**UNIVERZITA PARDUBICE**

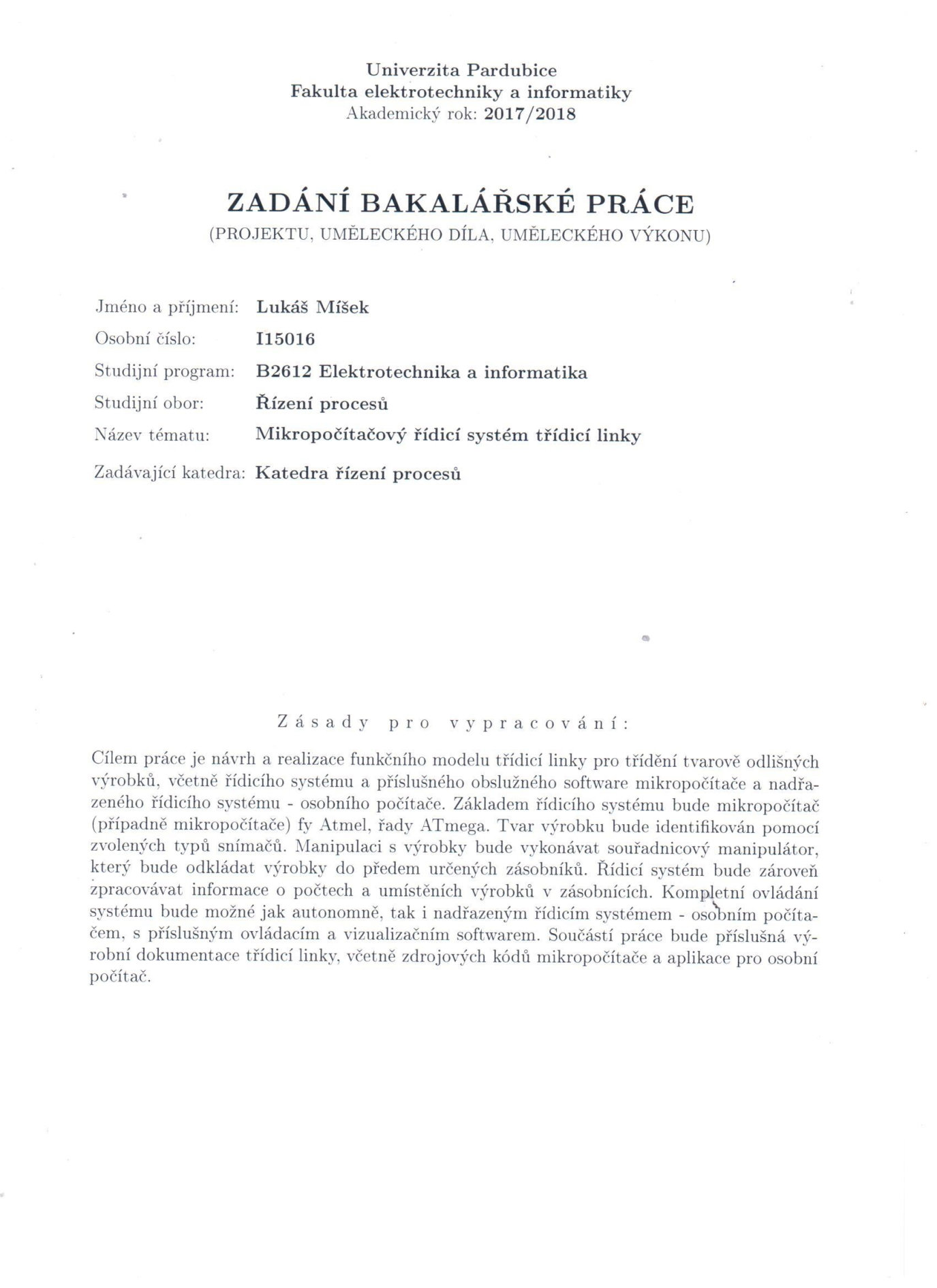
Fakulta elektrotechniky a informatiky

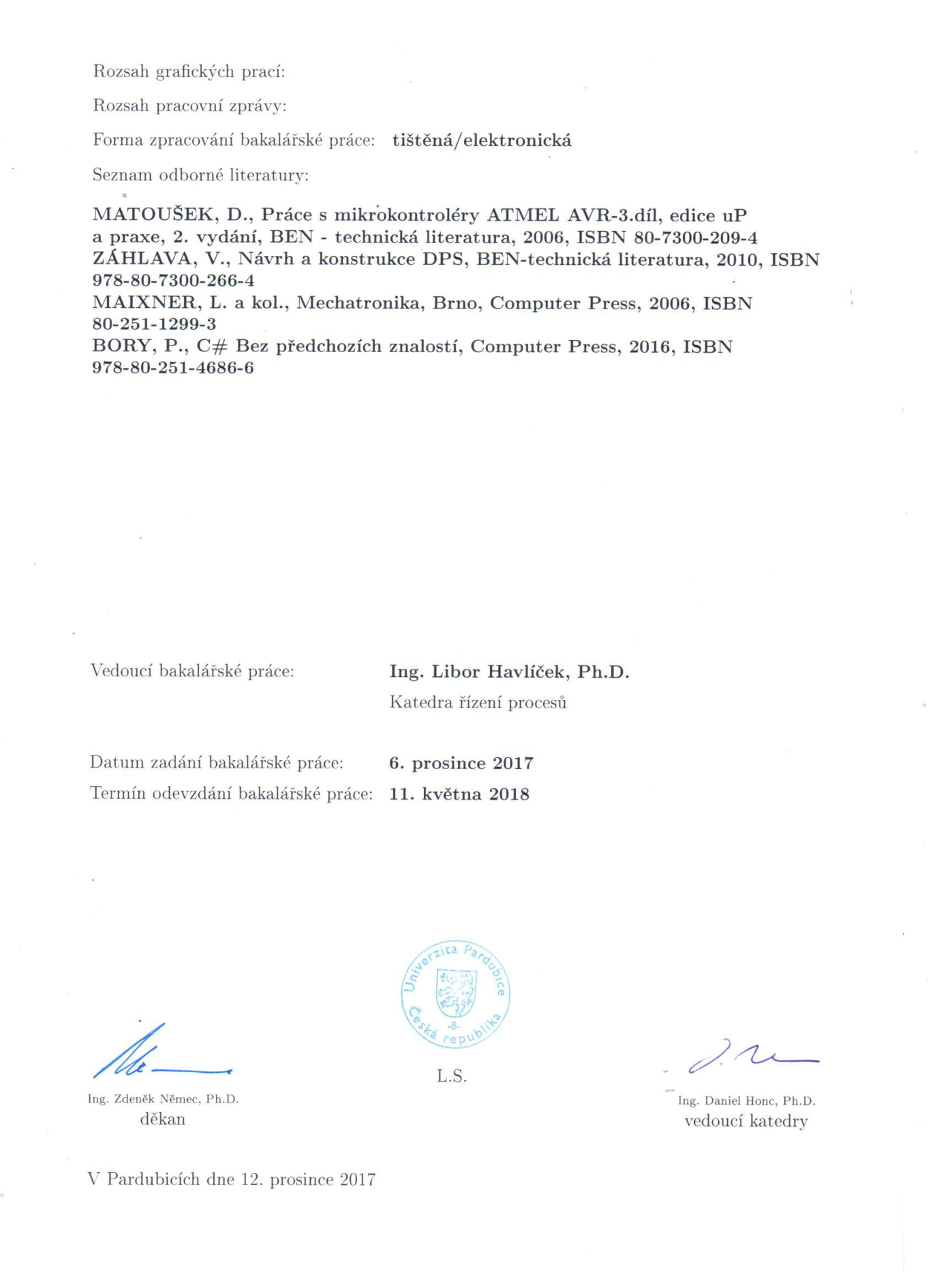
**Mikropočítačový řídicí systém třídící linky**

Lukáš Míšek

Bakalářská práce

2018



**

**Prohlášení**

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 29. 11. 2018

Lukáš Míšek

**ANOTACE**

*Cílem práce je návrh a realizace funkčního modelu třídicí linky pro třídění tvarově odlišných výrobků, včetně řídicího systému a příslušného obslužného software mikropočítače a nadřazeného řídicího systému – osobního počítače. Základem řídicího systému bude mikropočítač (případně mikropočítače) fy Atmel, řady ATmega. Kompletní ovládání systému bude možné jak autonomně, tak i nadřazeným řídicím systémem na osobním počítači, s příslušným ovládacím a vizualizačním softwarem. Součástí práce bude příslušná výrobní dokumentace třídicí linky, včetně zdrojových kódů mikropočítače a aplikace pro osobní počítač.*

