

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Modelování a simulace – Technická zpráva

T8: CA v dopravě

Obsah

1	Úvod	2
1.1	Účastníci a zdroje	2
1.2	Prostředí a validace	2
2	Rozbor tématu a použitých metod/technologií	2
2.1	Použité postupy	2
2.2	Použité technologie	3
3	Koncepce	3
3.1	Způsob vyjádření modelu	3
3.2	Popis konceptuálního modelu	3
4	Architektura simulačního modelu/simulátoru	4
4.1	Mapování konceptuálního modelu do simulátoru	4
4.2	Argumenty příkazové řádky	5
5	Podstata simulačních experimentů a jejich průběh	6
5.1	Postup experimentování	6
5.2	Dokumentace experimentů	7
5.2.1	Výsledky se základními parametry 10×20	7
5.2.2	Výsledky parkoviště 8×15	8
5.2.3	Výsledky parkoviště 10×17	9
5.2.4	Výsledky parkoviště 10×18	10
5.3	Závěry experimentů	11
6	Shrnutí simulačních experimentů a závěr	11

1 Úvod

Tato práce řeší projekt zaměřený na implementaci a simulaci modelu parkoviště. Projekt byl vyvíjen v rámci kurzu Modelování a Simulace. Cílem projektu je ukázat, jak parkování a strategie řidičů ovlivňují využití parkovacích ploch u obchodních domů.

Pomocí experimentů je v práci zkoumáno, jak různé konfigurace parkovišť s chováním řidičů ovlivňují efektivitu a využití parkovacích ploch.

1.1 Účastníci a zdroje

Autorem tohoto projektu je Ondřej Lukášek, student Fakulty informačních technologií Vysokého učení technického v Brně.

Pro vypracování projektu bylo využito článků a výzkumů z oblasti parkování u obchodních domů. Dále byly získány informace o průměrné vytíženosti obchodního domu, prostřednictvím jednoho ze zaměstnanců. To přispělo k možnosti přesnější simulace vytíženosti parkovacích ploch.

Všechny použité zdroje a odkazy jsou dostupné na konci tohoto dokumentu v části Použité zdroje, na straně 12.

1.2 Prostředí a validace

Vývoj byl prováděn za využití jazyka C, který umožňuje efektivně implementovat a simulovat celulární automaty.

Simulace a experimenty byly prováděny pod různými konfiguracemi, aby odrážely reálné scénáře parkování. Validita byla ověřena pomocí snímků¹ reálných parkovišť přilehlých u obchodních domů, čímž se zjistilo, že simulované parkoviště věrně napodobuje své reálné předlohy, čímž jsou využitelné v praxi, případně je možné je snadno upravit podle individuálních požadavků.

2 Rozbor tématu a použitých metod/technologií

Tato kapitola detailně popisuje rozbor systému parkování a obsahuje v sobě popis metod a technologií pro jeho modelování a simulaci.

2.1 Použité postupy

Model parkoviště a chování řidičů bylo vytvořeno na základě nastudovaných principů celulárních automatů [5], za využití Von Neumannova okolí, které bylo ideální pro hledání místa na parkovišti.

Samotné parkoviště je dále implementováno pomocí dvourozměrného pole o proměnlivé velikosti podle toho, což umožňuje dále ladit simulaci tak, aby se co nejvíce vyoptymalizovalo vytížení parkoviště a nebylo moc velké, případně moc malé. Je to důvodu, že v dvourozměrném poli se lze dobře orientovat pomocí souřadnic a není potřeba pracovat s nějakými složitými strukturama, čímž se toto řešení jeví jako nejideálnější.

Průměrné parkovací časy byly vybrány na základě již provedených výzkumů [2] [3] [7], což zlepšuje validitu celého modelu, nicméně jsou přebrány z různých částí světa, tudíž je pro lokální využití vhodnější využít soukromé průzkumy uživatele.

