ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Využití umělé inteligence pro analýzu úniku osob z hořící místnosti
Student:	Bc Jiří Kukačka
Vedoucí:	doc. RNDr. Ing. Marcel Jiřina, Ph.D.
Studijní program:	Informatika
Studijní obor:	Znalostní inženýrství
Katedra:	Katedra teoretické informatiky
Platnost zadání:	Do konce zimního semestru 2018/19

Pokyny pro vypracování

Optimalizace prostor z hlediska možností úniku osob z těchto prostor je algoritmicky zajímavým problémem i prakticky orientovanou úlohou.
1) Seznamte se s problémem úniku z hořící místnosti a stávajícími přístupy k řešení této problematiky.
2) Navrhněte vhodný model pro reprezentaci hořící místnosti.
3) Identifikuje a po diskuzi s vedoucím práce vyberte algoritmy umělé inteligence vhodné pro řešení problému optimalizace únikových cest z místnosti.
4) Vybrané algoritmy implementujte ve vhodném programovacím jazyku.
5) Srovnejte jednotlivé algoritmy mezi sebou z hlediska dosažených výsledků, včetně řešení navrženého člověkem. Dosažené výsledky diskutujte.

Seznam odborné literatury

Dodá vedoucí práce.

doc. Ing. Jan Janoušek, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
děkan

V Praze dne 2. března 2017

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Diplomová práce

Využití umělé inteligence pro analýzu úniku osob z hořící místnosti

Bc. Jiří Kukačka

Katedra teoretické informatiky
Vedoucí práce: doc. RNDr. Ing. Marcel Jiřina Ph.D.

9. května 2018

Poděkování

Děkuji svému vedoucímu za vstřícnost a podporu, stejně tak své rodině, přátelům, kolegům a vedení v práci.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 9. května 2018 .....

České vysoké učení technické v Praze
Fakulta informačních technologií
© 2018 Jiří Kukačka. Všechna práva vyhrazena.
Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Kukačka, Jiří. *Využití umělé inteligence pro analýzu úniku osob z hořící místnosti.* Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.

Abstrakt

Cílem této práce je probádat možnosti využití algoritmů umělé inteligence a strojového učení v odvětví požární optimalizace budov, a položit tak základy pro samočinnou kontrolu návrhů budov.

Klíčová slova umela inteligence, neuronova sit, geneticky algoritmus, optimalizace mravenci kolonii, optimalizace budovy, simulace pohybu osoby

Abstract

Aim of this thesis is to explore possibilities of using artificial intelligence algorithms and machine learning in field of fire protection related optimalization of buildings, and set basics for automated design checking of buildings.

Keywords artificial intelligence, neural network, genetic algorithm, ant colony optimalization, building optimalization, person movement simulation

Obsah

Úvod **1**

1 Cíl práce **3**
1.1 Původ myšlenky 3
1.2 Inovativní přístup 4

2 Rešerše **7**
2.1 FSES 7
2.2 FIERAsystem 8
2.3 Tabulkové vyhodnocování 8

3 Analýza **11**
3.1 Problematika hořících místností a jejich optimalizace z pohledu požární bezpečnosti 12
3.2 Rozbor algoritmů vhodných pro řešení dílčích problémú 14

4 Návrh **31**
4.1 Nástroj pro tvorbu místností 33
4.2 Optimalizace místnosti 34
4.3 Ohodnocení rizik a poskytnutí doporučení 36

5 Realizace **41**
5.1 Uživatelské rozhraní 41

6 Výsledky **45**
6.1 Optimalizace místnosti 45
6.2 Ohodnocení rizik a poskytnutní doporučení 48

7 Diskuse **57**
7.1 Kvalita výsledků 57

7.2 Problematické vstupy 58
7.3 Možná rozšíření 59

Závěr **61**

Literatura **63**

A Seznam použitých zkratek **67**

B Obsah přiloženého CD **69**

Seznam obrázků

4.1 Diagram workflow pro uživatele 33

5.1 Uživatelské rozhraní s vzorem vytvořeného domu - skladiště v zástavbě 42
5.2 Vygenerovaná mapa požárního nebezpečí - skladiště v zástavbě . . 42
5.3 Vygenerovaná mapa průchodnosti - skladiště v zástavbě 43
5.4 Ukázka místnosti k modelování 43
5.5 Hotový model pro vyhoodnocení rizik 43

6.1 Rychlost optimalizace vzhledem k rostoucí náročnosti problému . . 47
6.2 Ukázka výstupu po shlukování 48
6.3 Jiný výstup po shlukování 48
6.4 Špatný výstup reprezentující špatný shluk 48
6.5 Rychlost ohodnocení vzhledem k rostoucí náročnosti problému . . 49
6.6 Reprezentace směrů pomocí barev PSO diagramů 49
6.7 Ukázka algoritmu ACO, první vzorek 50
6.8 Ukázka algoritmu ACO, druhý vzorek 50
6.9 Ukázka algoritmu ACO, třetí vzorek 50
6.10 Ukázka algoritmu ACO, čtvrtý vzorek 50
6.11 Ukázka algoritmu PSO, první vzorek 51
6.12 Ukázka algoritmu PSO, druhý vzorek 51
6.13 Ukázka algoritmu PSO, třetí vzorek 51
6.14 Ukázka algoritmu PSO, čtvrtý vzorek 51
6.15 Ukázka algoritmu GWO, první vzorek 52
6.16 Ukázka algoritmu GWO, druhý vzorek 52
6.17 Ukázka algoritmu GWO, třetí vzorek 52
6.18 Ukázka algoritmu GWO, čtvrtý vzorek 52
6.19 Ukázka umístění bezpečnostního značení 53
6.20 Ukázka umístění bezpečnostního značení - zvýraznění 53
6.21 Závěrečné ohodnocení výsledků algoritmů člověkem 55

7.1 Ukázka problematické mapy, červená zóna tvoří problém 59

Seznam tabulek

6.1 Rychlost algoritmů optimalizace na základě složitosti problému - čas potřebný k nalezení všech řešení [s] 45
6.2 Rychlost algoritmů optimalizace na základě složitosti problému - čas potřebný k nalezení prvního platného řešení [s] 46
6.3 Ohodnocení výstupu člověkem [1-5, 1 nejlepší] 46
6.4 Čas shlukování podle počtu shluků a velikostí instance [s] 46
6.5 Rychlost vyhodnocení různých situací různými algoritmy [s] 48
6.6 Kvalita vyhodnocení jedné situace různými algoritmy - průměrný čas jedné iterace 48
6.7 Ohodnocení pomocí NN - referenční hodnocení 54

Úvod

V moderním světě je běžné, že velké množství budov spadá do kategorie se zvýšeným požárním rizikem. To je zapříčiněno velkým množstvím hořlavin koncentrovaných v budově, velkým množstvím lidí navštěvující budovu a nebo velkou výškou budovy. Zákon v tomto případě rozpoznává 3 kategorie budov: 1) bez zvýšeného požárního nebezpečí, 2) se zvýšeným požárním nebezpečím, 3) s vysokým požárním nebezpečím. Pro zařazení do kategorie 2 stačí, aby budova měla 8 či více pater, shromažďovalo se v ní větší množství lidí nebo hořlavý materiál. A protože v dnešní době je častá stavba vysokých budov, mnoho obchodních center a bezpočet skladů, spadá do rizikových kategorií čím dál větší množství objektů. S požární bezpečností je tedy proto dobré počítat již při plánování budov, a to vzhledem k jejich cílenému účelu. Zde je potřeba počítat s dobrou dostupností nouzových požárních úniků, kvalitním požárním značením a dobrým rozložením místností a jejich obsahu.

Všem výše uvedeným věcem lze samozřejmě pomoci již před stavbou budovy, a kvalitě a rozmístění požárního značení lze pomoci i u budov hotových. Pro rozmístění požárního značení existuje velké množství doporučení a požárních předpisů, nicméně jedná se o základní sadu pokynů a je možné umísťovat značení navíc. Právě značení navíc může být velký přínos, protože v místech se sníženou viditelností (například prostory zaplněné dýmem) nemusí být všechna značení rychle k nalezení. Část práce se tedy věnuje právě analýze možnosti umístění značení tak, aby byla usnadněna orientace. Protože při opouštění hořícího domu je rychlost evakuace přímo závislá na orientaci, správné umístění bezpečnostního značení může hrát klíčovou roli pro záhranu. Je tedy nutno značení umístit tak, aby byla největší šance jeho nalezení a následování značení bylo díky dobrému umístění usnadněno. Při simulaci se požáry budou šířit v závislosti dostupnosti hořlavin a umístění značení tedy bude odpovídat nejen statickému modelu, ale i na základě předpokládaného šíření požárů a prvků náhody.

Velkým požárním nebezpečím jsou také nesprávně umístěné předměty v místnosti, které tvoří překážky v cestě za bezpečím. Nelze sice požární bez-

pečnosti plně podřizovat účelnost rozmístění, ale na druhou stranu je minimálně dobré mít přehled o míře rizikovosti vzniklé umístěním vybavení po budově. Tomuto tématu se věnuje druhá část práce. Cílem je vytvořit analýzu budovy, a následně navrhovat vylepšení rozmístění vybavení po budově tak, aby klesla míra rizikovosti.

V závěru je vyhodnocení použitých metod a návrhy pro další postup, stejně jako diskuze o rozšiřitelnosti použitých modelů a metrik. Dále jsou též diskutovány možnosti využití těchto analýz v praxi.

Kapitola 1

Cíl práce

Jak již bylo naznačeno v úvodu, cílem této práce je vytvořit sadu analytických nástrojů pro budovy za pomoci algoritmů umělé inteligence a strojového učení. Pro tyto účely je třeba nejprve vytvořit model budovy, na kterém později půjde provádět analýzu, a tento model připravit dostatečně robustní pro všechny potřebné operace a pro všechny následující operace na něm prováděné. Po vytvoření základního modelu je potřeba v budově rozmístit nábytek a další vybavení. Toto lze pro plánované budovy rozvrhovat předem, a pro existující budovy jej lze vytvořit dle skutečnosti. Dříve vytvořený model tedy musí podporovat přidávání nábytku a vybavení. Pro plánované budovy je navíc zajímavé testovat i strojové plánování a optimalizaci místnosti pomocí algoritmů strojového učení. Počítačem vytvořené modely mohou pomoci člověku v navrhování finálního rozmístění nábytky a vybavení a přitom uvažovat i požární bezpečnost (distribuovat potenciální ohniska požáru). Nakonec je možno finální návrh ohodnotit a získat tak představu o míře rizikovosti tohoto návrhu. K tomuto ohodocení může sloužit jednak statická analýza, a navíc také simulace požáru a jeho šíření a simulace evakuace osob při požáru. Jako přidaná hodnota k tomuto ohodnocení lze ještě získat doporučení na umístění prvků požárního značení, což usnadní evakuaci.

1.1 Původ myšlenky

Myšlenka na tuto práci vznikla při požárním cvičení v zaměstnání, kde zaměstnanci zrovna zarovnali východ testovanou technikou, a evakuace byla v tu chvíli zbytečně zdlouhavá. Snaha o vytvoření podobných nástrojů zřejmě zatím není příliš častá, neboť analýza domů z pohledu požární bezpečnosti je v současné době spíše soupisem pravidel pro architekty, návrháře a požární revizní techniky. Po delším hledání bylo zjištěno, že jednotlivé státy mají vlastní sady pravidel a nařízení, stejně jako doporučení pro požární bezpečnost. Navíc strojové vyhodnocování budov pro tyto účely není praxí, a existující programy

nejsou volně dostupné, a nijak zvlášť rozšířené. Cílem je tedy vytvořit otevřenou sadu nástrojů pro analýzu a návrh.

Při studiu internetových zdrojů na dané téma se objevuje zajímavá statistika, která podporuje vizi potřeby analytických nástrojů. Podle této statistiky například v Číně každoročně stoupá ekonomická cena každého požáru, i když ztráty na životech klesají. Například v roce 2010 došlo k 130 tisícům požárů. Navíc ze statistiky vyplývá, že většina z požárů vzniká v obytných domech a ubytovnách, a jen 10% ze všech požárů vzniká v průmyslových budovách. A je velký rozdíl mezi využitím domů, zda se jedná o obytné domy nebo průmyslové komplexy, obě kategorie mají odlišný přístup k vyhodnocování požárních rizik [1]. Z praxe je navíc známo, že obytné domy často mívají špatné nebo dokonce nemají vůbec žádné požární značení. O účelnosti vybavení vzhledem k požární bezpečnosti v tomto případě vůbec nemůže být řeč. Navíc špatná dostupnost prvků požární ochrany může snadno způsobit větší škody a podpořit rychlé rozšíření požáru. Proto by tedy analytické nástroje mohly být vhodným pomocníkem pro bytové architekty, kteří často žijí vytrženi z reality, a zajímají se především o estetickou stránku návrhu.

1.2 Inovativní přístup

Cílem této práce je tedy místo prosté analýzy přidat i doporučení, která by mohla vést k zlepšení požární bezpečnosti. Protože se jedná o první fázi celkové myšlenky, je kategorie možných zlepšení omezena na umístění nouzového požárního značení a dostupnosti prvků požární ochrany. Není předmětem práce do detailu uvažovat materiály a jejich požární vlastnosti, jako spíše se zaměřit na kvantitativní cíle - na základě velkého množství simulací jsou hledána slabá místa, nezávisle na míře pravděpodobnosti vzniku požáru v daném místě. Jako primární zdroj požáru se předpokládá lidská chyba, a požár tedy může vzniknout prakticky kdekoli. Na základě parametrů místnosti se ovšem požár šíří na základě jednoduchého fyzikálního modelu, a tedy jeho šíření už bude částečně (zjednodušeně) reflektovat realitu. Práci tedy je možné rozdělit na 3 hlavní části:

1. navržení modelu pro hořící místnosti, řešení jeho využitelnosti pro různé vyhodnocovací mechanismy
2. řešení optimalizace místností z pohledu požární bezpečnosti
3. vyhodnocení požárních rizik hotového modelu domu, doporučení umístění prvků protipožární ochrany a nouzového značení

1.2.1 Model domu

Model domu musí být jednoduchý, snadno vizualizovatelný a přenositelný mezi různými analytickými platformami. Pro jeho tvorbu je využito integrované

GUI. Od modelu domu se očekává jeho snadné vytvoření a možnost kopírovat stavební plány budov, přehledné zobrazení rizik v rámci budovy a snadnou rozšiřitelnost. Při modelování budov vzniká často problém moderní architektury hojně využívající oblých tvarů a neobvyklých prvků. S tím je potřeba počítat, a podporovat i místnosti v jiných tvarech než je jen obdélník. Navíc je v rámci modelu dobré počítat s možnými budoucími problémy v podobě stavebních materiálů, protože i samotná stavba přináší požární rizika, nejen nábytek v domě umístěný.

1.2.2 Optimalizace rozložení nábytku v domě

Cílem optimalizace je vytvořit sadu možných návrhů rozmístění nábytku v domě tak, aby odpovídalo požadavkům na minimalizaci požárních rizik, a tedy aby bylo dům možné snadno opustit. Je totiž prakticky nutné, aby nábytek nebyl umístěn před únikovými východy a neblokoval možnost úniku nebo zbytečně prodlužoval únikovou cestu. Navíc i samotné rozmístění nábytku může ovlivňovat požární riziko, a přispívat k snadnému rozšíření požáru kupením snadno hořlavých materiálů na jednom místě. Hořlavý materiál je lépe rozložit po domě, čímž se riziko sníží.

Pro optimalizaci domu bude využito optimalizačních algoritmů, jedná se v podstatě o pozměněný problém batohu, tedy o problém NP úplný. Protože cílem optimalizace je dostupnost objektů v místnosti, je velice pravděpodobné, že bude mnoho podobných místností, lišících se o drobné posunutí. Bude proto také řešena otázka, zda využitím shlukování dojde ke zlepšení výsledků, nebo alespoň zmenšení množství relevantních výsledků.

1.2.3 Ohodnocení požárních rizik

Cílem zde je číselné ohodnocení (podobné tabulkovému přístupu) vytvořeného modelu domu, a navíc doporučení míst, kde by bylo vhodné umístit prvky požární ochrany. Ne každé rozložení nábytku se může ukázat jako vhodné, navíc uživatel má vždy poslední slovo a může si rozložení vybrat sám a pouze chtít toto rozložení ohodnotit. Inspirací pro testovací metodiku jsou systémy modelující chování v reálném čase, i zde se totiž jedná o simulaci úniku před vzniklým požárem a následné vyhodnocení této simulace.

Kapitola 2

Rešerše

Pro tuto práci bylo velice těžké sehnat data o existujících programech umožňujících vyhodnocení požární bezpečnosti, protože existující programy jsou chráněné autory. Existující programy se zpravidla nedají stáhnout a vyzkoušet, volně dostupné jsou pouze příručky a zmínky o programech. Tyto existující programy jsou převážně interní programy firem zabývajících se požární bezpečností, nebo různé pomůcky pro státní orgány využívané k auditům. Protože výsledky práce budou uveřejněné a volně k dispozici, i zde se tedy kladen důraz na volně dostupné zdroje.

2.1 FSES

FSES [2] je zastaralý nástroj pro analýzu budov, nicméně poskytující základní potřebná data. Na základě statického modelu vyhodnocuje poskytnutá data, jako je počet místností, počet chodeb, využití místností, informací o sprinklerech, požárním alarmu, interiéru a dalších parametrech. Vyhodnocuje rizikovost podle zadaných parametrů. Protože veškeré vyhodnocování probíhá staticky, jedná se prakticky pouze o integraci požárních tabulek do jednoho programu. Program na svém vstupu vyžaduje tabulkové údaje, na výstupu dává tabulkové hodnoty vyhodnocení. Jako nástroj na požární prevenci je program jistě vhodný, nicméně poskytuje spíše základní údaje pro pojišťovny a kontrolory, spíše než praktické možnosti zlepšení.

Na druhou stranu, tabulkové vyhodnocování má nespornou výhodu v šíři zkoumaných faktů. Zatímco cíl této práce je spíše teoretický, a jeho praktické nasazení vyžaduje ještě další integraci, tabulkové vyhodnocování sice bagatelizuje fakta, zato však zkoumá všechny možnosti. Přínosem je zde jistě zkoumání vedení kouře a jeho detektorů, a také kontrola sprinklerů a protipožárních opatření.

V závěru tento program funguje spíše jako checklist požární bezpečnosti. Úkol prevence požárů tedy plní pouze za spolupráce s odborníky, kteří vyhodnotí výstupy a na jejich základě provedou potřebná opatření. Co je však na

tomto programu nejzajímavější je, že byl vytvořen jako státní zakázka, a tedy že zkoumaná problematika je v praxi skutečně řešena.

