TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta strojní

Optimalizace procesů výukového modelu chytré továrny s principy Průmyslu 4.0.

Diplomová práce

Studijní program: N2301 – Strojní inženýrství
Studijní obor: 2301T049 – Výrobní systémy a procesy

Autor práce: **Bc. František Manlig**
Vedoucí práce: Ing. František Koblasa, Ph.D.

Liberec 2019

Technická univerzita v Liberci
Fakulta strojní
Akademický rok: **2018/2019**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. František Manlig**
Osobní číslo:	**S17000230**
Studijní program:	**N2301 Strojní inženýrství**
Studijní obor:	**Výrobní systémy a procesy**
Název tématu:	**Optimalizace procesů výukového modelu chytré továrny s principy Průmyslu 4.0.**
Zadávající katedra:	**Katedra výrobních systémů a automatizace**

Zásady pro vypracování:

Zásady pro vypracování:

1. Analýza principů umělé intelligence, chytré továrny a Průmyslu 4.0 s ohledem na optimalizovaný výukový model.
2. Navržení optimalizace řízení jízdy vozíčku založené na sledování černé čáry.
3. Navržení optimalizace výrobního postupu kompletace náramku s využitím více vozíčků a více zásobníků pomocí vybraných prvků umělé intelligence.
4. Aplikace vybraného algoritmu na řídicím systému na platformě Arduino.

Rozsah grafických prací:

Rozsah pracovní zprávy: **50-60 stran**

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

[1] Začínáme s Arduinem: příručka. In: Snail Instruments: www.hobbyrobot.cz [online]. 2014 [cit. 2015-01-09]. Dostupné z:http://www.snailshop.cz/literatura/1537-zaciname-s-arduinem-prirucka.html.
[2] Arduino Learning: Getting Started with Arduino. In: Arduino [online]. 2014 [cit. 2015-01-09]. Dostupné z: http://arduino.cc/en/Guide/HomePage.
[3] BURIAN, P. Webové a agentové technologie. Praha: Grada, 2012. Průvodce (Grada). ISBN 978-80-247-4376-9.
[4] Automated guided vehicle systems. New York: Springer, 2014. ISBN 9783662448137.

Vedoucí diplomové práce: **Ing. František Koblasa, Ph.D.**
Katedra výrobních systémů a automatizace

Konzultant diplomové práce: **Ing. Radek Votrubec, Ph.D.**
Katedra výrobních systémů a automatizace

Datum zadání diplomové práce: **15. listopadu 2018**

Termín odevzdání diplomové práce: **15. května 2020**

prof. Dr. Ing. Petr Lenfeld
děkan

L.S.

Ing. Petr Zelený, Ph.D.
vedoucí katedry

V Liberci dne 15. listopadu 2018

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že texty tištěné verze práce a elektronické verze práce vložené do IS STAG se shodují.

26. 4. 2019 Bc. František Manlig

Poděkování

Tímto bych chtěl poděkovat všem učitelům a lidem, kteří mi pomohli s touto diplomovou prací dojít ke zdárnému konci.

Abstrakt

Tato diplomová práce se zaměřuje na dvě problematiky. První problematikou je optimalizace řízení jízdy vozíčku založené na sledování černé čáry. Druhá část se zaměřuje na návrh optimalizace výrobního postupu kompletace náramku s využitím více zásobníků a vozíčků.

Řízení vozíčku založené na sledování černé čáry je pomocí třech senzorů. Prostřední senzor detekuje, zda je vozíček na čáře a dva boční detekují hranu čáry a korigují vozíček zpět na čáru. Ve výpočetním programu byla provedena analýza vlivu vstupních parametrů na jízdní vlastnosti. Mezi vstupní parametry patří vysunutí senzorů, rozteč bočních senzorů, tloušťka čáry a minimální křivost čáry v zatáčkách. Následně byly provedeny ověřovací praktické simulace / jízdy. Závěrem je doporučení pro tvorbu dráhy a tvorbu vozíčků.

Optimalizace výrobního postupu kompletace náramků je složitou problematikou. V průmyslu se využívá metoda nejkratšího procesního času. Tato metoda má však velké nedostatky, proto se přechází na heuristické algoritmy. Nejprve byla provedena analýza heuristických metod, které spadají do umělé inteligence a využívají se pro tyto účely v Průmyslu 4.0. Využit byl genetický, neboli evoluční algoritmus. Vycházelo se z projektu na katedře výrobních systémů a automatizace, kde se touto problematikou zabývají. Cílem práce bylo navrhnout optimalizaci části tvorby nových populací. Nejprve byla provedena analýza využívaných metod, následovala analýza principů genetiky a obecně inspirace přírodou, ze které tyto algoritmy vycházejí. Závěrem bylo navrženo optimalizační řešení. Řešení vychází z pochopení problematiky tvorby nových populací, a že je vhodné křížit začátek chromozomu v prvních generacích a konec chromozomu ke konci všech vypočítávaných generací. Návrh byl otestován a porovnán s jednoduchým genetickým algoritmem (SGA) a s metodou nejkratšího procesního času (SPT). Navrhovaný memetický genetický algoritmus (MGA) vykazuje oproti SGA rychlejší zlepšování populace. Je doporučeno v optimalizaci MGA pokračovat a ověřovat ho na složitějších kombinatorických problémech.

Klíčová slova: optimalizace, sledování černé čáry, jízda po černé čáře, heuristické algoritmy, genetické algoritmy, evoluční algoritmy, umělá inteligence, průmysl 4.0

Summary

This thesis focuses on two issues. The first issue is the optimization of wheelchair driving based on black line tracking. The second part focuses on the design optimization of the manufacturing process of bracelet assembly using multiple cartridges and carts.

Controlling a black-line tracker is using three sensors. The center sensor detects whether the carriage is on the line and the two side lines detect the line edge and correct the carriage back to the line. In the computational program, an analysis of the influence of input parameters on driving properties was performed. Input parameters include sensor ejection, side sensor spacing, line thickness, and min. curvature of a line in curves. Subsequently verification practical simulations were performed. The conclusion is a recommendation for the creation of the track and the creation of wheelchairs.

Optimizing the manufacturing process of bracelet assembly is a complex issue. The industry uses the shortest process time method. However, this method has large inadequacies, so it is changing to heuristic algorithms. Firstly, heuristic methods were analyzed, which fall into artificial intelligence and are used for these purposes in Industry 4.0. A genetic or evolutionary algorithm was used. It was based on the project at the Department of Production Systems and Automation, where they deal with this issue. The aim of this work was to propose optimization of the creation of new populations. At first, the methods used were analyzed, followed by analysis of the principles of genetics and generally the inspiration of nature from which these algorithms are based. Finally, an optimization solution was proposed. The solution is based on the understanding of the issue of the formation of new populations, and it is advisable to cross the beginning of the chromosome in the first generation and the end of the chromosome to the end of all calculated generations. The design was tested and compared with a simple genetic algorithm (SGA) and the shortest process time method. The proposed memetic genetic algorithm (MGA) shows a faster population improvement than SGA. It is recommended to continue to optimize MGA and validate it on more complex combinatorial issues.

Keywords: optimization, black line tracking, black line driving, heuristic algorithms, genetic algorithms, evolutionary algorithms, artificial intelligence, industry 4.0

:

Obsah

1 Úvod....17

2 Systémy autonomního zásobování....19

3 Návrh optimalizace jízdy vozíčku....20

3.1 Principy řízení vozíčků....20

3.1.1 Vstupní parametry....20

3.1.2 Souhrn a řešený případ....25

3.2 Metody pro analýzu....26

3.3 Analýza....28

3.3.1 Simulace v matlabu....28

3.3.2 Reálná jízda vozíčkem....36

3.3.3 Porovnání výsledků reál vz. simulace....38

3.4 Hodnocení vlivu vstupujících parametrů....39

4 Principy umělé inteligence a Průmyslu 4.0....41

4.1 Ověřování umělé inteligence....42

4.2 Neuronové sítě....43

4.3 Evoluční a genetické algoritmy....44

4.4 Multiagentní systémy (MAS)....46

5 Optimalizace výroby náramku s využitím více vozíčků....48

5.1 Model....48

5.2 Návrh evolučního algoritmu....51

5.2.1 Definování vstupních parametrů....51

5.2.2 Generování první populace....52

5.2.3 Ohodnocení kvality jedinců....52

5.2.4 Výběr rodičů (selekce)....53

5.2.5 Tvorba potomků/jedinců 56
5.2.6 Mutace nových jedinců 59
5.2.7 Eliminace populace 60
5.3 Jednoduchý GA (SGA) 61
5.4 Memetický GA (MGA) 62
5.5 Aplikace a zhodnocení 65
6 Závěr 68

Seznam obrázků

Obr. 1 Různé typy vozíčků 21

Obr. 2 Vývojový diagram pro logiku řízení a) LP, b) LPR 26

Obr. 3 Výstup simulace při zadané rovné dráze 30

Obr. 4 Kvalita výstupů dle složitosti dráhy.....30

Obr. 5 Pohyb vozíčku při různé rozteči senzorů a vysunutí 31

Obr. 6 Vliv rozteče a vysunutí senzorů na min. poloměr jízdy vozíčku.....32

Obr. 7 Pohyb vozíčku v zatáčce s rozdílnou roztečí senzorů vůči tloušce čáry 33

Obr. 8 Rozdíl jízdy vozíčku při odlišné logice řízení a) LP b) LPR 34

Obr. 9 Natočení vozíčku v kontaktu senzoru s čárou - natočení a) malé b) velké.....34

Obr. 10 Vývojový diagram pro logiku řízení LPR 36

Obr. 11 Foto z reálné jízdy vozíčkem.....37

Obr. 12 Neuron 43

Obr. 13 Struktura neuronové sítě.....44

Obr. 14 Výrobní hala 49

Obr. 15 Proces genetického algoritmu.....51

Obr. 16 Vliv mutace jedinců.....59

Obr. 17 Posun pravděpodobnosti křížení částí chromozomu v průběhu generací.....64

Obr. 18 Ukázka jedinců v průběhu generací MGA 65

Obr. 19 Výsledný průběh zakázek ve výrobním procesu 67

Obr. 20 Výsledné uspořádání zakázek na strojích 67

Obr. 21 Porovnání SGA a MGA.....69

Seznam tabulek

Tab. 1 Třídění logiky řízení dle možného pohybu vozíčku 23

Tab. 2 Porovnání metod pro analýzu vstupních parametrů jízdních vlastností 27

Tab. 3 Porovnání logiky řízení LP s logikou LPR 35

Tab. 4 Měření doby jízdy před a po optimalizaci 38

Tab. 5 Hodnocení rozteče a vysunutí senzorů na charakter jízdy 39

Tab. 6 Nárůst času výpočtu s nárůstem problematiky 41

Tab. 7 Uskupení zakázek použité pro simulaci 48

Tab. 8 Vzdálenosti mezi stanovišti v sekundách 50

Tab. 9 Přiřazení ID operaci, tvorba náhodného jedince 52

Tab. 10 Seznam zakázek a jedinec řešený pro hodnotu fitness funkce 53

Tab. 11 Křížení jedinců 57

Tab. 12 Oprava potomků 58

Tab. 13 SGA souhrn vstupních parametrů 62

Tab. 14 MGA - Potřebná pravděpodobnost ke křížení dle generace a části chromozomu 64

Tab. 15 MGA souhrn vstupních parametrů 65

Tab. 16 Data z testů SGA a MGA 66

Seznam zkratek

AGV	Automaticky řízené vozidlo (Automated guided vehicle)
CPS	Kyberneticko-fyzikální systémy (Cyber-Physical Systems)
GA	Genetické algoritmy
HW	Hardware
JIT	Doprava materiálu v požadovaný čas (Just in Time)
LP	jízda doleva či doprava
LPR	jízda doleva, doprava či rovně
MAS	Multiagentní systémy
MGA	Memetický genetický algoritmus (individuální optimalizace či učení)
M2M	Komunikace mezi stroji (machine-to-machine)
SF	Software
SGA	Jednoduchý genetický algoritmus (Simple Genetic Algorithm)
SMART Manufacturing	Inteligentní továrna
SPT	Nejkratší procesní čas (Shorter processing time)
TSP	Problém obchodního cestujícího (Travelling Salesman Problem)
UI	Umělá inteligence (AI Artificial Intelligence)
VRP	Vehicle Routing Problem
1P	pro 1 kolo pohon + podpůrné
2P	pro 2 kola + podpůrné
4P	pro 4 kola a natáčení předních kol
4Z	pro 4 kola a natáčení zadních kol
4V	pro 4 kola a natáčení všech 4 kol
4S	pro 4 kola a natáčení všech 4 kol (speciální varianty)

1 Úvod

Od 18. Století, kdy přišla 1. průmyslová revoluce, prochází výrobní procesy velmi rychlými a zásadními změnami. Obecně lze říci, že množství produktů stejného typu rostlo do nasycení trhu a následně klesá a bude klesat i nadále. Dalším stejným specifikem je optimalizace, zrychlení a zjednodušení všech procesů. Rozdílná je především situace, ve které jednotlivé průmyslové etapy žily. Problémy s vývojem, pokrokem a konkurencí byly však stále. Dnes se jen vše žene do extrémů. Trh je přesycený, konkurence obrovská, změny se dějí rychleji a koloběh produktu je příliš rychlý. Vše má zásadní vliv na přírodu a celé lidstvo. Proto se i mnohem více řeší životní prostředí a další oblasti jako jsou problematiky sociálního charakteru a obecně dopadů dění lidstva [1], [2], [3], [4], [5], [6].

Toto vše napomáhá rozvoji kyberneticko-fyzikálních systémů (CPS) a umělé inteligenci. Je zapotřebí vše digitalizovat, propojovat a řešit autonomní systémy, které fungují bez zásahu člověka. Tedy je zapotřebí rozvoj k inteligentním továrnám (SMART manufakturing). Základem chytré továrny je, že zákazníkův požadavek je systémem zpracován a všechny výrobní a organizační instrukce odeslány automaticky dál. Je zapotřebí kooperace s dalšími podsystémy, takzvanými agenty [7]. Jeden z velkých prvků inteligentní továrny je pokročilé plánování, využívající heuristické metody[8], [9]. Jeden z mnoha dalších prvků je autonomní vozík [10], který zajišťuje logistiku a který je v této diplomové práci také více rozebírán a řešen [1], [2], [4], [7], [10].

Autonomní vozíky

Potřeba autonomních vozíků vychází z potřeby automatizovat dopravu tak, aby v průběhu jízdy nebyl potřeba zásah člověka. Nakládka a vykládka jsou okrajovými podmínkami, které se řeší individuálně. V případě výukového modelu, je nakládka a vykládka automatizována a navigace vozíčku je pomocí černé čáry. Všechny zásobníky a obecně všechny trasy jsou proto vyznačeny černou čarou, která slouží vozíčku jako navigace. U této navigace je zapotřebí, aby vozíček byl schopen plynule jezdit po čáře a aby alespoň na kontrolních bodech věděl, kde se nachází [11], [12].

Plánování

Maximalizace zisků se přesouvá do problematiky minimalizace nákladů. Bohužel nároky na plnění zakázek jsou čím dál větší především z důvodu přechodu k téměř kusové vý-

robě, čímž dochází k růstu organizačních nákladů. Při organizaci se hledají co nejlepší postupy práce tak, aby se minimalizovaly náklady, zkracovaly dodací termíny a stihl se co největší objem za co nejkratší čas. Tento trend spěje ke snaze maximálně vytížit všechny zdroje, tedy například stroje a dopravní prostředky, aby jich bylo zapotřebí co nejméně.

Při plánování proto dělají velké komplikace zakázky charakteru kusové výroby. Dochází k velkému střídání strojů a tedy i možnostem kombinování pořadí zakázek na jednotlivých strojích. Čekání zakázky na stroji na dokončení jiné je vždy ztráta, proto se řeší optimální pořadí zakázek tak, aby byly splněny dané požadavky, mezi které patří tyto [1], [8]:

- Doba dodání dané zakázky určitému zákazníkovi.
- Kapacita vozidla je omezená.
- Existuje více využitelných skladů.
- Co nejkratší čas na splnění všech zakázek.

V případě výukového modelu chytré továrny se začíná s problematikou co nejkratšího celkového času na splnění všech zakázek, čímž se zabývá i tato diplomová práce.

Shrnutí

V projektu výukového modelu chytré továrny, která má myšlenku kompletaci náramků z korálků, se pro dopravu materiálu a zajištění sběru jednotlivých korálků využívají vozíčky. Aktuální myšlenka je, že vozíčky budou jezdit mezi sklady korálků po černé čáře. V této diplomové práci budou řešeny dvě problematiky:

- Optimalizace řízení vozíčku založené na sledování černé čáry rozebráno v kapitolách 2 a 3.
- Optimalizace výrobního procesu s využitím více vozíčků a více zásobníků rozebráno v kapitolách 4 a 5.

Obě oblasti mají nejprve teoretickou část, viz kapitoly 2 a 4. Praktické části jsou v kapitolách 3 a 5.

2 Systémy autonomního zásobování

Chytré továrny vyžadují maximální eliminaci chyb, plánování v reálném čase a s tím souvisí i změna plánů strojů a logistiky dle potřeby, tedy i dovezení materiálu v daný čas (JIT). V logistice se proto pro autonomní přepravu využívají různé AGV vozíky [13], [14], [15].

Tyto AGV vozíky mají kromě dopravení materiálu na dané místo za další úkol komunikovat s centrálním řízením. Tuto centrálu informuje a o svém stavu:

- Aktuální poloha.
- Stav: řešená zakázka, nabíjení, čekání, atd…
- Stav baterie.

Tyto informace slouží centrále pro stálou kalibraci a aktualizaci plánu. Dalším stádiem je komunikace vozíků a zařízení mezi sebou, tzv. machine-to-machine (M2M) komunikace [14].

Vozíky musí zároveň komunikovat s okolím [10], [11], [15], kde se nacházejí v prostoru a kudy mají jet. Jedna z možností je, že vozík ví prostor a naviguje se lasery [16]. Další možností je, že vozíky jezdí po černé čáře na podlaze [12].

3 Návrh optimalizace jízdy vozíčku

V projektu výukového modelu chytré továrny po firmě dopravují materiál vozíčky. Jednotlivá stanoviště, která jsou ve výrobním procesu, jsou pro navigaci dopravního prostředku propojeny černou čarou/páskou. Vozíčky tedy musí umět jezdit po této černé čáře/pásce. Optimální je, aby vozíček byl schopen jezdit co nejrychleji za splnění dalších požadavků:

- Bezpečnost – včasné zastavení v případě problému.
- Plynulost jízdy – aby se vozíček držel co nejlépe čáry a moc nekmital.
- Nedošlo k poškození převáženého materiálu.
- Přesnost – cílové polohy.

Bezpečnostní požadavek je v tomto případě požadován za splněný, pokud vozíček při vyjetí z dráhy zastaví. Následně může například počkat na obsluhu, která problém vyřeší. V našem případě má vozíček vpředu senzor, který v případě, že se dostane mimo trať, tedy černou čáru, okamžitě zastaví.

Čím plynuleji bude vozíček po čáře jezdit, tím méně vibrací/kmitů vozíček bude mít. Tím pojede i plynuleji a se správnou rychlostí jízdy dopraví převážený materiál v požadované kvalitě. Zároveň bude vozíček schopen přesněji přijet ke stanovištím.

Bezpečnost převáženého materiálu a přesností najetí do cílové polohy zajistíme přesnou jízdou. Tím pádem se hlavní problematika týká plynulostí jízdy.

V kapitole jsou nejprve rozebrány obecné principy různých vozíčků a konkretizována problematika pro analýzu. Před samotnou analýzou jsou zmíněny možnosti provedení samotné analýzy, hodnocení a volba postupu. Po analýze je hodnocen vliv vstupních parametrů na plynulost jízdy a je doporučen postup při volbě parametrů.

3.1 Principy řízení vozíčků

Na trhu se vyskytuje velké množství vozíčků, které lze s menšími či většími úpravami použít. Projekt výukového modelu chytré továrny s kompletací korálků díky drobným dílům nepotřebují velké zařízení.

3.1.1 Vstupní parametry

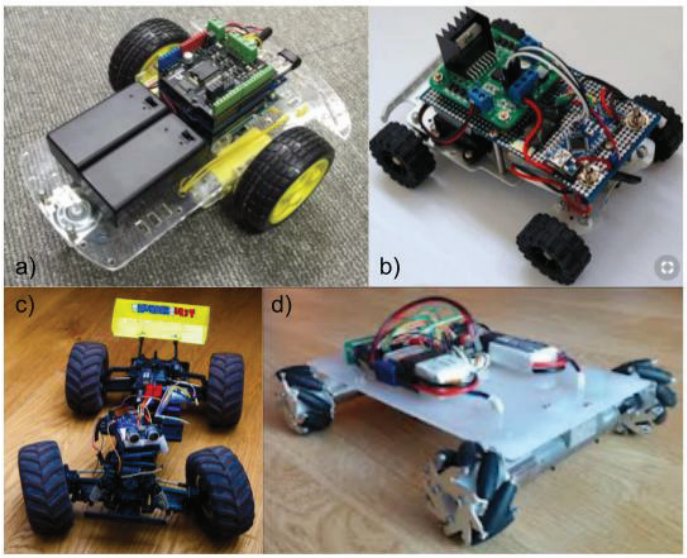
Základem pro definování vstupních parametrů je zásadní definovat typ vozíčku, definovat dráhu a definovat logiku řízení vozíčku. Logika řízení hodně navazuje na typu senzorů, které hodnotí polohu vozíčku.

Typ vozíčku:

U vozíčku je řešen princip pohonné jednotky, tedy zajištění pohybu, zajištění směru jízdy a metoda snímání polohy vozíčku vůči čáře, některé vozíčky jsou znázorněny na obr. 1:

- Pohonná část:
  - 1 kolo + podpůrné →natáčení kola,
  - 2 kola + podpůrné → možnost i natáčení kol obr a),
  - 4 kola → natáčení předních kol obr b/c),
  - 4 kola → natáčení zadních kol,
  - 4 kola → otáčení všech 4 kol (natáčení do stran, či různé speciální varianty typu regulace každého kola zvlášť atd …) obr d).
- Snímání čáry:
  - 2 senzory krajní hodnotící logické 1/0 (je a není na čáře senzor),
  - více senzorů krajních hodnotící logické 1/0,
  - senzory detekující vzdálenost od středu/osy čáry.