Typy řidičů, které jsou implementovány v programu jsou převzaté z praxe, aby jejich chování více zpřesňovalo simulační model. Některým řidičům nezáleží na tom, zda parkují blízko jiných. Někteří chtějí okolo sebe při parkování více místa, jiní zase místo kolem sebe vyloženě vyžadují.

¹ Snímky byly prohlíženy na <https://www.google.com/maps/>, platných k 9.12.2023.

2.2 Použité technologie

Pro vytvoření simulačního modelu bylo využito jazyka C, spolu s využitím některých z jeho standardních knihoven².

Auta jsou implementována formou struktury, obsahující důležité informace, jako čas příjezdu, odjezdu, a podobně. Struktury jsou v jazyce C běžně využívány.

Kromě výše zmíněného dvourozměrného pole bylo také využito jednosměrně vázaného seznamu [6] pro vytvoření seznamu vozidel, která jsou aktuálně obsažena v systému.

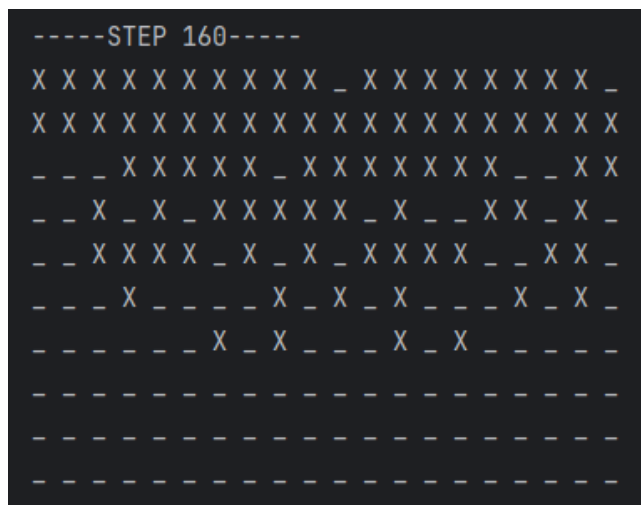
3 Koncepce

Tato kapitola se zaměřuje na abstrakci reality do formy modelu a popisuje, jak se promítly vlastnosti reálného světa do simulačního modelu.

3.1 Způsob vyjádření modelu

Model parkování je reprezentován dvourozměrným celulárním automatem, jehož základní strukturu lze vidět na Obrázku 1. Každé parkovací místo je reprezentováno jako buňka v mřížce, která může nabývat dvou různých stavů: obsazeno (`true`), volno (`false`). Způsob parkování řidičů je modelován pomocí pravidel, která jsou se aplikují na určitá místa, na základě buněk parkoviště (parkovacích míst) a jejich okolí.

Obsazení parkoviště se neustále mění. Auta na parkoviště přijíždějí (tyto časy jsou převzaty z podkladů získaných od jednoho ze zaměstnanců obchodního domu, jak je zmíněno v kapitole 1.1) a odjíždějí v různé časy (auta parkují na parkovišti různou dobu [2] [3] [7]), což napomáhá větší autentičnosti modelu.



Obrázek 1: Dvourozměrný model parkoviště

3.2 Popis konceptuálního modelu

V tomto modelu jsou parkovací místa buňky, které v čase mění svůj stav podle toho, kde v parkovišti se nacházejí a kdy přijíždějí o odjíždějí vozidla, snažící se zaparkovat na parkovací ploše, a jaký je aktuální typ tohoto vozidla.

Model bere v úvahu různé typy vozidel (respektive jejich řidičů) a jejich preference při parkování. Typ vozidla je generován při příjezdu do systému na základě pravděpodobnostního určení. Zároveň vozidla zahrnuje

²stdio, stdlib, time, stdbool, unistd. Seznam všech standardních knihoven s jejich popisy je zde: <https://en.cppreference.com/w/c/header>

i nějakou trpělovist řidiče, která je vyjádřena v parkovacích pokusech, které je schopné vozidlo uskutečnit. Pokud se nepodaří vozidlu zaparkovat do jemu individuálně vygenerovanému počtu pokusů, poté se mohou jeho pravidla parkování změnit (vizte senzám typů vozidel v kapitole 3.2), případně opouští systém. Typy vozidel jsou následující:

1. Zkušený řidič - vyhledává jakékoliv jedno volné místo, pokud možno co nejbližší vchodu do obchodního domu. Nezáleží na tom, jak vypadají místa okolo.
2. Opatrný řidič - vyhledává parkovací místo na základě toho, jestli je vedle volného parkovacího místa volné ještě jedno. Pokud ano, potom tam zaparkuje. Pokud se mu nepodaří na parkovišti zaparkovat, na svůj poslední parkovací pokus zaparkuje na jakékoli volné místo, co nejbližší ke vchodu.
3. Nový řidič - vyhledává alespoň 3 volná místa vedle sebe a parkuje do prostředního z nich. Pokud se mu ani v jednom z jeho parkovacích pokusů nepodaří zaparkovat, opouští systém.

Neopomenutelnou součástí modelu je také způsob, jakým jsou vozidla generována a odstraňována ze systému, což odpovídá realistickému příjezdu a odjezdu vozidel v reálném světě, přičemž tyto údaje o parkování vycházejí z reálných průzkumů [2] [3] [7]. Tento aspekt modelu je zásadní pro simulaci skutečných podmínek parkování a umožňuje analýzu vlivu různých faktorů, jako je velikost parkoviště, na celkovou efektivitu, kapacitu a zatížení parkovací plochy.

4 Architektura simulačního modelu/simulátoru

Tato kapitola se zaměřuje více konkrétně na technické prvky simulačního modelu a především tomu, jak je konceptuální model mapován do konkrétních forem simulátoru.

4.1 Mapování konceptuálního modelu do simulátoru

Model ve svém základu pracuje se strukturou parkoviště, pro kterou je alokována paměť a je inicializována funkcí `initializeParkingLot()`, na konci programu je alokovaná paměť uvolněna pomocí funkce pro vyčištění paměti `clearParkingLot()`.

Následně se spustí simulace v čase, která probíhá podle počtu zvolených kroků simulace, kdy jeden krok odpovídá přibližně jedné minutě reálného světa. Dobu simulace si může uživatel nastavit pomocí argumentů příkazové řádky (kapitola 4.2).

V každém kroku simulace se v systému generuje náhodný počet aut od 3 do 4 v průměru. To vychází ze získaných informací od jednoho ze zaměstnanců (kapitola 1.1), který sdělil, že jedním z obchodních domů³ projde za den přibližně mezi 2700 až 3100 zákazníky, což přibližně odpovídá 3-4 autům za minutu při otevírací době 7-22 hodin ($7 - 22 = 15$ otevíracích hodin, $15 * 60 = 900$ otevíracích minut, což znamená $2700/900 = 3$ až $3300/900 \approx 3.67 \approx 4$ auta za minutu).

Každému z aut se při vygenerování přiřadí náhodně typ (vizte kapitolu 3.2), maximální počet pokusů pro zaparkování (což signalizuje trpělivost řidiče). Dále se zapisuje čas příjezdu (ve kterém čase simulace přijelo auto do systému) a jak dlouho bude v systému setrvávat, pakliže se mu podaří zaparkovat (doba parkování je vybrána náhodně podle časů získaných z průzkumů [2] [3] [7]).

Zároveň se také při každém generování auta inkrementuje počítadlo aut, která přijela, což je užitečné pro získání statistik.

Po příjezdu vozidel do systému se vozy pokusí postupně zaparkovat (funkce `attemptToPark()`, která dále volá funkci `parkCar()`). Vozidla mohou mít různé typy, které jsou detailněji popsány v kapitole 3.2 a typ vozu definuje pravidla, kterými se bude při parkování řídit. Pokud se autu nepodaří zaparkovat, inkrementuje

³Jedná se konkrétně o Kaufland Kuřim, na ulici Blanenská

se jeho počítadlo pokusů o zaparkování. Pokud se zaparkovat podaří, potom se buňka (souřadnice) parkoviště nastaví jako obsazená (`true`) a ve struktuře auta se proměnná `parked` nastaví na `true`.