2.2 FIERAsystem

FIERAsystem [3] je sada modelů pro vyhodnocování požární bezpečnosti. Jedná se o soubor behaviorálních modelů, které umožňují spouštět analýzu nad nadefinovaným reálným modelem budovy. Zajímavostí na tomto vyhodnocování je široké spektrum modelů: šíření ohně, pohyb kouře, detekci ohně, rozpad budovy, účinnost hasení, reakci požárníků a jejich efektivitu, reakci postižených a jejich evakuaci, rizika pro zdraví, předpokládaný počet obětí, ekonomické ztráty, odhad likvidace následků.

Celkově je tento program navržen zdá se na širokou analýzu požáru. Bohužel modely nejsou dále specifikovány, a není tedy známo, zda se jedná o modely statické nebo dynamické. Vzhledem k datu vydání tohoto nástroje lze však předpokládat, že se jedná o modely statické. Toto nemusí být nutně nevýhoda, protože modely lidského chování, modely hoření a modely budov jsou popsány a zkoumány již dlouhou dobu. Navíc statické modely bývají oproti strojově učeným modelům rychlejší, a netrpí chybou při učení. Nevýhodou statických modelů jsou právě jejich předem určená pravidla, kdežto lidé lépe reagují na vzniké situace a mění přístupy k problémům.

Nevýhodou tohoto programu je však pouhá možnost analýzy, program jako takový nedává žádná doporučení. Výsledky programu tedy stejně jako u předchozí varianty musí zkoumat odborník, a možná opatření jsou opět navrhována manuálně. Další potencionální nevýhodou je implementace na míru kanadskému prostředí, není tedy zaručeno, že modely budou pracovat správně i pro jiné země (možné rozdíly ve složení betonu, použití dřevin, odlišný izolační materiál, jiné vnější podmínky).

Tento framework vypadá velmi zajímavě a slibně vzhledem k požadavkům na požární ochranu, bohužel má jednu podstatnou chybu - není k dispozici ke stažení, a nezdá se ani, že by byl stále podporován. Podle dostupných zdrojů poskytuje velké množství informací, a je tedy rozhodně přínosem v požárné ochraně.

2.3 Tabulkové vyhodnocování

Další, a bohužel nejčastější metodikou vyhodnocování požárních rizik je manuální metoda tužky a papíru, a tabulek. Každá země má svou vlastní metodologii na vyhodnocování rizik (například kanadské [4] vyhodnocování se liší od českého [5]), a je tedy nutné zaměstnávat experta v této oblasti. Obvykle se jedná o desítky stránek, které je potřeba vyplnit, a množství různých tabulek, ze kterých je třeba vybrat nejvhodnější odpověď. Jak je zřejmé, tato metoda bohužel neposkytuje žádná doporučení, a je tedy opět na zvážení experta,

jaký bude další postup. Nejsnáze dostupnými tabulkami jsou materiály pro Kanadu, Česká republika nic takového publikovaného nemá.

Kapitola 3

Analýza

Před snahou o řešení problému je nejprve nutno tento problém pochopit a co nejvíce prozkoumat. O hořících domech víme vždy dva základní údaje: dům byl postaven a může být vybaven, a vznikl v něm požár. Toto samozřejmě není žádné převratné zjištění, ale je to nutný základ, na kterém budeme stavět počátek analýzy. V první řadě je totiž nutno určit, co mohlo být příčínou vzniku požáru. Nikoliv ohně jako takového, ale požáru, tedy ohně rozšířeného mimo kontrolu a bez triviálního způsobu jeho uhašení. Tímto důvodem je hořlavina, která se nacházela v blízkosti jiskry či drobného plamínku. Díky hořlavému prostředí se oheň rozšířil, a stal se z něj požár. Protože však hořlaviny je třeba skladovat, a z hořlavěhé materiálu jsou často tvořeny i domy samotné, není reálně možné v rámci požární prevence hořlaviny odstranit a zajistit tak požární bezpečnost. Nabízí se tedy možnost velké zdroje hořlavin rozmístit po domě tak, aby jich nebylo velké množství na jednom místě, a bylo tedy možné ohně lépe hasit. V tomto může pomoci automatizace rozložení vybavení místnosti, což bude tedy jeden z cílů této práce.

Protože však nejde stoprocentně požárům čelit, je lépe se připravovat i na možnost, že k nim dojde. A pokud k požáru dojde, je třeba uvažovat nad možnými následky. Krom materiálních škod mohou nastat i škody na životech, pokud v budovách v době požáru jsou lidé. A protože lidský život je vzácnější než materiál, práce se dále bude směrovat k možnostem záchrany lidských životů. Pokud jsou v domě lidé, jsou při vědomí nejsou omezováni zdravím a nejsou-li schopni oheň zvládnout, nejsnažší věc kterou mohou udělat je utéct. Mohou se však nacházet v neznámém prostředí, a nemusí vědět, kde najít unikový východ. Pro tuto situaci je vymyšleno a používáno požární značení. Toto značení by mělo být viditelně umístěno, a mělo by navést lidi v neznámém prostředí k nejbližší možnosti úniku. Bohužel se může stát, že toto značení není umístěno dobře, proto nemusí být na první pohled vidět. Navíc toto značení může být na místech, kde ho lidé nebudou čekat nebo kde se bude vyskytovat jen malá část lidí. Smutná pravda je, že stavitelé často snaží ušetřit, a to i na bezpečnostních prvích. Proto se dělají různé audity, zda je

požární značení dostatečné nebo ne. I toto může být příležitost pro automatizaci auditů, tedy automatizované vyhodnocování požárních rizik. Na základě různých výpočtů a simulací lze ohodnotit stávající prvky požární bezpečnosti, případně doporučit nové.

Je tedy vidět, že se jedná o problém opuštění bludiště. Tento problém je již dlouhou dobu zkoumaný a existují různé algoritmy pro navigaci robotů [6] nebo pomocí neuronových sítí [7], ale hořící budova přidává do problému nový rozměr - bludiště se totiž v průběhu času mění, a navíc se stává nebezpečné a omezující. Zatímco v bludišti se předpokládá, že existuje možnost dostat se k východu, zde se tato možnost v průběhu času ztrácí.

3.1 Problematika hořících místností a jejich optimalizace z pohledu požární bezpečnosti

Jak již bylo v úvodu této kapitoly řečeno, první z úkolů je vyřešit distribuci možných ohnisek požáru, aby se rozložilo akutní riziko vzniku. Pro tento účel je nejprve potřeba navrhnout model, nad kterým bude možno optimalizaci provádět. Tento model nemusí stoprocentně přesně reflektovat dům samotný, ale musí poskytovat dostatečnou představu o problému, a také musí umožňovat analýzu a vyhodnocení bez nutnosti přepracování formátu. V první řadě je potřeba pochopit lidské požadavky na software, a očekávání pro tvorbu a zobrazení. Zde je nutno pracovat s určitou mírou abstrakce, protože snažit se postavit dokonalý model bez postupných iterací bývá prakticky nemožné. Proto bude volen v první řadě méně přesný model, ale zato snadno použitelný. Aby však model byl dostatečně vhodný pro vyhodnocení a analýzu, je potřeba pozorumět také nejprve nárokům možných nástrojů k těmto účelům.

3.1.1 Model domu

Pro model domu je třeba vytvořit co nejjednodušší rozhraní, aby bylo možné snadno a rychle přenášet výkresy vytvořené architekty do programu pro vyhodnocení. Protože však různé stavební prvky přinášejí různá rizika, nabízí se dvě možnosti. Buď podchytit všechny možné varianty materiálů a nebo využít určitou míru abstrakce a pracovat s materiály na základě rizikových skupin. I u skupin se však míra rizika bude lišit, protože existují například různé varianty betonu, železobeton, nebo různá dřeva. A tyto různé odchylky v materiálu mohou výrazně měnit teplotu hoření [8]. S dostatečnou mírou abstrakce lze tedy předpokládat, že místo řešení všech různých kategorií dokáže uživatel odhadnout přibližnou hořlavost z dané lineární škály. Kromě údaje o hořlavosti místa je nutno v modelu udat ještě informaci o tom, jak složitý je průchod tímto místem. Místa mohou být trojího typu: volná místa, zdi, předměty. Vybavení je možno sice rozebírat ještě dále a určovat různé typy předmětů, nebo dokonce konkrétní předměty, ale to je zbytečně. Protože v práci je jedním

z cílů opuštění hořícího domu, stačí pouze evidovat, jak moc velkou překážku předmět tvoří. Velké překážky nelze snadno překročit, a osoby v domě budou tedy hledat spíše cestu kolem překážek, než skrz ně. Naopak malé překážky budou osoby pravděpodobně překračovat nebo přelézat, místo aby je obcházely. I zde je tedy jistá míra abstrakce, a opět místo kategorií překážek stačí uvažovat pouze jak velkou překážkou v průchodu předmět tvoří.

3.1.2 Oheň

K některým výpočtům a simulacím se využívá ohně jako dynamické překážky. Oheň má určité ohnisko, tedy místo kde vznikl. V průběhu času se šíří a postupně pohlcuje celý dům. Šíření ohně lze abstrahovat jako náhodný jev. Protože model místnosti pracuje se škálou hořlavosti, je zbytečné dělat fyzikální model hoření. Místo toho stačí využít jednoduchý pravděpodobnostní model. Oheň se šíří na základě zahřátí látky na teplotu vznícení. Materiál se zahřívá přenášením tepla ze svého okolí. V reálném světě však působí více vlivů, které ovlivňují přenos teploty. Nejsnažší přenos je dotekem mezi horkým materiálem a chladným materiálem. Chladný materiál se od horkého ohřeje, ačkoliv tím se zchladí horký. Pokud je horký materiál hořící, vlivem hoření se bude stále ohřívat, a zahřeje i chladný materiál k bodu hoření. Protože ale nedochází vždy ke kontaktu materiálů, hraje roli mnoho dalších vlivů jako třeba teplota okolí nebo proudění vzduchu. Protože tento fyzikální model je složitý a komplexní, a již dříve bylo využito abstrakce, stačí i zde zjednodušit model na šíření ohně na základě teploty zdroje a pravděpodobnosti přenosu. Pokud by byl model shledán jako nedostatečný, lze využít například pokročilého nahodilého modelování [9].

3.1.3 Vybavení domu

Riziko požáru nevybaveného domu je nízké, dané pouze materiálem stavby domu. Protože ale prázdný dům je prakticky k ničemu, je třeba do domu umístit nějaké vybavení. Přidáním vybavení se však zvyšuje riziko požárního nebezpečí. Toto je známo již u staveb, a věnuje se tomu litaratura [10]. Ačkoliv navrhovaný program lze použít i pro stavby, jeho využití je daleko zajímavější pro hotové domy. Totiž například umístěním knihovny do domu vzroste požární riziko, protože se na jednom místě hromadí papír, který je dost hořlavý. Nebo v ještě horším případě skladiště hořlavin zvýší riziko ještě mnohem více. Krom toho že vybavení zvyšuje riziko požáru také navíc zhoršuje pohyb po domě. Pokud je vybavení nevhodně umístěno, blokuje východy, nebo minimálně znesnadňuje přístup k nim. Proto je potřeba k vybavení domu přistupovat hned s ohledem na dvě skutečnosti: úroveň požárního rizika a rozsah překážky. Požární riziko je zřejmé, zvyšuje riziko vzniku požáru, a stejně tak zvyšuje riziko šíření požáru. Protože však není cílem hodnotit vybavení jen na základě požárního rizika, ale je třeba i řešit evakuační plán, je nutno řešit i jak

moc překážka znesnadňuje pohyb. Překážky mohou být různých typů, některé překážky mohou být nízké, některé vysoké, s některými se dá pohnout, jiné se dají prolézt, některé naopak zůstávají na místě. Protože takových kategorií by se dalo vymyslet větší množství, je tato kategorizace opět převedena na lineární škálu určující obtížnost překážky. Ve výsledku uživatele zajímá riziko dané překážky, nikoli její charakter. Samozřejmě by se dalo uvažovat že překážkami může jít posouvat a tím budou blokovat okolní místa, ale i tuto vlastnost lze zachytit při klasifikaci překážky, či při jejím zadávání do schématu.

3.2 Rozbor algoritmů vhodných pro řešení dílčích problémů

Systém vyhodnocování hořících domů se skládá ze sady nezávislých skriptů pro jednotlivé funkce systému, tedy vytvoření modelu domu, rozmístění nábytku a jeho optimalizace, vyhodnocení požární bezpečnosti pro použitý model a simulace požáru. Prvotní část systému bude uživatelské rozhraní pro vytvoření modelu domu a definice požadovaného vybavení domu. Vybavení by mělo být možné umístit na předem požadované místo, nebo jej nechat v inventáři, ze kterého bude rozmístěno automaticky. K dispozici bude i možnost částečného vybavení a naplnění inventáře pro dodělání kompletní místnosti pomocí automatického rozmístění. Dále bude součástí systému sada skriptů pro optimalizaci budovy (vyžaduje předměty v inventáři) a pro vyhodnocení požárních rizik a získání doporučení k umístění bezpečnostních prvků. Protože se jedná o systém typu PoC, jednotlivé části do sebe nebudou integrované, ale bude možné je spouštět nezávisle. Předem vytvořené vzorové modely tedy bude možno snadno vyhodnotit na požadované výstupy nezávisle na ostatních komponentách. Protože se bude jednat o nezávislé komponenty, neexistuje přímé workflow pro práci s tímto nástrojem, pouze doporučený postup. Pro demonstraci celého nástroje je:

1. vytvořit prázdnou budovu
2. naplnit budovu pomocí nástrojů pro optimalizaci nebo manuálním vkládáním
3. vyhodnotit vytvořený model (prázdný nebo naplněný)

3.2.1 Nástroj pro tvorbu modelu domu

Návrh budovy bude probíhat na čtvercové mřížce o předem určených rozměrech, s omezením na jedno patro budovy. V modelu budou černou barvou zaznamenány zdi, bílou volná plocha, zelené budou bezpečné zóny (východ, možné nouzové východy). Po uzavření budovy dojde k jejímu automatickému zarámování do černé barvy, a uvažována bude pouze bílá plocha. Za spojité

zdi se budou považovat vodorovně či svisle sousedící černé čtverce, diagonálně sousedící souvislé nejsou. Navíc bude možnost naznačit bezpečná a nebezpečná místa v místnosti, a to pomocí červené (nebezpečné) a modré (bezpečné) barvy. Značící barvy budou v několika odstínech, a budou určovat míru bezpečí či rizika. Na základě těchto barev bude při generování požárových scénářů volen počátek scénáře. Pokud nebudou značící barvy použity, budou všechna bílá místa vyhodnocována se stejnou měrou rizika.

Druhou fází návrhu budovy bude rozmístění předmětů. Předměty budou umísťovány na plochu domu, nebo budou umísťovány do inventáře pro automatizaci jejich rozmístění. Automatické rozmístění bude řešeno v separátním kroku. Umisťování předmětů bude dle možností řešeno metodou drag & drop nebo umístěním levého horního roku objektu na plochu domu, alternativně možností objekt modelovat po čtvercích. Objekty budou škálovatelné, aby snadno vyhověly měřítku plánovaného domu. Objekty navíc mohou mít různé parametry (hořlavost, vysoká překážka). Pro další vyhodnocování navíc mohou mít určenu i minimální šířku vyžadovaného volného prostoru v různých směrech, aby bylo možné s objekty manipulovat. Například jídelní stůl musí být dostupný ze všech stran, zato psací stůl musí být dostupný jen z jedné.

Tento nástroj bude umožňovat uložení v textovém formátu nebo matici pro vyhodnocení, a bude umožňovat oba formáty načíst. Pro účely GUI nebude využito žádných algoritmů strojového učení, a GUI bude mít pouze estetický efekt.

3.2.2 Nástroje pro optimalizaci rozložení nábytku v domě

Pro optimalizaci místnosti slouží soupis předmětů v inventáři a matice pro vyhodnocení budovy. Jedná se o obdobu problému batohu, jen ve dvojrozměrné ploše. Řešení těchto úloh je již dlouhodobě velký výzva, všeobecně se k nim však využívá některý z optimalizačních algoritmů [11]. Jedná se již na první pohled o NP-úplný problém, a je řešitelný základními algoritmy pro tuto sadu problémů. Protože NP-úplné problémy mají velice rozsáhlý stavový prostor, řešení optimalizace bývá často velice zdlouhavé. Proto mohou být přidávány například hranice pro dobu trvání jednotlivých algoritmů, a očekává se vysoká paměťová náročnost.

Seznam metod k použití pro optimalizaci (zkoumané metody jsou označeny *):

- Prohledávání do šířky *
- Prohledávání do hloubky *
- Metoda větví a hranic *
- Metoda simplexu
- Řízené náhodné prohledávání

- Horolezecký algoritmus
- Simulované žíhání
- Diferenciální evoluce
- Evoluční prohledávání
- SOMA - samoorganizující stěhující algoritmus

Na problém se díváme jako na stavový prostor všech možných řešení, která musíme vyhodnotit. Jedná se tedy o stromovou strukturu plochy, a postupného umisťování předmětů do ní. Dívat se na problém na úrovni jednotlivých čtverců je zbytečně pracné, a neposkytuje to dobrou představu, protože umisťované předměty jsou kompaktní a není možné je dělit. Vznikalo by tak spoustu možností chybně umístěnych předmětů. Sledovat místo toho rozmístění objektů na ploše se jeví jako lepší možnost, nebude totiž vznikat fragmentace objektů, a půjde snadno omezovat neplatné stavy (překrytí objektů). Prohledávaný stavový prostor je tedy strom umístění jednotlivých předmětů.

Pro usnadnění a urychlení možnoství vyplnění předmětů na plochu se jeví jako nejlepší varianta nejprve požadované objekty seřadit podle velikosti. Velikostí rozumíme celkovou plochu objektu. Navíc se může stát, že uživatel zadal do inventáře příliš mnoho objektů, a na připravenou plochu se nevejdou. Takovou situaci bude nutno detekovat a uživateli ohlásit.

Pokud se zamyslíme nad možnými výsledky algoritmů, zjistíme že jich bude velké množství podobných. Protože není dobré uživatele přehltit množstvím výsledků, je na finální sadu výsledků aplikováno shlukování, a předváděny jsou jen reprezentativní vzorky.

3.2.2.1 Prohledávání do šířky

Prohledávání do šířky [12] je jeden z nejzákladnějších algoritmů. Prohledávání stavového prostoru začíná nejprve vybírat polohu velkých předmětů, a až následně umisťuje drobné předměty do okolí. Na první pohled by se mohlo zdát, že tato varianta bude nejlepší, a není potřeba dalších metod, ale nevhodné umístění velkých předmětů může zablokovat prostor pro předměty malé, takže není zaručen kvalitní výsledek, ba dokonce není zaručen ani úspěch této metody. Prohledávání do šířky prohledává stavový prostor úplně, a je tedy zaručeno, že pokud existuje ve stavovém prostoru přípustný výsledek, tato metoda jej najde.