Ukázky vozíčku jsou na obr. 1 [17], [18], [19], [20].



Obr. 1 Různé typy vozíčků

Z těchto druhů vozíčků je nejběžnější typ a), tedy dvě kola pohonná s jedním podpěrným kolem. Tento typ vozíčku je také zadán pro analýzu.

Každý vozíček by měl obsahovat senzor, který hlídá, zda je vozíček na čáře či není. To je pro případy, kdy nastane problém a vozíček vyjede z čáry. Problém může být i v chybě čáry, která zmizela. Senzor je však potřeba, aby se s daty mohlo nadále pracovat. Bez tohoto senzoru vozíček postrádá hlavní požadavek pro hlídání kolizních stavů pro následné zpraco-

vávání, vyhodnocování a případnou nápravu. První variantou snímání čáry jsou dva krajní senzory: levý a pravý. Má-li vozíček pouze dva senzory detekující hranu čáry, posílají senzory vozíčku logickou hodnotu 1/0, kterou logika řízení vozíčku dále vyhodnocuje. Další variantou je osázení vozíčku více senzory detekující hranici čáry jako první případ. V tomto případě však vozíček může mít informaci, že vyjíždí z dráhy ještě dříve, než zcela vyjede a může se dle toho zachovat. Například intenzita zatáčení je dle toho, kolik senzorů zaznamenalo detekci hranu čáry. Tímto se dá vytvořit určitá PID regulace, která je však velmi omezena na počet senzorů. Tento problém se dá vyřešit senzory detekující vzdálenost od osy/středu čáry, díky čemuž se dá využít PID regulace využitelná skutečně u náročných aplikací a u přesného vedení vozíčku po definované dráze.

Definování dráhy:

U definování dráhy je zásadní zadání tvaru dráhy, její křivost, různorodost a možnosti, se kterými se vozíček bude muset potýkat. Dalším parametrem je tloušťka čáry:

- Tvar dráhy:
  - dráha tvořena pouze rovnými úseky[21],
  - rovné úseky propojeny tečně s oblouky o definovaném minimálním poloměru [22],
  - obecná dráha.
- Tloušťka čáry:
  - definovaná (v celé trati konstantní),
  - proměnná.

Tvar dráhy velmi souvisí se zadáním celého projektu, možnostech prostoru. Může se stát, že prostory budou velké a veškeré změny směru budou pozvolné a vždy stejné. Další variantou je, že budou pozvolné, ale odlišné, například minimální poloměr oblouku 1 m. Též je možnost, že prostory nejsou a směna směru bude náhlá, prudká a nečekaná, což je pro logiku řízení také dost zásadní parametr. Poslední zmíněnou variantou je obecná dráha, kdy vozíček musí počítat se vším.

Tloušťka čáry má také svůj vliv. Obecně je lepší, když je stálá, konstantní, neměnná. Může se však stát, že bude různá, buďto chybou při výrobě, opotřebením či jakoukoli další vinnou. Další alternativou typu čáry je vícebarevná čára, tedy že barva čáry řekne vzdálenost od osy čáry, to jsou však další komplikace pro výrobu dráhy.

Logika řízení vozíčku – souvisí s typem vozíčku.

Logika řízení vozíčku se váže k typu vozíčku, konkrétně dle pohonné jednotky, tedy zda má vozíček 2 či 4 kola, a zatáčení je zajištěno natočením předních, zadních, všech kol. Vše má své specifické řízení. Možnosti lze roztřídit dle třech kategorií následovně viz tab. 1, kdy jsou využity následující zkratky:

1P pro 1 kolo pohon + podpůrné

2P pro 2 kola pohonné + podpůrné

4P pro 4 kola a natáčení předních kol

4Z pro 4 kola a natáčení zadních kol

4V pro 4 kola a natáčení všech 4 kol

4S pro 4 kola a natáčení všech 4 kol (speciální varianty)

Dle pohonu	**1P**	**2P**	**4P**	**4Z**	**4V**	**4S**
Vždy jen jedno kolo	x	x				
2 kola současně		x	x	x	x	x
4 kola současně					x	x
Kola v pohonu jinou rychlostí		x	x	x	x	x
Dle logiky řízení	**1P**	**2P**	**4P**	**4Z**	**4V**	**4S**
PID řízení	x	x	x	x	x	x
Jízda jakýmkoliv směrem bez natočení vozíčku					x	x
Přejetí čáry – návrat - pokračování v jízdě	x	x	x	x	x	x
Jiný typ řízení	x	x	x	x	x	x
Dle možného směru jízdy	**1P**	**2P**	**4P**	**4Z**	**4V**	**4S**
Rovně	x	x	x	x	x	x
Oblouk	x	x	x	x	x	x
Boční pohyb					x	x
Otáčení na místě	x	x			x	x

Tab. 1 Třídění logiky řízení dle možného pohybu vozíčku

Vozíček lze řídit obecně řečeno jakkoli, záleží na každém programátorovi a zadání. Jsou možnosti, kdy vozíček jede velmi rychle, přejede například svou setrvačností čáru, vrátí se, upraví zatáčení a pokračuje v jízdě. Naopak může jet pomalu a plynule, to souvisí s požadavkem na přesnost polohy vozíčku vůči čáře a tedy na snímání této odchylky, kdy mohou nastat tyto možnosti:

- 2 senzory logické 1/0 → detekováno pouze zda je vozíček moc na kraji.
- Více senzorů logiky 1/0 → dle počtu lze detekovat v krocích odchylku vozíčku, využitelné pro omezené PID řízení.
- Senzory detekující vzdálenost od osy čáry → vhodné pro PID řízení.

S tímto je spojené i umístění senzorů a definice čáry, kdy nám jde o tyto parametry:

- Rozteč senzorů, případně vzdálenost jednotlivých senzorů od sebe.
- Vysunutí senzorů.
- Tloušťka čáry.
- Definice čáry, tvar, plynulost, minimální poloměr, …

Na plynulost jízdy a rychlost mají tyto parametry vliv. S tím je spojen i požadavek na maximální zrychlení vozíčku, aby nedošlo k poruše, například poškození křehkého materiálu.

3.1.2 Souhrn a řešený případ

Vstupní parametry jsou v rámci HW vozíčku, dále v rámci FW, tedy logiky řízení vozíčku. Následují vstupní parametry v rámci celé trati. Lze je rozdělit takto:

Vozíček:

- HW:
  - 2 kola (rozteč 135 mm) + podpůrné,
  - 3 senzory pro jízdu, prostřední hlídá, zda je vozíček na čáře, dva korigují vozíček na čáru,
  - rozteč senzorů = 35 mm,
  - vysunutí senzorů = 55 mm.
- FW:
  - logika řízení jízdy doleva či doprava, případně možnost i jízdy rovně,
  - poloměr pro zatáčení.

Trať:

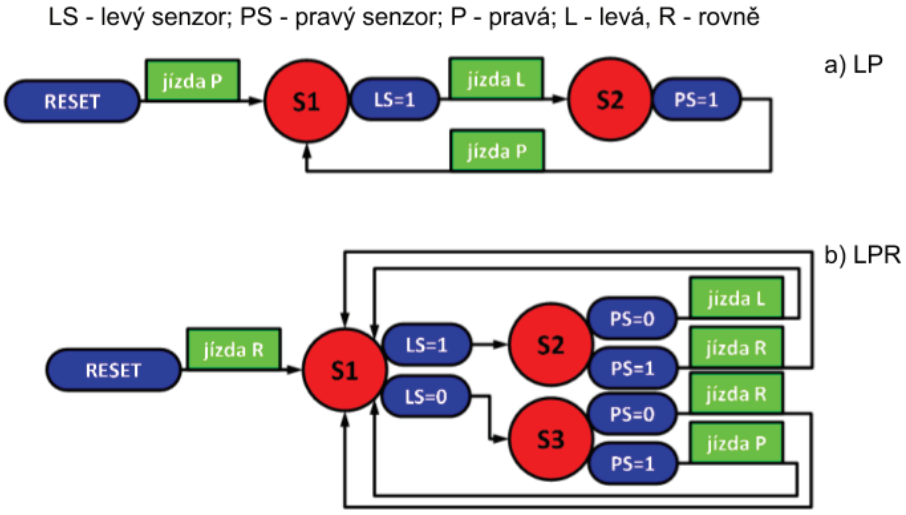
- Minimální poloměr dráhy.
- Šířka čáry.

Analýza bude provedena pro různé rozteče senzorů, vysunutí senzorů a šířky čáry. Očekává se velký vliv poloměru dráhy, který má vliv na poloměr zatáčení logiky řízení vozíčku. Zde bude nejzajímavější procentuální rozdíl pro případ řízení logiky s možností jízdy rovně a bez možnosti jízdy rovně. Poloměr pro zatáčení je tvořen pro minimální poloměr dráhy, dráha tedy ve skutečnosti může být i s větším poloměrem.

Simulace bude provedena pro logiku řízení jak pro jízdu doleva či doprava (LP), viz obr. 2 a), tak i pro jízdu doprava či doleva s možností jízdy rovně (LPR), viz obr. 2 b). Využívají se tyto zkratky pro logiku řízení:

LP jízda doleva či doprava

LPR jízda doleva, doprava či rovně



Obr. 2 Vývojový diagram pro logiku řízení a) LP, b) LPR

I takto drobný rozdíl v logice řízení vozíčku může způsobit velké rozdíly v jízdních vlastnostech. Očekává se výrazné zlepšení jízdních vlastností u LPR. Zlepšení se očekává především při jízdě v rovných úsecích, kde LP stále kmitá doleva a doprava. Tedy kde jízda rovně má možnost se nejvíce projevit. Další zlepšení LPR oproti LP se dá očekávat v rychlosti jízdy, kdy maximální rychlost LPR je max. rychlost kola, kdežto u LP je to průměrná hodnota min. a max. rychlosti kol, protože na rovině může rychlostí arduina kmitat jízda doleva a doprava, čímž se rychlost dostane na průměrnou hodnotu min. a max. rychlosti.

3.2 Metody pro analýzu

Postupů je vždy více, mohou být hodnoceny různě, například dle těchto parametrů:

- Náročnost, komplikovanost.
- Požadovaná odbornost, znalost teorie.
- Náročnost na vstupní data – kvalita, množství.
- Výstupní data – přesnost, kvalita, formát.
- Náročnost zpracovávání dat.

Jsou metody, které vyžadují vysokou vstupní časovou investici a vše další je již snadné a přesné. Naopak metody, které jsou rychlé pro jeden, dva pokusy, ale výstupní data jsou jen orientační. Pro analýzu vlivu vstupních parametrů pro řízení vozíčku jsou tyto možnosti:

- Vytvořit reálné motely tratí a reálné parametry vozíčků a testovat projetí/vyjetí vozíčku z dráhy.
- Vytvořit reálné modely, v každém intervalu hodnot udělat několik pokusů a odvozovat přibližně závislosti.
- Vytvořit digitální model a simulovat jízdy pomocí výpočetních systémů.

Jsou tedy v základu dvě možnosti, reálná simulace a virtuální simulace. Charaktery obou metod znázorňuje následující tab. 2.

Hodnocení je v bodech, kde počet bodů je znázorněn souhlasem od 0 od 10.

Charakteristika	Reálná	Virtuální
Náročná vstupní příprava	0	10
Náročná příprava každé simulace	8	1
Časová náročnost každé simulace	5	1
Nízká kvalita výstupních dat	5	6
Náročné zpracování dat	10	1
Nejednoznačnost výstupních dat	4	1

Tab. 2 Porovnání metod pro analýzu vstupních parametrů jízdních vlastností

Tabulku rozebírají dvě následující podkapitoly.

Charakteristika reálné simulace

Reálná simulace vyžaduje přípravu každé simulace daných parametrů. Tato náročnost se pohybuje od 5 do 30 minut. Každý test lze vyhodnotit dvěma způsoby. Jednak zda vozíček vyjel z dráhy či nea a v případě, že jízda trvá dostatečně dlouho, zda vozíček vyjede například až po ujetí 10/100 m. Rozdíl očekávám z důvodu problematiky nahodilostí, které lze velmi náročně odstranit. Tento výstup, tedy vyjetí či nevyjetí vozíčku směřuje postupně k úpravě algoritmu jízdy tak, aby vozíček nevyjel, což je dost zdlouhavé. Také lze jízdu vozíčku natáčet a hodnotit video. Získat však velké množství dat, které by se velmi snadno a srozumitelně vykreslily, je velmi zdlouhavé a náročné.

Charakteristika virtuální simulace

Virtuální simulace má vysoké požadavky na přípravu. Tato příprava obsahuje vytvoření simulačního programu, lze využít Matlab. Následně však lze vše zautomatizovat a lze v krátkém čase vytvořit velké množství dat, které lze snadno znázornit v grafické formě. Navíc samotné simulace a vykreslování dat probíhají na výpočetním systému bez potřeby zásahu. Data mají při malém rozlišení simulace velmi malý rozptyl výstupních dat a kvalita se dá očekávat maximálně v určitém posunu. Tento posun je možné vyřešit tím, že se uvede tolerance například 10%, která by měla být dostatečná a až reálná simulace případné problémy může nalézt.

Volba metody a postupu práce

Nejprve bude vytvořena virtuální simulace v Matlabu, díky tomu se získá mnoho dat s přehlednými grafickými výstupy. Následně bude provedena reálná simulace pro ověření virtuální simulace. Následně se vyhodnotí simulace a vytvoří závěry.

3.3 Analýza

V této kapitole je rozebrána simulace ve výpočetním programu Matlab. Následně je provedena reálná jízda vozíčkem a nastavení parametrů vozíčku a v závěru je porovnání.

3.3.1 Simulace v matlabu

Hlavní analýza bude provedena virtuálně ve výpočetním systému Matlab. Výsledky se částečně ověří a zhodnotí.

Pro kvalitní data je nutné si uvědomit vstupující parametry na kvalitu výstupních dat. Je to především škálování, respektive jaký krok simulace má. Pro simulace bylo využito škálování 1 mm, který vychází jako dostačující. Dalším velmi zásadním parametrem je tvar dráhy. Dobu simulace ovlivňují další parametry, například množství simulovaných vstupních parametrů. Pracuje se především s těmito parametry:

- Tloušťka čáry (TČ) → 10 – 100 mm.
- Poloměr dráhy (PD) →.135 – 1000 mm.
- Vysunutí → 30 – 100 mm.
- Rozteč senzorů → TČ *0,5 – TČ *2-1.

Černé pásky, pomocí kterých se může trať vyrábět, jsou vyráběny v různých šířkách [23], [24]. Poloměr dráhy je min. rozteč kol 135 mm, pokud nemá vozíček složitější algo-

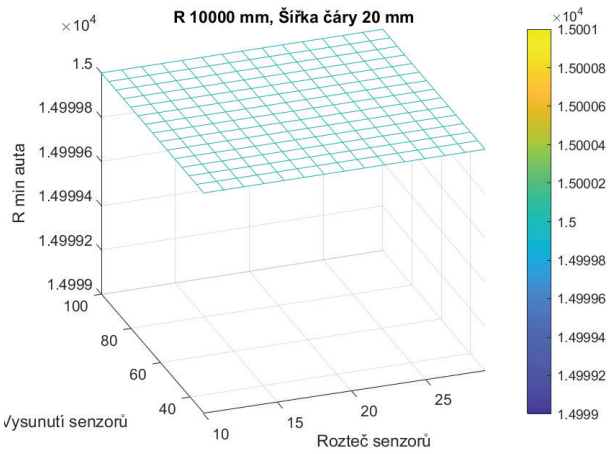
ritmy, které by umožňovali jet menší poloměry. Pro potřeby výukového modelu není potřeba. Vysunutí z konstrukčních důvodů stačí analyzovat od 30 mm a nad 100 mm byla simulací zjištěna příliš vysoká náročnost na výpočet a zároveň se výstupní data nijak výrazně neměnila. Rozteč senzorů je simulována od 50 % tloušťky čáry. Je to z důvodu přehledné grafiky, kdy nižší hodnoty způsobí nepřehlednost výstupního grafu. Maximální rozteč senzorů je dán dvojnásobnou tloušťkou čáry, protože stále musí být na čáře prostřední senzor, který hlídá, zda je vozíček na čáře. Stačilo by, aby na čáře byl alespoň jeden ze třech senzorů, ale pro zvýšení přesností jízdy vozíčku je definováno, že na čáře musí být vždy prostřední senzor. Od dvojnásobku tloušťky čáry je odečten 1 mm pro posun od maximálního parametru pro snazší řešení hranic simulace.

Dále v kapitole budou rozebrány tyto kategorie

- Volba tvaru dráhy pro simulaci.
- Porovnání logiky řízení vozíčku LP a LPR.
- Analýza vlivu rozteče a vysunutí na jízdu vozíčku.
- Analýza vlivu vstupních parametrů na min. poloměr logiky řízení vozíčku.

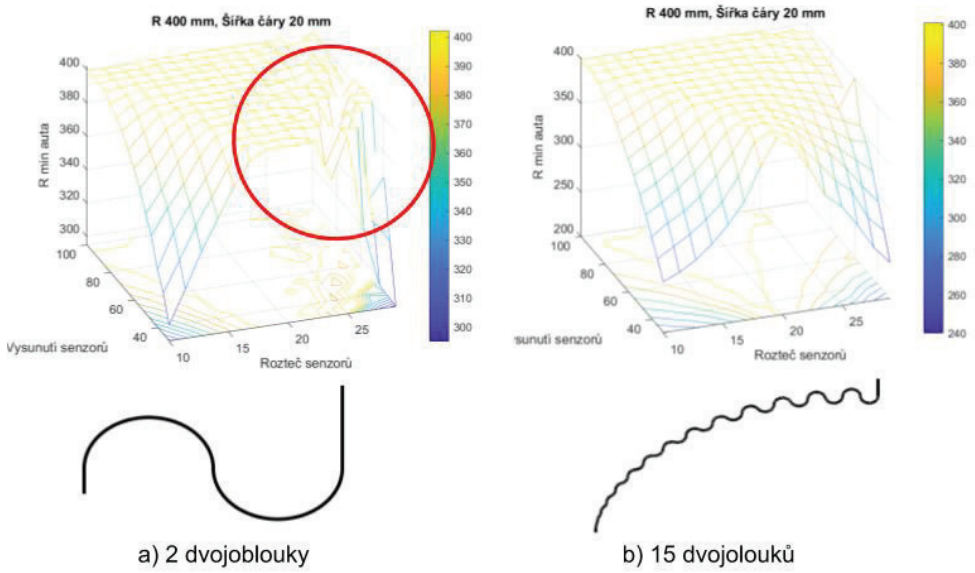
Volba tvaru dráhy pro simulaci

Založíme-li logiku řízení vozíčku na jízdu vždy doprava či doleva (LP), a vezmeme si dráhu v podobě rovné čáry, dostaneme optimální nastavení parametrů následovné, viz obr. 3.



Obr. 3 Výstup simulace při zadané rovné dráze

Tento graf ukazuje, že nezáleží na vysunutí senzoru ani na rozteči senzorů. Optimální nastavený poloměr jízdy vozíčku bude vždy stejný a to co největší. V případě simulace bylo nastaveno R = 10 m, což po úpravě SF vyšlo na 15 m. Simulovat jízdu na rovné trati tedy nemá smysl. Je tedy zapotřebí vytvořit dráhu s oblouky. Bylo otestováno, že počet použitých oblouků, zakřivení, komplikovanost dráhy zvyšuje kvalitu výstupních dat, viz obr. 4.

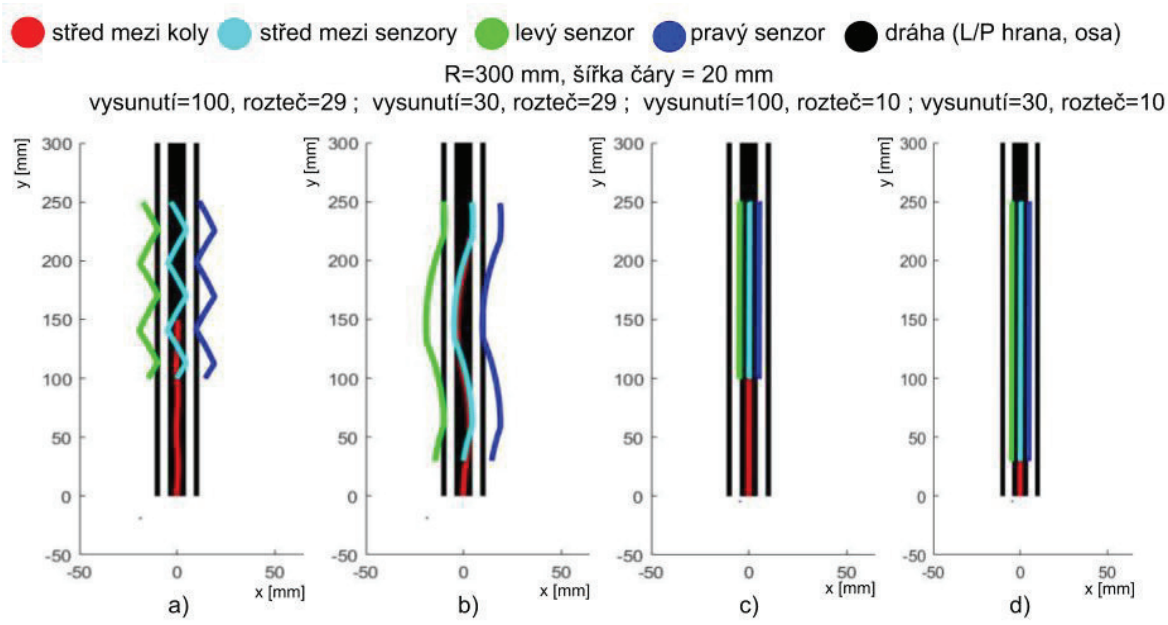


Obr. 4 Kvalita výstupů dle složitosti dráhy

Na obrátku je vidět zkvalitnění výstupních dat při využití komplikovanější dráhy pro simulaci jízdy vozíčku. Zkvalitnění je nejen na vyhlazení výstupního grafu, ale i zpřesnění maximálního použitelného poloměru pro jízdu vozíčku.