Následně program projde seznam všech vozidel (funkce `updateCars()`) a vyřadí ze seznamu všechna vozidla, kterým vypršel čas, který jim byl vyhrazen na parkovací dobu. Také jsou ze seznamu (a tedy i ze systému) vyřazena všechna auta, která přesáhla svůj maximální počet pokusů o zaparkování.

V této funkci se také inkrementuje počítadlo aut, kterým se nepodařilo zaparkovat, avšak inkrementuje se pouze v případě, kdy nějaké z vozidel přesáhlo svůj maximální počet pokusů o zaparkování. Toto počítadlo je důležité pro koncové statistiky, protože podává informaci o tom, kolika vozidlům se vůbec zaparkovat nepodařilo, a tak je možné zkusit příští běh simulace s jinými parametry (například s velikostí parkoviště)(vizte kapitolu 4.2).

Každých 10 běhů simulace se také spustí funkce `printStatus()`, která vykreslí aktuální stav parkoviště na standardní výstup programu (vizte Obrázek 1). Na obrázku znázorňuje znak `X`, že je parkovací místo obsazené. Znak `-` zase znázorňuje, že místo je volné.

Po proběhnutí všech kroků simulace se na standardní výstup vypíše statistiky pomocí funkce `printStats()`. Mezi tyto statistiky patří konfigurace parkoviště (celkový počet míst, počet řádků, počet sloupců), počet vozidel, která vstoupila do systému, počet vozidel, kterým se v průběhu simulace nepodařilo zaparkovat a opustila systém předčasně, a počet aut, kterým se zaparkovat podařilo. Příklad tabulky statistik lze vidět na obrázku 2.

```
===== SIMULATION STATISTICS =====
Configuration:           10 rows      20 columns
Number of parking spots: 200
Created cars:            629
Expired cars:             0
Parked cars:             629
Average parking attempts: 0.00
```

Obrázek 2: Příklad tabulky statistik

4.2 Argumenty příkazové řádky

Program umožňuje rychlé nastavení klíčových prvků simulace, jako je počet kroků simulace, konfigurace parkoviště (nastavení počtu řádků a sloupců), nastavení vchodu do obchodního domu (určí se, které parkovací místo z prvního řádku je ke vchodu nejbližší), což umožňuje větší flexibilitu a použitelnost pro různé obchodní domy. Nakonec je také program schopen vypsát nápovědu k jeho využití.

- `-s` – počet kroků simulace (celé, kladné číslo). Pokud není zadáno, je v základu nastaveno 180 kroků (3 hodiny).
- `-r` a `-c` – počet řádků a sloupců parkovací plochy (celé, kladné číslo). Musí být zadány oba současně. Pokud nejsou zadány, konfigurace parkoviště je základně 10 řádků a 20 sloupců.
- `-e` – vstup na parkoviště (celé, kladné číslo). Označuje sloupec v prvním řádku parkovacích míst, ke kterému je vstup nejbližší. Pokud není zadán, vstup do obchodního domu je nastaven na střed, což znamená `pocet_sloupcu/2`.
- `-h` – Vypíše nápovědu na standardní výstup, pokud je zadán.

5 Podstata simulačních experimentů a jejich průběh

Cílem následujících experimentů je optimalizovat rozměry parkoviště pro scénář, kde v průměru přijíždí 3-4 auta za simulační krok v simulaci trvající 15 hodin a dobu setrvání na parkovišti 20-40 minut, jako to odpovídá realitě. Experimenty byly zaměřeny na zjištění, jaké rozměry parkoviště nejlépe vyhovují těmto podmínkám a jak se parkoviště chová v různých konfiguracích.