Pseudokód prohledávání do šířky:

1. Do fronty přidat všechny sousedy prohledávaného stavu
2. Odebrat první prvek z fronty

3. Vyhodnotit odebraný prvek
4. Označit prvek jako prohledaný
5. Přidat do fronty dosud neprohledané sousedy
6. Opakovat od bodu 2) dokud není zásobník prázdný

3.2.2.2 Prohledávání do hloubky

Prohledávání do hloubky [13] je opačný případ k prohledávání do šířky. V prohledávaném stavovém prostoru se vybere jedna z cest, projde se na její konec, a následně se dělají drobné změny, v rámci kterých se prohledává nejbližší okolí. V podstatě se jedná o prohledávání od konce. Budeme-li předpokládat seřazení předmětů podle velikosti, největší předměty se umístí jako první, a změny se budou provádět převážně na úrovni drobných předmětů. Stejně jako v případě prohledávání do šířky se jedná o úplné prohledávání, a tato metoda tedy zaručuje nalezení přípustného výsledku.

Pseudokód prohledávání do hloubky:

1. Do zásobníku přidat všechny sousedy prohledávaného stavu
2. Odebrat první prvek ze zásobníku
3. Vyhodnotit odebraný prvek
4. Označit prvek jako prohledaný
5. Přidat do zásobníku dosud neprohledané sousedy
6. Opakovat od bodu 2) dokud není zásobník prázdný

Jak je vidět z pseudokódu, prohledávání do hloubky se od prohledávání do šířky liší jen v typu použitého úložiště pro další prohledávání.

3.2.2.3 Metoda větví a hranic

Metoda je založena na prohledávání stavového prostoru obdobně jako předchozí metody, ale navíc stanovuje meze, které omezují možnosti prohledávání [14]. Pokud je nějaký podstrom stavového prostoru jednoznačně horší než již nalezené řešení, je zbytečné ho prohledávat. Problém je, že zatímco u problému batohu je jasně definovaná podmínka úspěchu, optimalizace místnosti nemá předem jednoznačně definovaný úspěšný formát, a správných výsledků může být mnoho.

Pseudokód větví a hranic:

1. Za použití heuristiky vyhledat nějaké řešení
2. Pro vyhledané řešení uložit pomocné proměnné indikující hranice
3. Při průchodu stavovým prostorem k možným řešením vždy vyhodnotit za využití pomocných proměnných množinu možností, určit zda tato řešení mohou být lepší než aktuálně existující nejlepší řešení
   - Pokud ano, vyhodnotit aktuálně prohledávané řešení, a pokud je lepší aktualizovat referenční řešení
   - Pokud ne, odříznout tuto větev řešení jako nepravděpodobnou
4. Opakovat od bodu 2 dokud existují neprohledané větve řešení

Použitá heuristika může být jednoduché prohledání do hloubky, které samo o sobě je pomalé stejně jako prohledávání do šířky. Za pomoci metody větví a hranic může být toto prohledávání několikanásobně urychleno, protože na první pohled nevhodné větve mohou být oříznuty hned, bez nutnosti prohledat všechny možnosti v nich.

Tato metoda skrývá velkou výzvu pro daný problém, protože není zcela jednoznačné, jak budou navržené pomocné metriky účinné pro ořezávání stavového prostoru, či zda dokonce nezpůsobí problémy s vyhodnocením. Mohlo by se snadno stát, že podle aplikovaných omezení bude zahozena větev obsahující nejlepší řešení, které ale nebylo dle omezení přímo patrné.

3.2.2.4 Shlukování

Protože neexistuje pouze jedna správná možnost, ale bude jich více, zůstává otázka, jak vybrat reprezentativní vzorky, na kterých poté může být provedena analýza. Jedním řešením je samozřejmě nechat uživatele, ať si vybere sám, ale vzhledem k teoretickému množství různých kombinací je toto řešení nevhodné. Nebo spíše, bez předchozího filtrování je nevhodné. Proto je potřeba množinu přípustných výsledků nějakým způsobem přefiltrovat. Protože výstupní data jsou ve formátu matice, je možné snadno aplikovat shlukovací algoritmy, a hledat reprezentativní vzorky matic. Na základě podoby reprezentativních vzorků si následně může uživatel vybrat jemu nejvíc vyhovující, nebo může otestovat všechny vybrané vzory.

Při shlukování [15] se využívá podobných vlastností zkoumaných objektů pro určení jejich podobnosti. V případě zkoumané matice to bude zaplnění matice objekty, bez nutnosti mezi objekty rozlišovat. Z hlediska problému je zajímavý spíše volný prostor mezi objekty než objekty jako takové, i když z hlediska míry hořlavosti bude výměna objektů tvořit velké rozdíly. Ve výsledku bude tedy nutno vyhodnocovat trojrozměrnou matici, dva rozměry budou určovat obsazení pole, a třetí rozměr bude určovat pohled na skutečnost

(překážky, hořlavost). Z pohledu překážek tedy bude záležet čistě jen na tvarech, a hledání podobností bude na základě poměru volné plochy ku ploše obsazené. Z pohledu mapy hořlavosti se bude jednat o porovnání existujících matic, a výsledkem může být rozdílová matice mapující změny hořlavosti.

Pro shlukování se používá velké množství algoritmů, příklady:

- K-means
- EM shlukování
- AP (affinity propagation)

Pro určení podobností matic můžeme použít například následující vzdálenosti:

- L2 norma
- Frobéniova norma
- Sloupcová norma
- Řádová norma
- Maximová norma

Algoritmus K-means pro shlukování:

1. Určit požadovaný počet shluků
2. Náhodně zvolit pozice středů
3. Určit příslušnost prvků k středům na základě vzdáleností od středů (nejbližší vyhrává)
4. Na základě prvků ve shluku posunout střed shluku
5. Opakovat od bodu 3 dokud neproběhne určený počet opakování, nebo dokud přiřazení prvků ke shlukům přináší změny

Algoritmus EM:

1. Inicializovat sadu parametrů Θ na náhodné hodnoty
2. Dopočítat matici Z za pomoci matice Θ
3. Použít matici Z k opravě matice Θ

4. Opakovat od bodu 2 dokud nedojde ke stagnaci

Algoritmus AP:

1. Inicializace matic Zodpovědnosti (Z) a Dostupnosti (D) nulami, určení podobnostní funkce (P)
2. Určení požadované podobnosti pro ukončení algoritmu
3. $R(i,k) \leftarrow P(i,k) - \max\limits_{l \neq k} \Big(D(i,l) + P(i,l) \Big)$
4. $D(i,k) \leftarrow \min \Big(0, Z(k,k) + \sum\limits_{j \neq \{i,k\}} \max \Big(0, Z(j,k) \Big) \Big)$ pro $i \neq k$
5. $D(k,k) \leftarrow \sum\limits_{j \neq k} \max \Big(0, Z(j,k) \Big)$
6. Opakovat od bodu 3 dokud nedojde k dosažené požadované podobnosti v určeném počtu kroků

3.2.3 Ohodnocení požárních rizik a návrh umístění požárního značení

Ohodnocení požárních rizik je testováno simulací různých situací požáru pro vytvořený model domu. Pro simulaci požáru byly vybrány metody optimalizační, konkrétně Optimalizace mravenčí kolonií (ACO), Optimalizace rojem částic (PSO), Optimalizace hejnem světlušek (GCO). Cílem těchto metod je optimalizovat únikovou cestu z náhodně vybraných míst v domě, aniž by došlo k popálení či uhoření simulovaného agenta. Všechny tyto algoritmy jsou určené k hledání minimálních či maximálních optim. Je tedy potřeba upravit hodnoty v prohledávaném domě tak, aby cílová místa měla hodnoty minim či maxim, a optimalizovala se cesta k nim. Protože všechny algoritmy pracují s prohledáváním okolí, mohou značky požární bezpečnosti být převedeny na hodnoty v prohledávaném prostoru. Pro rozmístění značek požární bezpečnosti je použit genetický algoritmus, který využívá zpětné vazby z optimalizačních algoritmů, a vyhledává nejlepší možnosti umístění značení. Genetický algoritmus připravuje umístění značení, které ovlivní směrování optimalizačních algoritmů. Pro různé scénáře se zkoumá rychlost konvergence optimalizačních algoritmů, a na základě toho se měří úspěšnost připraveného rozmístění.

Alternativní přístup je vytvořit vyhodnocení pomocí neuronové sítě, a zkoumat, zda se naučí úspěšně rozpoznat dobré a špatné scénáře. Jako referenční řešení budou sloužit člověkem navržená, ale také generovaná předchozí metodou. Protože však není možné vygenerovat velké množství kvalitních učících materiálů, je potřeba učit neuronové sítě pomocí metody posilovaného (zpětnovazebního) učení. Touto metodou je možno učit neuronové sítě i na malém

množství vstupních dat, protože nedostatek ohodnocených trénovacích dat nahrazuje pomocnými ukazateli průběhu učení. Oproti metodě učení s učitelem, kdy předem známe přesné ohodnocení a snažíme se pouze upravit klasifikátory tak, aby odpovídaly požadovaným výstupům, podporuje tato metoda větší míru autonomního prohledávání sítí, a pouze označuje kvalitu celkového výstupu. Posilované učení je možné popsat následujícícm Markovským procesem:

1. Množina prostředí a stavy agenta S
2. Množina akcí A agenta
3. $P_a(x, y) = Pr(x_{t+1} = y \| x_t = x, a_t = a)$ je pravděpodobnost přechodu ze stavu x do stavu y při akci a
4. $R_a(x, y)$ je okamžitá odměna při přechodu z x do y při akci a
5. Pravidla popisující co agent pozoruje

3.2.3.1 Optimalizace mravenčí kolonií

Optimalizace mravenčí kolonií, neboli Ant colony optimalization (ACO) [16] je metoda používaná od roku 1992 a založená na reálném chování mravenčí kolonie. Prvotní cíl tohoto algoritmu bylo hledání optimálních cest v grafu, později se jeho aplikace rozšířila na širší třídu numerických problémů. Myšlenka algoritmu vychází ze základního komunikačního prostředku pro mravence - feromonu. Tento feromon je mravenci vypouštěn na zem, a informuje ostatní mravence (mimo jiného) o cestě. Feromon je chemická látka a časem degraduje až úplně zmizí, neprojde-li po stejné cestě jiný mravenec, který by jej obnovil. Naopak frekventovaná cesta je přesycena feromonem a je tedy pro ostatní mravence velmi přitažlivá. Tento princip se přenáší do počítačové techniky, kdy mravenci při prohledávání okolí při svém pohybu umisťují feromon, a směrují tím ostatní mravence na stejnou cestu. Ve svém primárním určení byly tímto algoritmem prohledávány grafy. Mravenec se tedy v jednom kroku vyskytoval v nějakém vrcholu grafu, a náhodně procházel po hranách do vrcholů ostatních. Při své cestě pokládal feromonové značky na cesty, které ostatní mravenci mohou s jistou pravděpodobností následovat. Tato pravděpodobnost se zvyšuje s tím, kolik feromonu na dané cestě je umístěno, a tedy s tím kolik mravenců již po této cestě prošlo. Pokud ale existuje kratší cesta než doposud nalezená, je stále šance, že po ní nějaký mravenec přejde, protože stoprocentní pravděpodobnost následování feromonové cesty nesmí nastat. Protože mravenec prošel po nové, kratší cestě, na kratší cestu umístil feromonové značky. Protože je omezený počet mravenců v systému, kratší cestu mravenci udržují, a z delší cesty pomalu mizí feromon, takže i když ji mravenec projde je menší šance její obnovy. Delší cesta tedy znamená vysoké nároky na obnovu, protože

je méně často obnovována (projít ji trvá delší dobu). Kratší cesta je obnovována častěji, protože umožní průchod více mravenců. Je tedy důležité správně vyvážit poměr počtu mravenců v systému a rychlost degradace feromonu.

Algoritmus ACO:

1. Pro všechny mravence:
   a) Vytvořit seznam možností
   b) Aktualizovat seznam podle feromonových stop (zajistit virtuální hodnotu i pro cesty bez feromonu)
   c) Vybrat jednu z možností (nelze použít 1 z N výběr, nutno použít výběr na základě pravděpodobností)
   d) Posunout mravence dle výběru
   e) Aktualizovat mapu feromonových stop v příštím cyklu po trase výběru
2. Aktualizovat staré feromonové stopy (degradace)
3. Opakovat od kroku 1 dokud nejsou splněny podmínky ukončení

Pravděpodobnost přesunu k-tého mravence hranou (x,y): $p^k_{x,y} = \frac{\tau^{\alpha}_{x,y} * \eta^{\beta}_{x,y}}{\sum_{z \in povolenéy} \tau^{\alpha}_{x,z} * \eta^{\beta}_{x,z}}$ $\tau_{x,y}$ je hodnota feromonu na hraně (x,y), α je parametr ovlivňující vliv feromonu na přechod, $\eta_{x,y}$ je žádanost přechodu (například ovlivněná délkou hrany), β je parametr ovlivňující vliv žádanosti přechodu.

Ovlivnění feromonu se vypočítává podle vzorečku: $\tau_{x,y} \leftarrow (1-\rho)\tau_{x,y} + \sum_k \Delta\tau^k_{x,y}$ $\tau_{x,y}$ je hodnota feromonu na hraně (x,y), ρ je koeficient degradace feromonu, $\Delta\tau^k_{x,y}$ je hodnota feromonu uloženého k-tým mravencem

3.2.3.2 Optimalizace rojem částic

Optimalizace rojem částic, neboli Particle swarm optimalization (PSO) [17], probíhá na základě přírodního modelu pohybu ptactva za potravou. Tento model je asi nejméně podobný svému přírodnímu předchůdci, protože zatímco ptactvo se pohybuje ve spojitém světě, model pracuje s diskrétní veličinou. V případě ptactva tedy dochází k rychlé konvergenci k požadovanému výsedku, zatímco v modelu je optimalizace pomalejší, způsobená diskrétním pohybem. Každý agent-částice je definován svým směrem, rychlostí a pamětí nejlepšího nalezeného výsledku. Na základě těchto údajů se agent posune na příští vyhledávané místo, a zde provede hledání. Podle úspěšnosti hledání agent upraví svůj směr a rychlost. Jako optima se vyhodnocují místa s největší koncentrací agentů po předem určeném počtu kroků, nebo na základě konvergence agentů.

Popis algoritmu PSO:

1. Vytvoření agentů-částic s přiřazenou polohou pro každého agenta i v prohledávaném prostoru x_0^i, vygenerovaným vektorem rychlosti v_0^i a určenou jejich aktuální pozicí jako nejlepší v paměti označenou jako p^i. Všechny náhodné volby jsou určeny na základě limitů prohledávaného prostoru a rovnoměrného rozdělení $U(a, b)$ v rozmezí od a do b.

2. Každý agent navíc vygeneruje množinu komunikujícího okolí L^i, která ho bude ovlivňovat (náhodný výběr ze všech agentů vyjma sebe sama).

3. Pro každého agenta se určí nejzajímavější optimum $g^i = \max_{j \in L}(p^i, p^j)$

4. Pokud platí že $g^i = p^i$ pak se vektor rychlosti agenta i upraví jako $v_t^i = w * v_{t-1} + U(0, c) * (p^i - x_t^i) + U(0, c) * (g^i - x_t^i)$, kde w a c jsou koeficienty pozorovací schopnosti

5. Pokud rovnice neplatí, vektor rychlosti agenta je určen zkráceným vztahem $v_t^i = w * v_{t-1} + U(0, c) * (p^i - x_t^i)$

6. Následně se aktualizují pozice agentů v systému na základě vztahu $x_t^i = x_{t-1}^i + v_t^i$. Zde je možno aplikovat omezení systému, a agenty mimo hranice posunout přesně na hranici, a jejich rychlostní vektor nastavit na 0

7. Pokud je aktuální pozice agenta lepší než jeho zaznamenaná nejlepší pozice, aktualizuje se jeho nejlepší pozice jednoduše jako $p^i = x_t^i$

8. Dokud nedojde k podmínkám zastavení systému opakuje se od kroku 2 celý algoritmus

3.2.3.3 Optimalizace hejnem světlušek

Optimalizace hejnem světlušek, v originále glowworm swarm optimalization (GSO) [18] je další algoritmus inspirovaný přírodou. Opět se jedná o algoritmus hledající globální optima. Hlavní hybnou silou pro světlušky je hladina svítivosti, určená tzv. luciferiem. Světluška se vydává za jasným světlem, málo jasné ignoruje. Pokud nevidí světlo žádné, stojí na místě. Algoritmus počítá s omezeným dohledem světlušky, a naopak nepočítá s fyzikálním modelem světla (degradace viditelnosti). Díky omezenému dohledu je možno tento algoritmus paralelizovat, protože jsou minimální komunikace mezi agenty-světluškami.

Popis algoritmu GSO:

1. Náhodně se rozmístí světlušky

2. Každá i-tá světluška nastaví svou hladinu luciferia dle funkce $l_i(t) = l_i(t-1)*\rho+\gamma*f(x_i(t))$, kde l je hodnota luciferia, t je čas, ρ je koeficient rozpadu luciferia, γ je koeficient excitace luciferia, $x_i(t)$ je umístění i-té světlušky v čase t a f je funkce určující zajímavost místa (tzv. účelová funkce)

3. Každá i-tá světluška se rozhlédne po okolních světluškách, které mají vyšší hladinu luciferia na základě vzorce $N_i(t) = \{j : 0 < \|\|x_j(t) - x_i(t)\|\| < d_i(t) and l_i < l_j; j \neq i\}$ kde $d_i(t)$ představuje dohled i-té světlušky v čase t a $\|\|x_j(t) - x_i(t)\|\|$ je norma vzájemné polohy světlušek i a j

4. Z množiny N si i-tá světluška náhodně vybere jednu j, kterou bude následovat, přesun k ní odpovídá vzorečku $x_i(t) = x_i(t-1) + k\left(\frac{x_j(t-1)-x_i(t-1)}{\|\|x_j(t-1)-x_i(t-1)\|\|}\right)$

5. Světluška si upraví svůj dohled dle rovnice $d_i(t) = \min\left\{k, \max\{0, d_i(t-1) + \beta*(s - \|N_i(t)\|)\}\right\}$ kde $\|N_i(t)\|$ je mohutnost množiny N z kroku 3 pro i-tou světlušku, s je parametr ovlivňující počet ideálně pozorovaných světlušek v okolí, β je magická konstanta pro počet sousedních světlušek

6. Opakování od kroku 2 dokud neskončí předepsaný počet iterací

7. Shluky světlušek jsou považovány za optima

3.2.3.4 Genetický algoritmus

Genetický algoritmus, genetic algorithm (GA) [19], stejně jako předchozí optimalizační algoritmy vychází z reálného vzoru v přírodě. Tento algoritmus vychází z evoluce samotné. Zatímco však genetika jako taková vychází prakticky z ničeho, a dlouhodobě přidává a upravuje vzorky k dokonalejšímu modelu, počítačový genetický algoritmus pracuje s řešením problému. Počítač náhodně vygeneruje různá řešení problému, a jejich následnými kombinacemi zkouší najít řešení optimální. Toto optimální řešení samozřejmě vždy nemusí kombinacemi najít, vše totiž záleží na náhodnosti generování různých kombinací, počátečních vzorcích a času věnovanému tvorbě různých řešení. Protože se jedná o počítačové řešení, jedinci se typicky pohybují v diskrétním prostoru, a nesou si genetickou informaci reprezentující řešení dané úlohy. Pro výběr jedinců se typicky používá výběrová funkce, náhodný výběr nebo náhodný ovlivněný výběr.