Analýza vlivu rozteče a vysunutí na pohyb vozíčku po dráze

Vliv na jízdní vlastnosti bez ohledu na tvar dráhy, má rozteč senzorů a jejich vysunutí. Následující obr. 5 ukazuje pohyb několika bodů na vozíčku. Je důležité si všímat především plynulosti a rozsahů pohybů. Logika řízení pro zvýraznění pohybu je jízda pouze doleva či doprava (LP).



Obr. 5 Pohyb vozíčku při různé rozteči senzorů a vysunutí

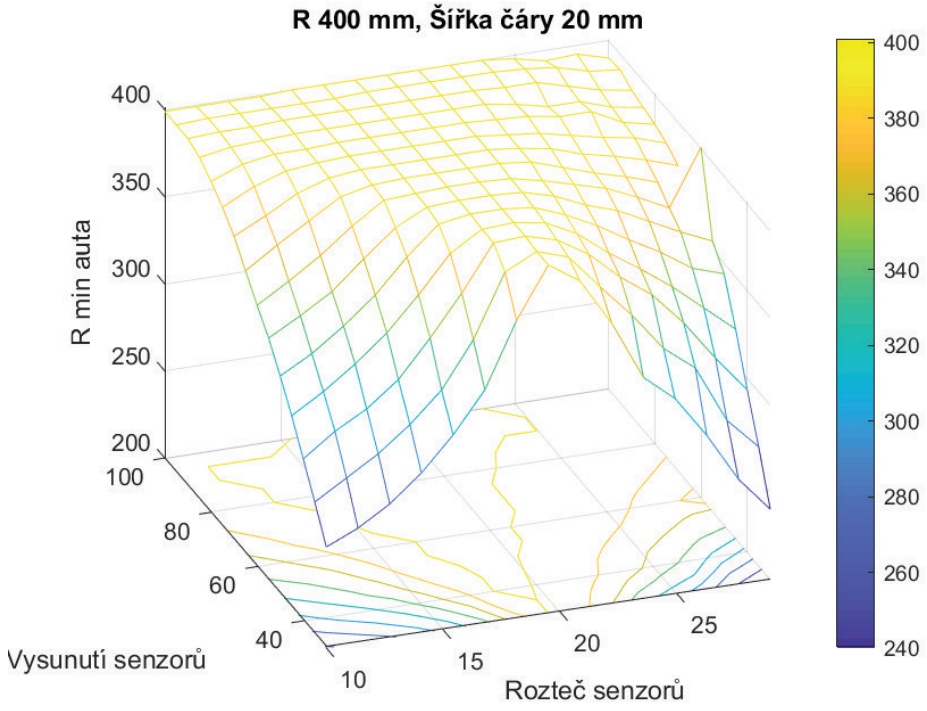
Bod ležící v ose mezi koly je červenou. Bod ležící též na ose vozíčku, ale leží mezi senzory, je světle modrou. Levý senzor znázorněn světle zelenou barvou, a pravý senzor tmavě modrou. Na obrázku je černou čarou znázorněn levý a pravý okraj čáry, po které vozíček jede. Pro zobrazení středu čáry je použita tlustá černá čára.

Lze sledovat plynulost či naopak rozklepanost předních senzorů. Je vidět, že velké vysunutí senzorů, viz obr a), způsobuje velké pohyby senzorů oproti málo vysunutým senzorům na obr b). Je to tím, že i drobná změna natočení vozíčku díky velkému ramenu (vysunutí) způsobí velký pohyb / odklon. Dle zdrojů [25], [26] se toto právě využívá pro vozíčky vyšších rychlostí, protože jedou více přímě, což je trochu vidět na červené čáře obr. a) a b). Lze tedy říci, že pro plynulejší jízdu je vhodné senzory vysunout víc.

Také je vidět zásadní rozdíl pohybu vozíčku mezi roztečí senzorů v případě, kdy je rozteč větší (obr. a, b) či naopak menší (obr. c, d), než šířka řídící čáry. Při větší rozteči vozíček jede jedním směrem, dokud nesepne senzor a jede na druhou stranu. Stejně to má vozíček se senzory blíže u sebe než je šířka čáry, jenže oba senzory jsou stále na čáře, takže vozíček stále kmitá doleva a doprava díky čemuž jede relativně rovně. Ve skutečnosti (není-li programem omezeno) je spínán v rychlosti platformy arduina MEGA 2560 pro jízdu doleva a doprava, čímž se dá očekávat určité zpoždění elektroniky a pohonů kol a skutečně vozíček pojede rovně, jen omezenou rychlostí. Rychlost se předpokládá omezená na průměrnou rychlost kol, protože neustále bude spínán pohon například na 50 % a 100 % čímž se dá očekávat skutečná rychlost $\frac{50+100}{2} = 75\ \%$.

Analýza vlivu rozteče a vysunutí na minimální poloměr logiky řízení vozíčku

Rozteč senzorů je omezen max. na dvojnásobek šířky čáry, protože chceme, aby prostřední senzor byl vždy na čáře. Vysunutí senzorů z předchozích zjištění urychlí zjištění následujícího směru dráhy, čímž se zvýší plynulost. Vliv těchto parametrů na max. poloměr pro logiku řízení vozíčku je znázorněno na následujícím obr. 6. Všechny grafy s tloušťkou čáry 10 až 80 mm a poloměr dráhy 200 až 1000 jsou v přílohách A a B.

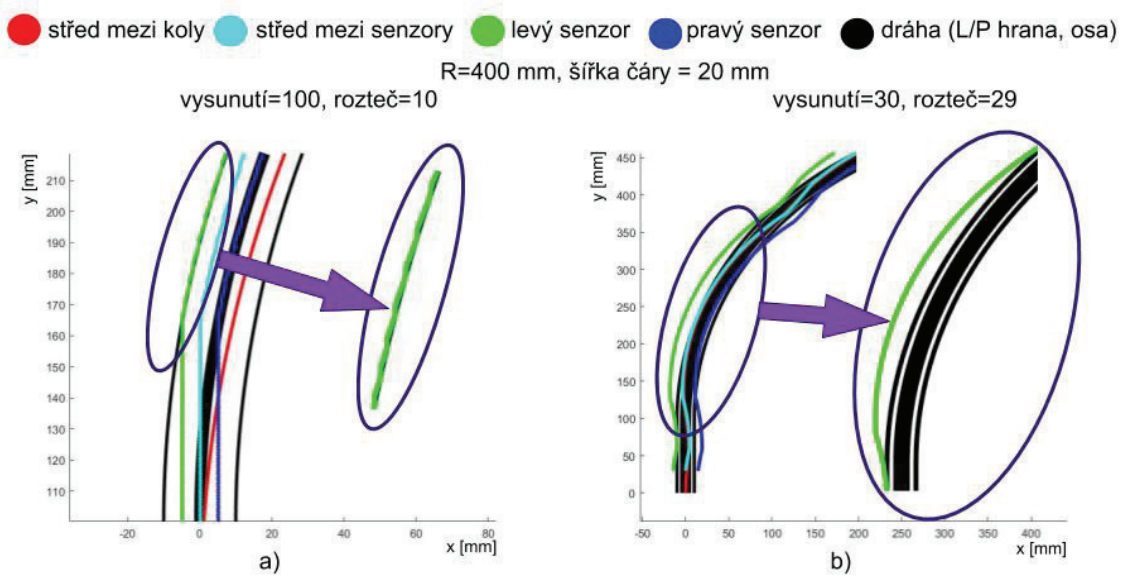


Obr. 6 Vliv rozteče a vysunutí senzorů na min. poloměr jízdy vozíčku

Pro graf výše byla využita dráha o poloměru 400 mm, rozteč senzorů 10 až 29 mm a pro vysunutí senzorů 10 až 100 mm. Je vidět obecný trend lepšího poloměru pro větší vysunutí senzorů. To však způsobuje vyšší nároky na rozměr vozíčku. Také je vidět trend lepšího, tedy většího min. poloměru pro jízdu blízko šířky čáry. Proto je doporučeno volit rozteč senzorů blízko rozměru šířky čáry.

Analýza vlivu rozteče senzorů na jízdu v zatáčkách

Rozteč senzorů může být větší než šířka čáry, rovna šířce čáry, nebo mohou být blíž, než je šířka čáry. Vozíček jede ve většině případů dost rozdílně, viz obr. 7.

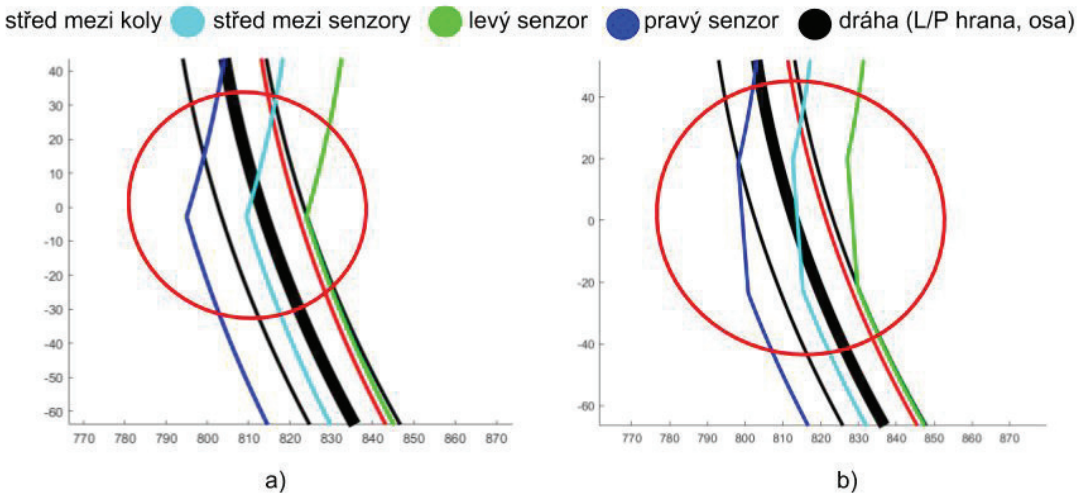


Obr. 7 Pohyb vozíčku v zatáčce s rozdílnou roztečí senzorů vůči tloušce čáry

Obr. a) znázorňuje jízdu vozíčku se senzory blíže než šířka čáry. Obr. b) naopak rozteč senzorů větší než šířka čáry. Je vidět, že u senzorů blíže obr. a), vozíček je stále kalibrován k čáře, naopak u rozteče senzorů větší, jezdí vozíček v obloucích. Z tohoto pohledu je lepší mít rozteč senzorů stejnou až menší než je šířka čáry, protože se vozíček drží lépe středu čáry.

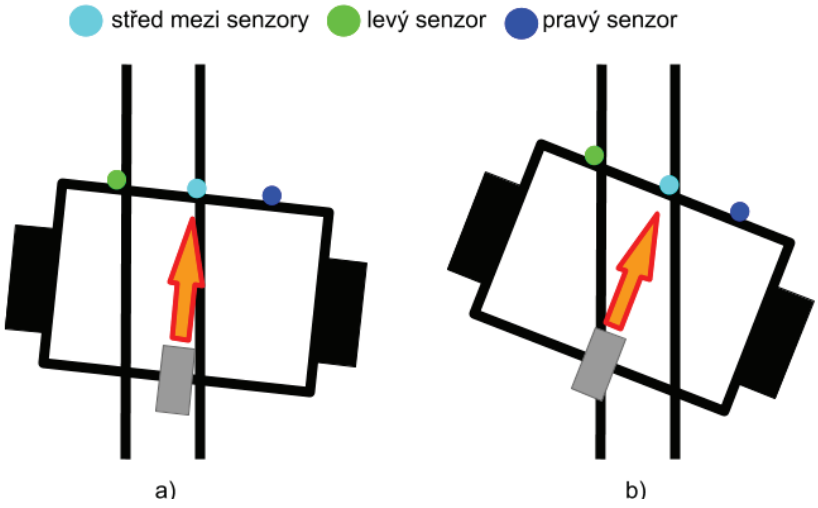
Porovnání logika řízení LP a LPR

U logiky řízení vozíčku jsme velmi omezeni. Pohon kol musí být pevně daný. Možná je jízda do stran (LP. Druhou variantou je umožnit vozíčku i jízdu rovně (LPR) v případech, kdy budou oba senzory sepnuté, nebo rozepnuté. Rozdíl pohybu vozíčku, respektive především senzorů je zobrazen následujícím obr. 8.



Obr. 8 Rozdíl jízdy vozíčku při odlišné logice řízení a) LP b) LPR

Tímto se dá očekávat u LPR b), oproti LP a) snížení vibrací vozíčku a snížení kmitání vozíčku. Je to vidět i na obrázku v místě navazování nové křivky, kdy ve variantě LP, viz a), je zlom větší, než u LPR, tedy obr. b). Toto vytváří najetí vozíčku na čáru pod menším sklonem, jak je znázorněno na obr. 9.



Obr. 9 Natočení vozíčku v kontaktu senzoru s čárou - natočení a) malé b) velké

Z důvodu tohoto odlišného natočení vozíčku při rozteči senzorů větší než šířka čáry je stanovena hypotéza 1. V případě, kdy vozíček bude umět jet i rovně LPR, bude mít zvětšení maximálního poloměru pro logiku řízení. Pro rozteč senzorů menší jak šířka čáry zlepšení se neočekává, protože vozíček bude v době, kdy by mohl jet rovně (oba senzory sepnuté), stále kmitat, tedy zdánlivě pojede téměř rychle. Tab. 3 ukazuje výsledky maximálních poloměrů pro dráhu s poloměrem R=400 mm, šířka čáry s=20, komplikovanost dráhy pro vyhlazení a zpřesnění hodnot bylo zvoleno 30 dvojoblouků, graf vykreslován na 5x5 bodů v rozsahu rozteče senzorů 10 až 29, vysunutí senzorů 30 až 100, škálování simulace zvolen 1 mm (krok).

R400, S20, 2Obl 30, bodu 5x5, [mm]					
LP	**10**	**14,75**	**19,5**	**24,25**	**29**
30	252	293	400	309	225
47,5	313	362	401	373	325
65	361	399	401	390	369
82,5	393	400	401	397	384
100	399	400	400	400	393
LPR	**10**	**14,75**	**19,5**	**24,25**	**29**
30	250	292	400	316	265
47,5	312	363	401	384	329
65	362	399	401	400	375
82,5	394	401	401	401	399
100	399	399	400	401	401
LPR/LP	**10**	**14,75**	**19,5**	**24,25**	**29**
30	0,992	0,997	1,000	1,023	1,178
47,5	0,997	1,003	1,000	1,029	1,012
65	1,003	1,000	1,000	1,026	1,016
82,5	1,003	1,003	1,000	1,010	1,039
100	1,000	0,998	1,000	1,003	1,020

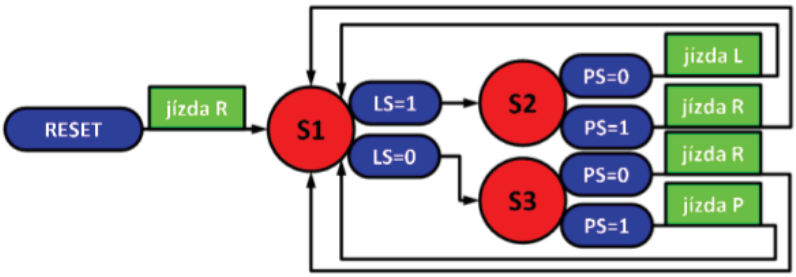
Tab. 3 Porovnání logiky řízení LP s logikou LPR

Tabulka zobrazuje nejprve maximální poloměry jízdy pro logiku řízení LP, následně LPR a nakonec je porovnání hodnot, které je vydělením LPR/LP. Porovnání hodnot je na 3 desetinná místa z toho důvodu, aby byl vidět trend. Hodnoty by byly přesnější při využití jemnějšího škálování simulace, pro potřeby je toto dostačující a lepší škálování je časově náročnější. Tabulka jednoznačně ukazuje, že pro tento případ zadaných parametrů dráhy je vhodnější logika řízení LPR, což potvrzuje hypotézu 1. Simulace ukazuje tři zhoršení a to při rozteči senzorů menší jak šířka čáry a vysunutí senzorů menší jak 50 mm. Zhoršení se přisuzuje velkému škálování, kdy zde bylo využito kroku 1 mm. V ostatních případech je zlepšení od 0 % do 17,8 %, což v průměru činí zlepšení o 1,4 %. V případě, že se do výpočtu průměr-

ného zlepšení nevezmou dvě nejvyšší a dvě nejnižší hodnoty, vyjde průměrné zlepšení 0,7 %. Vyšší zlepšení je při vyšších roztečích senzorů, protože zde může docházet k očekávanému efektu velkého natočení vozíčku na čáru při logice řízení LP, kdy vozíček na čáru najíždí pod menším úhlem. Kdežto při roztečích menších jak šířka čáry dochází při sepnutých obou senzorech dle rychlosti arduina k neustálému kmitání jízdy doleva a doprava, čímž vozíček zdánlivě jede rovně. Největší rozdíl byl očekáván při rozteči senzorů větších jak šířka čáry, tedy v tomto případě 20 mm. Zde dochází ke zlepšení v průměru o 3,6 %, bez započtení dvou nejvyšších hodnot zlepšení o 1,7 %. Zásadní rozdíl se však předpokládá v celkové rychlosti vozíčku a především vyhlazení jízdy po rovných úsecích dráhy, protože LP by stále kmital. Oproti tomu LPR má snahu jet rovně. Dá se očekávat zlepšení logiky řízení LPR oproti LP i za jiných parametrů dráhy a vozíčku. Data neukazují trend zhoršení, proto je doporučeno vždy využít logiku řízení s možností jízdy rovně.

3.3.2 Reálná jízda vozíčkem

Byl vytvořen program pro platformu Arduino MEGA 2560, dle vývojového diagramu viz obr. 10.



Obr. 10 Vývojový diagram pro logiku řízení LPR

Testování proběhlo na čáře šířky 40 mm, poloměr oblouku dráhy 400 mm, rozteč senzorů 35 mm vysunutí senzorů 55 mm. Pro jízdu do zatáčky bylo využito nejprve pohonu vnitřního kola na 62 %, což je odvozeno od poloměru dráhy a rozteče kol RK = 135 mm. Z předchozích simulací byl pro tyto parametry určen max. poloměr jízdy Rmax = 360 mm, započítáme alespoň 20% rezervu, aby spolehlivě vozíček nevyjel z čáry, viz následující vzorce.

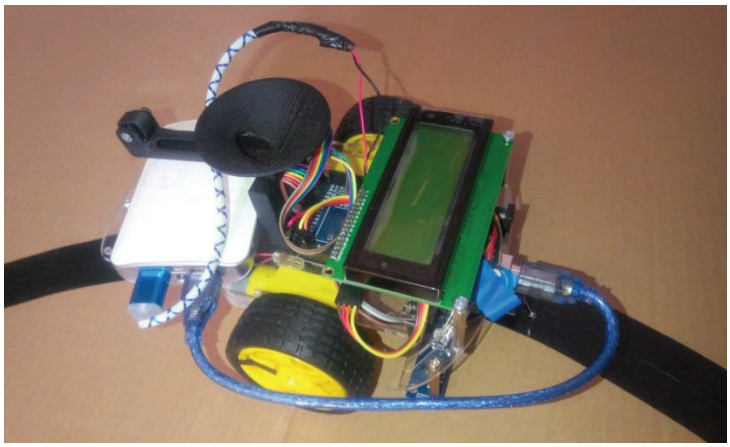
RK = 135 mm rozteč kol

Rmax = 360 mm maximální poloměr jízdy získané ze simulace

k = 0,8 koeficient pro započtení rezervy 20% na poloměr jízdy

$$\frac{Rmax * k - \frac{RK}{2}}{Rmax * k + \frac{RK}{2}} = \frac{360 * 0{,}8 - \frac{135}{2}}{360 * 0{,}8 + \frac{135}{2}} = 0{,}62 = 62\ \%\ vnitřní\ kolo$$

Následující obr. 11 je ukázka z reálné jízdy vozíčkem.



Obr. 11 Foto z reálné jízdy vozíčkem

Při reálné jízdě se vyskytlo několik problémů:

- Kola vozíčku prokluzují.
- Pravé kolo je o 10-20 % silnější než levé.
- Kolo bez zátěže se roztočí až při výkonu motoru 47%, se zátěží od 59 %.

Vozíček jezdí po kluzkém povrchu, čímž dochází k prokluzu pneumatik a dále při změně otáček motoru není reakce okamžitá. Toto se projevuje především při brždění a rozjezdu, kdy kola mění otáčky. Další vliv může mít stav baterie. Z těchto důvodů je vhodné zpomalit jízdu tak, aby k těmto efektům nedocházelo. Dalším problémem bylo zjištění, že levé kolo vozíčku je slabší o 10-20 %. Toto se kalibrovalo v řídícím programu, bohužel každý vozíček bude mít jiné parametry. Testování bylo provedeno spočtením doby, za kterou kolo bez zátěže udělá 40 otáček. Dalším zjištěním bylo, že motory nejsou dostatečně silné a motor má pro jízdu sílu až od 59 %, což znemožňuje otáčení obou kol v zatáčkách ve chvíli, kdy není požadavek jízdy na maximální rychlost.

Z důvodu všech těchto potíží byl do algoritmu přidán vlastní kód pro regulaci otáček na principu PWM regulace. Po testování se došlo do nastavení otáček kol v zatáčce v poměru 30 % a celková rychlost nastavena na 50%. Nyní vozíček úspěšně jezdí.

Testování před a po optimalizaci

Testování proběhlo na rovném úseku dráhy dlouhé 3,5 m. Měření bylo provedeno vždy 10x, naměřené hodnoty ukazuje následující tab. 4.