5.1 Postup experimentování

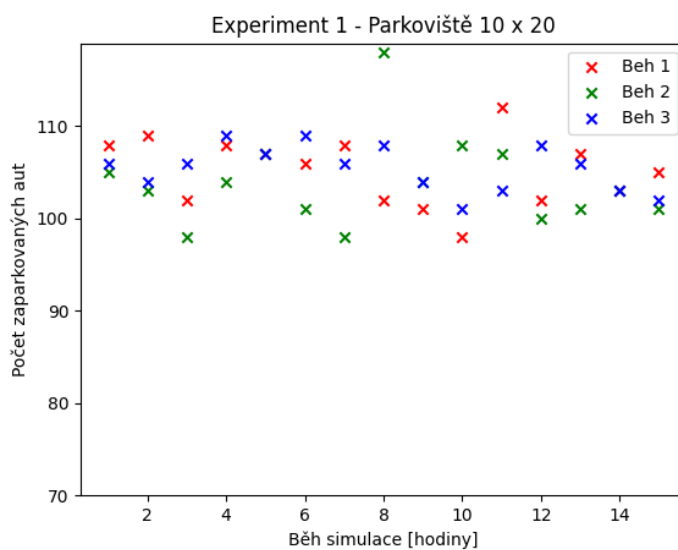
Experimenty byly postaveny na postupném upravování rozměrů parkoviště a sledování efektivity parkování vozidel. Začalo se s výchozími rozměry (tedy 10 řádků a 20 sloupců), které byly postupně měněny s cílem zvýšit efektivitu využití prostoru, zkrátit dobu hledání volného místa pro přijíždějící vozidla a pokud možno eliminovat množství aut, kterému se nepodařilo zaparkovat. Zároveň ale žádoucí, aby parkovací plocha nebyla příliš velká, protože nákup jakékoli plochy je nákladný. V každém experimentu bylo zaznamenáno, kolik aut úspěšně zaparkovalo, kolik opustilo systém kvůli neúspěšnému hledání parkovacího místa a jak dlouho trvalo najít místo. Také byl každý experiment proveden třikrát a byly z něj zaznamenány grafy pro každou hodinu běhu. Grafy byly generovány v jazyce Python za použití knihovny matplotlib [4].

5.2 Dokumentace experimentů

5.2.1 Výsledky se základními parametry 10×20

Tento experiment proběhl bez jakýchkoliv potíží, každému vozidlu se podařilo zaparkovat na první pokus a u žádného z nich se nestalo, že by odešlo ze systému na základě vyprchaného počtu pokusů o zaparkování. Po extrakci průměrných množství zaparkovaných aut z Obrázku 3 vychází, že v průměru se na parkovišti nacházelo 105 aut.

Lze tedy usoudit, že parkoviště je možné ještě optimalizovat a trochu zmenšit, proto se v dalším experimentu parkoviště zmenší ve svém rozměru.



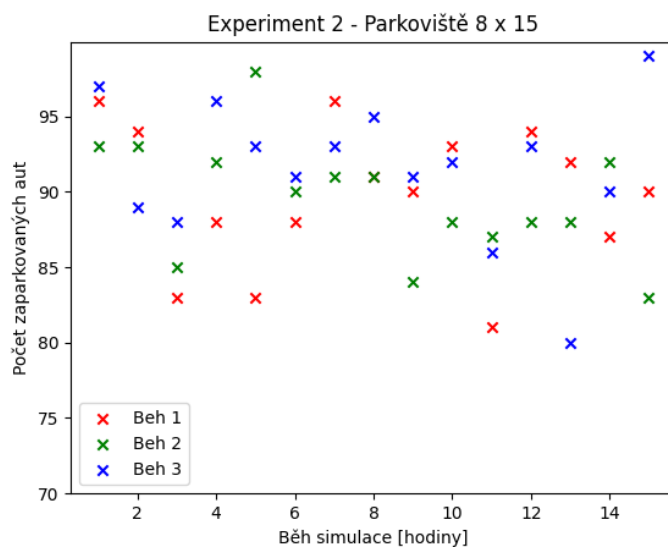
Obrázek 3: Experiment se základními parametry

5.2.2 Výsledky parkoviště 8×15

Zde byly výsledky poměrně nepříznivé. Oproti předchozímu experimentu bylo míst výrazně méně. Z 200 míst se počet snížil na pouhých 120. To se také výrazně promítlo na obsazených místech parkoviště. Obsazená místa nyní v žádném z experimentů nepřesáhla 100 míst (vizte Obrázek 4), protože průměrně byly ze 120 míst obsazeno pouhých 90 míst.

