí v nově vytvořené tabulce.
arcpy.AddField_management( path_tab2,"Num_area","DOUBLE")
arcpy.AddField_management( path_tab2,"Heigh","DOUBLE")

# Adding information about each depression in created table.
# Doplnění údajů o každé depresi v nově vytvořené tabulce.
with arcpy.da.InsertCursor(path_tab2, ["Num_area","Heigh"])
as cursor:
    for i in List2:
        cursor.insertRow([int(i[0]),float(i[1])])

# Linking the table to the polygon layer of depression.
# Propojení tabulky s polygonovou vrstvou depresí.
arcpy.JoinField_management(output_flat, "GRIDCODE",
path_tab2, "Num_area")

# Deleting unnecessary fields in the table.
# Smazání nepotřebných polí v tabulce.
arcpy.DeleteField_management(output_flat,["GRIDCODE",
"Field1"])

# Deleting a created table.
# Smazání vytvořené tabulky.
os.remove(path_tab2)
os.remove(path_tab2+".xml")
os.remove(path_tab2[:len(path_tab2)-4]+".cpg")

else:
    arcpy.AddMessage("-----")
    arcpy.AddMessage("DEM does not contain any flat areas")
    arcpy.AddMessage("DEM neobsahuje žádné plošiny")

pls = Dep_layer(raster, output, output_flat)

```

11.4. fill_DEM

```
# -*- coding: cp1250 -*-
"""
This script contains a "fill_DEM" function that loads the DEM raster,
fills all of its closed depressions to the elevation of their pour
point. It is possible to enter a polygon layer of depression and
determine in advance what depression will be subject to filling.

Tento skript obsahuje funkci "fill_DEM", která načte rastr DEM
a vyplní všechny uzavřené deprese na hodnotu elevace jejich bodu
přetečení. Je možno zadat polygonovou vrstvu depresí a předem určit
jaky deprese budou předmětem plnění.
"""

# Load modules.
# Načtení modulů.
import arcpy, numpy, sys, os, rta, heap, neighbors, border_of_raster

# Linking the parameters specified in the window toolbox to the script.
# Propojení parametrů zadaných v okně nástroje se skriptem.
raster = sys.argv[1]
output = sys.argv[2]
depression = sys.argv[3]

arcpy.env.overwriteOutput = True

def fill_DEM(raster , output, depression):

    arcpy.env.extent = raster

    # Loads the raster to numpy.array.
    # Načtení rastru do numpy.array.
    aar_dem = rta.raster_to_numpy(raster)

    DEM = aar_dem[0]
    Xmax = aar_dem[1]
    Ymax = aar_dem[2]
    vCell = aar_dem[3]
    LeftX = aar_dem[4]
    LeftY = aar_dem[5]

    # Creates a zero array of the same size as the converted raster.
    # Vytvoření pole nul o stejné velikosti, jako převedený rastr.
    ra = numpy.zeros((Xmax,Ymax))

    # Creates a heap (x,y,z) sorted by the "z" value from values on
    the border of the input raster.
    # Vytvoření haldy (x,y,z) řazenou podle hodnoty "z" z hodnot na
    okraji vstupního rastru.
    ha = []
    border_of_raster.border(DEM,ha,Xmax,Ymax)
```

```

# Checks the neighbors of the first element in the heap.
# Kontrola sousedů prvního prvku v halde.
while len(ha) != 0:
    # Sets the first element in the heap as processed.
    # Nastavení prvního prvku v halde jako zpracovaný.
    ra[ha[0][0],ha[0][1]] = 1

    # Detects coordinates and elevation of neighboring cells.
    # Zjištění souřadnic a elevací sousedních buněk.
    sou = neighbors.neighbors_z(DEM,ha[0][0],ha[0][1],Xmax,Ymax)

    for i in sou:
        # Checks if the neighbor is not already processed.
        # Kontrola souseda, jestli není už zpracovaný.
        if ra[i[0],i[1]] != 1:
            # Check if the neighbor contains NoData.
            # Kontrola, jestli soused neobsahuje NoData.
            if i[2] != -999:
                if i[2] < ha[0][2]:
                    ra[i[0],i[1]] = 1
                    # Override neighbor elevation.
                    # Přepsání elevace souseda.
                    DEM[i[0],i[1]] = ha[0][2]
                if i not in ha:
                    # If neighbor is not in the heap, adds
                    # Pokud není soused v haldě, přidá se.
                    heap.ins_el(ha,[i[0],i[1], DEM[i[0],i[1]]],
                                index=2)