Měření	Čas před optimalizací	Čas po optimalizaci
1	25,5	18,9
2	26,2	18,5
3	25,8	17,6
4	26,9	17,4
5	26,5	18,2
6	26,2	17,8
7	26,1	18,6
8	26,5	18,1
9	25,8	17,6
10	25,4	18,5
Průměr	**26,1**	**18,1**

Tab. 4 Měření doby jízdy před a po optimalizaci

Zlepšení je o 31 % v rychlosti a k tomu je jízda plynulejší, tedy je cíl splněn.

3.3.3 Porovnání výsledků reál vz. simulace

Fyzický vozíček měl mnoho problémů, se kterými simulace nemohla počítat. Po úpravě kódu lze říci, že je nutné data ze simulace opatřit dostatečnou ochranou a je zapotřebí reálná kontrola. Pro simulované parametry poloměru dráhy 400 mm vychází potřebná bezpečnost / ochrana na 50 %. Výstupní data ze simulací slouží pro přehled vlivu vstupních parametrů a dá se očekávat jen určité snížení poloměru pro logiku řízení.

3.4 Hodnocení vlivu vstupujících parametrů

Shrnutí vlivu rozteče a vysunutí senzorů na jízdní vlastnosti shrnuje následující tab. 5. Hodnocení je ve stupnici 0-10 (10 = nejlepší).

Vysunutí senzorů:	**↓**			**ca 50**			**↑**			**Váha**
Rozteč senzorů:	**↓**	**ca šč.**	**↑**	**↓**	**ca šč.**	**↑**	**↓**	**ca šč.**	**↑**	
Plynulejší jízda	3	4	0	7	7	5	10	10	10	10
Min. poloměr jízdy vozíčku	0	8	0	5	9	5	7	10	7	5
Přesnost vozíčku na čáře	10	8	2	10	8	2	10	8	2	8
Hodnocení	0,48	0,63	0,07	0,76	0,78	0,4	0,93	0,93	0,66	

Tab. 5 Hodnocení rozteče a vysunutí senzorů na charakter jízdy

Příklad výpočtu pro malé vysunutí a malou rozteč senzorů je následující:

První souřadnice je vysunutí senzoru: 1 = malé (0 až ca 50 mm), 2 = ca 50 mm (+/- 10 %), 3 = velké (do ca 150 mm)

Druhá souřadnice je rozteč senzorů: 1 = malá (menší jak 90 % šířky čáry), 2 = šířka čáry (+/- 10 %), 3 = větší jak šířka čáry (10% nad šířku čáry až dvojnásobek šířky čáry).

PJ[1,1] = 3 Plynulejší jízda

MP[1,1] = 0 Min. poloměr jízdy vozíčku

PV[1,1] = 10 Přesnost vozíčku na čáře

V[1] = 10 Váha hodnotícího parametru „plynulejší jízda“

V[2] = 5 Váha hodnotícího parametru „Min. poloměr jízdy vozíčku“

V[3] = 8 Váha hodnotícího parametru „Přesnost vozíčku na čáře“

$$\frac{PJ[1,1] * V[1] + MP[1,1] * V[2] + PV[1,1] * V[3]}{10 * (V[1] + V[2] + V[3])} = \frac{3 * 10 + 0 * 5 + 10 * 8}{10 * (10 + 5 + 8)} = 0{,}48 = 48\ \%$$

V tabulce je vidět vhodnost využití většího vysunutí senzorů. Zároveň využití rozteče senzorů blízké šířce čáry. V případě, že se využije rozteč senzorů blízké šířce čáry a menší, lze pro dobré výsledky využít i střední vzdálenost vysunutí senzorů, tedy ca 50 mm, čímž se dají snížit prostorové nároky na zařízení. Při využití středního vysunutí senzorů je vhodnější využít rozteče senzorů blízké šířce čáry, kde nedochází k velkému snížení min. poloměru jízdy vozíčku.

Shrnutí doporučení pro tvorbu vozíčku a dráhy je následující:

- Dráha s větším min. poloměrem → alespoň 400 mm.
- Logika řízení LPR, možnost jízdy vozíčku i rovně.
- Rozteč senzorů blízké šířce čáry, ideálně o malinko menší.
- Vysunutí střední až velké → 50 – 100 mm.

Tyto doporučení jsou pro zadání, kdy vozíček má jet po čáře bez vyjetí, problémů, určité plynulosti a s omezením vozíčku.

Původní vozíček využíval jen jízdu LP a neřešit umístění senzorů. Změnou vstupních parametrů se docílilo výrazného zlepšení plynulejší jízdy.

4 Principy umělé inteligence a Průmyslu 4.0

Vývoj a optimalizace nabírá rychle na náročnosti. Po elektrifikaci a automatizaci se přidávají prvky schopné rychle měnit výrobní procesy. K tomuto firmy sbírají data všeho druhu, dostávají se však do problému zpracování a vyhodnocování těchto dat. Jednak je dat obrovské množství, za další komplikovanost zadání nezná mezí a současné metody by problematiky nespočítaly do konce světa. Proto se vývojáři opět vrací k přírodě a hledají inspiraci [10], [27].

V případě kompletace korálků, kde máme *n* zakázek, existuje m! možných uspořádání zakázek. V našem případě však zakázky nejsou stejné a tedy se dá očekávat, že na jeden zásobník s danou barvou kuličky (počet zásobníků *n*) mnohdy přijedou dva vozíčky zároveň. V tomto případě se musí řešit, který vozíček přijede dříve. Tím vzniká až $m!^n$ variant. Sice všechny varianty nejsou reálné, ale variant výrazně přibylo. Následující tab. 6 ukazuje navýšení času s rostoucím počtem zakázek *m*. Pro výpočet byl využit pouze faktoriál a čas na výpočet jednoho řešení je použit čas 1 s.

m	Čas [s]	Čas [dny]	Čas [let]
5	120		
8	$4{,}0*10^{4}$	0,47	
10	$3{,}6*10^{6}$	42	0,12
15	$1{,}3*10^{12}$	$1{,}5*10^{7}$	$4{,}1*10^{4}$
20	$2{,}4*10^{18}$	$2{,}8*10^{13}$	$7{,}7*10^{10}$

Tab. 6 Nárůst času výpočtu s nárůstem problematiky

Z tabulky je vidě exponenciální nárůst doby výpočtu s lineárním růstem komplikovanosti zadání. Již při pouhých deseti prvků v zadání by výpočet trval 42 dní. Již při 20 prvcích v zadání by výpočet trval déle, než je existence vesmíru, který je vědci vypočten na 13,8 miliardy let [27]. Není reálné u všech možností časovou náročnost spočítat a je tedy nutné přijít s jinými algoritmy pro hledání řešení, které sice není obvykle nejoptimálnější, ale v rámci rychlosti výpočtu má dostatečnou kvalitu.

Problematika uspořádání zakázek je natolik komplikovaná, že je nutné využít pokročilejších algoritmů. V tomto případě budou využity genetické algoritmy, které jsou k tomu vhodné. Dochází tedy k návratu k přírodě a vývoj algoritmů se opět nechává inspirovat živými organizmy.

Živé organizmy zde existovaly mnohem dříve než lidé. Všemožné organizmy byly nuceny se adaptovat na prostředí a poradit si s všemožnými problémy. Dnes bychom mohli říci, že si musely vytvořit algoritmy přežití. Proto je nutné, aby lidé byli pro pokrok inspirování přírodou a hledali metody, které sice nemusí být nejoptimálnější ze všech variant, ale budou dostatečně kvalitní a zároveň rozhodnutí přijde v reálném čase. Umělá inteligence (UI) se opírá o principy biologické a heuristické [10], [9], [28], [29], [30].

Homo sapiens sapiens, neboli člověk rozumný, rodu člověk je primát, který je též velkou inspirací pro tvorbu a řešení problematiky UI. Jeho mozek má obrovské schopnosti a slouží jako velký hodnotitel kvality UI. K hodnocení se využívá například Turingova testu, který je více rozebrán v následující podkapitole [9], [31], [32], [33], [34].

4.1 Ověřování umělé inteligence

Problematika UI je velmi rozsáhlá a velmi mladá. Prostudovaná literatura se shoduje na tom, že je velmi náročné stanovit, zda stroj, respektive přesněji řečeno program, již skutečně myslí. Přesto je níže snaha nastínit, jak se na problematiku dnes nahlíží a o čem se již dá uvažovat jako o úspěšné UI. Protože i lidé mnohdy nemyslí a u stoje se dá snáze zhodnotit, zda v daném úkolu obstál či nikoliv [10], [9], [34].

„Myslící prohlásíme počítač tehdy, když jeho chování nebudeme schopni rozeznat od chování člověka.“ [35]. Na tomto zakládá Turingův test [9], [34], [35], který funguje na principu komunikace. Hodnotitelé/rozhodčí mají za úkol otázkami určit, zda na druhé straně je člověk, nebo počítač (program). Komunikace probíhá psanou formou a za úspěšné se považuje, když se hodnotitelé u počítače alespoň z 30 % domnívají, že je člověkem. Problém se obvykle komplikuje tím, že se počítač přímo za někoho vydává, například jako 7. 6. 2014 na univerzitě v Readingu v Anglii, kde se vydával za třináctiletého Ukrajince a přesvědčil hodnotitele v 33 % odpovědí [34], [36].

Velké populární testování umělé inteligence probíhá na kombinatorických hrách, jako jsou šachy či asijská hra go. U šachů variant tahů je velké množství, ale ve chvíli, kdy počítač vyhrál, se začalo řešit, zda se již nestalo, že počítač má pouze graf, ve kterém se pohybuje dle tahů soupeře. Neboli že jde jen o omezené možnosti kombinací. Závěr na omezené možnosti kombinací se vyskytovali i v případech, kdy počítače dokázaly psát básně či povídky. I zde totiž jde „pouze“ o určitou kombinaci již známých slov [35].

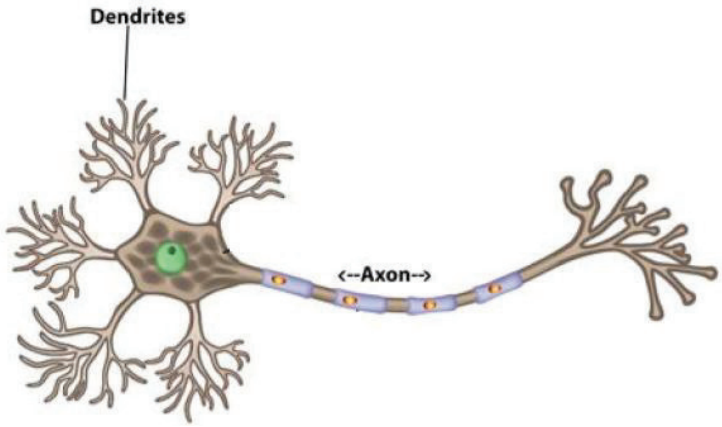
Počítače dokáží překvapit nejen chybami, ale i velmi zajímavými strategiemi, které lidem unikají. Velký úspěch počítače je považován u asijské hry go, kdy počítač/program nazýván AlphaGo vyhrál v roce 2016 nad nejlepším hráčem světa [37], [38], [39], [40].

Hlavní využití je tedy tam, kde kombinací je tolik, že se musí využívat určité myšlení a učení. Metody, které se tímto zabývají, jsou neuronové sítě viz kap. 4.2 a evoluční algoritmy, viz kap. 4.3. Některé možnosti využití multiagentních systémů je rozebráno v kap. 4.4.

4.2 Neuronové sítě

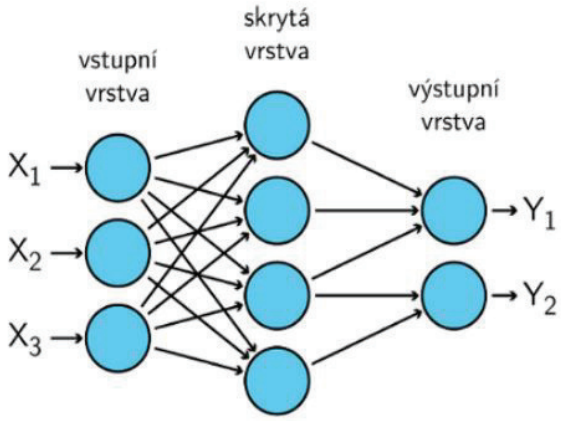
Velkým potenciálem mozku je učení se. Je fantastické, že dítě narozené dnes má principiálně stejný mozek, jako ve starověku, jde tedy pouze o to, jaké informace se mozku dají. Proto je snaha zjistit, jak mozek funguje a využít jeho principů [41], [42].

Mozek má velké množství neuronů, které mohou mít mnoho vstupů (dentritů), ale vždy mají jen jeden výstup (axon) [10], [9], [41], viz obrázek obr.12 [43].



Obr. 12 Neuron

Pro učení je třeba velké množství dat s informací, co data znamenají. Například fotky s ozubenými koly. Síť prochází fotky stále dokola a upravuje a zlepšuje parametry daných neuronů. Některé vrstvy sítě se zaměřují na barvy, tvary či třeba struktury další, komplexnější vrstvy se zaměřují na rozpoznávání oka, čumáku a ve finála, zda se jedná skutečně o ozubené kolo [10], [27]. Schéma struktury neuronových sítí je na obr. 13.



Obr. 13 Struktura neuronové sítě

Kvalita učení závisí především na kvalitě vstupních dat. To může znamenat jak množství, ale především rozmanitost. Může se stát, že dáme neuronové síti novou fotografii, kde ozubené kolo vůbec není a ona prohlásí, že zde ozubené kolo je, protože se nenaučila rozpoznat ozubené kolo, ale třeba oblohu, nebo pouze ozubené kolo ocelové s bílým pozadím [10], [27].

Pro optimalizaci výrobního procesu výukového modelu chytré továrny se neuronové sítě využívat nebudou. Dá se však přemýšlet o využití v dalších fázích optimalizace, kde by právě tyto sítě mohly pomoci s optimalizací parametrů evolučních algoritmů, které se využívat budou.

4.3 Evoluční a genetické algoritmy

Evoluční algoritmy, pro potřeby této práce lze říkat genetické algoritmy (GA), se inspirují evolucí. Tedy se v nových generacích vyskytují tyto evoluční operace:

- Dědičnost.
- Křížení.
- Mutace.
- Přirozený výběr.

Nový jedinec má dva rodiče. Jeden rodič se bere jako hlavní, ze kterého se v základu dědičností skládá, tedy přebírá chromozon. Od druhého rodiče přejímá křížením část vlastností, tedy genů. Mutace je dobrá, aby vzniklo něco zcela nového, úplně nový druh, který by mohl mít nové, zcela odlišné a pozitivní účinky, kterých by nové generace jen křížením těžko či vůbec nedosáhly. Na konci generace před tvorbou další generace je nutné zvolit, kteří jedinci zůstanou a kteří se budou reprodukovat do dalších generací [9], [10], [44], [45], [46].

V problematice GA jsou využívány tyto pojmy [8], [10]:

Chromozom	vektor jednoho řešení
Gen	část chromozomu
Alela	množina hodnot, kterých mohou geny nabývat
Účelová funkce	funkce, u které hledáme extrém, obvykle globální minimum, vyjadřuje například potřebný čas pro dokončení všech zakázek
Fitness funkce	vyjadřuje úspěšnost řešení, bývá rovna převrácené hodnotě účelové funkce, čím je vyšší, tím je řešení lepší

Jednoduše řečeno lze říci, že výhodou GA je, že vezmeme určitá řešení (jedince), ty zkřížíme, zkombinujeme. Tím vytvoříme další generace, ze které vybereme nejlepší a vytvoříme nové generace. Celý proces se několikrát opakuje a vznikne výsledek, který sice není ze všech možných řešení nejlepší, ale je získán v relativně reálném čase. To znamená, že se obětuje nejlepší řešení za nereálný čas a získá se obstojné řešení téměř okamžitě. Problém je však v nastavení všech algoritmů křížení, mutace a výběru jedinců pro další generace, čímž se zabývá kapitola 5.2.

Tím, že se v úvodu vygenerují náhodná řešení, tedy jedinci, kteří se v dalších generacích vylepšují, lze tyto GA využít všude, kde dochází k zlepšování řešení. Využití je například v těchto oblastech [44], [46], [47]:

- Dělení tyčí.
- Uspořádání výpalků na plech.
- Doprava, logistika, problematika obchodního cestujícího.
- Řízení výroby.

Dělení tyčí a uspořádání výpalků na plech optimalizuje uspořádání zakázek do materiálu tak, aby bylo zapotřebí co nejméně nakupovaného materiálu. Vždy se jedná o zadání úče-

lové funkce, kdy například zbytek kulatiny větší jak půl metrů může být pro nás hodnotnější, než 30 cm, protože malý zbytek již zřejmě nevyužijeme.

V dopravě tyto GA denně navrhují a rozvrhují trasy kamionů, aby vše stíhaly, zvládaly a základem problematiky je obchodní cestující, kde se řeší problém jak navštívit všechny města tak, aby se každé navštívilo jen jednou a cesta byla co nejkratší.

U řízení výroby GA rozvrhují zakázky tak, aby se maximalizovalo využití strojů, minimalizovaly ztrátové časy a uspořádávají zakázky dle dalších pokynů. Toto s problematikou v logistice lze shrnout do Vehicle Routing Problem (VRP) [1], [48]:

- Vozidlo má omezenou kapacitu.
- Časová okna – zákazník musí být obsloužen v daný čas.
- Je možné využívat více skladů/strojů.
- Zákazník zboží nemusí převzít, reklamační hospodářství.
- Nahodilé, neplánované požadavky.
- Pravidelné svozy/rozvozy.

V řešené problematice optimalizace výroby náramku je možné v dalších vývojových etapách pokračovat a rozšiřovat o výše zmíněné problematiky. V aktuálním stavu se bude pouze řešit rozvržení a naplánování zakázek v pravidelných intervalech. Tedy systém vše vyhodnotí například každý den v 6:00. Existuje více možností, jak řídit výrobu, zda centrálně, či decentralizovaně. Tyto možnosti jsou rozebrány v následující kapitole multiagentní systémy.

4.4 Multiagentní systémy (MAS)

Řízení podniků, logistiky, a dalšího bylo centralizované, tedy všechny pokyny pocházely z jednoho centrálního mozku, který vše naplánoval, rozvrhnul a zadal úkoly. Co když se toto centrum porouchá? Navíc plánování je čím dál náročnější. Vzniká tedy myšlenka agentních systémů a využívání kyberneticko-fyzikálních systémů (Cyber-Physical Systems – CPS). Ve firmách je také několik stupňů managementu, kde každý stupeň řeší své a větší detaily se posílají k řešení dál. Multiagentní systémy se dají rozdělit dle integrace [7], [9], [7], [49]:

- Reaktivní agent – pouze reaguje na podněty, nevyužívá paměť a algoritmy.
- Autonomní agent – učí se, využívá UI, ale pracuje samostatně.
- Sociální agent – dokáže spolupracovat a kooperovat s ostatními agenty.
- Intencionalitní agent – agent má na paměti dlouhodobé cíle, problémy rozkládá a má chování k dosahování vytyčených cílů.

Pro případ chytré továrny a plánování uspořádání zakázek do výroby bude využit principiálně autonomní agent. Tento agent je centrálním mozkem plánování, který vše navrhne a pokyny pro výrobu dává dál vozíčkům. Pro vylepšování chytré továrny se dá systém vylepšit na sociální agenty, kdy každý vozíček bude agent. Toto řešení je náročnější, řeší však mnoho problémů, jako je například výpadek systému či výpadek vozíčku a řeší i všechny nahodilosti a nesrovnalosti, protože každý agent ví v daném čase, kde se nachází s jakými parametry a je schopen se domluvit s ostatními tak, aby se vše stále synchronizovalo a optimalizovalo.

5 Optimalizace výroby náramku s využitím více vozíčků

Proces začíná zákazníkem, který si na e-shopu zvolí náramek z korálků. Zákazník může zvolit barvy jednotlivých korálků. Zakázka je odeslána řídící centrále, která všechny zakázky vyhodnocuje, zpracovává a určuje parametry pro výrobu. Po sestavení náramku se expeduje zákazníkovi.

Celý algoritmus vychází z projektu na katedře výrobních systémů, kde se touto problematikou zabývají a je nad rámec této práce řešit kompletní genetický algoritmus. Cílem je navrhnout optimalizaci části tvorby nových populací. Proto je nejprve provedeno základní pochopení principu genetického algoritmu a analýza využívaných metod viz kapitola 5.2, kde je i návrh na úpravu. Následně je v kapitole 5.3 popsán jednoduchý GA (SGA), se kterým bude návrh úpravy GA (MGA), viz kapitola 5.4, srovnán. Navrhovaný MGA bude také porovnán s metodou nejkratšího procesního času (SPT), která se běžně využívá pro svou jednoduchost. V závěru je v kapitole 5.5 aplikace a zhodnocení.

5.1 Model

Tato část kapitoly se zabývá problematikou zpracování zakázek. Výstupem má být takové uspořádání, které docílí dokončení všech zakázek v co nejkratším čase. Tab. 7 znázorňuje ukázku zakázek, které ještě nebyly odeslány do výroby. Kde ČZ = pořadové číslo zakázky, B1 až B5 je barva korálku (případ o pěti barvách kuliček).