Také by bylo na místě si nyní zmínit průměrný počet aut, která byla v systému vytvořena. Těch bylo v průměru ze všech 3 běhů 3148. Z těchto aut v průměru nezaparkovalo 462, což znamená, že přibližně 14.67% aut na parkovišti vůbec nezaparkovalo.

Lze tedy jednoznačně usoudit, že parkoviště musí být opět o něco zvětšeno v dalším experimentu.

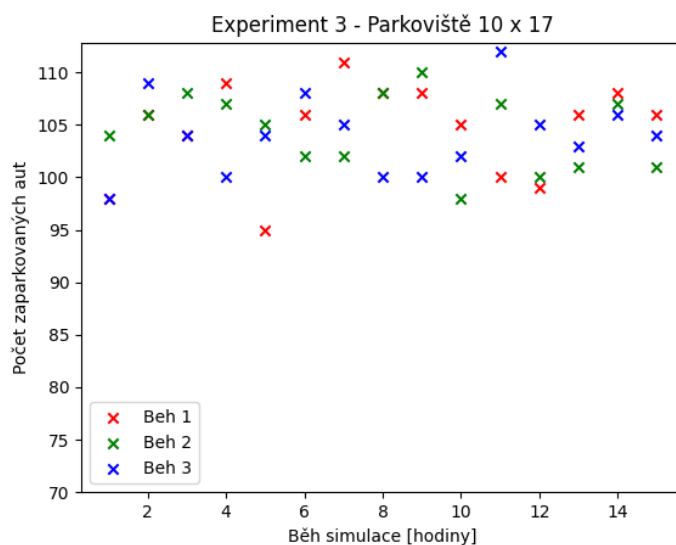


Obrázek 4: Experiment s rozměry 8×15

5.2.3 Výsledky parkoviště 10×17

V tomto experimentu byly oproti předchozímu výsledky opět o něco příznivější (jak se dalo očekávat). Počet vozidel na parkovišti znovu běžně překračoval 100 a průměrně množství zaparkovaných aut v simulaci bylo 104 (vytížení v jednotlivých hodinách lze pozorovat na Obrázku 5).

Nicméně se ani zde nejedná o naprosto přesvědčivý výsledek, protože průměrně se za jeden běh nepodařilo vůbec zaparkovat 5 vozidlům, což je přibližně 0.16% ze všech aut v systému. Toto číslo stále není v naprostém souladu s cílem těchto experimentů (vizte kapitolu 5), a je tedy potřeba parkoviště ještě o něco zvětšit, nicméně už to stačí jen velmi málo, aby byla úspěšnost zaparkování 100%.