Obecný popis genetických algoritmů:

1. Vytvoření počáteční generace

2. Výběr jedinců z generace na základě zvolené metody výběru, zahození nevybraných jedinců
3. Vytvoření nových jedinců v generaci, generované pomocí základních elementů:
    - Křížení - kombinace dvou jedinců v nové
    - Mutace - úprava genetické informace jedince k vytvoření odchylky, umožňuje prozkoumat okolí jedince
4. Vypočtení zdatnosti jedinců na základě hodnotící funkce, označované jako fitness
5. Kontrola podmínek zastavení algoritmu, jinak opakování od bodu 2
6. Výběr nejzdatnějšího jedince jako finálního řešení

Metody výběru jedinců:

- Náhodný výběr - jedinci se vybírají na základě rovnoměrného rozdělení
- Turnajový výběr - vybere se náhodná skupina jedinců, nejlepší z ní vyjde jako vítěz do finálního výběru, opakuje se dokud není finální výběr dostatečně velký
- Vážená ruleta - podobné náhodnému výběru, ale jedinci se vybírají s přihlédnutím k jejich kvalitě - fitness
- Elitismus - nejlepší zůstanou
- Stárnutí - staří uvolňují místo mladým

Různé metody křížení:

- Jednobodové - určí se jeden bod, před ním se genetická informace bere z jednoho rodiče, za ním z druhého
- Vícebodové - střídá se získávání informace od jednoho, následně od druhého rodiče
- Na základě rovnoměrného rozdělení - pro každý prvek genetické informace se urči, z kterého rodiče se bude brát

Různé metody mutací:

- Přehození bitu

- Náhodná hodnota jednotky genetické informace
- Záměna dvou jednotek informace
- Permutace více jednotek informace
- Zrcadlení více jednotek informace

U genetických algoritmů hrozí zaseknutí v lokálním optimu, je tedy vhodné rozšířit algoritmus o prevenci proti takovému zaseknutí. Různé možnosti pro taková řešení jsou:

- Metoda ostrovů - pracuje se s více nezávislými populacemi, jednou za čas si tyto populace vyměni část jedinců
- Katastrofa - část populace se odstraní, nahradí náhodně vygenerovanými jedinci
- Změna selekčního tlaku - pokud se populace ustaluje, zvyšuje se míra mutace

3.2.3.5 Neuronové sítě - MLP

Vícevrstvé perceptrony, multilayer perceptron (MLP) [20], se skládají z neuronových jednotek známých jako perceptrony. Perceptrony jsou matematické modely reálnách neuronů. Jejich přírodní varianta se skládá z neuronových vstupů, filtračních dendritů, neuronové buňky porovánávající sílu všech vstupů vůči své referenční hodnotě, a konečně z axonu produkujícího výslednou hodnotu stavu neuronu. Perceptrony jsou tvořeny odpovídajícími prvky: vstupními jednotkami opatřenými vahou daného vstupu, jednotkou počítající sumu všech vstupů, prahovou hodnotou určující referenční hodnotu vybuzení vstupu a konečně přechodovou funkcí určující přímo hodnotu výstupu. Funkce perceptronu je tedy dobře popsatelná matematickým vztahem $out = activation\Big(\sum_{i \in vstupy} (in_i * weight_i) - bias\Big)$. Výstup jednoho neuronu tvoří přímku v prohledávaném prostoru. Základním problémem neuronu je jeho nastavení, protože pro malé sítě sice lze nastavení neuronů odhadnout lidsky, ale není to dobré řešení. Správné řešení je naučit neuron, aby dělal požadovanou práci. Je tedy potřeba aplikovat učící algoritmus.

Aktivační funkce:

- Skoková funkce - 0 pro neaktivní neuron, 1 pro aktivní neuron
- Znaménková funkce - -1 pro neaktivní neuron, 1 pro aktivní
- Lineární funkce - výstup je lineární modifikace vstupů

- Sigmoidální funkce - více odpovídá přírodnímu modelu, není skoková změna, dobře derivovatelná, rozsah 0 až 1
- Hyperbolický tangens - podobná sigmoidální fci, ale v rozsahu -1 až 1
- ReLU - lineární funkce x^+

Učící algoritmus perceptronu:

1. Náhodné vygenerování vah pro jednotlivé vstupy a prahové hodnoty
2. Na základě známé výstupní hodnoty d_i pro vstup z trénovací sady $x_i \in X$ vypočítat výstup out_i dle vztahu uvedeného výše
3. Upravit váhy pro vstupy dle $weight_j^{t+1} = weight_j^t + (d_i - out_i) * x_{i,j}$ pro všecchny vstupy j perceptronu pro vstup i
4. Opakovat učení dokud $\frac{1}{|||X|||} \sum_{j=1}^{|||X|||} \|d_j - out_j(t)\| < \gamma$, kde γ je uživatelem zvolená maximální přípustná chyba, nebo dokud nedojde ke stagnaci

Princip učení MLP - zpětná propagace, gradient descent:

1. Vypočítat výstupní hodnotu sítě
2. Určení chyby jednotlivých výstupů $E_k = \frac{1}{2}(d_{i,k} - out_k)^2$ pro všechny výstupní neurony k, celková chyba $E_{total} = \sum_k E_k$
3. Pro určení dílčích chyb se uvažuje derivace celkové chyby vůči konkrétní váze $\frac{\partial E_{total}}{\partial weight_l}$ pro všechny váhy l. Tato derivace se jinak také nazývá gradient vůči váze. Pro určení gradientu vůči váze se používá tzv. pravidlo řetězu, kde se postupuje přes všechny stupně až ke konkrétní váze
4. Takto vypočítaná hodnote se přičte k aktuální váze, a tím se její hodnota upraví. Běžně se nepočítá s plnou hodnotou, ale používá se ještě η násobitel, kterým se omezuje přímý vliv na učení. Tento násobitel se nazývá učící koeficient, a v základní podobě má hodnotu 0,5
5. Po úpravě výstupní vrstvy je nutno upravit skryté vrstvy a vstupní vrstvy. Algoritmus se používá stejný, počítá se stále s celkovou chybou, ale tentokrokrát se výpočet omezuje pouze na danou vrstvu
6. Pro každou iteraci učení se opakuje od kroku 1

3.2.3.6 Neuronové sítě - RBF

Radiálně bázické funkce, radial basis function (RBF) [21] nacházejí své uplatnění u dalšího typu neuronové sítě. Touto sítí je radial basis function neural network (RBFNN), která je oproti vícevrstvé perceptronové síti striktně určena. Určeny jsou počty vrstev v této síti, a její základní struktura. První vrstva sítě je vstupní, jedná se prakticky jen o vstupní data. Následně je skrytá vrstva RBF neuronů, kde každý neuron je určen svým centrem c_i. Poslední vrstvou je lineární výstupní vrstva, která určuje váhy neuronů z prostřední vrstvy a počítá jejich vážený součet. Tento druh neuronových sítí má obvykle rychlejší křivku učení a bývá úspěšnější při hledání nových stavů, navíc je snadno rozšiřitelný pro evoluci neuronové sítě. Výpočetní funkce

pro RBF: $out_j = \sum_{i=0}^{|||N|||} weight_{i,j}\rho(|||in - c_i|||)$, kde j je index výstupního neuronu a ρ je funkce radiální báze. Typicky se používá Gaussovská funkce $\rho(|||in - c_i|||) = \exp\Big(-\beta|||in - c_i|||\Big)$. Parametry $weight_{i,j}$, c_i a β se určují tak aby funkce ρ odpovídala dobře datům.

Učení RBF:

1. Z učících vzorků se vyberou reprezentativní středy, které se použijí jako množina C. Tento výběr je buď náhodný, nebo na základě výpočtu shluků

2. Úprava váhových koeficientů tak, aby výstupy skryté vrsty odpovídaly lineárnímu modelu na výstupu. Toho se docílí implementací rovnice $K(w) = \sum_{t=1}^{T}[y(t) - out\Big(in(t), w\Big)]$, kde $y(t)$ je očekávaný výstup pro daný testovací vstup

3.2.3.7 Neuronové sítě - RNN

Rekurentní neuronové sítě, recurrent neural network (RNN) [22] jsou neuronové sítě se zvláštní organizací tvořící orientovaný kruh. Kruh v rekurentních neuronovývch sítítch tvoří zpětnou vazbu, takže sít je schopná reagovat nejen na svůj aktuální stav, ale i na výsledek předchozího stavu. Toto pomáhá sítím pracovat s pamětí, a tedy reagovat přesněji na sekvenci vstupů, nejen na jednotlivé vstupy. Z toho důvodu mají rekurentní sítě potenciál dosáhnout lepších výsledků při práci s sekvenčními daty. Základním stavebním kamenem této sítě je neuron podobný perceptronu, pouze s rozdílem možnosti zpětné vazby. Perceptronová a Radiálně bázická síť jsou obě sítě dopředné, rekurentní síť je však zpětnovazební, čímž se od předchozích dvou velice liší. Díky možnosti zpětné vazby tak síť umí zpracovávat nejen okamžité vstupy, ale i sekvence. Protože však síť obsahuje i zpětnou vazbu, mohou se po této šířit chyby, a síť

tak může mít větší sklony k špatnému učení. To se snaží řešit různé varianty rekurentních sítí.

Varianty rekurentních sítí:

- Plně rekurentní
- Rekurzivní
- Hopfieldova
- Elmanova a Jordanova
- Síť odezvy (Echo síť)
- Shlukovač nervové historie (Neural history compressor)
- Dlouhá krátkodobá paměť (LSTM)

Učení rekurentních sítí probíhá podobně jako u dopředných sítí. Stejně jako vícevrstvá perceptronová síť i rekurentní síť používá metodu gradientního sestupu a zpětnou propagaci. Rozdíl oproti zpětné propagaci u dopředných sítí je propagování s přihlédnutím k času. Ještě existuje varianta učení nazývaná Rekurentní učení v reálném čase, který je výpočetně náročnější. Jeho výhodou je učení v reálném čase, chyby se tedy nepropagují do dalších časových úseků. Dalším problémem při učení je ztráta gradientu chyby při řídkých vstupech pro síť. Pro učení zajímavé události se vlivem tohoto problému rozpouští do sady nezajímavých událostí, nemají tedy velký vliv na učení. LSTM jednotky se snaží této chybě vyhnout použitím tzv. "zapomínacích"hradel.

3.2.3.8 Neuronové sítě - VIN

Síť iterující hodnoty, value iteration network (VIN) [23], je neuronová síť s integrovaným plánovačem pro výběr nadcházejících akcí. Skládá ze ze dvou modulů. První modul tvoří konvoluční neuronová síť, tento model je přímo napojen na pozorovaná vstupní data. Tato konvoluční síť je vícevrstvá, a zpracová data v iteracích. Výsledkem každé iterace je určená výstupní hodnota sítě, která se propaguje zpět do další vrstvy. Toto se opakuje tolikrát, kolik je vrstev v síti. V každé iteraci se tedy vezmou vstupní data s nějakou výstupní hodnotou, z nich se vytvoří nová hodnota pro další iteraci. Následně se vybere vhodná akce na základě vektoru hodnot a podmíněného seznamu akcí. Použitý vektor hodnot je ještě možno omezit na jeho podmnožinu, protože všechny hodnoty nemusí být význačné pro výběr akce.

Kapitola 4

Návrh

Cíle této práce jsou tři:

- Vytvořit model domu pro umožnění analýzy požárních rizik
- Optimalizovat rozmístění vybavení domu v daném modelu
- Vyhodnotit rizika požární bezpečnosti na vytvořeném modelu včetně vybavení a přidat možná doporučení pro snížení rizik

Pro realizaci těchto cílů je nutno postupovat dle uvedeného pořadí, nelze vyhodnocovat požární rizika bez existujícího modelu. První cíl spočívá čistě v návrhu. Model místnosti je přímočarý, pracuje se dvěmi základními veličinami: průchodností a hořlavostí. Pomocí těchto dvou hodnot je vytvořena dostatečná simulace domu, která poskytuje všechny informace uživateli i algoritmům pracujícím s těmito modely.

Krom modelu bylo potřeba ještě upravit algoritmy samotné. Protože řešení úlohy nejsou standardní doménou žádného ze zkoumaných algoritmů, bylo potřeba každý z nich změnit dle následujícího přehledu.

Nejmenší úpravy je nutno provést v případě algoritmů pro optimalizování místnosti, protože prohledávání jsou přímo určena k řešení tohoto problému. Jediným problémem zde je omezení při umísťování předmětů, protože je potřeba počítat s většími rozměry umísťovaného předmětu. Zatímco běžný algoritmus počítá s obsazováním pouze jediného místa při kladném rozhodnutí o umístění, zde je nutno do stavového prostoru reflektovat širší změny. Nejnáročnější změna v tomto případě je potřeba vytvoření vhodné vyhodnocovací funkce, která pro určité rozmístění vrací úspěšnost.

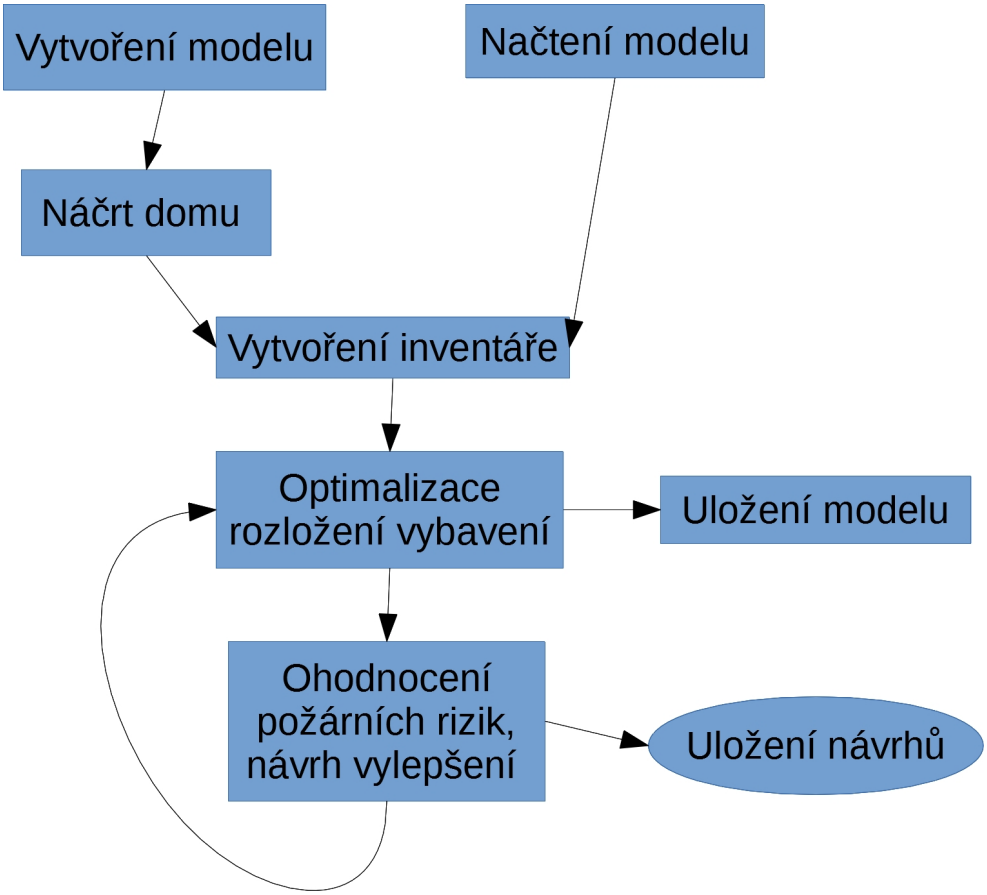
Drobnějších úprav doznaly neuronové sítě, protože každá ze sítí požadovaným operacím vyhovuje dobře. Problém je však u vstupních dat, protože ta obsahují dvě různé matice, které ale na sobě závisí, a v rámci jednodušší sítě je dobré tato data dodávat v jednom vstupu. Také je potřeba upravit učící algoritmus pro neuronové sítě, protože je nutné spoléhat se na vzorky dodané

z jiných zdrojů. Dalším problémem u neuronových sítí je velikost vstupu. Protože testované neuronové sítě jsou vytvořené s pevným počtem neuronů, je třeba upravit velikost vstupní vrstvy. Jednou z možností je využít tzv. Přesunutí učení, které spočívá v rozdělení neuronové sítě na dvě vrstvy, a nahrazení jedné vrstvy jinou (jiným rozměrem). Poté je potřeba krátké učení, ale předtrénovaná část toto učení urychluje.

Daleko větší výzvou je úprava optimalizačních algoritmů, protože tyto jsou vytvořené k práci ve stavovém prostoru, který je omezený, ale spojitý. Protože však zkoumaný prostor budov spojitý být nemusí (oddělení zdmi, překážkami, ohněm), je třeba tyto algoritmy upravit. Při směrování je tedy nutné uvažovat s omezeným prostorem, při orientaci s omezeným dohledem. Je diskutabilní a nutné ověřit, zda při vzájemné komunikaci mezi agenty je nutno uvažovat překážky, nebo zda tato komunikace může být prospěšná. Navíc je třeba zachovávat data z průběhů jednotlivých iterací, protože z nich vychází směrování genetického algoritmu. A právě to je další potřebná změna. Genetický algoritmus obecně snadno řeší problémy v jednodimenzionálním prostoru, tento prohledávaný prostor je však vícedimenzionální. Je potřeba brát v úvahu obecnou mapu budovy, dále také teplotní mapu budovy. Protože obecná optimamlizace pomocí genetického algoritmu by trvala dlouho, jeden z pokusů je upravit tento algoritmus, aby využíval výstupů z ostatních algoritmů, a směřoval rychleji k optimální variantě. Protože algoritmy optimalizace hejnem v jednotlivých krocích (v případě hejna částic a světlušek) nebo ve svém finálním stavu (v případě optimalizace mravenčím hejnem), je možno náhodné jevy genetického algoritmu upravit na pseudonáhodné závislé na určitém vstupu.

Jak tedy funguje nástroj jako celek? Nejlépe je jeho fukce popsat pomocí diagramu workflow pro uživatele. Nejprve je potřeba vytvořit model. Pro tento model je dále třeba vytvořit invenvář a ten následně optimalizovat, nebo jej prohlásit za optimální. Poté je možno hodnotit požární rizika a získat doporučení.