**KLÍČOVÁ SLOVA**

*robotický manipulátor, mikropočítač, programování.*

**TITLE**

*Microcomputer control system of the sorting line*

**ANNOTATION**

*Goal of this work is to design and construct functional model of the sorting line to sort objects of the various shapes. Main building block of this system will be microcontroller (or microcontrollers) of company Atmel, series ATMEGA. Shape of the sorted object will be decided by group of chosen sensors. Complete control of the system will be both autonomous, or by superior system, the personal computer with corresponding elements and visual software.*

**KEYWORDS**

*microcontroller, programming, control system.*

[1 Rozbor zadání 11](#_Toc6787610)

[1.1 Rešerše literatury 11](#_Toc6787611)

[1.2 Návrh provedení 11](#_Toc6787612)

[1.3 Blokové schéma 13](#_Toc6787613)

[1.4 Historie automatizace 14](#_Toc6787614)

[1.5 Úvod do projektu 15](#_Toc6787615)

[2 Hardware 17](#_Toc6787616)

[2.1 Snímače 17](#_Toc6787617)

[2.1.1 Binární snímač polohy 17](#_Toc6787618)

[2.1.2 Clonící senzor rozměru 18](#_Toc6787619)

[2.2 Transportní zařízení 19](#_Toc6787620)

[2.2.1 Válečkové tratě 19](#_Toc6787621)

[2.2.2 Pásové dopravníky 19](#_Toc6787622)

[2.2.3 Řetězové dopravníky 19](#_Toc6787623)

[2.2.4 Karuselový dopravník 20](#_Toc6787624)

[2.3 Třídící zařízení 20](#_Toc6787625)

[2.3.1 Přesuvna 20](#_Toc6787626)

[2.3.2 Otočný stůl 21](#_Toc6787627)

[2.3.3 Výklopný shrnovač a tlačka 21](#_Toc6787628)

[2.3.4 Robotické rameno 22](#_Toc6787629)

[2.4 Mikrokontroler ATMEGA 23](#_Toc6787630)

[2.5 Programovací deska Arduino Uno 24](#_Toc6787631)

[2.6 Multiplexor a Demultiplexor 74HC4067 25](#_Toc6787632)

[2.7 Posuvný registr CD4021B 26](#_Toc6787633)

[2.8 Senzor tvaru 27](#_Toc6787634)

[2.9 Senzor tvaru s posuvnými registry 28](#_Toc6787635)

[2.10 LED Matice 8x8 KVM-20881CVB 29](#_Toc6787636)

[2.11 Krokový motor 31](#_Toc6787637)

[2.12 Servo motory SG90 a MSG90 32](#_Toc6787638)

[2.13 Řídící jednotka TB6560 33](#_Toc6787639)

[2.14 Testovací třídicí linka od firmy Merkur 34](#_Toc6787640)

[3 Konstrukce modelu třídící linky 35](#_Toc6787641)

[3.1 Konstrukce dopravníkového pásu 35](#_Toc6787642)

[3.2 Řídící obvod motoru GM37-3530 36](#_Toc6787643)

[3.3 Řídící obvod krokového motoru 28Byj-48 37](#_Toc6787644)

[3.4 Řídící obvod pro servo motory SG90 a MSG90 38](#_Toc6787645)

[3.5 Senzor tvaru 4x4 39](#_Toc6787646)

[3.6 Ovládací deska 40](#_Toc6787647)

[3.7 Tvorba vlastních mechanických součástek 42](#_Toc6787648)

[4 Programování mikrokontroleru 44](#_Toc6787649)

[4.1 Programování Mikrokontroleru pomocí Atmel studia 44](#_Toc6787650)

[4.2 Programování mikrokontroleru pomoci Arduino IDE 46](#_Toc6787651)

[4.3 Použité algoritmy mikrokontroleru 47](#_Toc6787652)

[4.3.1 Detekce stisku tlačítka mikrokontrolerem 47](#_Toc6787653)

[4.3.2 Rozšíření počtu vstupních pinů pomocí posuvného registru 47](#_Toc6787654)

[4.3.3 Rozsvícení LED, nebo LED matice 47](#_Toc6787655)

[4.3.4 Ovládání krokového motoru 28byj-48 48](#_Toc6787656)

[4.3.5 Ovládání servo motoru SG90 48](#_Toc6787657)

[4.3.6 Ovládání GM37-3530 motoru 48](#_Toc6787658)

[4.3.7 Řízení toku programu 49](#_Toc6787659)