ČZ	B1	B2	B3	B4	B5
1	5	2	1	3	4
2	2	3	5	1	1
3	2	1	3	5	5
4	4	1	3	2	4
5	2	5	1	2	2
6	3	5	4	2	5
7	4	1	1	5	5
8	4	4	3	3	2
9	4	2	3	1	4
10	3	4	5	2	5

Tab. 7 Uskupení zakázek použité pro simulaci

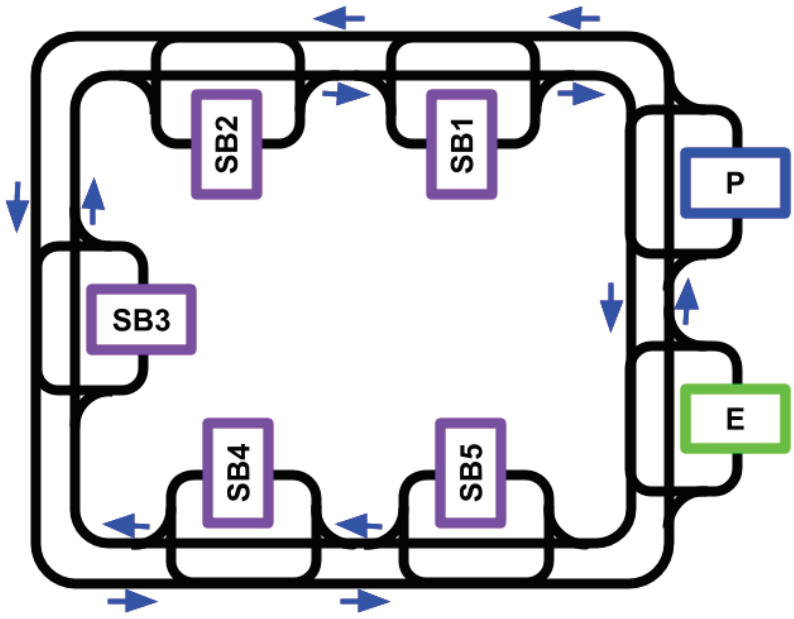
Cílem je uspořádat zakázky do takového pořadí, aby celkový čas, na vyrobení všech zakázek, byl co nejnižší. Může se tedy stát, že zakázka, která přišla od zákazníků první, půjde do výroby jako poslední.

Výroba funguje následovně:

- Volný vybraný vozíček jede pro korálky v pořadí dané zakázkou.
- Vozíček jezdí nejkratší možnou cestou.
- Ve chvíli, kdy před vozíčkem stojí jiný vozíček, zastaví a čeká.
- Ve chvíli, kdy vozíček přijede k zásobníku s korálky, zastaví a čeká na uvolnění korálku.
- Po uvolnění korálku vozíček odjíždí nejkratší povolenou cestou pro další korálek.
- Ve chvíli, kdy vozíček má všechny potřebné korálky, jede na expedici, kde korálky expeduje.
- Prázdný vozíček si zažádá o novou zakázku, pokud ji dostane, jede zakázku zhotovit. Pokud nedostane zakázku, jede na parkoviště.
- Vozíček na parkovišti stále žádá o novou zakázku.

Celý problém je nyní zjednodušen tak, že centrála si předem spočítá, v jakém čase se nachází který vozíček a v jakém stavu zakázky, tedy i kdy bude opět k dispozici. Ve chvíli, kdy vyhodnotí, že je vozíček schopen zakázku zhotovit, vyšle jej. Možné vylepšení do budoucna je právě v online zpracovávání dat o současném stavu vozíčků a jejich stavu baterie.

Následující obr. 14 znázorňuje uspořádání výrobní haly. Nachází se zde stanoviště expedice (E), sklady, neboli zásobníky či stroje jednotlivých kuliček dané barvy (SB1 až SB5) a parkoviště (P), kde vozíčky čekají a nabíjejí se. Tyto stanoviště spojuje dráha. Po dráze je možné jezdit pouze v povoleném směru (modré šipky).



Obr. 14 Výrobní hala

Jízda je po okruhu umožněna v obou směrech. Je nutné dle reálného modelu zadat vzdálenosti mezi jednotlivými stanovišti. Jedná se o následující tab. 8 , která je nyní vyplněna odhadovanými časovými nároky na jízdu v jednotkách sekund.

Z \ do	E	P	SB1	SB2	SB3	SB4	SB5
E	xxx	15	30	33	33	27	24
P	30	xxx	27	36	45	39	30
SB1	30	24	xxx	33	42	36	30
SB2	27	21	15	xxx	33	36	33
SB3	42	39	24	18	xxx	33	42
SB4	33	36	30	24	18	xxx	30
SB5	27	30	30	27	21	15	xxx

Tab. 8 Vzdálenosti mezi stanovišti v sekundách

Hodnoty v této tabulce se musí experimentálně ověřit reálnou jízdou vozíčku. K dispozici je pro plnění zakázek 10 vozíčků.

Posledním parametrem při jízdě je čas vyskladňování korálku, který se také musí experimentálně ověřit. Pro potřeby se bude počítat 10 s na vyskladnění.

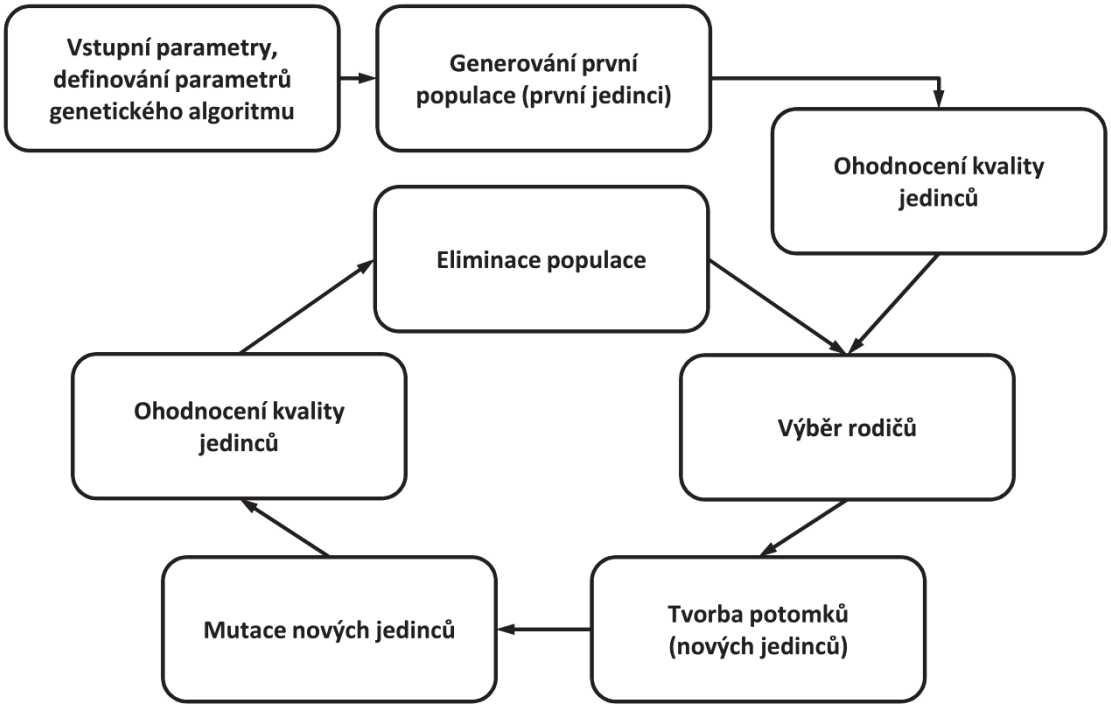
Číslování zásobníků / pozic v simulaci je od nuly včetně expedice a parkoviště, tedy E = 0, P = 1, SB1 = 2, SB2 = 3, SB3 = 4, SB4 = 5, SB5 = 6.

Cíl pro algoritmus:

Seskupit zakázky tak, aby celková doba všech zakázek byla co nejkratší, neboli minimalizovat čekání vozíčků při vyskladňování korálků jiným vozíčkům. Což se docílí také lepším vytížením skladů.

5.2 Návrh evolučního algoritmu

Algoritmus pracuje s jedinci, kteří se vyvíjení jednotlivými generacemi, neboli cykly výpočtu. V každé generaci dochází k volbě rodičů pro nové potomky/jedince, kteří se kříží a vytvářejí nové potomky. Geny nových potomků mohou zmutovat. Dalším krokem je selekce populace, kdy je nutné ponechat jen některé jedince, aby populace nebyla příliš velká. Celý proces zobrazuje následující obr. 15 [8], [50].



Obr. 15 Proces genetického algoritmu

Jednotlivé kroky algoritmu budou popsány v dalších částech této kapitoly.

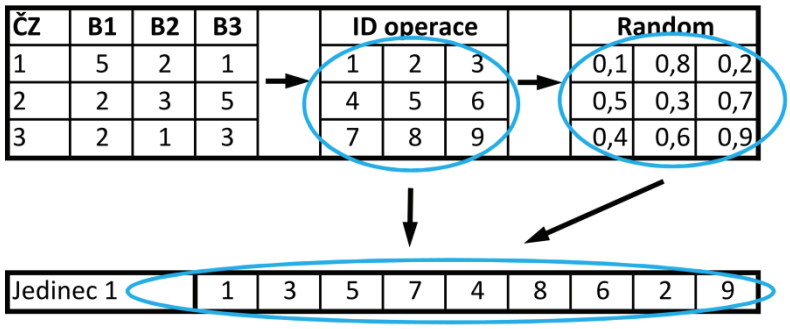
5.2.1 Definování vstupních parametrů

Vstupními parametry jsou všechny parametry potřebné v procesu výpočtu. Lokální parametry budou zmíněny dále. Mezi globální parametry patří počet generací, který bude použit. Vyšší počet generací zvyšuje pravděpodobnost nalezení lepších jedinců, ale zvyšuje čas výpočtu. Dalším možným parametrem je velikost populace. Toto číslo nemusí být striktně dodržováno, obvykle velikost populace osciluje kolem tohoto čísla. Více v části selekce potomků.

5.2.2 Generování první populace

První jedinci jsou generováni náhodně. Z tohoto důvodu se jich do první generace negeneruje velké množství, protože jejich kvalita je pouze náhodná. Kvalita a velikost populace se vyvíjí v dalších generacích.

Generování první populace probíhá tak, že zakázky uspořádáme například dle čísla zakázky (ČZ). Jedinec se skládá z genů, které v tomto případě znamenají ID operace. Tab. 9 znázorňuje tři zakázky, kde každá zakázka má požadavek na tři kuličky (B1 až B3) pěti barev. Následně je přiřazeno chronologicky ID každé operaci, tedy každé kuličce. Následuje přiřazení náhodného čísla každému ID, dle kterého je sestaven jedinec.



Tab. 9 Přiřazení ID operaci, tvorba náhodného jedince

Tedy ID operace s nejnižší náhodnou hodnotou jde na první místo genu jedince atd… Celému řetězci genů (ID) se říká chromozom.

5.2.3 Ohodnocení kvality jedinců

Kvalita jedinců je označována jako hodnota fitness funkce a je dána účelovou funkcí. Neboli je jedinec, tedy konkrétní řešení spočteno/simulováno. Výsledkem simulace v případě kompletace náramků je čas potřebný pro vyrobení všech zakázek. Fitness funkce je upravená převrácená hodnota zmíněného potřebného času na vyrobení.

Příklad vysvětlení aplikace jedince je následující: nejprve se vezmou všechny zakázky a jedinec, zde „Jedinec 1“, viz tab. 10.

ČZ	B1	B2	B3		ID operace		
1	5	2	1		1	2	3
2	2	3	5		4	5	6
3	2	1	3		7	8	9

Jedinec 1	1	3	5	7	4	8	6	2	9

Tab. 10 Seznam zakázek a jedinec řešený pro hodnotu fitness funkce

Nyní se vypustí volný vozík. Přiřazení zakázky vozíku je dle dostupných zakázek, v úvodu jsou k dispozici zakázky Z1, Z2 a Z3. Volba zakázky je dle chromozomu. Tedy který gen, neboli ID z těchto zakázek je první. V chromozomu je na prvním místě gen 1, který patří zakázce číslo 1. Dále se volí ze zakázek Z2 a Z3. V chromozomu je první ID 5, která patří zakázce číslo 2. Jako poslední pojede vozík se zakázkou 3. Zakázky tedy pojedou v pořadí Z1, Z2 a Z3.

Ve chvíli, kdy na operaci / stroj, tedy zásobník s korálky přijedou dva vozíky, musí být rozhodnuto, který vozík pojede jako první. První pojede ten, který má dané ID operace v chromozomu dřív.

V chromozomu si lze všimnout, že ID pro první zakázku je v pořadí 1 → 3 → 2. Z technologického postupu však má být 1 → 2 → 3. Zde je vidět problém náhodného generování pořadí genů a dalších procesů. Tato problematika je nad rámec diplomové práce a využívá řešení projektu na katedře.

5.2.4 Výběr rodičů (selekce)

Výběr rodičů patří jako tvorba nových jedinců a eliminace populace k velmi náročné problematice. V literatuře se vyskytuje běžně několik variant selekce rodičů:

- Náhodný výběr.
- Ruletové kolo.
- Rank Selection.
- Turnaj.
- Distribuované.

Všechny tyto varianty jsou dále více popsány.

Náhodný výběr

Náhodný výběr jedinců nepracuje nijak s kvalitou jedinců, jedná se tedy více o náhodné prohledávání.

Ruletové kolo.

Metoda ruletového kolo pracuje s myšlenkou, že se hodí mít v populaci nejen nejsilnější, ale i střední a nižší vrstvu neboli jedince s horší hodnotou fitness funkce. Princip výběru jedinců pro další populaci a i jako rodiče je obdobné ruletě. Pravděpodobnost výběru jedince je obdobná kvalitě fitness funkce [51], [52], [53], [54], [55].

Tento mechanismus nezaručuje vybrání nejlepších jedinců, kteří mají i velký potenciál ke zlepšení, je-li populace skutečně velká, ca od 1000 jedinců [53], [55].

Problém tohoto mechanismu nastává také ve chvíli, kdy mají všichni jedinci velmi podobnou hodnotu fitness funkce, což může být v rozmezí 20 %. V tuto chvíli se prohledávání začne jevit spíše jako náhodně, než aby se generacemi postupně spělo k lepším jedincům [53].

Obdoba předešlého problému je, má-li jen pár jedinců mnohem lepší, například třikrát, fitness hodnotu než ostatní jedinci. Dochází k uvíznutí v lokálních optimech, protože tito jedinci jsou téměř vždy kříženi. Řešením je využití mechanismu Rank Selection [53]. Obecně je metoda ruletového kola považována za mnohochybnou a pomalou a proto byly vyvinuty další metody [56].

Metoda ruletového kola se dá využít i pro eliminaci populace, kde vyšší fitness funkce jedince znamená vyšší pravděpodobnost přežití do další generace.

Rank Selection.

Metoda Rank Selection zakládá na úpravě metody rulety. Základem je změna pravděpodobnosti výběru jedince jako rodič. Pravděpodobnost není s fitness hodnotou, ale dle pořadí jedinců, které je dáno fitness hodnotou. Dochází tedy k rovnoměrnějšímu výběru rodičů i v populacích s velmi rozdílnými jedinci, což je právě hlavní problém metody ruletového kola [51], [53], [54], [56], [57].

Rank Selection také eliminuje předčasnou konvergenci, což je problematika metody rulety, je to dáno tím, že se mohou častěji křížit jedinci s velmi rozdílnými vlastnostmi, což do jisté míry může částečně představovat mutaci. Mutace je však zcela náhodné prohledávání,

kdežto Rank Selection využívá jedinců ze současné populace, ke kterým generacemi postupně dospěla [56].

Metoda Rank Selection se dá využít i pro eliminaci populace, kde pořadí jedince znamená vyšší pravděpodobnost přežití do další generace.

Turnaj.

Metoda turnaje se inspiruje rytířskými turnaji, kde princeznu získal nejlepší rytíř. Metoda tedy funguje tak, že se například zvolí náhodně první rodič. Následně se náhodně vybere určitý počet jedinců, kteří mezi sebou zápasí. Vyhrává jedinec s nejlepší fitness funkcí. Alternativou může být, že turnaj vyhraje náhodně vybraný, nebo jedinec v pořadí prostřední [51], [54], [56].

Alternativou je, že je náhodná volba rodičů a náhodné zvolení rodiče pro dědění a rodiče pro křížení [51], [53].

Distribuované.

Tato metoda se inspiruje osídlováním měst, venkova, kdy lidé nebydlí všichni spolu. Někteří si jsou blíže a proto je i vyšší pravděpodobnost, že se dají dohromady. Tato distribuovaná metoda tedy sice pracuje s celou populací, ale rozmísťuje jedince do určitého prostoru, obvykle 2D či 3D. Následný výběr rodičů je v okolí. V rámci pár generací mají jedinci omezené možnosti výběru partnerského rodiče, ale s rostoucím počtem generací se geny mohou přemisťovat libovolně v prostoru a tedy i křížit s aktuálně velmi vzdálenými jedinci [53].

Návrh selekce rodičů

GA jsou inspirovány přírodou, proto následuje návrh, který je takto inspirován.

Myšlenka z přírody: V přírodě dochází ke křížení jedinců. Jedinců, kteří mají málo společného. Potomek má většinu genů děděnou jedincem se silnějším genomem. Druhý jedinec je přikřížen. Potomci neojediněle mají lepší vlastnosti z obou rodičů, čímž dochází k potomkům s úžasnými vlastnostmi, ke kterým by nedošlo křížením stejného druhu [58], [59], [60], [61].

Optimalizační návrh: Rodiče by měli být z různých vrstev, tedy v rámci populace by měly být vybíráni jedinci s velkým rozdílem hodnoty fitness funkce.

Tato myšlenka využívá silnějších a slabších jedinců v populaci. První rodič je vybrán s pravděpodobností odpovídající fitness funkci. Druhý jedinec se vybírá s pravděpodobností odpovídající převrácené hodnotě fitness funkce. Tedy nejvyšší pravděpodobnost výběru mají jedinci s nejnižší hodnotou fitness funkce. Pravděpodobnost výběru prvního a druhého rodiče je následující:

$$pravděpodobnost\ výběru\ 1.rodiče\ PVJ_i = \frac{FF_i}{\sum_{i=1}^{n} FF_i}$$

$$pravděpodobnost\ výběru\ 2.rodiče\ PVJ_i = \frac{\frac{1}{FF_i}}{\sum_{i=1}^{n} \frac{1}{FF_i}}$$

kde:

PVJ_i	pravděpodobnost výběru jedince
FF_i	hodnota fitness funkce daného jedince
n	počet jedinců

5.2.5 Tvorba potomků/jedinců

Když jsou vybráni dva jedinci, tedy rodiče, dochází k dědičnosti a křížení. Rodič pro dědičnost a množství potomků může být řešeno těmito způsoby:

- Jeden potomek. Dědičnost je z rodiče s vyšší hodnotou fitness funkcí.
- Jeden potomek. Dědičnost je z rodiče s nižší hodnotou fitness funkce.
- Jeden potomek. Náhodná volba rodiče pro dědičnost.
- Dva potomci. Jeden má geny děděny z jednoho a druhý z druhého rodiče.
- Zbytkový stochastický výběr (Reminder stochastic sampling) -rodiče mají více potomků.

Počet potomků daného jednoho páru rodičů obvykle bývá jeden či dva. Jsou-li dva potomci, bývá dědění řešeno tak, že je chromozom děděn u prvního potomka z prvního rodiče a druhý potomek má zděděn chromozom druhého rodiče. Další variantou je zbytkový stochastický výběr. Tato metoda vychází z mechanismu rulety. Snaží se odstranit její nedostatky a předpokládá, že nejlepší jedinci mají nejlepší geny a mají proto nejvyšší pravděpodobnost

stvořit potomky ještě lepší než jsou oni sami. Zároveň se toto nepředpokládá u slabších jedinců. Proto návrh zbytkového stochastického výběru pracuje s množstvím potomků každého rodiče, který je dán na základě kvality fitness hodnoty. Silnější jedinci tedy budou mít více potomků než jedinci s nižší hodnotou fitness funkce. Nová populace je následně tvořena pouze novými potomky [56].

V problematice algoritmu pro kompletaci náramku se využívá generování dvou potomků, kdy jeden je z jednoho rodiče a druhý potomek z druhého rodiče.

Pro křížení samotných genů se využívá uniformní křížení, které projíždí gen po genu a náhodné číslo udává, zda gen bude překřížen či nebude. Gen je překřížen, když je náhodné číslo větší jak definované číslo, například 0,8, tedy dochází ke křížení u 20 % genů [62].

Křížení je komplikovanější záležitostí. Pro křížení může být dán počet překřížených rodičů větší jak jedna. Také záleží na množství prokřížených genů. Dá se dělat i křížení určitých celků chromozomu. Dále je ukázán princip křížení genů a oprava jedince.

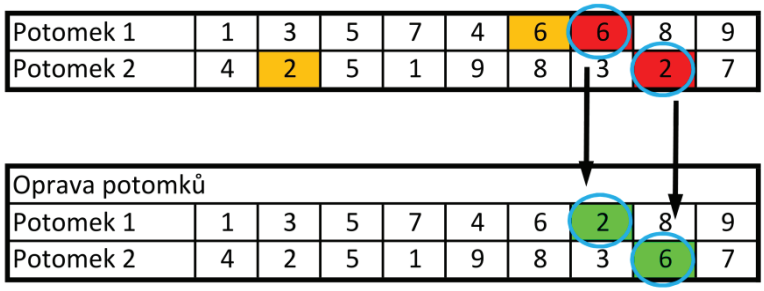
Ukázka křížení genů

Křížení probíhá mezi dvěma geny. Zda ke křížení dojde, určuje vygenerované číslo, které musí překročit 80 %. Tab. 11 ukazuje dva jedince, kteří předávají geny dvěma potomkům a zároveň dochází ke křížení některých genů dle vygenerovaného náhodného čísla.

Random	0,1	0,5	0,6	0,2	0,6	0,9	0,5	0,8	0,3
Jedinec 1	1	3	5	7	4	8	6	2	9
Jedinec 2	4	2	5	1	9	6	3	8	7
Potomek 1	1	3	5	7	4	6	6	8	9
Potomek 2	4	2	5	1	9	8	3	2	7

Tab. 11 Křížení jedinců

Při tomto způsobu křížení zde dochází k problému nereálnosti řešení, kdy některé ID v chromozomu chybí. Proto musí dojít k opravě, viz tab. 12.



Tab. 12 Oprava potomků

Překřížený gen s ID 6 potomka 1 se vyskytl v chromozomu dvakrát a operace s ID 2 z chromozomu vypadla. Opravou chromozomu dochází k přepsání druhého genu s ID 6 na gen ID 2. Totéž u potomka 2, jen je nahrazen gen s ID 2 na 6, který vypadl.

Tato oprava je jedna z možných, zároveň stále nejsou ID jednotlivých zakázek v pořadí dle technologického postupu. Oprava chromozomů je nad rámec této diplomové práce, je však vhodné jisté pochopení principů.