Následující odstavce popisují konkrétní úpravy pro zvolené algoritmy. Značení jsou převzata z příslušných kapitol popisujících standardní funkci algoritmů.



Obrázek 4.1: Diagram workflow pro uživatele

4.1 Nástroj pro tvorbu místností

Nejprve je nutno zamyslet se nad významem jednotlivých údajů o domě. Dům obsahuje informace o požárním riziku a také o průchodnosti. Základní průchodnost je daná tím, zda se jedná o zeď nebo o prázdný prostor. V prostoru může být umístěno nějaké vybavení, které přímo určuje požární riziko v daném místě. Existuje tedy přímá vazba mezi oběma údaji. Inspirací k modelu jsou vícekanálové obrázky, například RGB. Takové obrázky mají více složek (kanálů), a tyto jsou reprezentovány jako matice, a existují nezávisle na ostatních složkách, ale přitom jsou na nich vlastně závislé. Nezávislé jsou proto, že pokud nějakou složku vynecháme, výsledek pořád může dávat smysl. Závislé jsou proto, že dohromady tvoří celek a bez jedné složky by byl celek nekompletní. Informaci o umístní podlahy a zdí lze snadno odvodit přímo z mapy průchodnosti. Kde je průchodnost nulová je zeď, kde je úplná je volná podlaha. Složitější může být reprezentovat různé výklenky. Ale i to je možné, například označením výklenku jako vybavení. Problémem je mapa požárních rizik, tedy

mapa hořlavosti. Problémem je hlavně proto, že některé materiály zvyšují riziko nejen v místě, kde jsou, ale mohou ho zvyšovat i v celé místnosti nebo ve svém okolí. Například hořlavé látky často produkují hořlavé výpary a ty zhoršují situaci v celé místnosti. A také díky riziku vytečení kapaliny zvyšují riziko ve svém okolí. Bohužel podchytit toto riziko se dá v modelu jen těžko, protože by bylo nutno do modelu při umísťování vybavení současně ještě uvádět, jaký je charakter požárního rizika v okolí. Proto je tato váha nechána na uživateli, který může pokrýt místnost volnou podlahou, ale současně k ní uvést zvýšené požární riziko. Většina algoritmů pracuje s maticovou reprezentací, a tento model je pro ně dostatečným.

4.2 Optimalizace místnosti

Pro optimalizaci místnosti bylo třeba provést nejméně změn. Protože vstupní data jsou přímo navržena k zpracování, není potřeba žádné předzpracování dat, pouze načtení do pracovních struktur. Pouze pro potřeby shlukování bylo třeba upravit vstupní data pro zjednodušení výpočtu vzdáleností, protože je snazší počítat vzdálenosti pro dvojrozměrnou matici, než pro vícerozměrné varianty. Výhodou sloučení je reprezentace místnosti v šedotonové podobě s podchycením požárních rizik. Pro sloučení vstupů byl použit výpočet normalizace $V_{i,j} = \max\Big(\min(\frac{50*-T_{i,j}}{128} + 170, \mu(M_{i,j}), 255), 255 * Z_{i,j}\Big)$. $Z_{i,j}$ je přítomnost značení v binární podobě, použita pro posun na hodnotu 255 v případě přítomnosti požárního značení. V opačném případě se použije varinta teplotní mapy nebo překážek. Překážky jsou reprezentovány mapou místnosti $M_{i,j}$, ve které se nacházejí hodnoty 0 reprezentující neprostupnou překážku (zeď), 255 reprezentující volné místo a hodnotami mezi 0 a 255 reprezentujícími překážky o různé prostupnosti. Normalizace je zde provedena funkcí μ, převádějící hodnoty 0 na 0, 255 na 255, a všechny ostatní hodnoty na 3 shodně velké kategorie reprezentující nepřekonatelné, obtížné a snadné překážky. Tyto jsou reprezentovány čísly 30, 60, 90 pro zobrazení ve stupních šedi. Pro dobré zobrazení teplotních map tedy zbývá rozsah 120-220. Teplotní mapy $T_{i,j}$ reprezentují očekávanou teplotu místa, a jsou centrovány na hodnotu 0. Kladné hodnoty jsou okolí hořlavých materiálů, a záporné jsou naopak nehořlavá bezpečnější místa. Jejich centrem (a tedy místem neutrální teploty) je hodnota 170, a tato bude použita v případě nepřítomnosti překážek v mapě místnosti M. Tato mapa je jedním z výstupů pro uživatele, a představuje mapu pro snadnou orientaci. Dalším výstupem optimalizační fáze je mapa místnosti samotná, seznam předmětů v místnosti a také barevnou mapu teplotních rizik, převednou z mapy T převodem a normalizací kladných hodnot na rozsah 0-255 červené barvy a záporných hodnot normalizací a negací na rozsah 0-255 modré barvy. Protože původní mapa obsahovala tyto hodnoty v rozmezí od -127 do 127, ve výsledné teplotní mapě nemůže nikdy nastat kombinace červené a modré, tedy jakýkoli odstín fialové.

4.2.1 Prohledávání do šířky

Zde není prakticky potřeba žádná úprava algoritmu jako takového, pouze rozšíření jeho bodu 1. o změnu sousedů. Sousedé jsou v tomto algoritmu nejbližší vrcholy pro případ prohledávání bez umístění objektu, a všichni sousedé hraničící s objektem v případě jeho umístění. Díky této úpravě lze vyloučit z prohledávání překryté vrcholy rovnou a také lze detekovat překrytí pro umístění objektů na vrcholy označující existující překážku.

Pro vypočítání vhodného umístění lze použít jednoduchý vzorec $v = d(x_+)+d(x_-)+d(y_+)+d(y_-)+d(xy_+)+d(xy_-)+d(yx_+)+d(yx_-)+k(xy)$. Funkce $d(z)$ reprezentuje vzdálenost od objektu k nejbližší překážce ve směru z, tedy vodorovně, svisle i po obou diagonálách a funkce $k(xy)$ reprezentuje vzdálenost objektu od nejblížší překážky ve všech směrech. Funkce v je reprezentována jako počet volných vrcholů od středu objektu po nejblížší překážku mínus počet vrcholů od středu k hranici objektu. Funkce k je počet volných polí okolo celého objektu, iterativně obsazovaných algoritmem vlny, a vrací počet iterací než bude nalezen obsazený vrchol.

4.2.2 Prohledávání do hloubky

Protože se jedná v podstatě o stejný princip jako v případě prohledávání do šířky, je vhodné pracovat se stejným omezením a stejnou metodikou jako v případě BFS.

4.2.3 Metoda větví a hranic

Při analýze bylo zjištěno, že pro úspěšnou aplikaci B&B je třeba určit metriky, které by plnily roli odhadů. Nutný je v první řadě odhad budoucích možností. Tyto odhady by měly být jednoduché, a přitom potencionálně informativní vůči budoucímu zaplnění plochy. Jako metriky se nabízí:

- Zbývající souvislá plocha v okolí přidaného objektu
- Poměr velikosti objektu ku velikosti volné plochy (největší, nejmenší dostatečný)
- Největší souvislá velikost zbývající volné plochy
- Rozložení hořlavých prvků na plochu

Na druhou stranu, zrovna tak bude potřeba vytvořit metriku pro odhadování zbývajících věcí v inventáři. Jako přímá metrika je možno počítat s velikostí sumy plochy všech předmětů, protože umisťování objektů se snaží optimalizovat i rozložení předmětů v rozumné vzdálenosti od sebe, bude volná plocha rychle ubývat. Dalším parametrem může být největší rozměr z pohledu šířky a

výšky, protože úzké objekty mohou být dlouhé, a naopak krátké objekty mohou být široké. A vzhledem k typu uspořádání předmětů v inventáři mohou dlouhé objekty přijít na řadu po kratších, ale rozlehlejších.

Efekt různých parametrů (metrik) je potřeba ověřit praktickým vyhodnocením, protože na první pohled se všechny zdají být opodstatněné, a ve výsledku nemusí hrát příliš velkou roli. Navíc některé metriky vychází ze stejného základu jako metrika používaná u BFS a DFS, takže přímo podporují iterace k podobným výsledkům. Navíc stejně jako v BFS a DFS i zde platí možnost automatického ořezání nevhodných větví vyhodnocením překrytí, podporovaná metrikami vyhodnocujícími volné plochy.

4.2.4 Shlukování

Oproti výše uvedeným algoritmům shlukování nedoznalo žádné změny oproti jeho standardnímu algoritmu, ale také jako jediné vyžaduje drobné předzpracování, a to v podobě převedení výsledné tepelné a orientační mapy do jedné sloučené mapy. Cílem shlukování je vytvořit sadu reprezentativních jedinců pro vyhodnocení, která bude uložena do složky k analýze. Uživatel bude moci ovlivnit požadované výstupy na základě práce se složkou, a také na základě nastavení parametrů shlukování. Zajímavým ukazatelem pro shlukování může být porovnánní vytvořených shluků a funkcí v vyhodnocující vhodné umístění, který ale nebude vyhodnocován jako součást vektoru shluku.

4.3 Ohodnocení rizik a poskytnutí doporučení

Protože každý z algoritmů pracuje s jiným typem agentů, je vhodné každému z nich předzpracovat vstupy. Konkrétní předzpraování bude rozebíráno pro každý algoritmus zvlášť.

Společným bodem pro všechny algoritmy však bude základní algoritmus šíření ohně, protože vyhodnocování probíhá v reálném čase, a je tedy potřeba brát v potaz i šíření ohně jako takového. Pro dostatečný model ohně stačí aplikovat Bernoulliho rozdělení ovlivněné teplotní mapou a hořením v aktuálním okamžiku. Oheň je definován intenzitou a svým počátkem. Pro šíření do okolí se bude používat algoritmus vlny. Každý hořící vrchol x zvyšuje svou intenzitu i jako funkci $i_x(t) = \max\Big(i_x(t-1), rand\Big(\omega * (t_x * i_x(t-1) + \sum_{y \in neighbours} t_y * i_y(t-1))\Big)\Big)$. Totéž bude platit pro vrcholy, které dosud nehoří, a to na vzdálenost 1 od hořících vrcholů. K ovlivnění rychlosti šíření požáru bude integrována skrytá konstanta ω, která může šíření zpomalovat i zrychlovat a která umožňuje korelaci mezi šířením a rychlostí prozkoumávání optimalizaních algoritmů. Vhodná hodnota tohoto parametru je předmětem experimentálního vyhodnocení. Navíc jako další vstupní parametr bude počáteční intenzita požáru, která umožňuje korigovat počáteční rychlost šíření. Ideálním počáteč-

ním nastavením je nízká hodnota ω a vysoká hodnota počátečního žáru. Toto nastavení představuje požár vysoce hořlavého materiálu, následnou aktivaci protipožárních opatření, která omezí riziko šíření požáru, a současně vyvolají požární poplach. Při tomto nastavení je vysoká šance optimalizačních algoritmů na nalezení úspěšných cest pro opuštění mapy, a její optimalizaci. Zvýšením rychlosti šíření je možno otestovat kvalitu navrženého umístění značek požární bezpečnosti a také míru úspěchu při učení neuronových sítí.

4.3.1 Optimalizace mravenčím hejnem

Na první pohled lze říci, že zkoumaný prostor se může podobat grafu, konkrétně mnoha grafům spojeným vzájemně mosty (dveře). Na takové situace již byli mravenci používáni [24], problém je zde ovšem trochu jiný. Zatímco mravenci v systémech hledají optimální nastavení v jednotlivých podgrafech, zde mravenci pouze hledají cestu z podgrafu do jiného. Podstatně bližší této úloze je tedy využití mravenců k navigaci z bludiště [25].

Pro optimalizaci mravenčím hejnem bude třeba pracovat s novým termínem - antiferomonem. Tato varianta feromonu dynamicky pokrývá mapu, a to konkrétně políčka sousedící s hořícími do vzdálenosti omezené vstupním parametrem. Bude tedy motivovat mravence nechodit od plamenů. Na druhou stranu, protože správná cesta může vést skrz hořící vrchol, umístění jakéhokoli feromonu na vrchol s antiferomonem bude znamenat úplné zrušení antiferomonu.

Druhou podstatnou změnou je vrchol obsahující požární značení. V rámci tohoto vrcholu bude umístěna trvalá feromonová stopa v okolí vedoucí k tomuto vrcholu. Hodnota této stopy bude nízká, aby pouze směrovala, ale nutně nesváděla mravence mimo. To umožní popojít k dalšímu značení, nebo od něj odejít ve snaze najít východ. Bude také třeba otestovat, jestli bude vyhovující aby všechna značená místa měla stejnou úroveň trvalého feromonu, nebo zda bude lépe intenzitu zvyšovat se vzdáleností k východu.

Výstupní metrika z tohoto modelu je počet mravenců úspěšně zachráněných, počet mravenců uhynulých v požáru a rychlost obou skutečností.

4.3.2 Optimalizace rojem částic

I pro optimalizaci pomocí roje částic je úloha navigace v bludišti již řešená [26], a je možné z ní čerpat inspiraci. Protože toto není přímo úloha navigace v bludišti (vlivem ohně se bludiště mění), je potřeba ještě myšlenku rozvést.

Obdobě jako ACO i PSO vyžaduje přípravu plochy pro vyhodnocení. První úpravou je potřeba LOS omezení pro množinu L^i. Do této množiny mohou spadat pouze agenti, na které dotyčný agent přímo vidí. Navíc do této skupiny budou spadat i vrcholy s požárním značením. Požární značení k sobě přitahuje agenty ze svého okolí, a pomáhá jim určovat směr. Pokud bude více požárních značení v dohledu, agenti budou mít snahu cestovat pomocí znační za vý-

chodem. Další experiment bude třeba provést s dohledem požárního značení. Ignorování dohledu by totiž mohlo potencionálně zavést agenty správným směrem, i pokud na konkrétní značení nevidí (například na základě paměti polohy tohoto značení při předchozím průchodu kolem něj).

Navíc je bezpomínečně nutné, aby částice narážely na překážky, a reagovaly na takovou srážku. Jsou tři možnosti, jak mohou částice na srážku reagoavat, a další testování bude nutné pro zjištění efektivnějšího přístupu. První variantou je nechat částice odrážet se od překážek, a přepočítávat finální umístění vzhledem k odrazům. Druhou variantou je částice při srážce s překážkou umístěné na hranici s překážkou, přepočítat odražený směrový vektor a počkat na další iteraci pro pohyb. Třetí variantou je využít původní myšlenky v obecném algoritmu, a to omezením rychlosti (nastavení rychlosti na 0).

Výstupní metrika PSO je totožná s metrikou pro ACO.

4.3.3 Optimalizace hejnem světlušek

Také GSO umožňuje úpravu plochy, aby bylo dosaženo lepšího a rychlejšího úspěchu. Základní myšlenkou zde je označovat hořící místa jako záporné světlo, a místa se značením jako kladné světlo. Díky tomu mají světlušky menší tendenci směrovat se k nebezpečným místům a naopak mířit k bezpečí. Jako nejsilnější zdroj světla lze považovat samotné východy, které mají být trvalým cílem. Pro vyhodnocení dohledu k zdrojům je potřeba počítat s překážkami, a detekovat zakrýtí zdrojů. Původní hypotéza k využití optimalizace hejnem světlušek byla její podobnost se zjednodušeným vnímáním. Člověk si tvoří obraz budovy, ve které se nachází, a přirozeně si udržuje v podvědomí cestu budovou. K realizaci tohoto paměťového efektu může pomoci koeficient stmívání, který s každou překážkou mezi zdrojem a agentem sníží intenzitu světla zdroje vnímanou pozorujícím agentem.

Kruh vnímaného světla se zvětšuje, je-li světluška dlouho na jednom místě. Delší přítomnost světlušky u značení způsobuje omezení světla produkovaného tímto bodem, aby se zvětšovala šance světlušky body značení opustit a vyhledávat další. Postupně po uhasnutí bodu dojde opět k jeho obnovení. Rychlost uhasínání je přímo úměrná počtu světlušek v jeho okolí. K uhasínání nedojde v případě, že se jedná o bod pro bezpečné opuštění domu. Světlušky, které dlouho nevidí žádný světelný bod postupně umírají, nejprve sníží svou intenzitu, a následně zmizí. Tímto je možno zamezit hromadění světlušek v místech bez úniku, pokud se v okolí tohoto místa nenachází žádné značení.

Oproti předchozím algoritmům je k výstupním metrikám ještě přidán počet uhynulých světlušek na základě problémů s orientací. Tento údaj je zajímavý pro možné další optimalizace rozmístění požárního značení.

4.3.4 Genetický algoritmus

Genetický algoritmus je použit pro umísťování bodů bezpečnostního značení. Jeho aplikace by mohla být sice využita i pro simulaci úniku z domu (algoritmus se dá použít i pro řešení problému bludiště [27]), ale pro tyto účely by bylo třeba ještě vytvářet agenta, kterého by genetický algoritmus vylepšoval. Protože půjde pouze o optimalizaci umístění bezpečnostního značení. Na počátku jsou tato značení umisťována náhodně, a jejich umístění je postupně vylepšováno. Protože se nejedná o typické využití GA, je použito vysoké míry mutace, a naopak nízké míry křížení, protože tím je zajištěno získávání nových zajímavých bodů spíše než tradičním přístupem.

Protože se jedná o čtvercovou síť, bylo nutno provést úpravy tradičního chápání operátorů. Mutace vytváří nové body v ose X nebo Y vůči referenčnímu bodu. Křížení vytváří 0-2 potomky na základě doplnění do obdélníku. Pokud na hranách obdélníku nejsou žádné překážky v podobě zdí (překážky v podobě vnitřního vybavení jsou ignorovány), a pokud uvažované vrcholy obdélníku jsou na zemi bez překážek.

Asi nejzajímavější částí zde je selekce. Stejně jako konvenční selekce vybírá na základě fitness, i tato musí nějakým způsobem ohodnocovat kvalitu umístění. Protože však v této úloze není snadné definovat fitness, nabízí se 3 možnosti - vzdálenost od evakuačního bodu, počet značení přímé viditelnosti od bezpečného bodu, vztah s provedeným algoritmem. Vzdálenost od evakuačního bodu je sice zajímavý údaj, ale bohužel také nechtěný dle diskuse v kapitole Analýza. Cílem algoritmů je totiž orientace bez přidané hodnoty směru k cíli. Totéž se dá říci i o druhé možnosti, počtu značení přímé viditelnosti. Na druhou stranu přímá viditelnost umožňuje určit jen možnost nánsledování značení do cíle. Díky tomu je tedy možné rozdělit značení na dvě skupiny, jednu na vedoucí do cíle, a druhou na oddělenou. Oddělená skupina bude daleko zajímavější cíl pro optimalizace než skupina od cíle, protože nalezením spojení mezi těmito dvěma skupinami dojde k pokrytí sjednocení obou prostorů. Úloha je tedy částečně podobná hledání minimální kostry v grafu. Poslední možností je využití znalostí z průběhu algoritmu provedeného k vyhodnocení. Například zjištěním cest mravenců mezi počátkem a cílem je možno najít silně využívané cesty a na ně následně umístit značení. Algoritmus pro selekci tedy využívá kombinace varianty dvě a tři, hledá propojení mezi viditelností a cestou vytvořenou algoritmem. Z toho důvodu jsou všechny tři optimalizační algoritmy upraveny k zachycování historie. Toho je v případě PSO docíleno zaznamenáváním trajektorie každé částice (velice paměťově náročné) a následným hledáním průniků mezi trajektoriemi. Pro ACO je situace jednodušší, pokládaný feromon se aditivně ukládá do matice historie. Stejně tak pro GWO je možné snadno určit historii na základě intenzity světla v daném bodě jako aditivní složky matice historie.