[4.4 Program pro mikrokontroler 50](#_Toc6787660)

[4.4.1 Hlavní program 50](#_Toc6787661)

[4.4.2 Inicializační blok Setup a blok SerialEvent 51](#_Toc6787662)

[4.4.3 Blok Loop 51](#_Toc6787663)

[4.4.4 Blok Manual mode 52](#_Toc6787664)

[4.4.5 Obslužné funkce akčních členů 53](#_Toc6787665)

[4.4.6 Detekce tvaru výrobku pomocí funkce SHAP 54](#_Toc6787666)

[4.4.7 Určení kombinace uložiště a tvaru pomocí funkce WHERE 55](#_Toc6787667)

[4.4.8 Automatický režim 56](#_Toc6787668)

[4.4.9 Funkce Default a bloky MoveDest 57](#_Toc6787669)

[4.5 Ukázka běhu programu 58](#_Toc6787670)

[5 Nadřazený systém 59](#_Toc6787671)

[5.1 Aplikace na počítači uživatele: 59](#_Toc6787672)

[5.2 Main window 59](#_Toc6787673)

[5.3 Menu Configuraiton 60](#_Toc6787674)

[5.4 Menu Data storage 61](#_Toc6787675)

[5.5 Menu Manual mode 62](#_Toc6787676)

[5.6 Menu Automat mode 63](#_Toc6787677)

[6 Seznam literatury: 64](#_Toc6787678)

[Obrázek 1: Blokové schéma systému 13](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787679)

[Obrázek 2: Kolo na hřídelí9 14](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787680)

[Obrázek 3: Páka10 14](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787681)

[Obrázek 4: Výroba pečiva11 15](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787682)

[Obrázek 5: Výroba automobilu12 16](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787683)

[Obrázek 6: Výroba integrovaných obvodů13 16](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787684)

[Obrázek 7: Světelná závora, binární snímač polohy14 17](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787685)

[Obrázek 8: Clonící senzor rozměru15 18](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787686)

[Obrázek 9: Princip clonícího senzoru rozěmru16 18](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787687)

[Obrázek 10: Válečková trať17 19](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787688)

[Obrázek 11: Pásový dopravník18 19](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787689)

[Obrázek 12: Řetězový dopravník19 19](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787690)

[Obrázek 13: Karuselový dopravník od firmy LOGSYS20 20](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787691)

[Obrázek 14: Přesuvna od firmy LOGSYS21 20](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787692)

[Obrázek 15: Otočný stůl od firmy FEIFER22 21](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787693)

[Obrázek 16: Výklopný shrnovač23 21](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787694)

[Obrázek 17: Robotický manipulátor od firmy Buddy Toys BCR 1024 22](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787695)

[Obrázek 18: Robotický manipulátor bez pohyblivých kloubů26 22](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787696)

[Obrázek 19: Mikrokontroler ATMEGA27 23](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787697)

[Obrázek 20: Vývojová deska Arduino Uno29 24](#_Toc6787698)

[Obrázek 21: Princip multiplexoru30 25](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787699)

[Obrázek 22: Integrovaný obvod 74HCT406732 25](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787700)

[Obrázek 23: Integrovaný obvod CD4014B34 26](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787701)

[Obrázek 25: Logika senzoru 27](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787702)

[Obrázek 26: Vnitřní zapojení LED Matice KVM-20881 CVB36 29](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787703)

[Obrázek 27: LED Matice KVM-20881 CVB37 29](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787704)

[Obrázek 28: Zapojení LED Matice k mikrokontroleru 30](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787705)

[Obrázek 29: Konstrukce krokového motoru39 31](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787706)

[Obrázek 30: Krokový motor 28byj-4840 31](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787707)

[Obrázek 31: Servo motor42 32](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787708)

[Obrázek 32: Servo motor43 32](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787709)

[Obrázek 33: Vnitřní zapojení krokového motoru45 33](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787710)

[Obrázek 34: Řídící jednotka TB656046 33](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787711)

[Obrázek 35: Motor GM37-353048 34](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787712)

[Obrázek 36: Dopravníkový pás firmy Merkur47 34](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787713)

[Obrázek 37: Tranzistor IRLZ44N49 36](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787714)

[Obrázek 38: GM37-3530 řídící obvod 36](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787715)

[Obrázek 39: Řídící jednotka pro 28Byj-48 37](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787716)

[Obrázek 40: Zapojení servo motorů 38](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787717)

[Obrázek 41: Senzor tvaru 4x4 39](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787718)