Návrh křížení genů

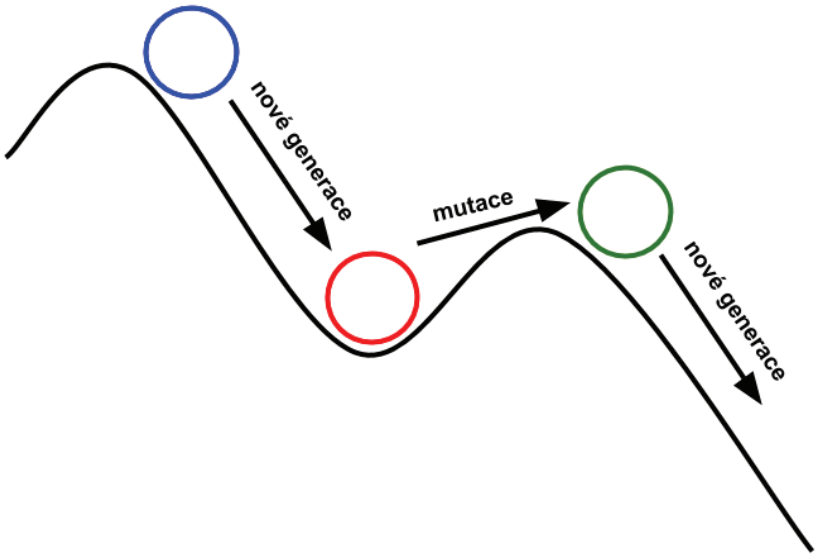
Současné metody křížení jsou náhodné. Náhodnosti se dá využít, když se nezná problematika a je tedy náročné odhadnout možný vývoj populace. V případě uskupení zakázek se problematika zná a je možné postup křížení modifikovat. Proto je snaha inspirovat se přírodou pro návrh optimalizace křížení.

Myšlenka z přírody: Lepší porozumění genetiky začalo mít snahu křížit rostliny a vše živé tak, aby využili to nejlepší. Této problematice se velmi věnoval pan Mendel, který pracoval s hrášky či včelami. V přírodě samotné či uměle člověkem, je snaha křížit tak, aby se využily nejlepší geny obou rodičů. Díky tomu mohou vzniknou noví jedinci zajímavých vlastností [63], [64], [65].

Optimalizační návrh: Chromozom se vyvíjí jednotlivými generacemi. Co se však stane, když se část genu změní na začátku evoluce či až ke konci evoluce? Výsledek se sice dá očekávat stejný, pokud jsou jedinci totožní, evoluce by však postupovala zcela odlišně. Proto je myšlenkou měnit pravděpodobnost křížení počtem generací. Možností je, že by křížení mohlo probíhat vždy v celém rozsahu chromozomu, ale v prvních generacích by byla vyšší pravděpodobnost křížení na začátku chromozomu a na konci evoluce by nejvyšší pravděpodobnost křížení byla na konci.

5.2.6 Mutace nových jedinců

Mutace je náhodná změna části jedince. Protože je mutace náhodná, neslouží k okamžitému zlepšení jedince, ale k zamezení uvíznutí nejlepšího řešení/jedince v lokálním optimu. Mutace má tedy za cíl náhodnou změnu velmi malé množství jedinců tak, aby se další generace mohly dostat z lokálních optim, viz obr. 16.



Obr. 16 Vliv mutace jedinců

Řešení se nejprve nachází někde mezi horšími řešeními, viz modrá koule. Postupnou tvorbou nových jedinců se řešení zlepšuje, ale může uvíznout v lokálním optimu jako červená koule. Potřebná energie, aby se aktuální nejlepší řešení (jedinec) mohlo dostat do lepšího řešení je mutace, která náhodné jedince posune do prostoru zelené koule a populace se díky těmto jedincům může přes další generace nadále zlepšovat.

5.2.7 Eliminace populace

Každou generací přibývá mnoho nových jedinců. Je vhodné celou populaci průběžně eliminovat, některé potomky nechat zemřít, neboli vymazat. Rozhodně neplatí, že se nechává jen třeba 10 jedinců s nejlepší hodnotou fitness funkce, protože pro křížení jsou potřeba jedinci s různou hodnotou fitness funkce. Mezi základní metody eliminace populace patří tyto:

- Elitářský mechanismus.
- Ruletové kolo.
- Rank Selection.

Elitářský mechanismus funguje na principu přežití nejlepšího, kdy v každé populaci přežijí jen ti nejsilnější, a i ti nejsilnější se mohou rozmnožovat. V genetickém algoritmu toto představuje to, že pro další cyklus výpočtu se nechá pouze určité množství nejlepších, například 50 % jedinců. Toto má velkou snahu skončit v lokálním optimu a nedochází k velkému zlepšování jedinců [53], [56], [66].

Metoda ruletového kola a Rank Selection pracují s celou populací, kdy každý jedinec má odlišnou pravděpodobnost na přežití. Více podrobně jsou tyto metody popsány v kapitole 5.2.4.

Jedním z alternativních způsobů je stárnutí populace, kdy jedinci umírají na stáří. Funguje to tak, že každou generaci jedinci stárnou a splňují-li definované podmínky, omladnou. Mezi tyto omlazovací podmínky může patřit:

- Jedinec patří do určitého procenta jedinců s nejlepší hodnotou fitness funkce.
- Jedinec stvořil potomka, který je lepší než on.
- Omlazeni jsou náhodní jedinci.
- Omlazen je každý n-tý jedinec.
- Jedinec patří do určitého procenta s danou hodnotou fitness funkce.

Eliminace je nutná, jsou-li velké populace, dochází k spíše náhodnému prohledávání. Pravidla pro eliminaci jsou důležité, protože se může stát, že jedinec, který nemá velkou hodnotu fitness funkce, může mít kvalitní potomky a je proto vhodné, aby bylo otestováno více potomků.

5.3 Jednoduchý GA (SGA)

Jednoduchý GA, anglicky Simple Genetic Algorithm (SGA), je základní princip genetického algoritmu, který využívá hojně pravděpodobnost [56]. Tento algoritmus bude využit pro porovnání zlepšení navrhovaného GA v následující kapitole 5.4.

Následuje popis všech částí genetického algoritmu.

Generování první populace

První populace je náhodně vygenerována. Velikost populace je dána na 100 jedinců.

Ohodnocení kvality jedinců

Noví jedinci se simulují ve výrobním procesu kompletace náramků a spočítá se hodnota fitness funkce, která je převrácenou hodnotou celkového času na vyrobení všech zakázek.

V průběhu výpočtu se vytvoří celkem 50 generací. Výsledkem výpočtu genetického algoritmu bude jedinec s nejlepší hodnotou fitness funkce na konci výpočtu.

Výběr rodičů (selekce)

Výběr obou rodičů je náhodný. Rodiče jsou vybírání tak dlouho, dokud není vytvořeno nových potomků stejné množství, jako je rodičů.

Tvorba potomků / jedinců

Každý pár rodičů vytvoří dva potomky. První potomek má děděn chromozom z prvního rodiče a křížené geny s druhým rodičem. Druhý potomek má děděn chromozom z druhého rodiče a křížené geny má s prvním rodičem. Nových jedinců bude celkem 100.

Pro křížení genů je využito uniformní křížení. Hodnota pro křížení je 80%, bude tedy docházek v průměru ke křížení 20% genů.

Mutace nových jedinců

Mutace není využita.

Eliminace populace

Pro eliminaci bude využit elitářský mechanizmus viz kapitola 5.2.7. Celá populace jedinců je seřazena dle hodnoty fitness funkce. Pro další generaci se ponechá 50 % jedinců, tedy 100 jedinců s nejvyšší hodnotou fitness funkce.

Souhrn vstupních parametrů

Vstupní parametry pro výpočet algoritmu jsou následující:

Velikost populace	100 jedinců
Počet generací	50
Pravděpodobnost křížení	80%
Nových jedinců	100

Tab. 13 SGA souhrn vstupních parametrů

5.4 Memetický GA (MGA)

Tato kapitola má za cíl navrhnout možnosti optimalizace genetického algoritmu. Z podstaty věci, že se genetické algoritmy inspirují přírodou je vhodné se i v tomto případě inspirovat přírodou a zvířaty.

Princip generování první populace je shodný s SGA. První populace je náhodně vygenerována. Velikost populace je dána na 100 jedinců.

Princip ohodnocení kvality jedinců, tedy vypočtení hodnoty fitness funkce je vždy stejný, tedy je i zde shodný s metodou SGA.

Mutace také nebude použita, tedy je postup stejný s metodou SGA.

Eliminace populace bude shodná s metodou SGA. Jedinci se tedy srovnají dle hodnoty fitness funkce a nechá se 100 nejlepších jedinců.

Výběr rodičů (selekce)

Metoda SGA nechává výběr rodičů náhodě. V návrhu MGA se bude pracovat naopak účelově s celou populací. První rodič se vybere s pravděpodobností odpovídající fitness funkci. Druhý jedinec se vybírá s pravděpodobností odpovídající převrácené hodnotě fitness funkce. Tedy nejvyšší pravděpodobnost výběru mají jedinci s nejnižší hodnotou fitness funkce. Pravděpodobnost výběru prvního a druhého rodiče je následující, viz návrh selekce rodičů v kapitole 5.2.4.

$$pravděpodobnost\ výběru\ 1.rodiče\ PVJ_i = \frac{FF_i}{\sum_{i=1}^{n} FF_i}$$

$$pravděpodobnost\ výběru\ 2.rodiče\ PVJ_i = \frac{\frac{1}{FF_i}}{\sum_{i=1}^{n} \frac{1}{FF_i}}$$

kde:

PVJ_i	pravděpodobnost výběru jedince
FF_i	hodnota fitness funkce daného jedince
n	počet jedinců

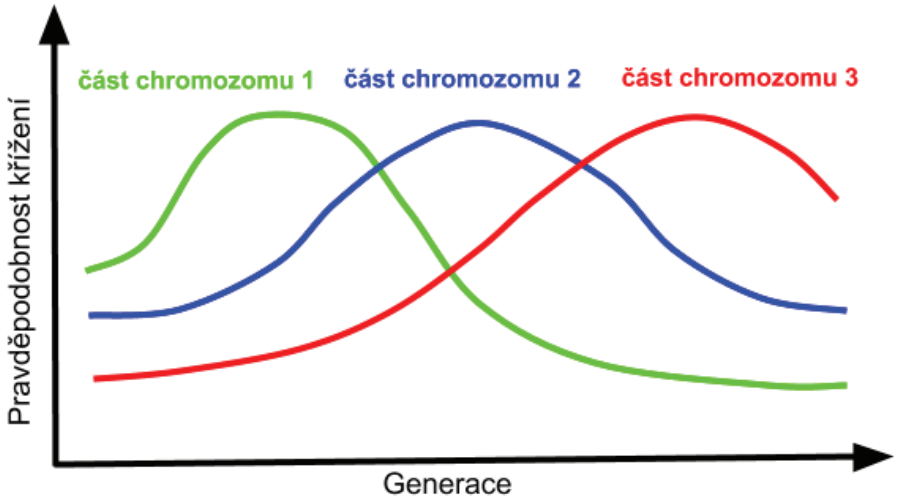
Myšlenkou je, aby noví jedinci měli geny z rozdílnějších rodičů a mohli se rychleji vyvíjet k lepším výsledkům.

Tvorba potomků / jedinců

Volba rodičů pro křížení a množství vzniklých potomků bude totožné s metodou SGA. Každý pár rodičů tedy vytvoří dva potomky. První potomek má děděn chromozom z prvního rodiče a křížené geny s druhým rodičem. Druhý potomek to má opačně a nových jedinců bude celkem 100.

Pro křížení genů se v tomto případě jeví jako velmi zajímavým návrhem využití znalosti řešeného problému, tedy návrhu uspořádání zakázek. S vysokou pravděpodobností se předpokládá, že změna pořadí zakázek na začátku výrazně změní výsledek řešení, kdežto konec uspořádání již takový vliv na výsledek řešení mít nebude. Z tohoto pohledu je proto vhodné se nejprve zabývat začátkem seskupení zakázek, postupně přecházet do střední části a ke konci výpočetních generací nejvíce pracovat s uspořádáním zakázek na konci. S touto myšlenkou pracuje návrh GA inspirovaný přírodou, viz kapitola 5.2.5.

Následující obr. 17 znázorňuje posun pravděpodobnosti křížení genu. Kdy chromozom je rozdělen na tři části. Část jedna je znázorněna zelenou barvou, část dvě modrou a část třetí červenou. Nejvyšší pravděpodobnost křížení genu v prvních generacích je v první části chromozomu, viz zelená linie. Třetí část (červená) se také může křížit, jen s nižší pravděpodobností.



Obr. 17 Posun pravděpodobnosti křížení částí chromozomu v průběhu generací

Chromozom nemusí být rozdělen jen na tři části, možné úpravy a další varianty jsou však nad rámec této práce. V této práci bude testována jednodušší varianta, kdy každá ze tří částí chromozomu bude mít tři pravděpodobnosti křížení pro tři části celé evoluce, tedy všech generací, viz následující tab. 14.

	Potřebná pravděpodobnost ke křížení dle generace		
Část chromosomu	0-16	17-33	33-50
1.	**70%**	85%	95%
2.	85%	**70%**	85%
3.	95%	85%	**70%**

Tab. 14 MGA - Potřebná pravděpodobnost ke křížení dle generace a části chromozomu

Počet překřížených genů je doplňkem k potřebné pravděpodobnosti, takže například první část chromozomu bude mít v generacích 0 až 16 zkříženo 30 % (100 %-70 %=30 %) a v posledních generacích 33 až 50 bude mít jen 5 % zkřížených genů (100 %-95 %=5 %).

Cílem tohoto návrhu je zvýšení účinnosti GA. Kdyby byla změna pouze v dané části, bude hledáno vždy jen optimum pro daný tvar chromozomu ve zbylých částech, čímž by se

hledalo lokální optimum. Je nutné, aby ke změně v ostatních částech chromozomu docházelo z toho důvodu, aby část, která je křížena nejvíce, byla stále prověřována pro různé varianty zbylé části chromozomu.

Souhrn vstupních parametrů

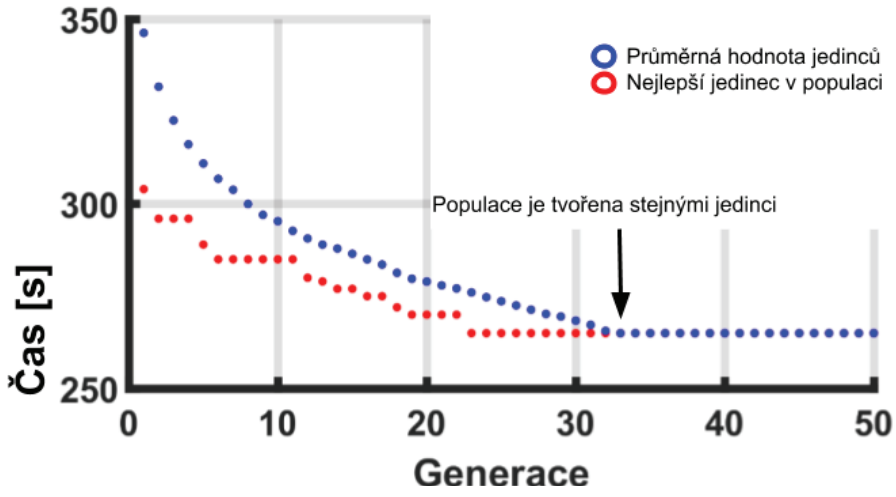
Vstupní parametry pro výpočet algoritmu jsou následující:

<table>
<tr><td>Velikost populace</td><td>100 jedinců</td></tr>
<tr><td>Počet generací</td><td>50</td></tr>
<tr><td>Pravděpodobnost křížení</td><td>
<table>
<tr><td></td><td colspan="3">Potřebná pravd. ke křížení dle generace</td></tr>
<tr><td>Část chromosomu</td><td>0-16</td><td>17-33</td><td>33-50</td></tr>
<tr><td>1.</td><td>70%</td><td>85%</td><td>95%</td></tr>
<tr><td>2.</td><td>85%</td><td>70%</td><td>85%</td></tr>
<tr><td>3.</td><td>95%</td><td>85%</td><td>70%</td></tr>
</table>
</td></tr>
<tr><td>Nových jedinců</td><td>100</td></tr>
</table>

Tab. 15 MGA souhrn vstupních parametrů

5.5 Aplikace a zhodnocení

Navrhovaná úprava genetického algoritmu MGA popsaná v kapitole 5.4 je postavena proti základní metodě SGA popsané v kapitole 5.3. Následující obr. 18 znázorňuje průběh jedinců generacemi v jednom z testů.



Obr. 18 Ukázka jedinců v průběhu generací MGA

Ve chvíli, kdy průměrná hodnota jedinců je rovna nejlepšímu jedinci, dochází k problému, kdy se populace již nemůže zlepšit (konvergence populace k lokálnímu extrému). To

je dáno jednoduchými pravidly pro eliminaci populace, kdy zůstávají jen ti nejlepší nehledě na to, zda si jsou jedinci podobní.

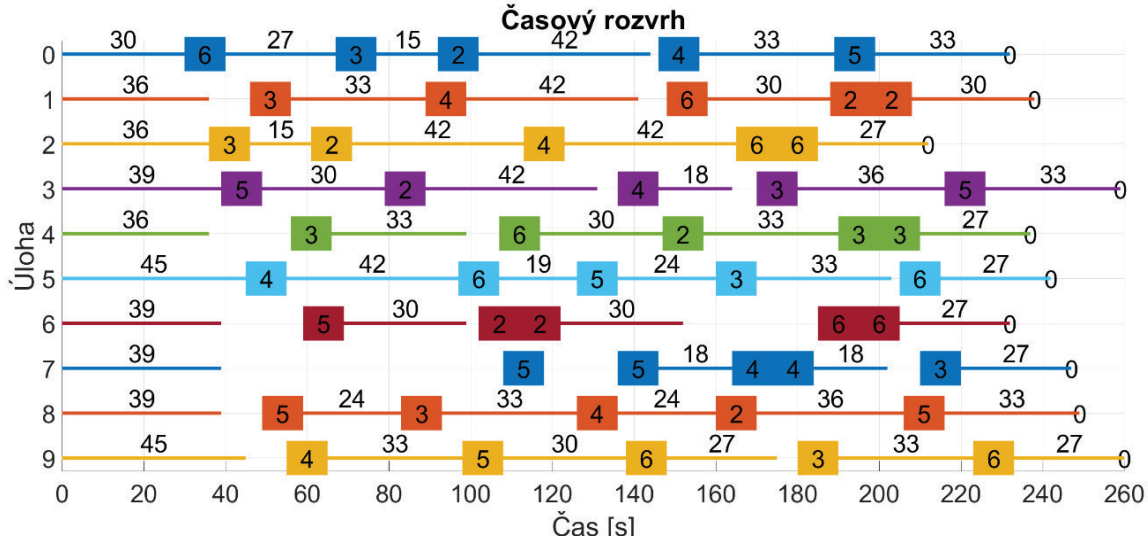
Následující tab. 16 zobrazuje 16 testování každé varianty. Grafy všech testování jsou v příloze. Pro SGA v příloze C a pro MGA v příloze D.

SGA																	
Pokus číslo	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Průměr
Generace, ve kterém byl poprvé nalezen lokálním extrémem	26	30	30	35	27	48	41	41	31	41	42	44	36	46	47	35	**37,5**
Generace, ve tkeré došla populace poprvé ke konvergenci	36	40	37	45	39	x	50	x	x	49	x	x	46	x	x	x	**46,4**
Hodnota účelové fce [s]	270	264	266	260	270	264	265	270	265	263	260	264	265	265	262	266	**264,9**
MGA																	
Pokus číslo	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Průměr
Generace, ve kterém byl poprvé nalezen lokálním extrémem	32	28	29	18	29	23	19	25	15	35	30	21	21	19	23	26	**24,6**
Generace, ve tkeré došla populace poprvé ke konvergenci	40	33	37	26	35	33	30	33	23	46	38	25	32	32	33	34	**33,1**
Hodnota účelové fce [s]	260	265	264	260	265	265	275	270	274	265	260	264	260	269	269	260	**265,3**

Tab. 16 Data z testů SGA a MGA

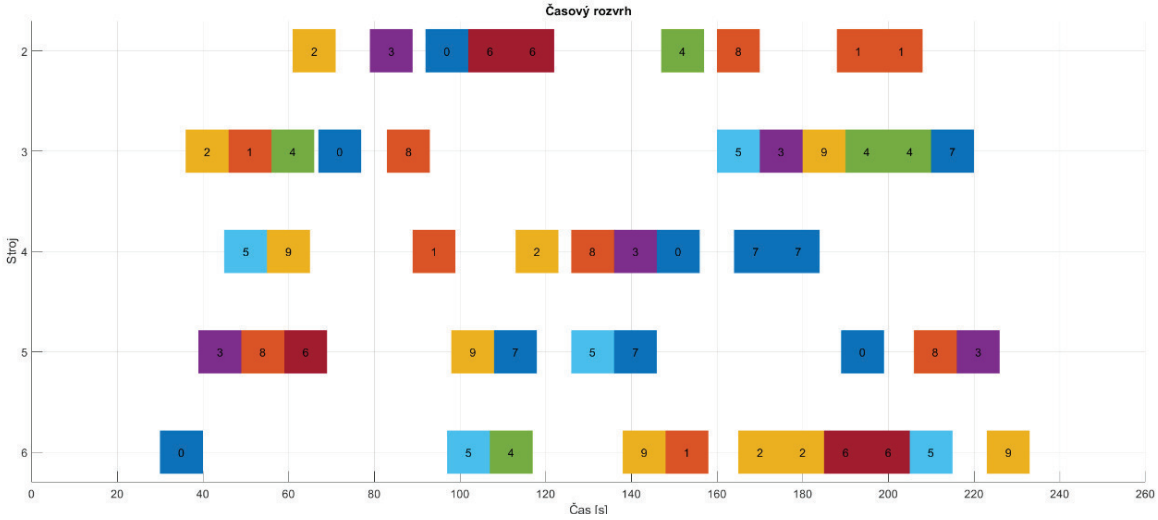
Oba GA se dostávají k hodnotě 260 [s]. Při využití jednoduché metody nejkratšího procesního času, která je velmi jednoduchá a rychlá, dojde výpočet na 286 [s]. Čas na výpočet metody nejkratšího procesního času je přibližně jedna sekunda. Výpočet obou GA trvá přibližně 2 minuty. Oba GA ukazují, že GA mají potenciál a smysl. Z tabulky je vidět, že metoda MGA rychleji konverguje. Při výpočtu se stává, že je populace tvořena stejnými jedinci, což je způsobeno striktním ponecháním nejlepších jedinců bez ohledu na podobnost s ostatními jedinci. Celkově se návrh MGA ukazuje jako velice zajímavý a je proto vhodné provézt jeho úpravy a testovat na komplikovanějších kombinatorických problémech.