Pro zachování variability řešení je implementována varianta omezené katastrofy. Ta náhodně vybírá prvky, pokud nejsou důležitým spojem, tak je

odstraňuje. Důležitý spoj je definován jako místo, ze kterého je vidět na více bezpečnostních bodů v okolí, které ale vzájemně nevidí na sebe.

4.3.5 Neuronové sítě

Neuronové sítě jsou využity ve svých standardních implementacích z knihovny tensorflow, s výjimkou sítě VIN z externího zdroje. Sítě jsou trénovány na základě dat vypočítaných předchozími algoritmy, poskytují orientační hodnoty, jakých by měly dosáhnout déletrvající běhy optimalizačních algoritmů.

Kapitola 5

Realizace

Finálním produktem je aplikace vytvořená v Pythonu verze 3 za pomocí knihovny PyQt a TensorFlow.

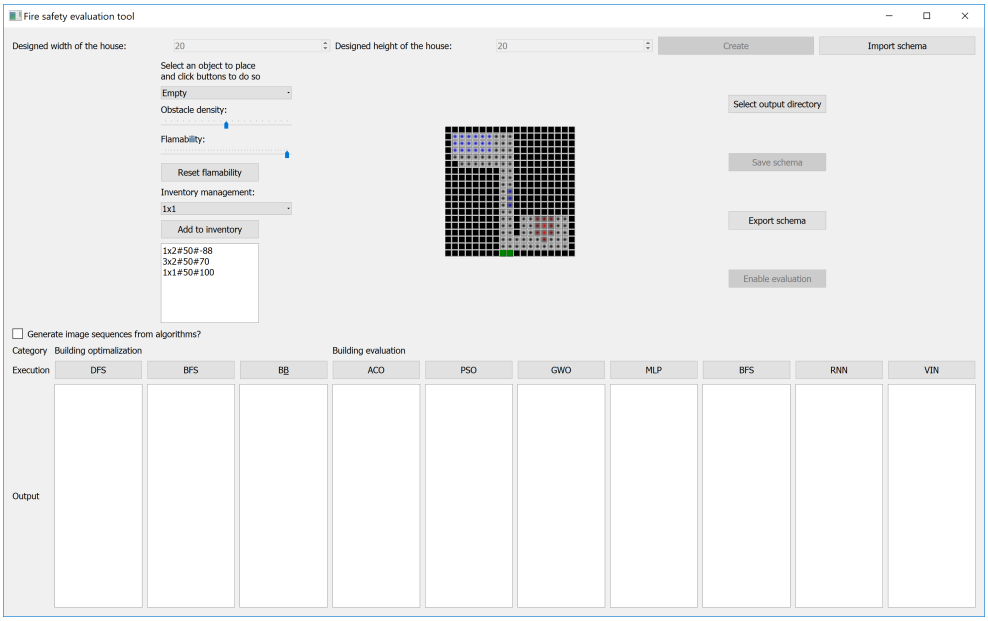
5.1 Uživatelské rozhraní

Uživatelské rozhraní je minimalistické a účelné, vytvořené za pomoci knihovny PyQt, a je tedy snadno rozšiřitelné o další prvky. Je optimalizováno pro 4k monitor s vysokým rozlišením.

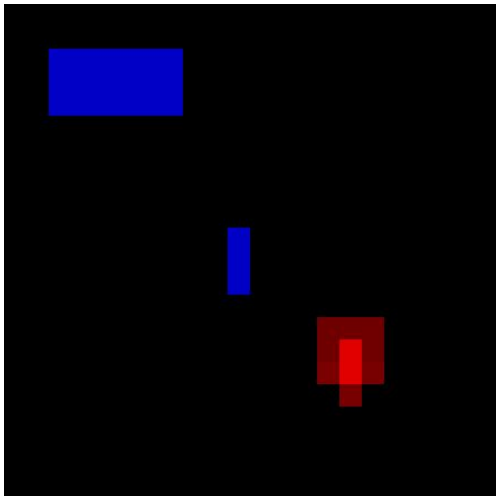
Rozhraní je vytvořeno s ohledem na snadnou orientaci a workflow. Jediným málo intuitivním prvkem je vkládání do inventáře a umísťování prvků na plán budovy. Kliknutí na prvky budovy umožňuje přidat libovolný tvar předmětu, ale je nutno vyplňovat budovu po čtvercích, zatímco přidávání předmětů do inventáře automaticky počítá s určeným tvarem.

Program automaticky ukládá data z výstupů jednotlivých algoritmů, je-li určena složka pro výstupy. V první řadě zde je možno získat teplotní mapu a mapu průchodnosti. Dále také kumulativní mapy feromonů, luciféria a orientace vektorů částic, na základě kterých se poté vytváří doporučení. Také je možno samozřejmě ukládat výstupy těchto doporučení v grafické formě.

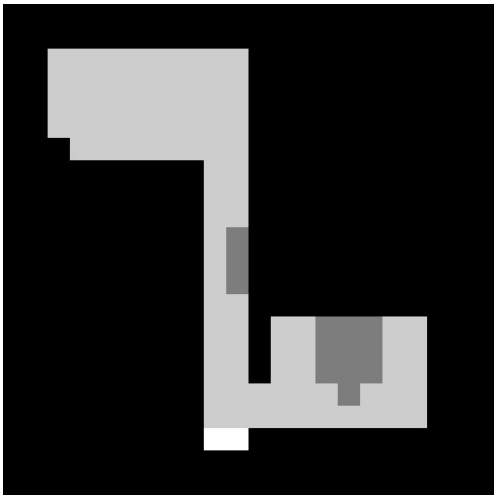
Pro jednodušší orientaci ve čtvercové síti jsou čtverce vykreslovány s barevnou tečkou uvnitř. Vnější okraj čtverce určuje, jak snadný je průchod čtvercem. Vnitřní tečka určuje, jak moc velké požární riziko je na tomto čtverci.



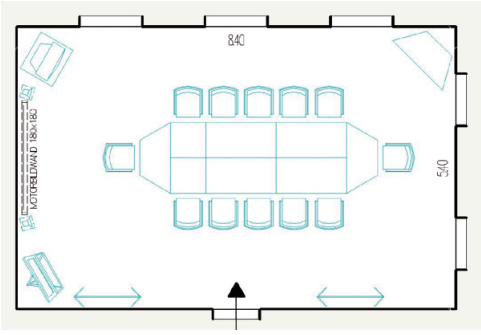
Obrázek 5.1: Uživatelské rozhraní s vzorem vytvořeného domu - skladiště v zástavbě



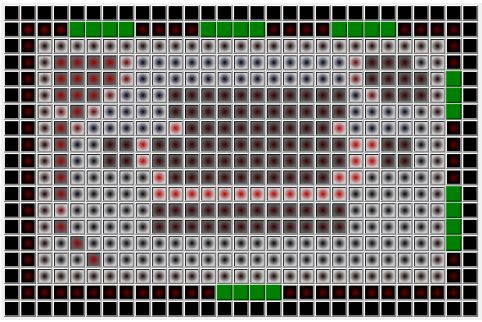
Obrázek 5.2: Vygenerovaná mapa požárního nebezpečí - skladiště v zástavbě



Obrázek 5.3: Vygenerovaná mapa průchodnosti - skladiště v zástavbě



Obrázek 5.4: Ukázka místnosti k modelování



Obrázek 5.5: Hotový model pro vyhodnocení rizik

Kapitola 6

Výsledky

6.1 Optimalizace místnosti

Přidávání věcí z inventáře probíhá po stlačení tlačítka pro jednotlivý algoritmus, vždy se vztahuje na původní vzor budovy. Po dokončení algoritmu jsou vybrány 3 řešení na základě shlukování, a tato řešení jsou následně uložena do výstupní složky. Uživatel si může načíst to řešení, které nejlépe odpovídá jeho představám. Nejprve je možnost vytvořená řešení prohlédnout standardním prohlížečem obrázků. Je tedy možné vyzkoušet všechny algoritmy a na základě jejich výstupů zvolit nejlepší vygenerované řešení.

Tabulka 6.1: Rychlost algoritmů optimalizace na základě složitosti problému - čas potřebný k nalezení všech řešení [s]

Rozměr budovy	Počet předmětů k umístění	DFS	BFS	B&B
10x10	2	0.124	0.153	0.101
10x10	5	0.141	0.139	0.095
10x10	10	0.216	0.197	0.120
20x20	2	0.934	0.894	0.646
20x20	5	1.250	1.309	0.976
20x20	10	1.762	1.846	1.412

Z testování algoritmů je patrno, že v mnoha případech využití shlukování nemělo žádný větší přínos. Dále praktické testy ukazují, že pro větší instance jde vytížení paměti do desítek GB kvůli rozsáhlé reprezentaci grafů, a program není schopen dokončit analýzu. Stejné chování je u všech tří algoritmů, protože jejich zvolené varianty pracují s mapami pro všechny přípustné stavy. Protože

Tabulka 6.2: Rychlost algoritmů optimalizace na základě složitosti problému - čas potřebný k nalezení prvního platného řešení [s]

Rozměr budovy	Počet předmětů k umístění	DFS	BFS	B&B
10x10	2	0.005	0.004	0.001
10x10	5	0.005	0.005	0.001
10x10	10	0.005	0.008	0.002
20x20	2	0.019	0.022	0.010
20x20	5	0.033	0.034	0.015
20x20	10	0.051	0.048	0.031

Tabulka 6.3: Ohodnocení výstupu člověkem [1-5, 1 nejlepší]

Rozměr budovy	Počet předmětů k umístění	DFS	BFS	B&B
10x10	2	2	2	2
10x10	5	3	2	3
10x10	10	3	3	4
20x20	2	2	2	2
20x20	5	2	3	3
20x20	10	4	3	4

Tabulka 6.4: Čas shlukování podle počtu shluků a velikostí instance [s]

Rozměr budovy	Počet předmětů k umístění	Čas
10x10	2	120.2
10x10	5	162.8
10x10	10	203.5
20x20	2	801.2
20x20	5	1239.0
20x20	10	1834.2

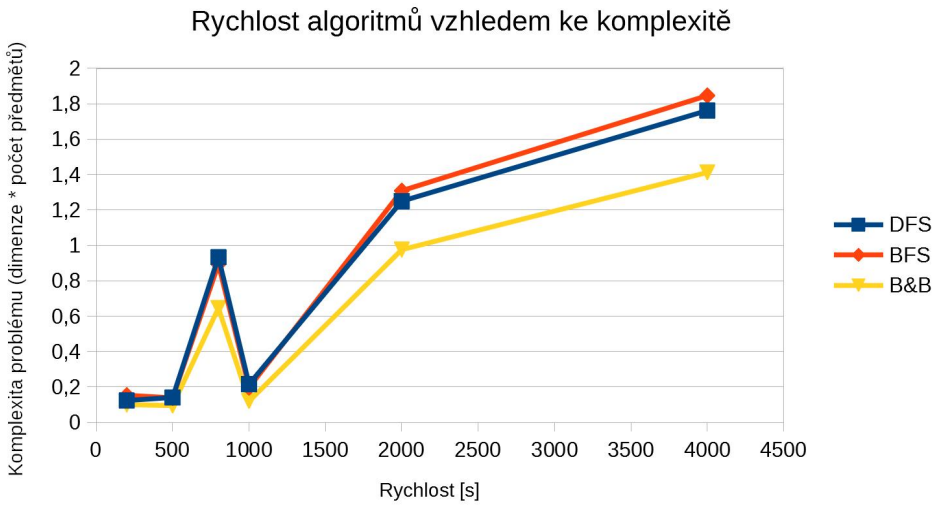
však stavů je mnoho (pro jednoduchost můžeme počítat velikost pole umocněný na počet předmětů), a do toho pro každý stav jsou různé konfigurace hořlavosti, stavový prostor řešení je velice rozsáhlý. Jen pro mapu 10x10 je to tedy 1 000 000 různých stavů, které každé obsahují dvě mapy 10x10, takže jen na minimální konfiguraci pro úplný stavový prostor je potřeba přibližně 1GB paměti.

Prohledávání je potřeba do budoucna zjednodušit. Místo udržování map by mohlo být praktické udržovat pouze informace o umístění, a pracovat jen s nimi s drobnější ztrátou přesnosti. Nevýhodou bude ovšem nezaměnitelnost podobných předmětů, protože každý předmět bude mít pevné umístění.

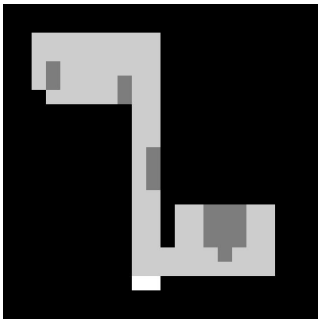
Překvapivým výsledkem je špatný výpočetní čas pro algoritmus B&B, protože tento algoritmus by čas měl podstatně snížit. Jediné možné vysvětlení je optimalizace vytváření objektů v jazyce Python. Ostatní varianty totiž kopí-

rují data až na konci běhu při tvorbě záznamu o úspěšném nalezení, a jedině tento algoritmus vytváří kopie v průběhu celého svého běhu.

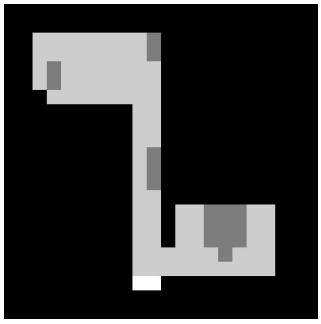
Kvůli využití shlukování pro poskytnutí výstupů má algoritmus tendenci dávat objekty dále ode zdí, čímž nejčastěji zaplňuje prostředky místností či chodeb. Při nastavení většího počtu shluků již mají předměty šanci se dostat ke zdi, což ale není příliš vhodné pro praxi, protože tím dochází k prodloužení času výpočtu, a to dosti značně. Protože lidé mají naopak raději předměty blíže u zdí, výsledky za použití shlukování nejsou vždy nejlepší, a žádný nebyl hodnocen známkou 1.



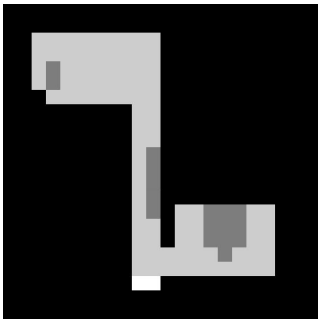
Obrázek 6.1: Rychlost optimalizace vzhledem k rostoucí náročnosti problému



Obrázek 6.2: Ukázka výstupu po shlukování



Obrázek 6.3: Jiný výstup po shlukování



Obrázek 6.4: Špatný výstup reprezentující špatný shluk

6.2 Ohodnocení rizik a poskytnutní doporučení

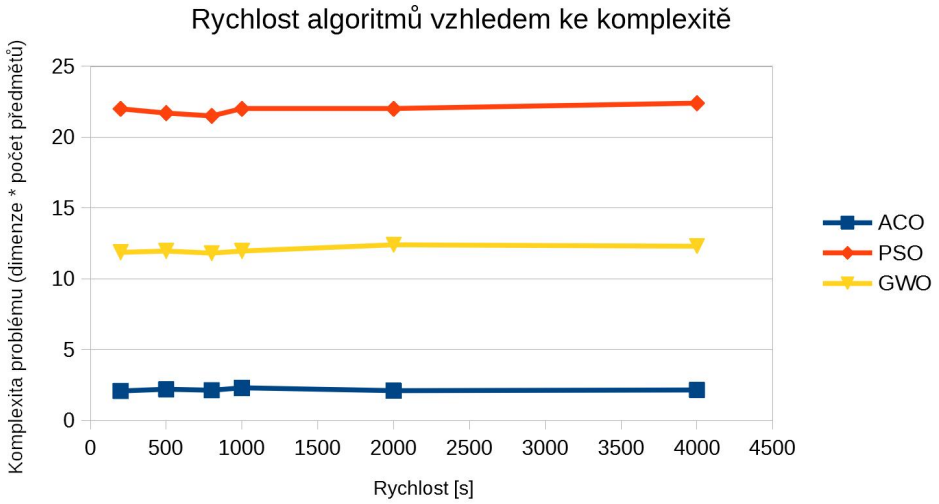
Výsledky vyhodnocení jsou ukládány spolu s doporučeními do výstupní složky, a pouze nejdůležitější vlastnosti jsou viditelné. Protože se však jedná o algoritmy závislé na inicializaci, je potřeba provést velké množství iterací než bude možno ohodnotit výsledek. Součástí výsledků je rozmístění bodů pro navigaci, které činí automatické ohodnocení velice těžké, a jeho hodnocení je prováděno ručně.