[Obrázek 42: Schéma ovládací desky 41](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787719)

[Obrázek 43: Návrh hlavice senzoru tvaru v programu DesignSpark Mechanical 4.0 42](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787720)

[Obrázek 44: Návrh výrobků v programu DesignSpark Mechanical 4.0 42](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787721)

[Obrázek 45: Návrh náhradních merkur součástek v programu DesignSpark Mechanical 43](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787722)

[Obrázek 46: Vývojové prostředí Atmel studio 7.0 44](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787723)

[Obrázek 47: AND-Load aplikace 45](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787724)

[Obrázek 48: Ukázka vývojového prostředí Arduino Uno 46](#_Toc6787725)

[Obrázek 49: Řadič ULN200358 48](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787726)

[Obrázek 50: Hlavní program 50](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787727)

[Obrázek 51: Blok Loop 51](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787728)

[Obrázek 52: Blok manual mode 52](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787729)

[Obrázek 53: Funkce S1UP 53](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787730)

[Obrázek 54: Funkce lineForward 53](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787731)

[Obrázek 55: Funkce S1DO 53](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787732)

[Obrázek 56: Funkce STUP 53](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787733)

[Obrázek 57: Funkce STDO 53](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787734)

[Obrázek 58: Detekce tvaru výrobku 54](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787735)

[Obrázek 59: Funkce WHERE 55](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787736)

[Obrázek 60: Automatický režim 56](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787737)

[Obrázek 61: Bloky MoveDest 57](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787738)

[Obrázek 62: Ukázka běhu programu 58](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787739)

[Obrázek 63: Main window 59](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787740)

[Obrázek 64: Menu Configuration 60](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787741)

[Obrázek 65: Menu Data storage 61](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787742)

[Obrázek 66: Menu manual mode 62](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787743)

[Obrázek 67: Menu Automatic mode 63](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787744)

**Seznam zkratek:**

Ucc ... Všechny řídící obvody, servo motory a krokový motor je napájen napětím 5V. Napájení těchto obvodů je označeno Ucc.  
UccLine ... Motor GM37-3530 dopravníkového pásu má jako jedinný napájecí napětí 12V.  
GND ... Označení pro uzemnění. Uzemnění spojuje všechny elektrické vodivé spojení se zemí.  
Latch ... Řídící signál pro použitý posuvný registr CD4021b. Signálem je řízena vnitřní pamět posuvného registru.  
Clock ... Hodinový signál pro použitý posuvný registr CD4021b. Signálem je řízeno čtení výstupu registru.  
Data ... Výstupní datový signál posuvného registru. Signál nese informace o vstupních pinech registru.

* 1. Rozbor zadání

Úkolem práce je vyzkoušet možnosti a využití optimalizace pro řešení vybraného logistického problému. Logistickým problémem je možné rozumět proces, kde je nutné přesunout věci z místa A do místa B, případně libovolného množství dalších míst.

* 1. Podobné problémy

Podobným problémem se zabýval Aleš Horký v práci „Tvorba rozvrhů pomocí genetických algoritmů“. Úkolem této práce bylo vytvořit program, který umožní uživateli vytvořit rozvrhy pro školu. Tyto rozvrhy jsou tvořeny s ohledem na mnohé požadavky. Například v jedné třídě nemůže být více vyučovaných hodin současně, jeden učitel v danou chvíli může učit pouze jednu třídu, žáci nemohou mít více než 1 hodinu současně a podobně.

Důvodem ke vzniku tohoto problému je komplexnost řešení. Pro požadavky, které jsou omezeny pouze několika podmínkami je řešení poměrně triviální. Například škola se 4 učiteli, 40 žáky, 1 místností a 4 vyučovacími předměty, kde každý učitel vyučuje právě jeden předmět po dobu jedné hodiny. Zde jsou žáci rozmístěni do jedné třídy a postupně je učí jeden učitel. Zde existuje mnoho správných řešení.

Problém nastane ve chvíli, kdy učitelé potřebují vyučovat hodiny v konkrétních třídách a mohou učit pouze v omezeném časovém úseku. Potom nelze předměty rozmístit libovolně jako v předchozím příkladu. Nemůžeme použít třídu, ve které je vyučován konkrétní předmět k výuce jiných předmětů. Nelze zároveň dát přednost méně prioritním předmětům, které je možné učit v pozdějších hodinách.