    # Removes the first element in the heap and replaces the next
    # with the smallest elevation value.
    # Odstranění prvního prvku v haldě a nahrazení dalším s
    # nejmenší hodnotou elevace.
    heap.ret_min(ha)

if depression != "#":
    aar_dem_2 = rta.raster_to_numpy(raster)
    Org_DEM = aar_dem_2[0]

    # Converting the polygon layer to depression and numpy.array.
    # Převadí polygonové vrstvy depresí na rastr a numpy.array.
    dep_raster = depression[:len(depression)-4]+".tif"
    arcpy.PolygonToRaster_conversion(depression, "Num_dep",
    dep_raster, "CELL_CENTER", cellsize = vCell)
    aar_dem_rb = rta.raster_to_numpy(dep_raster)
    Dep_raster = aar_dem_rb[0]

    Final_DEM = numpy.zeros((Xmax,Ymax))

```

```

# Scan the raster layer for depression and assign the
# appropriate elevation value to the newly created raster.
# Skenování rastrové vrstvy depresí a přidělení příslušné
# hodnoty elevace nově vytvořenému rastru.
for i in range(Xmax):
    for j in range(Ymax):
        if Dep_raster[i,j] == -999:
            Final_DEM[i,j] = Org_DEM[i,j]
        else:
            Final_DEM[i,j] = DEM[i,j]

# Convert numpy.array to raster and save.
# Převedení numpy.array na rastr a jeho uložení.
if depression != "#":
    fill_raster =arcpy.NumPyArrayToRaster(Final_DEM,arcpy.Point(
        LeftX,LeftY),vCell,value_to_nodata = -999)
    fill_raster.save(output)
else:
    fill_raster =arcpy.NumPyArrayToRaster(DEM,arcpy.Point(LeftX,
        LeftY),vCell,value_to_nodata = -999)
    fill_raster.save(output)

if depression != "#":
    # Removing the raster layer of depression.
    # Odstranění rastrové vrstvy depresí
    os.remove(dep_raster[:len(dep_raster)-4] + ".tfw")
    os.remove(dep_raster + ".aux.xml")
    os.remove(dep_raster + ".ovr")
    os.remove(dep_raster + ".vat.cpg")
    os.remove(dep_raster + ".vat.dbf")
    os.remove(dep_raster + ".xml")
    os.remove(dep_raster)

pls = fill_DEM(raster, output, depression)

```

11.5. carving

```
# -*- coding: cp1250 -*-

"""
This script contains a "fill_DEM" function that loads the DEM raster
and creates a path from each bottom of the depression bottom to the
outside of the depression.

Tento skript obsahuje funkci "Carving", která načte rastr DEM a
vytvoří cestu od každého dna dna deprese směrem ven z prohlubně.
"""

# Load modules.
# Načtení modulů.
import arcpy, numpy, os, sys, rta, heap, neighbors, border_of_raster

arcpy.env.overwriteOutput = True

# Linking the parameters specified in the window toolbox to the script.
# Propojení parametrů zadaných v okně nástroje se skriptem.
raster = sys.argv[1]
output = sys.argv[2]

def Carving(raster, output):

    # Loads the raster to numpy.array.
    # Načtení rastru do numpy.array
    aar_dem = rta.raster_to_numpy(raster)

    DEM = aar_dem[0]
    Xmax = aar_dem[1]
    Ymax = aar_dem[2]
    vCell = aar_dem[3]
    LeftX = aar_dem[4]
    LeftY = aar_dem[5]

    # Creates a zero array of the same size as the converted raster.
    # Vytvoření pole nul o stejné velikosti, jako převedený rastr.
    ra = numpy.zeros((Xmax, Ymax))
    rx = numpy.zeros((Xmax, Ymax))
    ry = numpy.zeros((Xmax, Ymax))