Tabulka 6.5: Rychlost vyhodnocení různých situací různými algoritmy [s]

Rozměr budovy	Počet předmětů k umístění	ACO	PSO	GWO
10x10	2	2.08	22	11.87
10x10	5	2.2	21.7	11.96
10x10	10	2.3	22	11.96
20x20	2	2.13	21.5	11.81
20x20	5	2.1	22	12.4
20x20	10	2.15	22.4	12.3

Tabulka 6.6: Kvalita vyhodnocení jedné situace různými algoritmy - průměrný čas jedné iterace

	ACO	PSO	GWO
Hodnocení	134/18562	764/53197	438/30106
Rychlost [s]	0.001	0.02	0.014
Doporučení [známka]	4	3	3
Rychlost dle doporučení [s]	2.1	25	13.2

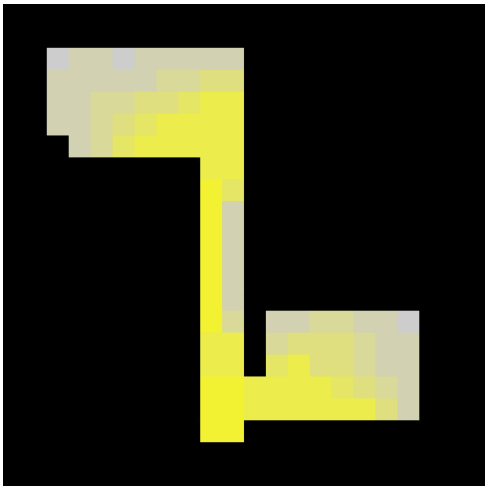


Obrázek 6.5: Rychlost ohodnocení vzhledem k rostoucí náročnosti problému

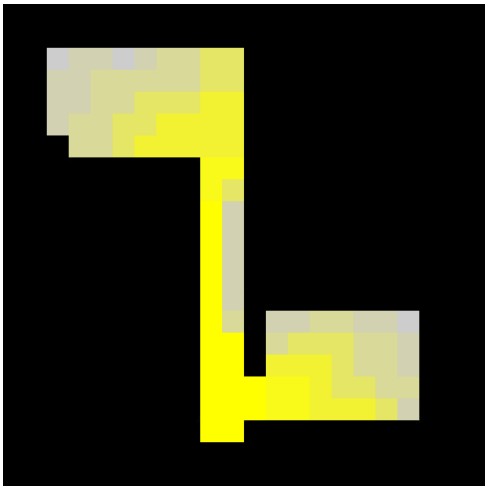
ACO je výrazně rychlejší algoritmus oproti PSO i GWO, protože uvažuje jen své nejbližší okolí. Všechny ostatní algoritmy totiž počítají viditelnost význačných bodů, kdežto ACO uvažuje pouze své okolí bez nutnosti dalších složitějších výpočtů. GWO je rychlejší než PSO, protože pracuje s omezeným dohledem, a je tedy lépe možno optimalizovat funkci pro výpočet dohledu. GWO i PSO pracují s vektorovým výpočtem pohybu, ACO pracuje s jednotkovým axiálním posunem.

Jednotkový posun u ACO se ukazuje jako nepříliš dobrý pro následné návrhy umístění značení požární bezpečnosti, protože pracovní plocha je potom příliš zanesena značkami. Nespojité značky zlepšují situaci, protože je poté jednoduché najít nejvýznačnější místa. Na druhou stranu nespojité značky mají jistou nevýhodu, protože všechna políčka nemají určenou hodnotu a tedy optimální místa mohou být nepokrytá.

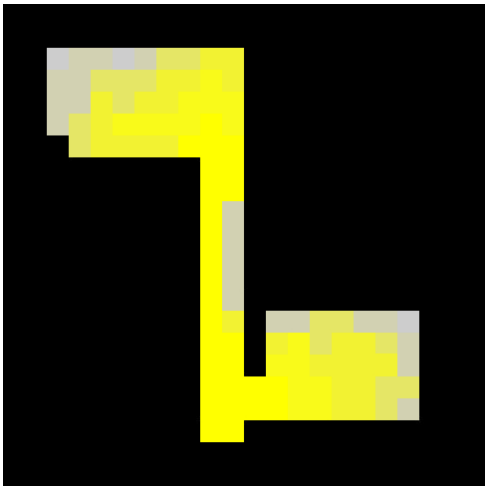
Obrázek 6.6: Reprezentace směrů pomocí barev PSO diagramů



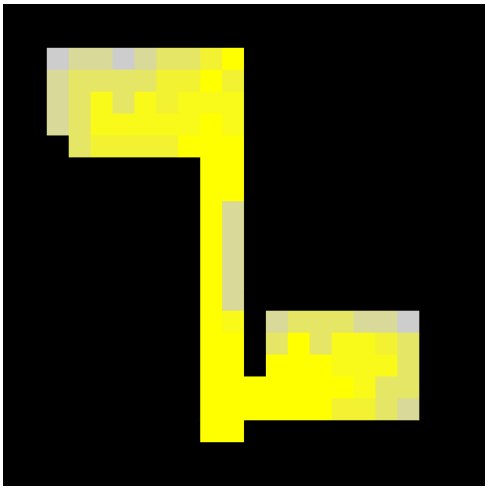
Obrázek 6.7: Ukázka algoritmu ACO, první vzorek



Obrázek 6.8: Ukázka algoritmu ACO, druhý vzorek

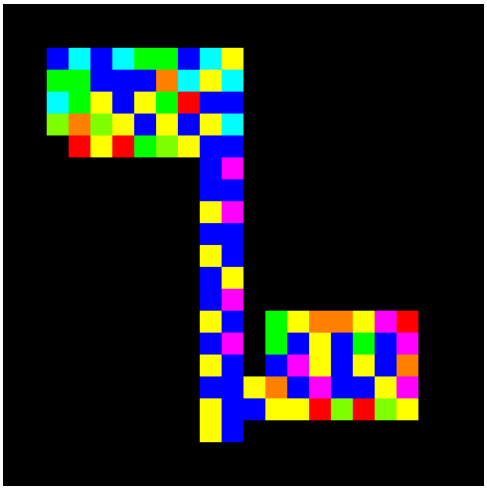


Obrázek 6.9: Ukázka algoritmu ACO, třetí vzorek

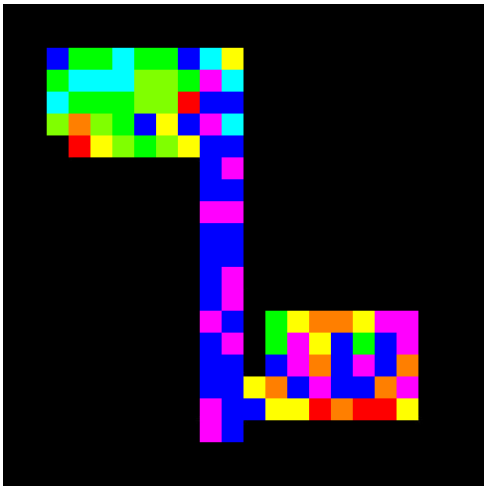


Obrázek 6.10: Ukázka algoritmu ACO, čtvrtý vzorek

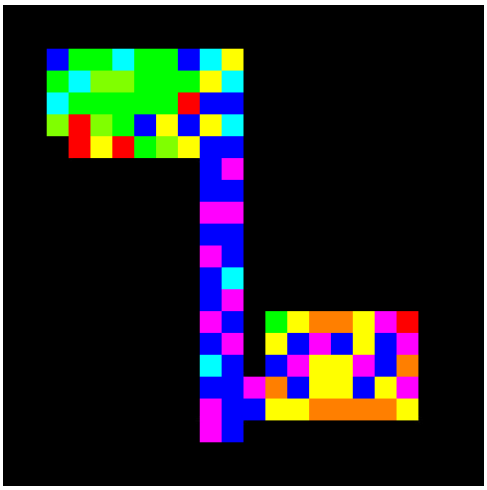
Žlutá barva zobrazuje intenzitu feromonu v daném místě, je vidět že chodbou procházelo nejvíce mravenců, takže kumulativní hodnota feromonu dosahovala nejvyšších hodnot. Obrázky jsou upravené aby lépe na první pohled reprezentovaly zesilování feromonu, program generuje obrázky normalizované vůči maximální hodnotě, takže výstupy všech 4 vzorků by byly stejně barevně intenzivní.



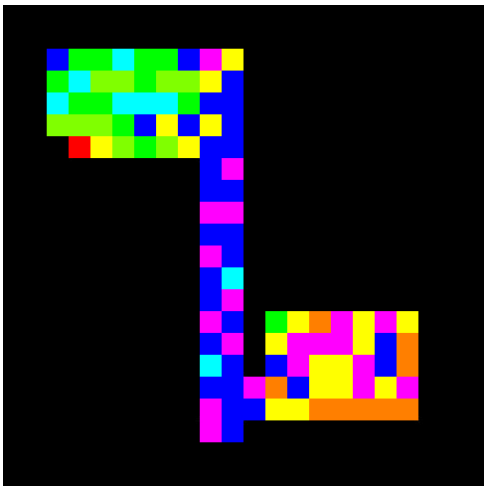
Obrázek 6.11: Ukázka algoritmu PSO, první vzorek



Obrázek 6.12: Ukázka algoritmu PSO, druhý vzorek

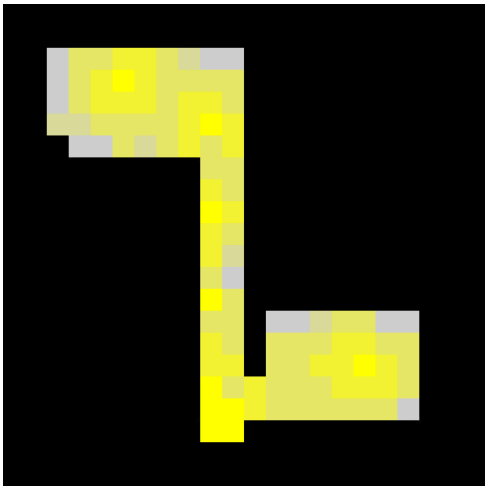


Obrázek 6.13: Ukázka algoritmu PSO, třetí vzorek

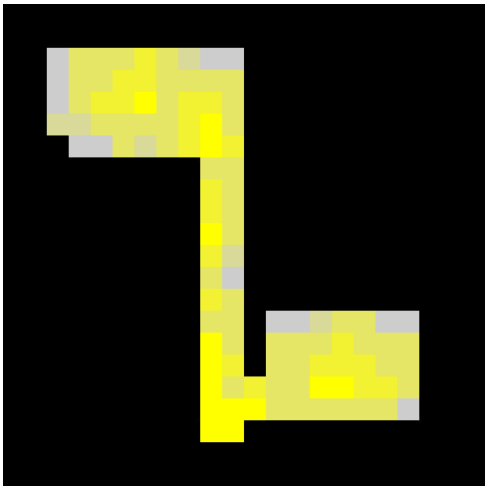


Obrázek 6.14: Ukázka algoritmu PSO, čtvrtý vzorek

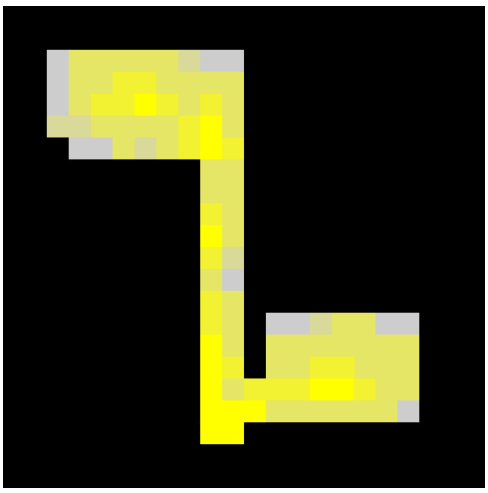
Na první pohled barevná změť reprezentuje orientaci vektorů natočení částic při prohledávání domu. Protože částice na sebe reagují, je vidět že ke konci algoritmu se částice naklánějí převážně na cestě k východu.



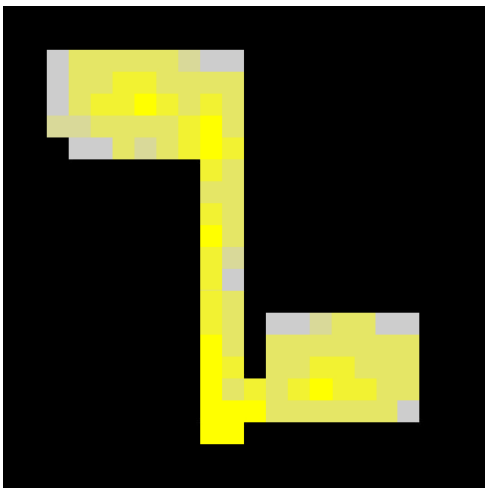
Obrázek 6.15: Ukázka algoritmu GWO, první vzorek



Obrázek 6.16: Ukázka algoritmu GWO, druhý vzorek



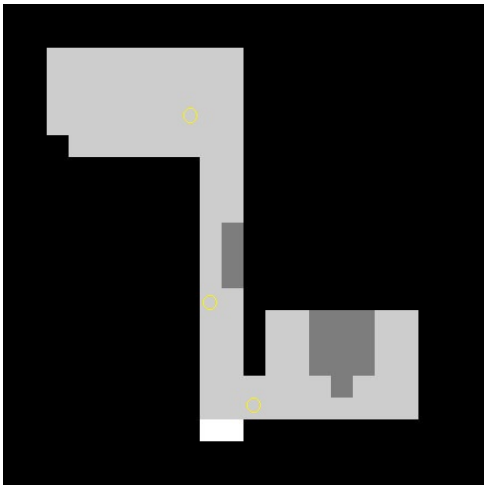
Obrázek 6.17: Ukázka algoritmu GWO, třetí vzorek



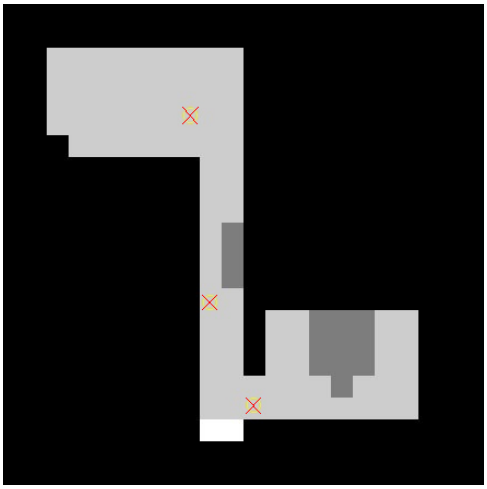
Obrázek 6.18: Ukázka algoritmu GWO, čtvrtý vzorek

Intenzita žluté barvy reprezentuje kumulativní intenzitu světlušek v daném místě. Jak je vidět, program negeneroval světlušky v rozích v dostatečné intenzitě, aby byly rozeznatelné. Dále lze pozorovat, že světlušky tvoří shluky, a nemají zřejmou snahu směřovat se k východu. Tyto shluky vypadají jako zajímavá místa pro umístění požárního značení.

Genetický algoritmus umísťuje požární značení, a jeho výstupy jsou závislé na kumulativních hodnotách poskytnutých optimalizačními algoritmy. Na základě těchto hodnot upravuje náhodně umístěná požární značení a generuje obrázek jejich umístění. Pohyb značení je velmi závislý na základním umístění, protože algoritmus nemá pevná kritéria, kde by se mělo značení nacházet. Jeho umístění lze hodnotit jako průměrné.



Obrázek 6.19: Ukázka umístění bezpečnostního značení



Obrázek 6.20: Ukázka umístění bezpečnostního značení - zvýraznění

Dále je potřeba vyhodnotit schopnost neuronových sítí ohodnotit plány obdobně, jako by to dělal optimalizační algoritmus. Všechna ohodnocení se prováděla na schématu, které nebylo zaznamenáno v trénovacích datech. Trénovací data byla vytvořena pomocí upraveného algoritmu DFS a kombinace ACO, PSO a GWO.

Tabulka 6.7: Ohodnocení pomocí NN - referenční hodnocení

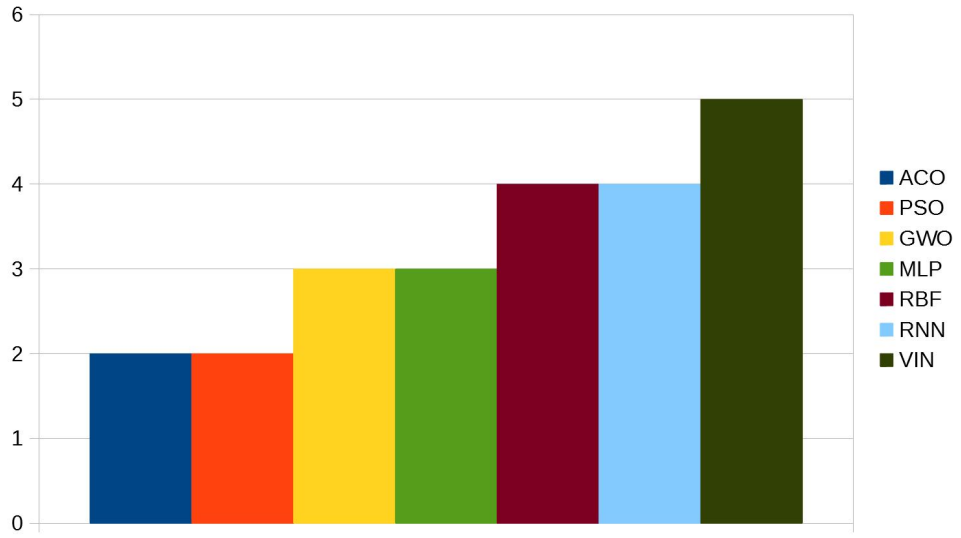
	MLP	RBF	RNN	VIN
Rychlost učení [s]	13 967.2	14 686.4	36 514	30 084.8
Rychlost výpočtu [s]	0.38	0.39	0.61	0.56
Ohodnocení/velikost 10 [známka]	5	5	5	5
Ohodnocení/velikost 100 [známka]	4	4	5	4
Ohodnocení/velikost 1000 [známka]	4	3	4	4
Ohodnocení/velikost 10000 [známka]	3	3	4	4

Ohodnocení/velikost X popisuje výsledné ohodnocení při velikosti množiny trénovacích dat X.

Nejrychlejší a nejjednodušší pro implementaci jsou sítě MLP a RBF. Obě tyto sítě mají nejvyšší rychlost učení a srovnatelné výsledky. Při detailnějším porovnání se zdá síť RBF být trochu přesnější, pravděpodobně díky skutečnosti, že místa s různým požárním rizikem tvoří přirozeně shluky.

Rekurentní neuronová síť má problémy s učením, protože pro efektivnější učení vyžaduje větší množství dat. Navíc nelze vyloučit, že síť se snaží na množinu trénovacích dat nahlížet jako na sekvenci, a díky paměti předpokládá ohodnocení ovlivněné předchozími hodnoceními.

Síť VIN dobře vyhodnocuje překážky díky integrovanému listu akcí pro detekci překážek, ale špatně počítá ohodnocení celého systému, protože je navržena primárně na navigaci.



Obrázek 6.21: Závěrečné ohodnocení výsledků algoritmů člověkem

Není překvapením, že výsledek neuronových sítí je výrazně horší než výsledek optimalizačních algoritmů, protože tyto algoritmy jsou zdrojem trénovacích dat pro neuronové sítě. Neuronové sítě se tedy učí z nepřesných dat.

Kapitola 7

Diskuse

Práce obsahuje mnoho míst a myšlenek, které by se daly vylepšit nebo rozvést. V tomto bodě bych rád upozornil, že rozsáhlá vylepšení nebyla předmětem této práce. Cíle práce byly stanoveny dost nízko, protože zkoumaná doména není typickým cílem, a je o ní známo jen velmi málo. Bylo tedy nutno nejprve vybudovat doménovou znalost, kterou by tato práce měla poskytnout dostatečnou, a prozkoumat základní myšlenky a principy. Teprve na základě těchto znalostí lze v budoucnu pracovat s problematikou více do hloubky, a vytvořit dobré a cílené metody na zvládnutí jednotlivých problémů řešených v této práci. Největším problémem však byly rozdílné požadavky uživatele a metodiky požární bezpečnosti. Zatímco uživatel očekává skvěle vypadající a efektivní vybavení domu (subjektivní), požární bezpečnost říká, že nejlepší je rozložit rizika, což ale často není efektivní z pohledu uživatele (např. hořlaviny rozmístěné v celém domě místo jednoho centrálního skladování). Je proto nutno hledat kompromis, což ale není vůbec snadné, je-li jeden ze dvou vstupů subjektivní.