Řešení je stále poměrně jednoduché. Co se ale stane, když máme školu o 15 učitelích, 20 třídách, kde má každá 20-30 žáků a k dispozici bude pouze 12 učeben? Zároveň zde budou požadavky na předměty, které mohou být vyučovány pouze v konkrétních třídách. Například hodiny chemie, tělocvik, nebo hudební výchova. Aleš Horký tuto problematiku podrobně popisuje v jeho práci.

* 1. Návrh řešení

Řešením by tedy měl být program, který je schopen vyřešit daný problém. Poskytnuté řešení by mělo vyhovět požadavkům zadaným zadavatelem. Dále by mělo poskytnou řešení v reálném čase a umožnit uživateli více než jednu možnost, případně předložené řešení modifikovat.

Tyto požadavky by měly být strukturovány do formátu, který je možné v elektronické formě uložit do souboru. Aplikace na základě tohoto programu stanoví požadavky uživatele. Tyto soubory by měly být navrženy takovým způsobem, že je možné je vytvořit, nebo naplnit daty běžným uživatelem, nebo analytikem.

Aplikace na základě požadavků vygeneruje výslednou kombinaci. Uživatel musí mít k dispozici ovládací prvky, které mu umožní vybrat z výsledných řešení. V případě, že žádné řešení neexistuje, tak bude vrácena taková kombinace, která obsahuje nejmenší množství konfliktů. Uživatel by potom mohl řešit problém nestandardním řešením.

Výstup aplikace bude řešen pomocí csv, nebo excel souboru. Tento soubor je potom dále možné použít v jiných aplikacích, které slouží k zadání samotného rozvrhu. Případně může být výstup ve formátu jpg, bmp, nebo pdf, které je možné vytisknou. Tato možnost by mohla být požadována uživateli, kteří nepoužívají žádné zvláštní systémy.

V neposlední řadě je důležité, aby byla aplikace prezentovatelné široké veřejnosti. Aplikace by neměla být příliš složitá, aby ji mohl ovládat pouze specializovaný pracovník, nebo být tak specializovaná, že ji je možné použít k řešení jenom několika málo problémů. Aplikace tedy bude rozhodně obsahovat grafické rozhraní a bude realizována v programovacím jazyce C#.

* 1. Blokové schéma

Vstupem aplikace budou soubory s požadavky ve formátu txt, csv, nebo xls. Soubory představují jednotlivé kapacity, které se musí sejít. Mějme tedy Kamion, který představuje nákladový prostor. Každý kamion má místo na 32 palet. Potom máme palety, které jsou dále předměty. Třetím souborem budou skladníci. Skladníci neboli pracovníci mohou nakládat do kamiónů pouze určité kamiony. Tyto tři soubory představují náš vstup.

Dále je aplikace samotná. Ta se skládá z vyhodnocení vstupu, samotného algoritmu a výstupu. Výstup v tomhle případě je kombinace skladníků, kamiónů a palet. Program pomocí algoritmu bude generovat velké množství výstupu. Tyto výstupy musí být ohodnoceny, aby bylo možné rozlišit, které jsou nevyhovující a které vyhovují. Program si bude nejlepších 20% výstupů pamatovat. Tohle bude dalším vstupem do algoritmu.

Budou zde tedy paměťové blogy. Blog k vyhodnocení výstupu a blok k uchování výstupu, který bude prezentován uživateli. V neposlední řadě bude mít program blok s kontrolním nastavením. Zde budou vstupy, které nepřímo ovlivňují výpočet. Například počet generací, které budou generovány, velikost populací, procento generací, které je označeno za úspěšné a uživatelské nastavení, jako formát výstupu.

* 1. Úvod do problematiky

Problematika hledání vhodné kombinace řešení, ať už je to rozložení palet do kamionu, tvorba rozvrhu pro školu, nebo dosazení doktorů na služby, je v dnešní době řešena specializovanými pracovníky. Tito pracovníci používají rozličné metody k nalezení těchto řešení.

Jedním z příkladu řešení je takzvaná hrubá síla. To znamená, že zkouším náhodné kombinace a vyhodnocuji, jestli jsou tyto kombinace vyhovující. Tento postup přesto, že funguje, tak není optimální, protože čas pro nalezení vhodné kombinace je exponenciálně zvyšuje s rostoucími vstupy.

Tyto vstupy jsou konkrétní požadavky zadavatelů. Aleš Horký v jeho práci zmiňuje dva druhy omezení. Takzvané Tvrdé podmínky a Měkké podmínky. Tvrdé podmínky musí být splněny vždy. Měkké podmínky nemusí být nutně splněny. Nicméně jejich splnění pozitivně ovlivní ohodnocení kombinace pomocí hodnotící funkce.