    # Creates a heap (x,y,z) sorted by the "z" value from values on
    the border of the input raster.
    # Vytvoření haldy (x,y,z) řazenou podle hodnoty "z" z hodnot na
    okraji vstupního rastru.
    ha = []
    border_of_raster.border(DEM, ha, Xmax, Ymax)
```



```

while len(ha) != 0:
    dep = False
    # Sets the first element in the heap as processed.
    # Nastavení prvního prvku v halde jako zpracovaný.
    ra[ha[0][0],ha[0][1]] = 1
    # Detects coordinates and elevation of neighboring cells.
    # Zjištění souřadnic a elevací sousedních buněk.
    sou = neighbors.neighbors_z(DEM,ha[0][0],ha[0][1],Xmax,Ymax)

    # Checking if a cell is a potential bottom of depression.
    # Kontrola, jestli buňka není potencionální dno deprese.
    if ha[0][0] != 0:
        if ha[0][0] != Xmax-1:
            if ha[0][1] != 0:
                if ha[0][1] != Ymax-1:
                    List = []
                    for i in sou:
                        if i[2] > ha[0][2]:
                            List.append(1)
                        if i[2] == ha[0][2]:
                            List.append(2)
                        if i[2] < ha[0][2]:
                            List.append(3)
                    if 3 not in List and 2 not in List and 1 in List:
                        dep = True
                    if 1 in List and 2 in List and 3 not in List:
                        dep = True
                    del(List)

    # Create a path from the bottom of depression out of sink.
    # Vytvoření cesty ode dna deprese ven z deprese.
    if dep == True:
        x = ha[0][0]
        y = ha[0][1]
        stop = False
        while not stop:
            if DEM[x,y] <= DEM[int(rx[x,y]),int(ry[x,y])]:
                DEM[int(rx[x,y]),int(ry[x,y])] = DEM[x,y]
                x2 = x
                x = int(rx[x,y])
                y = int(ry[x2,y])
                del (x2)
            else:
                stop = True
        if x == 0 or x == Xmax-1 or y == 0 or y == Ymax-1:
            stop = True
        if DEM[x,y] == -999:
            stop = True

    del(x)
    del(y)

```

```

x = ha[0][0]
y = ha[0][1]

# Removes the first element in the heap and replaces the next
with the smallest elevation value.
# Odstranění prvního prvku v haldě a nahrazení dalším s
nejmenší hodnotou elevace.
heap.ret_min(ha)

# Unprocessed neighbors are added link to the processed cell.
# Nezpracovaným sousedům se přidá odkaz na zpracovávanou
buňku.
for i in sou:
    if ra[i[0],i[1]] != 1:
        if i[2] != -999:
            ra[i[0],i[1]] = 1
            rx[i[0],i[1]] = int(x)
            ry[i[0],i[1]] = int(y)
            if i not in ha:
                # If neighbor is not in the heap, adds
                # Pokud není soused v haldě, přidá se.
                heap.ins_el(ha, [i[0], i[1],DEM[i[0],i[1]]],
                    index=2)

# Convert numpy.array to raster and save.
# Převedení numpy.array na rastr a jeho uložení.
Fin_DEM = arcpy.NumPyArrayToRaster(DEM, arcpy.Point(LeftX, LeftY),
    vCell, value_to_nodata = -999)
Fin_DEM.save(output)

pls = Carving (raster, output)

```

11.6. Obsah přiloženého CD

Součástí diplomové práce je CD obsahující vytvořené nástroje a všechny skripty potřebné k jejich spuštění.

Toolbox	složka obsahující nástroje a skripty
HyCoTer.dbx	toolbox ArcGISu
*.py	zdrojový kódy skriptů
Ukázka použití	složka obsahující ukázková data
HyCoTer.mxd	ArcMap GIS Project
Vstupni_vrstva_DEM	složka obsahující vstupní DEM
DEM	rastr DEM
Output_Depression_layer	složka obsahující výstupy Depression layer
depressions.shp	polygonová vrstva depresí
flat_areas.shp	polygonová vrstva plošin
Output_Fill_DEM	složka obsahující výstup Fill DEM
Fill	rastr DEM s vyplněnými depresemi
Output_Carving	složka obsahující výstup Carving
carving	rastr DEM s odstraněnými depresemi
Diplomová práce.pdf	textová část diplomové práce