Následující obr. 19 znázorňuje nejlepší spočtenou variantu uspořádání zakázek. Obrázek znázorňuje, v jakém čase je stav zakázky (vozíček).



Obr. 19 Výsledný průběh zakázek ve výrobním procesu

Plná čára s číslem nad čarou znázorňuje přejíždění vozíčku mezi stroji (sklady s kuličkami). Obdélník s číslem uvnitř značí vyskladňování kuličky na daném stroji, kdy číslo barvy odpovídá o jedna menší než uvedené číslo (barva 1 má číslo 2, …). Vynechané místo je prostoj / čekání. Pořadí zakázek na strojích zobrazuje následující obr. 20. Číslo stroje odpovídá opět barvě kuličky, jen o hodnotu o jedna posunutá (barva 1 má číslo stroje 2)



Obr. 20 Výsledné uspořádání zakázek na strojích

Na tomto grafu je dobře vidět, jak se zakázky na stroji sejdou, tedy jdou zakázky hned za sebou, nebo jsou stroje bez zakázek.

6 Závěr

Práce pokračuje v projektu výukového modelu chytré továrny, který běží na Katedře výrobních systémů a automatizace. Projekt je zaměřen na kompletaci náramků vyráběný z korálků barev daných zákazníkem. Tato diplomová práce se zabývá dvěma problémy:

- Optimalizací jízdních vlastností vozíčku – návrh optimalizace.
- Optimalizace pořadí zpracování zakázek.

Optimalizace jízdních vlastností je založena na zlepšení plynulosti a rychlosti jízdy vozíčku, který je mezi operacemi navigován pomocí černé čáry, a který jednotlivé zakázky kompletuje.

Uspořádání zakázek má za cíl zvolit pořadí zpracování zakázek tak, aby čas na vyrobení všech zakázek byl co nejkratší.

Závěr optimalizace jízdních vlastností

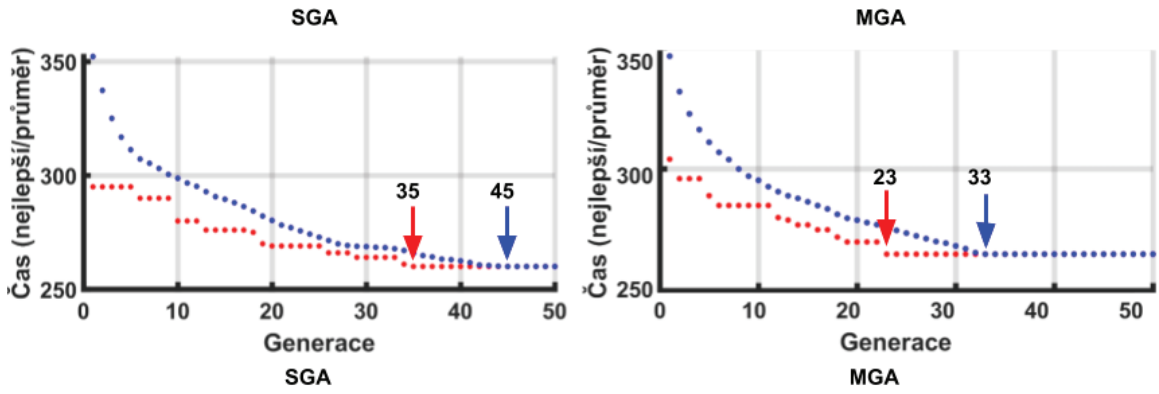
Optimalizace jízdních vlastností byla úspěšně provedena. Vozíček jezdí plynuleji a tím se docílilo i zvýšení rychlosti o ca 30 %, viz kapitola 3.3.2. Dále byl v rámci analýzy navržen postup při návrhu vozíčku a dráhy mezi stanovišti. Postup je více rozebrán v kapitole 3.4. Shrnutí je následující:

- Dráha s větším min. poloměrem: alespoň 400 mm, větší je lepší.
- Logika řízení: v zatáčkách se točí obě kola, na rovině se kola točí stejně rychle.
- Rozteč senzorů: blízké šířce čáry, ideálně o malinko menší.
- Vysunutí senzorů: 50 až 100 mm, větší umožňuje vyšší rychlost jízdy.

Závěr optimalizace uspořádání zakázek

Navrhovaný optimalizační krok, kdy se pravděpodobnost křížení mění v průběhu generací, se jeví jako smysluplný. Za testovaných podmínek se jedinci zlepšovali rychleji než u metody SGA. Metoda MGA také dosahovala častěji nejlepšího řešení. Nevýhoda v SGA a MGA je v tom, že dochází velmi brzy ke konvergenci. Při odstranění tohoto problému se může MGA jevit velmi dobře. Proto je navrhováno dále optimalizovat navržený MGA a testovat jej na komplikovanějších kombinatorických problémech.

Následující obr. 21 znázorňuje vybrané testy blížící se průměrným výsledkům.



Obr. 21 Porovnání SGA a MGA

Z obrázku je vidět posun hodnot u MGA oproti SGA směrem k dřívější generaci, což znamená rychlejší zlepšování jedinců.

Použitá literatura

[1] Průmyslová revoluce – *Od Průmyslu 1.0 k Průmyslu 4.0* - Desoutter Industrial Tools. Průmyslové nářadí Desoutter – Průmyslové pneumatické nářadí pro letecký a automobilový *průmysl* [online]. Copyright © 2019 Desoutter Industrial Tools. Všechna práva vyhrazena. [cit. 03.04.2019]. Dostupné z: https://www.desouttertools.cz/prumysl-4-0/novinky/563/prumyslova-revoluce-od-prumyslu-1-0-k-prumyslu-4-0

[2] CEJNAROVÁ, Andrea. *Od 1. průmyslové revoluce ke 4.* | Technický týdeník. TT | Technický týdeník 2015 [online]. Copyright © Business Media CZ, Nádražní 32, 150 [cit. 03.04.2019]. Dostupné z: https://www.technickytydenik.cz/rubriky/ekonomika-byznys/od-1-prumyslove-revoluce-ke-4_31001.html

[3] NĚMEC, Václav. SURÝ, Jan. *Anglická průmyslová revolue a její rozšíření.* dějepis.com. dějepis.com [online]. Copyright © 1997 [cit. 03.04.2019]. Dostupné z: http://www.dejepis.com/ucebnice/anglicka-prumyslova-revoluce-a-jeji-rozsireni/

[4] ČERNÝ, Michal. *Kvalita a Industry 4.0_cz* - IPA Czech. Firemní vzdelávaní, Inovace, Strategický rozvoj, Výrobní manažment, Optimalizáce výroby, Soft skills - IPA Czech 2017 [online]. Copyright © 2012 [cit. 03.04.2019]. Dostupné z: https://www.ipaczech.cz/cz/tisk-a-media/aktualni-vzdelavani/kvalita-a-industry-4-0cz

[5] *Práce 4.0*. Integrovaný portál MPSV 2018 [online]. [cit. 03.04.2019]. Dostupné z: https://portal.mpsv.cz/sz/politikazamest/prace_4_0

[6] KOŠTURIAK, Ján. *Štíhlý podnik* - IPA Slovník - IPA Czech. Firemní vzdelávaní, Inovace, Strategický rozvoj, Výrobní manažment, Optimalizáce výroby, Soft skills - IPA Czech [online]. Copyright © 2012 [cit. 03.04.2019]. Dostupné z: https://www.ipaczech.cz/cz/ipa-slovnik/stihly-podnik

[7] *Multiagentní systémy (MAS)*. University information system MENDELU [online]. [cit. 25.03.2019]. Dostupné z: https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=32363

[8] KOBLASA, František. *Uplatnění heuristických optimalizačních metod v oblasti rozvrhování strojírenské výroby středních a malých podniků.* Liberec, 2013. Disertační práce. Technická univerzita v Liberci.

[9] ČERMÁK, Petr. *Umělá inteligence*. MVŠO - Moravská vysoká škola Olomouc [online]. Copyright ©EO [cit. 22.03.2019]. Dostupné z: https://mvso.cz/wp-content/uploads/2018/02/Umělá-inteligence-studijní-text.pdf

[10] SIDORA, Juraj. *Logistika 4.0* - IPA Slovník - IPA Czech. Firemní vzdelávaní, Inovace, Strategický rozvoj, Výrobní manažment, Optimalizáce výroby, Soft skills - IPA Czech 2017 [online]. Copyright © 2012 [cit. 03.04.2019]. Dostupné z: https://www.ipaczech.cz/cz/ipa-slovnik/logistika-4-0-cz

[11] *Industry 4.0 je na vzestupu: naše senzorová řešení pro mobilní platformy* | SICK. 301 Moved Permanently [online]. Copyright © 2019 SICK AG [cit. 03.04.2019]. Dostupné z: https://www.sick.com/cz/cs/senzorova-reseni-pro-mobilni-platformy/w/mobile-platforms/

[12] Automatizované systémy. Vysokozdvižné vozíky, vzv, servis - Manipulační technika Toyota [online]. Dostupné z: https://toyota-forklifts.cz/nase-nabidka/produktova-rada/automatizovane-systemy/

[13] CIGÁNEKOVÁ, Monika. *Optimalizace logistického systému* - IPA Czech. Firemní vzdelávaní, Inovace, Strategický rozvoj, Výrobní manažment, Optimalizáce výroby, Soft skills - IPA Czech [online]. Copyright © 2016 [cit. 03.04.2019]. Dostupné z: https://www.ipaczech.cz/cz/tisk-a-media/aktuality/optimalizace-logistickeho-systemu

[14] *Inteligentní intralogistika | Tři fáze automatizace vnitropodnikového zásobování* | EMANS. Development of customized software, IT services: ANASOFT 2018 [online]. Copyright © Anasoft [cit. 03.04.2019]. Dostupné z: https://www.anasoft.com/emans/cz/home/Novinky-blog/Blog/Inteligentni-intralogistika

[15] *Na cestě k automatizovanému řízení skladu* | EMANS. Development of customized software, IT services: ANASOFT 2018 [online]. Copyright © Anasoft [cit.

03.04.2019]. Dostupné z: https://www.anasoft.com/emans/cz/home/Novinky-blog/Blog/na-ceste-k-automatizovanemu-rizeni-skladu

[16] *Automatizace. Klíč k optimalizaci Vaší intralogistiky.* AGV CZ | Jungheinrich. Homepage | Jungheinrich [online]. Copyright © 2019 Jungheinrich AG [cit. 03.04.2019]. Dostupné z: https://www.jungheinrich.cz/landingpages/agv-cz/

[17] *DIY RC Car*. Arduino Create [online], [cit. 03.03.2019]. Dostupné z: https://create.arduino.cc/projecthub/gyusik-song/diy-rc-car-fbd87a

[18] *Simple RC Car for Beginners (Android Control Over Bluetooth)* | Electronic | Pinterest | Arduino, Android and Arduino bluetooth. Pinterest [online], . [cit. 03.03.2019]. Dostupné z: https://cz.pinterest.com/pin/359865826456332035/?autologin=true

[19] *Arduino controlled RC-car*. Svenbluege [online], [cit. 03.03.2019]. Dostupné z: https://www.svenbluege.de/blog/hardware/135-arduino-controlled-rc-car

[20] GRIGARČÍK, David. *Konstrukční návrh všesměrového kola pro velká zatížení.* VŠB-Technická univerzita Ostrava. 2018 [online]. Copyright © [cit. 04.03.2019]. Dostupné z: https://dspace.vsb.cz/bitstream/handle/10084/129658/GRI0023_FS_N2301_2301T013_2018.pdf?sequence=1&isAllowed=n

[21] Line Follower Tutorial - CarduBot - Linienverfolger mit Arduino - YouTube. YouTube [online], [cit. 03.03.2019]. Dostupné z: https://www.youtube.com/watch?v=H2RAuL6r1Jg

[22] *Line Follower Tutorial – CarduBot – Linienverfolger mit Arduino*. Project Robot Test - YouTube. YouTube [online], [cit. 03.03.2019]. Dostupné z: https://www.youtube.com/watch?v=Rh8hH6rWYGk

[23] *Elektro izolační lepící pásky.* [online]. [cit. 5.02.2019] Dostupné z: https://www.az-hobby.cz/c/naradi/lepici-pasky-a-izolace/elektro-izolacni-lepici-pasky/?view_style=grid&id=product-list-c856&view_count=15&price_from=9&price_to=295&view_sort=stock

[24] *Páska samolepící PP 48 mm x 66 m, černá.* [online]. [cit. 12.02.2019] Dostupné z: https://www.paketo.cz/paska-samolepici-pp-48-mm-x-66-m-cerna/?gclid=EAIaIQobChMIw-GEnofV4AIVzeF3Ch2Cuwx7EAQYAyABEgKyoPD_BwE

[25] *Arduino Mikrocontroller: Selbstfahrendes Auto mit Rücksetzfunktion, dunkle Linie* - YouTube. YouTube [online]. [cit. 04.03.2019]. Dostupné z: https://www.youtube.com/watch?v=YfBn-ELiBpE

[26] CUTA, Andrei. *Test Line Follwer Custom Arduino* - YouTube. YouTube [online]. [cit. 04.03.2019] Dostupné z: https://www.youtube.com/watch?v=Dg2CX_qs1Ys

[27] TYL, Jan. *Všudypřítomná umělá inteligence bude megatrendem příštích deseti let*. Systemonline [online]. [cit. . 22.03.2019] Dostupné z: https://www.systemonline.cz/business-intelligence/vsudypritomna-umela-inteligence-bude-megatrendem-pristich-deseti-let.htm

[28] *17VUIB - Umělá inteligence pro biomedicínské inženýrství* | predmety.fbmi.cvut.cz. Biomedicínský technik - prezenční a kombinovaná forma | predmety.fbmi.cvut.cz [online]. Copyright © ČVUT v Praze [cit. 22.03.2019]. Dostupné z: https://predmety.fbmi.cvut.cz/cs/17vuib

[29] *Heuristika*. [online]. Copyright © [cit. 22.03.2019]. Dostupné z: https://it-slovnik.cz/pojem/heuristika

[30] *Heuristika.* [online]. Copyright © [cit. 22.03.2019]. Dostupné z: http://lidovyslovnik.cz/index.php?dotaz=heuristika

[31] NĚMEC, Václav. ČÍŽEK, Tomáš. *Vznik a vývoj člověka a lidské společnosti* .dějepis.com [online]. Copyright © 1997 [cit. 22.03.2019]. Dostupné z: http://www.dejepis.com/ucebnice/vznik-a-vyvoj-cloveka-a-lidske-spolecnosti/

[32] *Vznik a vývoj člověka* | Genetika - Biologie. Genetika - Biologie | Váš zdroj informací o genetice a biologii [online]. Copyright ©2010 [cit. 22.03.2019]. Dostupné z: http://www.genetika-biologie.cz/vznik-vyvoj-cloveka

[33] MEFISTOFELES. *Člověk rozumný - Homo sapiens sapiens* - PŘÍRODA.cz. PŘÍRODA.cz - příroda, ekologie, životní prostředí [online]. Copyright © 2005 [cit. 22.03.2019]. Dostupné z: https://www.priroda.cz/clanky.php?detail=501

[34] *Turingův test* - ManagementMania.com. [online]. Copyright © 2011 [cit. 22.03.2019]. Dostupné z: https://managementmania.com/cs/turinguv-test

[35] MICHÁLEK, Bedřich. *Turingův test – může počítač myslet?* – Masarykova univerzita. 2003 [online]. [cit. 22.03.2019]. Dostupné z: https://www.fi.muni.cz/usr/jkucera/pv109/2003p/xmichal4.htm

[36] *Paradoxy umělé inteligence: Turingův test 50 let poté* - Root.cz. Root.cz - informace nejen ze světa Linuxu [online]. Copyright © 1998 [cit. 22.03.2019]. Dostupné z: https://www.root.cz/clanky/paradoxy-umele-inteligence-turinguv-test-50-let-pote/

[37] *Počítač rozdrtil světového šampióna, ve hře go vyhrál čtyřikrát* – Novinky.cz . Novinky.cz – nejčtenější zprávy na českém internetu 2016 [online]. Copyright © 2016 [cit. 22.03.2019]. Dostupné z: https://www.novinky.cz/internet-a-pc/hry-a-herni-systemy/397721-pocitac-rozdrtil-svetoveho-sampiona-ve-hre-go-vyhral-ctyrikrat.html

[38] TRLICA, David. *Úspěch umělé inteligence: AlphaGo od Googlu porazila nejlepšího Go hráče na světě*. Homepage - Svět Androida 2017 [online]. Copyright © 2017 [cit. 22.03.2019]. Dostupné z: https://www.svetandroida.cz/alphago-umela-inteligence/

[39] *Umělá inteligence porazila nejlepšího hráče ve starověké hře GO*. 6D Hub [online]. Dostupné z: https://www.6dhub.cz/smart-systemy/4516id/umela-inteligence-porazila-nejlepsiho-hrace-ve-staroveke-hre-go/

[40] HOFERKOVÁ, Kateřina. *Umělá inteligence od Googlu už nemá ve hře Go lidskou konkurenci* | Svět hardware. Svět hardware | homepage 2017 [online]. Copyright © 1998 [cit. 22.03.2019]. Dostupné z: https://www.svethardware.cz/umela-inteligence-od-googlu-uz-nema-ve-hre-go-lidskou-konkurenci/44529

[41] *Biologické algoritmy (4) - Neuronové sítě* - Root.cz. Root.cz - informace nejen ze světa Linuxu [online]. Copyright © 1998 [cit. 24.03.2019]. Dostupné z: https://www.root.cz/clanky/biologicke-algoritmy-4-neuronove-site/

[42] *Umělé neuronové sítě a neuroevoluce*. [online]. [cit. 24.03.2019]. Dostupné z: https://users.fit.cvut.cz/~rehorto2/otevrena-fakulta/neural-networks.html

[43] Traumatic Brain Injury Resource Guide - *Neuronal Firing*. Traumatic Brain Injury Rehabilitation and Resource Guide [online]. Copyright © 2019 Copyright Centre for Neuro Skills [cit. 24.03.2019]. Dostupné z: https://www.neuroskills.com/brain-injury/neuronal-firing.php

[44] LEHOCKÝ, Zdeněk. *Genetické algoritmy a jejich aplikace v praxi*. Programujte.com — odborný web zaměřený na oblast vývoje, návrhu a designu webových, mobilních a desktopových aplikací [online]. Copyright © 2005 [cit. 22.03.2019]. Dostupné z: http://programujte.com/clanek/2005072601-geneticke-algoritmy-a-jejich-aplikace-v-praxi/

[45] *Genetické algoritmy.* [online]. Copyright © [cit. 22.03.2019]. Dostupné z: http://physics.ujep.cz/~mmaly/vyuka/MPVT_II/Heuristiky/Ga.pdf

[46] MADĚROVÁ, Michala. *Gentické algoritmy ve zvoleném softwarovém prostředí*. Pardubice, 2007. Bakalářská práce. Univerzita Pardubice. [online]. Copyright © [cit. 22.03.2019]. Dostupné z: https://dk.upce.cz/bitstream/handle/10195/24765/text.pdf?sequence=1&isAllowed=y

[47] NEKVINDA. Michal. *Genetické algoritmy a jejich využití v řízení výroby, I. Část*. Systemonline. , 2001 [online]. [cit. 22.03.2019]. Dostupné z: https://www.systemonline.cz/clanky/geneticke-algoritmy-a-jejich-vyuziti-v-rizeni-vyroby-i-cast.htm

[48] DUŠAN, Salaj. *Vehicle Routing Problém Metódy riešenia*. Bratislava 2007. Diplomová práce. Univerzita Komenského, Bratislava. [online]. Copyright © [cit. 25.03.2019]. Dostupné z: http://www.dcs.fmph.uniba.sk/diplomovky/obhajene/getfile.php/Diplomovka.pdf?id=175&fid=290&type=application%2Fpdf

[49] *Úrovně řízení a typy managementu* - ManagementMania.com. [online]. Copyright © 2011 [cit. 25.03.2019]. Dostupné z: https://managementmania.com/cs/urovne-rizeni-a-typy-managementu

[50] CHUROVÁ, Vendula. *Optimalizační algoritmy inspirované živou přírodou*. Brno 2015. Institut biostatistiky a analýz LF a PřF MU, Masarykova Univerzita [online]. [cit. 22.02.2019].. Dostupné z: https://is.muni.cz/th/v7hck/uprava.pdf#page=45&zoom=100,0,437

[51] *Genetické algoritmy* [online]. Copyright © [cit. 27.03.2019]. Dostupné z: http://bio.felk.cvut.cz/~huptycm/Vyuka/IKTZ_prednasky/GAIKTZ0910.pdf

[52] PILNÝ, Petr. *3.0 Genetické algoritmy*.OriTas. Vítejte - Petr Pilný - OriTas [online]. Copyright © 2012 [cit. 03.04.2019]. Dostupné z: http://petrpilny.cz/?30-geneticke-algoritmy,47

[53] *Genetické algoritmy* [online]. [cit. 27.03.2019]. Dostupné z: https://flurry.dg.fmph.uniba.sk/webog/SuboryOG/bohdal/9-GenetickeAlgoritmy.pdf