Mezi tvrdé podmínky platí základní podmínky, jako na jedno paletové místo musí být dosazena pouze jedna paleta, kamion nemůže obsahovat více palet, než je jeho kapacita a skladník nemůže nakládat více než jeden kamion v daný okamžik. Splněním těchto požadavků dosáhneme správného řešení. Toto řešení může být jediné, zároveň jich může existovat více než jedno a zároveň nemusí být ani jedno.

Měkké podmínky jsou takové, které nemusí být splněny. Nicméně jejich splnění pozitivně ovlivní ohodnocení výsledné kombinace. Může to být například zvláštní požadavek skladníka, který chce nakládat pouze některé kamiony, palety, které mohou být naloženy pouze do některých kamiónů. Některé kamiony mohou být prioritní a je nutné je naložit dříve a podobně.

Vyhodnocením těchto vstupů získáme kombinaci, která může být správným řešením. Pokud kombinace vyhoví všem tvrdým podmínkám, tak se jedná o správné řešení. Pokud nevyhoví, tak se jedná o nesprávné řešení. Zároveň je kombinace pomocí ohodnocující funkce hodnocena a je jí přiřazeno takzvané fitness skóre. Toto fitness skóre udává, jak vhodné je řešení. Vyšší fitness skóre představuje vhodnější řešení.

Kombinaci správných řešení získáme náhodným vygenerování. Po vyhodnocení kombinace je možné hledání dalších kombinací ukončit, nebo pokračovat. V případě pokračování je další generace nových kombinací vytvořena s ohledem na ty nejúspěšnější kombinace. Zároveň jsou kombinace náhodným způsobem nepatrně pozměněny. Tato změna za účelem nezaseknutí se v lokálním extrému.

1. Genetické algoritmy a programování

V této kapitole jsou popsány genetické algoritmy, jejich návrh, programování a princip. Tato kapitola se opírá o již vydané publikace. Zejména o publikaci s názvem Genetické algoritmy a genetické programování od Josefa Hynka.

Tato kniha slouží jako dobrý nástroj pro úvod do problematiky a nastiňuje jejich aplikaci. V knize jsou podrobně popsány postupy a principy, které se používají při tvorbě genetických algoritmů.

V této kapitole jsou popsány pouze ty postupy, které byly použity v realizaci algoritmu. Obsáhlejší popis genetických algoritmů, kódování a operací je popsán v knize Genetické programování a genetické algoritmy, ze které tato práce vychází.

* 1. Princip genetických algoritmů

Libovolnou reálnou úlohu je možné matematickým, nebo symbolickým způsobem popsat. Řekněme, že obchodník potřebuje navštívit 3 místa. Řešením této úlohy je posloupnost 3 měst. Tyto města označíme jako A, B, C. Řešení potom může vypadat následovně:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | A | B | C | |  |
|  |  |

Tato posloupnost měst je vhodným řešením a zároveň je nejkratší vzdáleností mezi těmito 3 body. Tuto posloupnost nazveme jedincem, kde každý jeho prvek bude genem. Zde jsou všechna možná řešení pro tuto úlohu:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | C |  | A | C | B |
|  |  |  |  |  |  |  |
| B | C | A |  | B | A | C |
|  |  |  |  |  |  |  |
| C | A | B |  | C | B | A |

Problém nastane ve chvíli, když počet měst se zvýší. Potom nemusí být zjevné, co je nejkratší vzdáleností. V tom případě vygenerujeme skupinu jedinců o konečném počtu, kde každý jedinec bude obsahovat náhodné geny. Tuto skupinu nazveme generací.

Pro nalezení nejvhodnějšího řešení budeme kombinovat jedince v aktuální generaci. Z generace 1 vybereme 2 jedince, kteří budou rodiči. Tyto 2 rodiče zkombinujeme a vytvoříme tím nového jedince. Tento nový jedinec bude potomkem součástí další generace. Tedy generace 2.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Rodič 1** | A | B | C | D | E | |  |  |  |  |  |  | | **Rodič 2** | D | A | C | B | E | | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  | | **Potomek** | A | B | C | B | E | |  |  |  |  |  |  | |
|  |  |

Zde jsme zkombinovali 2 jedince (rodiče) z generace 1 a vytvořili jsme z nich 1 jedince (potomka) z generace 2. Podobným způsobem naplníme celou generaci 2 kombinováním náhodných rodičů. Tento postup se opakuje pro daný počet generací.