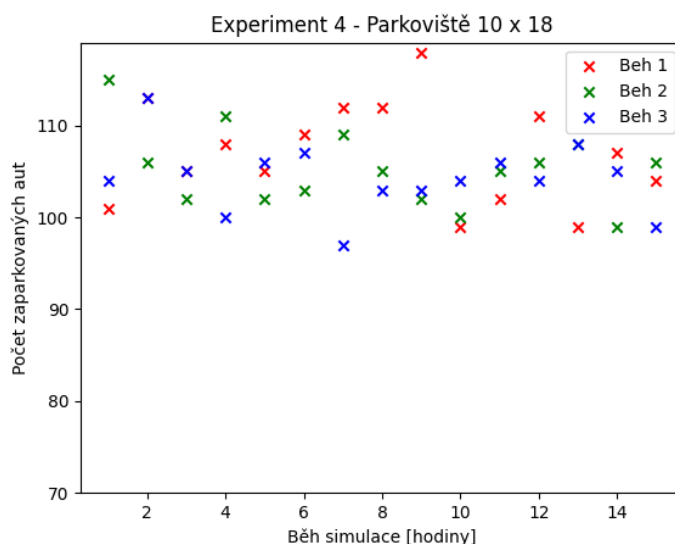


Obrázek 5: Experiment s rozměry 10×17

5.2.4 Výsledky parkoviště 10×18

Tento experiment už se jeví jako ideální, protože se opět, stejně jako v prvním experimentu (kapitola 5.2.1), podařilo zaparkovat všem vozidlům, která vstoupila do systému. Těchto vozidel bylo v průměru za jeden běh 3149 (vytíženost v jednotlivých hodinách lze vidět na Obrázku 6).

Na první pohled jsou čísla velice podobná, jako při prvním experimentu, protože je parkovací plocha zúžená o pouhé 2 sloupce. Nicméně to při průměrné velikosti parkovacího místa $5 \times 2,3$ metrů čtverečních [1] dělá ušetření přibližně 230 čtverečních metrů, což není při ceně stavební plochy malý rozdíl.



Obrázek 6: Experiment s rozměry 10×18

5.3 Závěry experimentů

Celkově byly provedeny 4 experimenty. Všechny byly odsimulovány s různou konfigurací parkoviště a postupně se blížily k ideálnímu výsledku, stanoveném v kapitole 5. K tomuto výsledku se došlo v posledním, čtvrtém experimentu (kapitola 5.2.4), který se svou konfigurací demonstroval ideální výsledky.

6 Shrnutí simulačních experimentů a závěr

Zjištění z výsledků simulačních experimentů ukazují, že parkoviště o rozměrech 10×18 poskytuje nejlepší vyváženost mezi kapacitou a efektivním využitím prostoru. Tento výsledek je relevantní zejména pro lokace, kde je jedním ze stěžejních úkolů efektivní vytížení omezených prostor.

Program by bylo možné i nadále ladit a upravovat. Například by v případě reálného využití v praxi zákazníkem bylo vhodné, aby byly do programu implementovány hodnoty jeho vlastních průzkumů, neboť tento simulační model funguje na obecné rovině, která vychází z veřejných zdrojů (ty jsou shrnuté v kapitole 6). Doplnění hodnot a úprava programu na základě vlastních soukromých šetření by přispěla k jeho největší přesnosti pro individuální využití.

Použité zdroje

- [1] What are the dimensions of a parking space? May 2022. Dostupné z: <https://www.interparking-france.com/en/what-are-the-dimensions-of-a-parking-space/>
- [2] Arndt, J.; Gronmo, S.: The time dimension of shopping behavior: Some empirical findings: ACR. Jan 1977. Dostupné z: <https://www.acrwebsite.org/volumes/9359/volumes/v04/NA-04>
- [3] Bawa, R.; Sinha, A. K.; Kant, R.: Emerging mall culture and shopping behavior of young consumers. Jun 2019. Dostupné z: <https://www.scirp.org/journal/paperinformation?paperid=93089>
- [4] Kumar, N.: Graph plotting in Python: Set 1. Jul 2023. Dostupné z: <https://www.geeksforgeeks.org/graph-plotting-in-python-set-1/>
- [5] Periger, P.; Hrubý, M.: Modelování a simulace. Sep 2023. Dostupné z: <http://www.fit.vutbr.cz/study/courses/IMS/public/prednasky/IMS.pdf>
- [6] pinkikrjad: Introduction to linked list - data structure and algorithm tutorials. Nov 2023. Dostupné z: <https://www.geeksforgeeks.org/introduction-to-linked-list-data-structure-and-algorithm-tutorial/>
- [7] Rodgers, E.: Grocery Store Statistics: Where, when, & how much people grocery shop. Apr 2023. Dostupné z: <https://www.driveresearch.com/market-research-company-blog/grocery-store-statistics-where-when-how-much-people-grocery-shop/>