7.1 Kvalita výsledků

Protože celá práce staví na předpokladech a obecných doporučeních, lze jen těžko hodnotit výstupy závazně a objektivně. Výsledky práce byly konzultovány s několika experty v oblasti požární bezpečnosti, a každý z nich navrhoval jiná a někdy i protichůdná vylepšení. Z toho lze usuzovat, že co odborník, to názor. Žádný z odborníků však neměl zásadní výhrady vůči výsledkům. Celkově tedy jejich názory lze shrnout jako fakt, že výsledky nejsou špatné, ale nelze určit jak moc jsou dobře. Veškerá vyhodnocování jsou tedy subjektivní, a je nutno k nim i takto přistupovat.

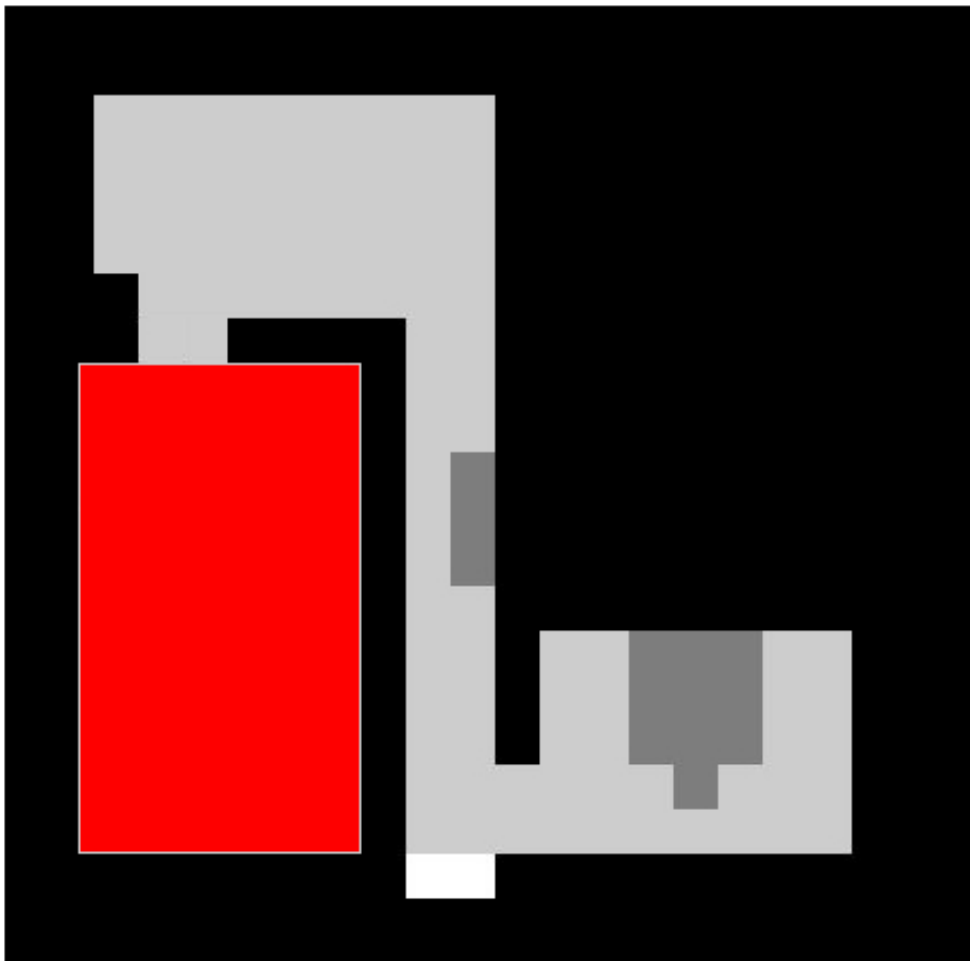
Model místnosti byl vypracován s ohledem na data i uživatele, a tato část je tedy nejvíce přesná z celé práce. Vyhodnocovat model místnosti lze tedy na základě UX a práce s daty při jednotlivých vyhodnoceních.

Optimalizace místnosti je úspěšná v podobě rozložení požárních rizik, ale nepraktická z pohledu reálného rozmístění vybavení. Není zde například uvažována přístupnost vybavení, pouze existence únikové cesty. Navíc jednotlivé kusy vybavení nejsou rozlišeny. Může se tedy stát, že algoritmus umístí jídelní stůl do rohu místnosti, kde budou dvě jeho strany nedostupné. Zrovna tak může umístit skříň, ke které je potřeba přistupovat pouze z jedné strany doprostřed místnosti. Proto tedy lze dobře vyhodnotit rozložení požárních rizik, ale zato dobře hodnocená rozložení nemusí být vůbec vhodná z hlediska požadavků lidí.

Nejvíce kontroverzní byla část ohodnocování bezpečnosti pro vytvořené modely. Jak již bylo uvedeno, každý z oslovených odborníků měl na výsledek jiný názor. Tyto názory neměly žádnou společnou základnu, která by se dala zapracovat. Proto všechna vyhodnocení jsou také subjektivní a vyhodnocená mnou jako autorem a osobou s dostatečnou doménovou znalostí. I přes toto omezení je výhodou výsledků upozornění na problém a zvýšení zájmu o požární bezpečnost. Navíc i částečně dobré výsledky usměrňují diskusi nad riziky a tedy plní účel upozornění na problémy.

7.2 Problematické vstupy

I přes snahu upravit algoritmy aby byly schopné fungovat za každé situace lze najít příklady, kde použité metody selhávají. Je to hlavně proto, že je nastavena vysoká přitažlivost cíle, a jeho návštěva je hlavní prioritou. Proto je-li místnost příliš blízko cíli a oddělena od něj zdivem, agenti mají snahu jít k cíli, a nedostanou se z místnosti.



Obrázek 7.1: Ukázka problematické mapy, červená zóna tvoří problém

7.3 Možná rozšíření

Možných rozšíření pro tuto práci je velké množství, a to na více místech. Z pohledu algoritmů strojového učení a umělé inteligence je prostor pro úpravy v programu například:

1. Simulace rozpadání budovy
2. Simulace kouře
3. Využití NN na simulaci evakuace
4. Přidat simulované agenty s různým chováním
5. Přidat simulaci zvládnutí požáru

6. Optimalizace pamětově náročnosti

Možnosti rozšíření pro použité algoritmy:

1. Podpora orientace v mísnosti - pro všechny optimalizační algoritmy
2. Předučené modely pro NN integrované v práci
3. Využití NN na simulaci evakuace

Dalším místem pro úpravy je frontend. Tyto úpravy už se přímo nedotýkají strojového učení a umělé inteligence, ale uživatelova pocitu z používání programu, neboli UX. Frontend pro základní odhady stačí, ale pro příjemnější práci by vyžadoval následující úpravy:

1. Nahrazení tlačítek pro zobrazení plánu budovy obrázkem, výpočet pozic nad obrázkem - zrychlení generování plánu
2. Click and hold kreslení nahrazující potřebu klikat na jednotlivá tlačítka plánu
3. Přidat parametrizaci algoritmů do UI
4. Přidat podporu různých typů objektů (dveře)
5. Přidat přímé zobrazování výsledků bez nutnosti otevírat externí prohlížeč obrázků
6. Přidat podporu načítání map z rysů a grafických výstupů z jiných aplikací
7. Umožnit drag and drop umísťování překážek
8. Přidat možnost definice místnosti a jejich pojmenování
9. Přidat možnost pojmenovat věci v inventáři
10. Přidat podporu orientace předmětů v inventáři (otevírání skříní)

Závěr

Cíle této práce byly tři: navrhnout model domů pro vyhodnocení, vytvořit nástroje pro optimalizaci místností z pohledu požární bezpečnosti, ohodnocení požárních rizik na vytvořeném modelu.

První z cílů, tedy vytvoření modelu, byl splněn již z počátku, kde byly postupně uvažovány všechny aspekty hořících domů. V úvahu byl vzat návrh domu samotného, šíření ohně, vybavení domu, nouzové evakuační cesty a nakonec též požadavky na data pro různé algoritmy. Na základě toho byl vytvořen model, který bez potřeby dalšího předzpracování dat slouží pro všechny algoritmy, a poskytuje dostatečnou představu pro uživatele.

Druhým cílem bylo vytvoření nástrojů pro optimalizaci místností, což bylo též splněno. Pro tyto účely bylo zvoleno několik základních algoritmů, které byly vzájemně porovnány. Také byla diskutována rizika spojená s optimalizací, jako výpočetní nároky nebo špatné vnímání navrhovaného využití z lidské strany. Přestože z hlediska rychlosti se ze zkoumaných algoritmů jevil jako nejlepší algoritmus B&B, jeho výsledky nebyly vždy nejlepší z hlediska lidského vnímání. Vylepšení jeho výstupů nebyly cílem této práce, protože neexistuje předem jednoznačné hodnocení výstupů, ale vše je hodnoceno subjektivně.

Třetím cílem bylo ohodnocení požárních rizik na vytvořeném modelu, a též tento cíl byl splněn. Ohodnocení bylo řešeno dvěmi způsoby - simulací a modelem. Simulace dává až na výjimku výsledky srovnatelné kvality a za podobný čas. Jako nejhorší se ukazuje ACO v původní podobě, protože než agenti najdou správnou cestu k východu, rozšíří se požár a feromonová stopa je slabá pro následování ostatními agenty. Vylepšená varianta už pracuje se směrováním lépe a efektivněji. PSO v základní podobě trpí podobným problémem, protože cílem práce bylo využívat pouze znalosti agenta bez nutnosti provádět rozsáhlé výpočty umístění (například agent za zdí je jednoduchým výpočtem bližší než agent na druhé straně stejné místnosti). Při vhodném počtu agentů se jako nejlepší jeví GWO, ale těžko tento počet lze určit předem. Krom simulací bylo využito i předem připravených modelů na ohodnocení. Tyto modely spočívají v naučených neuronových sítích. Jejich výhodnou je rychlost vyhod-

nocení, a poměrně dobrá přesnost. Jejich nevýhodou je nutnost naučení, a to pro různě velké budovy, takže pro každou kombinaci rozměrů modelu existuje jeden naučený model. Výsledky neuronových sítí se špatně vyhodnocovaly, protože jako svá trénovací data využívaly ohodnocení z předchozích metod, tedy na základě simulací.

Ze zadání práce byly splněny všechny úkoly, a to následovně:

- Seznamte se s problémem úniku z hořící místnosti a stávajícími přístupy k řešení této problematiky - Splněno v kapitolách Rešerše a Analýza.
- Navrhněte vhodný model pro reprezentaci hořící místnosti - Splněno v kapitole Návrh v podkapitole Nástroj pro tvorbu místností
- Identifikuje a po diskuzi s vedoucím práce vyberte algoritmy umělé inteligence vhodné pro řešení problému optimalizace únikových cest z místnosti - Spěno v kapitole Analýza v podkapitole Rozbor algoritmů vhodných pro řešení dílčích problémů
- Vybrané algoritmy implementujte ve vhodném programovacím jazyku - Spěno v kapitole Realizace v programovacím jazyce Python
- Srovnejte jednotlivé algoritmy mezi sebou z hlediska dosažených výsledků, včetně řešení navrženého člověkem - Splněno v kapitole Výsledky
- Dosažené výsledky diskutujte - Splněno v kapitole výsledky a Diskuse

Celkově byla práce úspěšná, ale vyhodnocení úspěchu podléhá stejné chybě jako všechna vyhodnocení v práci - jsou subjektivní. Tato skutečnost ovšem nevadí, protože před vytvořením práce nebylo jasno, zda existuje nějaké pevné hodnocení a možnost určit kvalitu výstupů. Hlavním cílem práce bylo získat doménovou znalost a vytvořit doporučení, ke kterým mohou uživatelé přihlížet, a to se povedlo.

Literatura

[1] Pokorný, M.: *Požární bezpečnost staveb. Sylabus pro praktickou výuku.* ČVUT, druhé vydání, 2018, ISBN 978-80-01-06394-1.

[2] Fire safety evaluation system (FSES) for business occupancies software. Jul 2000, [cit. 2017-12-28]. Dostupné z: `http://fire.nist.gov/bfrlpubs/fire00/PDF/f00101.pdf`

[3] Benichou, N.: FIERAsystem: A Fire Risk Assessment Tool to Evaluate Fire Safety in Industrial Buildings and Large Spaces. *Journal of Fire Protection Engineering*, ročník 15, č. 3, Jan 2005: str. 145–172, doi: 10.1177/1042391505049437, [cit. 2017-12-27].

[4] Fire risk analysis of residential buildings based on scenario clusters and its application in fire risk management. Oct 2013, [cit. 2017-12-27]. Dostupné z: `https://www.sciencedirect.com/science/article/pii/S0379711213001653`

[5] AION: 23/2008 Sb. Vyhláška o technických podmínkách požární ochrany staveb. Dostupné z: `https://www.zakonyprolidi.cz/cs/2008-23`

[6] Dracopoulos, D.: Robot path planning for maze navigation. *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98CH36227)*, doi:10.1109/ijcnn.1998.687180.

[7] Mannucci, T.; Kampen, E.-J. V.: A hierarchical maze navigation algorithm with Reinforcement Learning and mapping. *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2016, doi:10.1109/ssci.2016.7849365.

[8] Kislinger, M. R.: *Požárně technické charakteristiky a technické informace pro potřeby ZPP.* MV-generální ředitelství Hasičského záchranného sboru ČR, první vydání, 2015, ISBN 978-80-86466-72-9.

[9] Stochastic Fire Risk Modeling. *Evaluation of Fire Safety*, 2006: str. 399–420, doi:10.1002/0470020083.ch15.

[10] NOVOTNÝ, K.: *Lexikon BOZP - stavebnictví*. ROVS Brno, vlastní vydání vydání, 2016, 130 s.

[11] Nagy, D.: Design optimization – Generative Design – Medium. Mar 2017. Dostupné z: `https://medium.com/generative-design/design-optimization-2ec2ba3b40f7`

[12] Kozen, D. C.: Depth-First and Breadth-First Search. *The Design and Analysis of Algorithms*, 1992: str. 19–24, doi:10.1007/978-1-4612-4400-4_4.

[13] Tarjan, R.: Depth-First Search and Linear Graph Algorithms. *SIAM Journal on Computing*, ročník 1, č. 2, 1972: str. 146–160, [cit. 2018-05-03].

[14] Gross, D.; Soland, R. M.: A branch and bound algorithm for allocation problems in which constraint coefficients depend upon decision variables. *Naval Research Logistics Quarterly*, ročník 16, č. 2, 1969: str. 157–174, doi:10.1002/nav.3800160202.

[15] Cluster analysis. May 2018. Dostupné z: `https://en.wikipedia.org/wiki/Cluster_analysis`

[16] Ant Colony Optimization. *Ant Colony Optimization and Constraint Programming*, Jul 2013: str. 101–104, doi:10.1002/9781118557563.part2.

[17] Kennedy, J.; Eberhart, R.: Particle swarm optimization. *Proceedings of ICNN95 - International Conference on Neural Networks*, doi:10.1109/icnn.1995.488968.

[18] Kaipa, K. N.; Ghose, D.: Glowworm Swarm Optimization: Algorithm Development. *Glowworm Swarm Optimization Studies in Computational Intelligence*, 2017: str. 21–56, doi:10.1007/978-3-319-51595-3_2.

[19] Genetic Algorithm Implementation. *Swarm Intelligence*, 2001: str. 451–456, doi:10.1016/b978-155860595-4/50013-9.

[20] The Multi-Layer Perceptron Model. *A Statistical Approach to Neural Networks for Pattern Recognition*, Jul 2006: str. 9–18, doi:10.1002/9780470148150.ch2.

[21] Liu, J.: RBF Neural Network Design and Simulation. *Radial Basis Function (RBF) Neural Network Control for Mechanical Systems*, 2012: str. 19–53, doi:10.1007/978-3-642-34816-7_2.

[22] Mashhadany, Y. I. A.: Recurrent Neural Network with Human Simulator Based Virtual Reality. *Recurrent Neural Networks and Soft Computing*, 2012, doi:10.5772/35538.

[23] Tamar, A.; Wu, Y.; Thomas, G.; aj.: Value Iteration Networks. *Neural Information Processing Systems*, 2016.

[24] Ostfeld, A.: Ant Colony Optimization for Water Resources Systems Analysis – Review and Challenges. *Ant Colony Optimization - Methods and Applications*, Apr 2011, doi:10.5772/13522.

[25] Zhang, X.; He, B.; Niu, X.: Path planning algorithm of maze based on Ant Colony Optimization. *Information Science and Engineering (ICISE), 2010 2nd International Conference*, Dec 2010, doi:10.1109/ICISE.2010.5689289.

[26] Chen, Y.; Li, M.; Fan, Q.: Particle swarm optimization algorithm to solve the maze problem. ročník 31, 01 2010.

[27] Su, S.; Tsuchiya, K.: Learning of a maze using a genetic algorithm. *Proceedings of IECON 93 - 19th Annual Conference of IEEE Industrial Electronics*, doi:10.1109/iecon.1993.339049.

Příloha **A**

Seznam použitých zkratek

PoC Proof of concept - důkaz konceptu, jinak též experiment dokazující realizovatelnost projektu

DFS Depth first search - prohledávání do hloubky

BFS Breadth first search - prohledávání do šířky

B&B Branch and bounds - větve a hranice

ACO Ant colony optimalization - optimalizace mravenčím hejnem

PSO Particle swarm optimalization - optimalizace rojem částic

GWO Glow Worm optimalization - optimalizace hejnem světlušek

GA Genetic algorithm - genetický algoritmus

NN Neural network - neuronová síť

MLP Multi layered perceptron - vícevrstvý perceptron

RBFN Radial basis function network - síť s funkcí radiální báze

RNN Recurrent neural network - rekurentní neuronová síť

VIN Value iteration network - Síť iterující hodnoty

LOS Line of sight - dohled

UX User experience - uživatelův pocit z používání programu

Příloha **B**

Obsah přiloženého CD

```
readme.txt ..... stručný popis obsahu CD
run ..... adresář se spustitelnou formou implementace
  runme.txt ..... návod ke spuštění jednotlivých částí
src
  impl ..... zdrojové kódy implementace
    ui ..... zdrojové kódy UI
    optimize ..... zdrojové kódy optimalizační části
    evaluate ..... zdrojové kódy vyhodnocovací části
  thesis ..... zdrojová forma práce ve formátu LaTeX
text ..... text práce
  thesis.pdf ..... text práce ve formátu PDF
  thesis.ps ..... text práce ve formátu PS
examples ..... příklady výstupů práce
```