[54] HORDĚJČUK, Vojtěch. *Genetický algoritmus*. Software Engineer and Bedroom Music Producer [online]. Copyright © 2008 [cit. 27.03.2019]. Dostupné z: http://voho.eu/wiki/geneticky-algoritmus/

[55] LEPS, Matěj. *Genetické algoritmy*. 2008 [online]. Copyright ©cWG [cit. 27.03.2019]. Dostupné z: http://mech.fsv.cvut.cz/~leps/teaching/mmo/prednasky/prednaska06_GA.pdf

[56] POŠÍK, Petr. *Genetické algoritmy.* České vysoké učení technické Praha. 2000 [online]. [cit. 27.03.2019]. Dostupné z: http://labe.felk.cvut.cz/~posik/pga/theory/ga-theory.htm

[57] *Genetické algoritmy* (GA) [online]. [cit. 27.03.2019]. Dostupné z: https://ksvi.mff.cuni.cz/~mraz/nn/slides/GenAlg/GenAlg1.pdf

[58] KUBÍČKOVÁ, Klára. *Šrámy, sexuální kanibalismus, střílení ostnů. Takové je zvířecí milování*. Idnes. 2017 [online]. [cit. 29.03.2019]. Dostupné z: https://www.idnes.cz/hobby/mazlicci/laska-zvireci-rise.A170506_2323563_hobby-mazlicci_bma?setver=touch

[59] *Mula* - Vše o plemeni. Hello Sandy - Hafíci, miciny a koníčci to dali dohromady [online]. [cit. 30.03.2019]. Dostupné z: https://www.hellosandy.cz/mula/

[60] ANDRLE, Michal. *9 nejzvláštnějších mezidruhových kříženců* – 21stoleti.cz. 21stoleti.cz – VĚDA KTERÁ VÁS BUDE BAVIT [online]. Copyright © 21stoleti.cz. All Rights Reserved. [cit. 05.04.2019]. Dostupné z: https://21stoleti.cz/2012/04/20/9-nejzvlastnejsich-mezidruhovych-krizencu/

[61] CHURAŇOVÁ, Eliška. *FOTO: Fascinující zvířecí kříženci* – ZOO Magazín. ZOO Magazín – Magazín o zvířatech v ZOO i mimo ně 2015 [online]. [cit. 05.04.2019]. Dostupné z: https://zoomagazin.cz/foto-fascinujici-zvireci-krizenci/

[62] STUDNIČKA, Vladimír. *Genetické algoritmy – Multi-Core CPU implementace.* Brno, 2010. Diplomová práce. Vysoké učení technické v Brně. [online]. [cit. 22.02.2019].. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=33586

[63] *Byl jednou jeden... vynálezce* - s01e16 - Mendel and Peas | Zavislak.to - CZ/SK Seriály a Filmy Online. Zavislak.to - CZ/SK Seriály a Filmy Online [online]. Copyright © Zavislak.to 2017 [cit. 05.04.2019]. Dostupné z: https://zavislak.to/serial/byl-jednou-jeden-vynalezce/s01e16

[64] OVESNÁ, Jaroslava. *Nové techniky ve šlechtění rostlin* | Úroda. Úroda | Aktuality z rostlinné produkce a zemědělského výzkumu 2016 [online]. Dostupné z: https://www.uroda.cz/nove-techniky-ve-slechteni-rostlin/

[65] *Gregor Johann Mendel (20.7.1822-6.1.1882): Otec genetiky*. ŽivotopisyOnline.cz | Osobnosti, životopisy, společnost 2019 [online]. Copyright © 2019. All Rights Reserved. [cit. 05.04.2019]. Dostupné z: http://zivotopisyonline.cz/gregor-johann-mendel-otec-genetiky/

[66] LEHOCKÝ, Zdeněk. *Genetické algoritmy a jejich aplikace v praxi*. Programujte.com — odborný web zaměřený na oblast vývoje, návrhu a designu webových, mobilních a desktopových aplikací [online]. Copyright © 2015 [cit. 27.03.2019]. Dostupné z: http://programujte.com/clanek/2005072601-geneticke-algoritmy-a-jejich-aplikace-v-praxi/

Seznam příloh

A Vliv rozteče a vysunutí senzorů a poloměru dráhy na poloměr logiky řízení 1 z 2

B Vliv rozteče a vysunutí senzorů a poloměru dráhy na poloměr logiky řízení 2 z 2

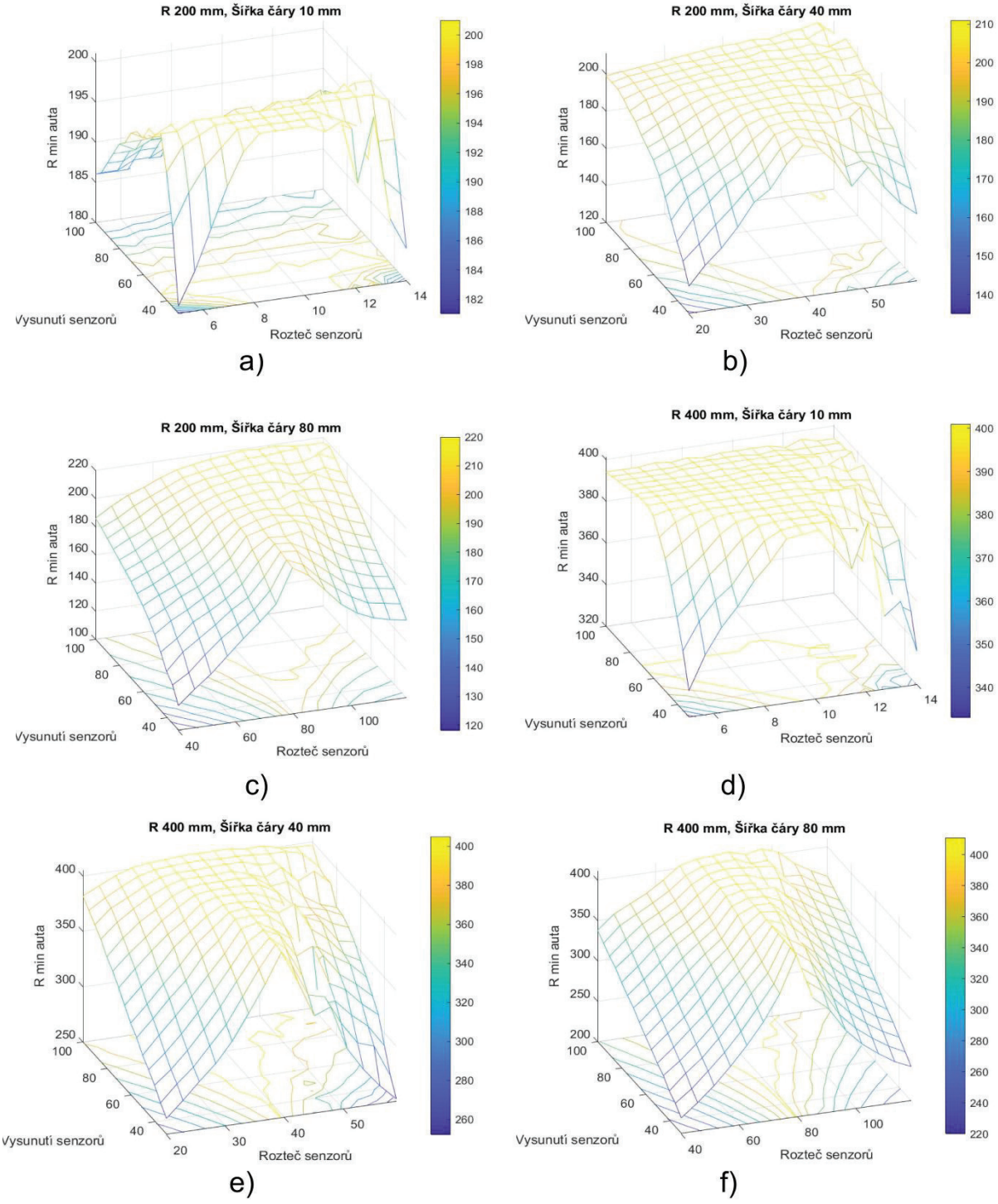
C Grafy jednotlivých testování SGA

D Grafy jednotlivých testování MGA

Obsah CD

- Text_diplomove_prace
  - Diplomova_prace_2019_Frantisek_Manlig.pdf
  - Diplomova_prace_2019_Frantisek_Manlig.docx
  - Kopie_zadani_diplomove_prace_2019_Frantisek_Manlig.pdf
- FW_vozicek-arduino
  - FW_vozicek-arduino
- Geneticke_algoritmy-matlab
  - TestSGA
  - TestMGA
  - Další potřebné podprogramy
- Simulace_jizdy_vozicku-matlab
  - START_jizda_po_15_dvojoblouku_R400
  - Další potřebné podprogramy
- Video-jizda_vozicku_po_optimalizaci.mp4

A. Vliv rozteče a vysunutí senzorů a poloměru dráhy na poloměr logiky řízení 1 z 2



a) R 200 mm, šířka čáry 10 mm

b) R 200 mm, šířka čáry 40 mm

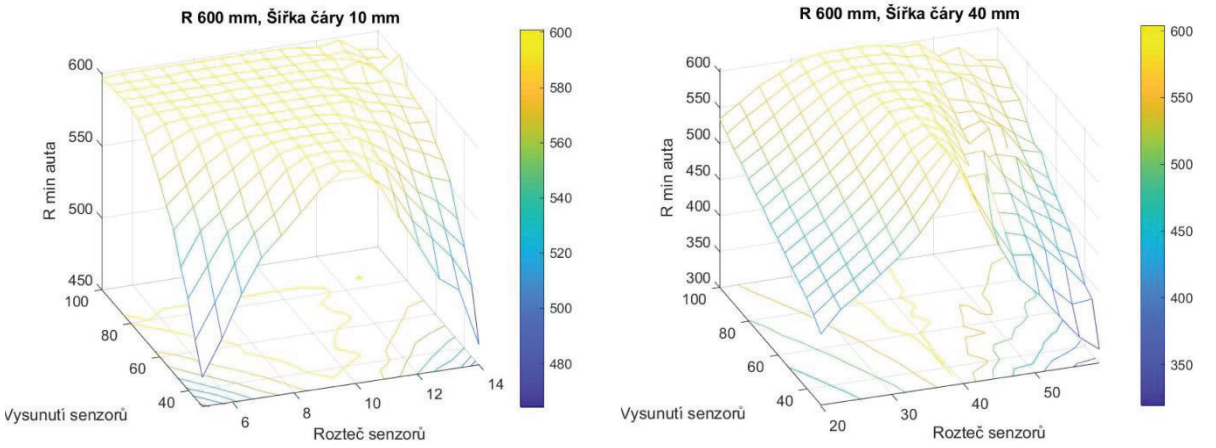
c) R 200 mm, šířka čáry 80 mm

d) R 400 mm, šířka čáry 10 mm

e) R 400 mm, šířka čáry 40 mm

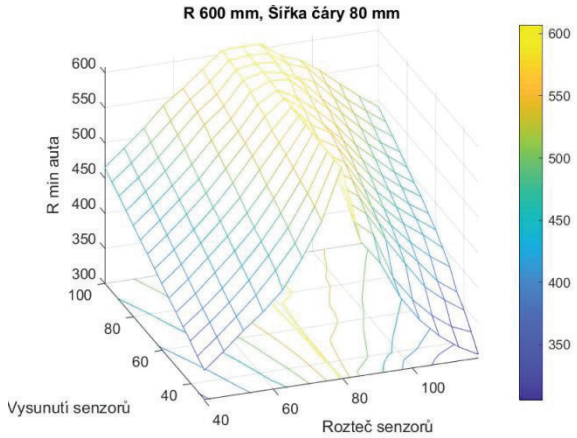
f) R 400 mm, šířka čáry 80 mm

B. Vliv rozteče a vysunutí senzorů a poloměru dráhy na poloměr logiky řízení 2 z 2

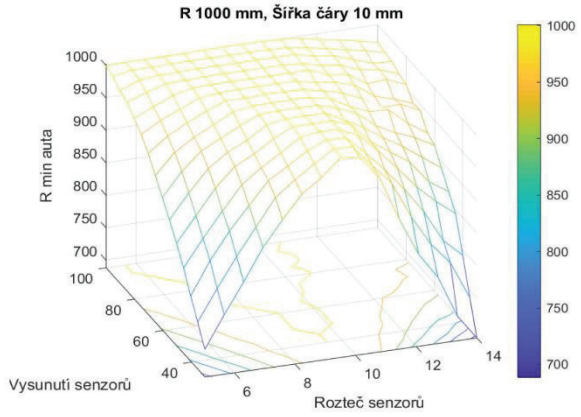


g)

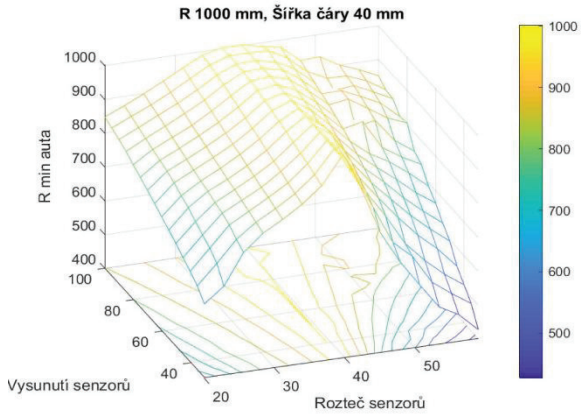
h)



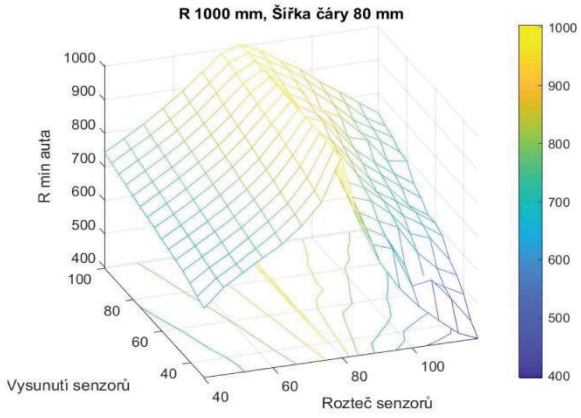
i)



j)



k)



l)

g) R 600 mm, šířka čáry 10 mm

h) R 600 mm, šířka čáry 40 mm

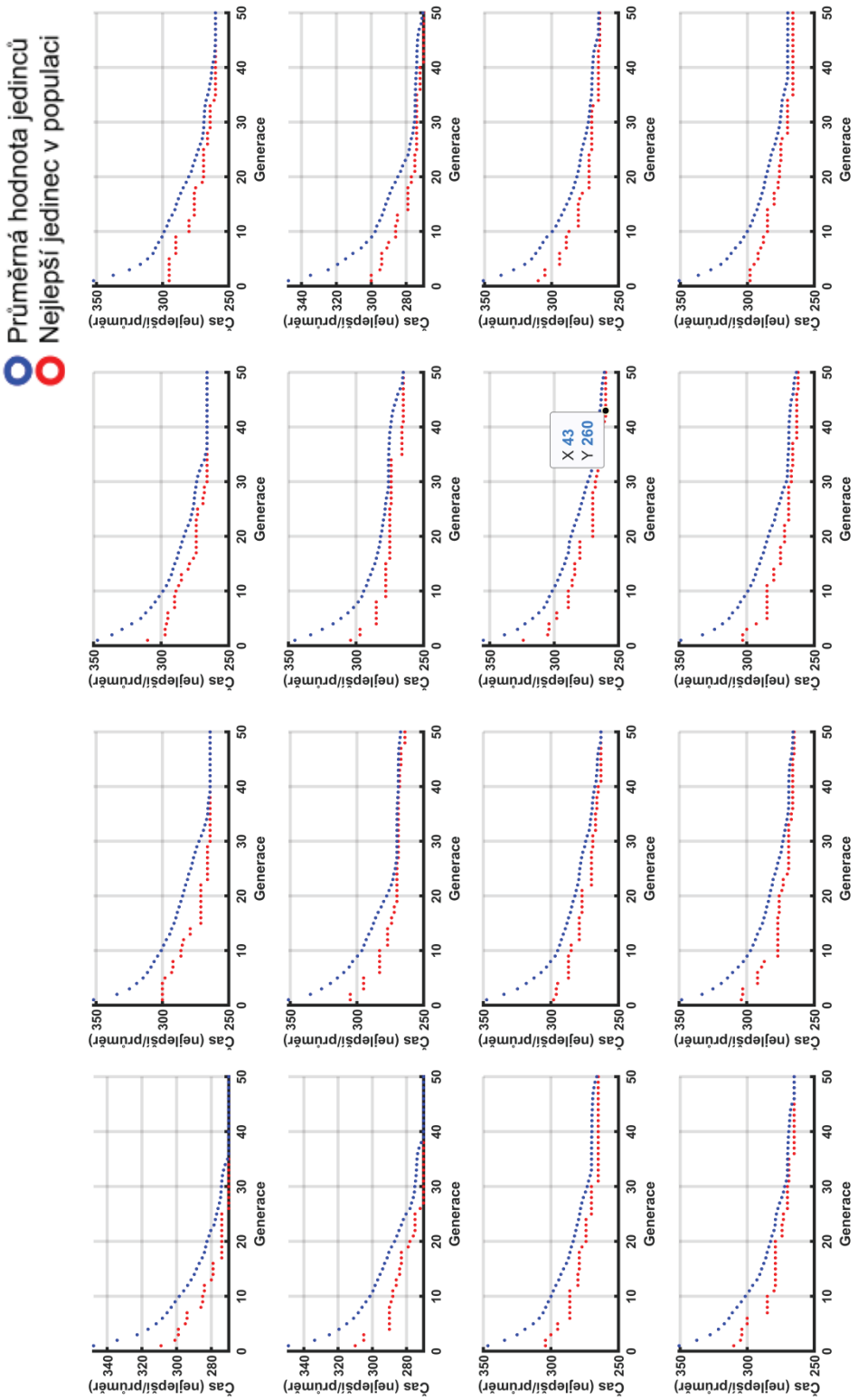
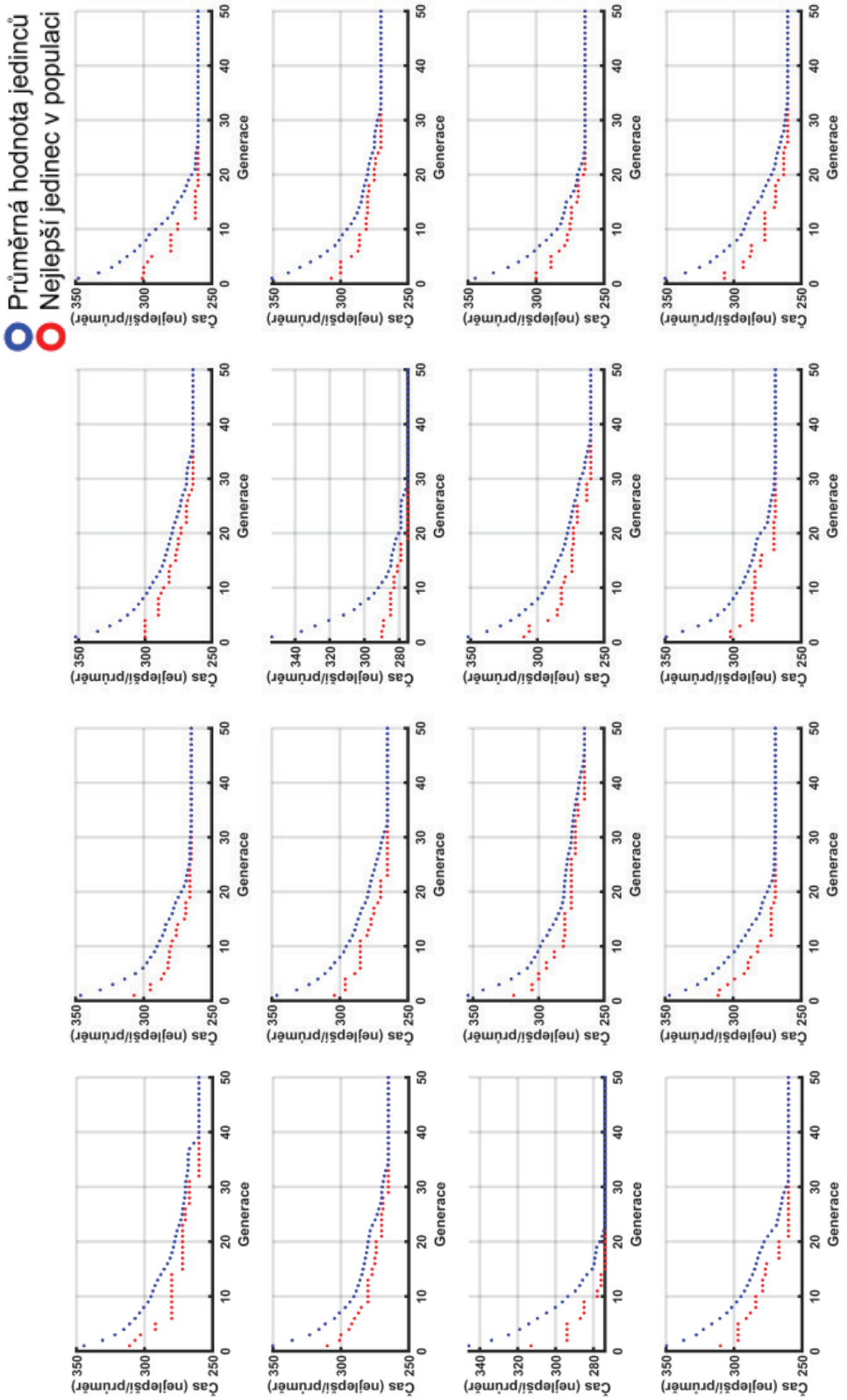
i) R 600 mm, šířka čáry 80 mm

j) R 1000 mm, šířka čáry 10 mm

k) R 1000 mm, šířka čáry 40 mm

l) R 1000 mm, šířka čáry 80 mm

C. Grafy jednotlivých testovaní SGA



D. Grafy jednotlivých testovaní MGA