Tuto kombinaci jedinců nazvěme křížením. Výběr dvou rodičů nazvěme selekcí. Celkovou vzdálenost mezi městy zobecníme a nezveme ji fitness funkcí. Celý algoritmus by potom fungoval tak, že by každý další generace byla tvořena potomky, kteří byli vybrání z rodičů podle fitness funkce. Právě díky fitness funkci by se do další generace dostaly geny pouze těch nejlepších jedinců.

* 1. Fitness funkce

Každý jedinec je reprezentací řešení. Jak dobré toto řešení je, popisuje fitness funkce neboli ohodnocení. Fitness funkce může být součet vzdáleností mezi městy, potom by menší fitness funkce byla lepším a vhodnějším řešením neboli silnějším jedincem.

Jiným příkladem fitness funkce může být počet konfliktů v rozvrhu. Jedinec by byl posloupností předmětů vyučovaných ve škole a každý konflikt by se započítal do fitness funkce. Fitness funkce také může být získanými body žákem. Každý žák by odpověděl na testové otázky jiným způsobem a získané body by byla právě fitness funkce.

V některých případech by vyšší fitness funkce mohla být negativní a jindy by mohla být pozitivní. Na základě této fitness funkce by potom byli vybrání ti nejlepší jedinci a z nich by byli vytvořeni jedinci další generace. Tento proces se nazývá selekce a může vypadat následovně.

Zde máme 2 generace. Obě generace mají celkem 12 jedinců. Číslice reprezentují fitness funkce každého jedince. V tomto případě vyšší fitness značí silnějšího jedince.

* Žlutě jsou označení nejsilnější jedinci z generaci 1. Tito jedinci mají největší šanci na vytvoření potomstva.
* Modře byl označen jedinec, který je silnější, než všichni z předchozí generace. Tento jedinec má nyní největší pravděpodobnost, aby vytvořil potomstvo do další generace.
* Červeně byl označen jedinec, který „se nepovedl“. Tento jedinec během selekce „odpadne“.

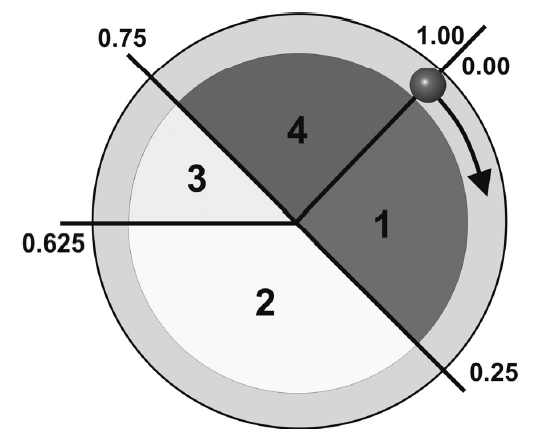
|  |  |
| --- | --- |
| **Generace 1:** | **Generace 2:** |
| |  |  |  | | --- | --- | --- | | 93 | 38 | 1 | | 40 | 5 | 65 | | 33 | 1 | 89 | | 32 | 34 | 57 | | |  |  |  | | --- | --- | --- | | 63 | 26 | 29 | | 58 | 0 | 67 | | 95 | 87 | 92 | | 21 | 19 | 63 | |

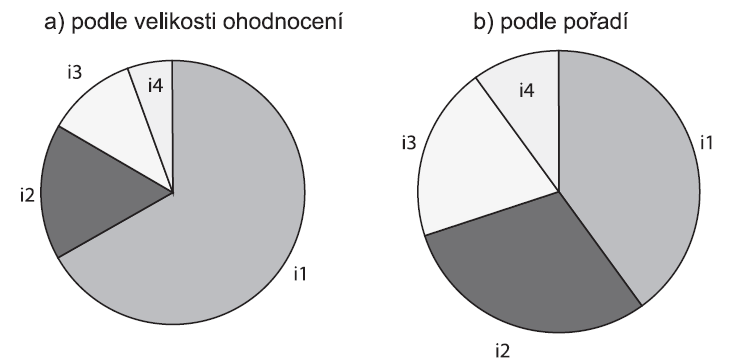
* 1. Selekce

Selekce je způsob, kterým jsou vybírání jedinci do další generace. Většinou se používá fitness funkce, podle které jsou označeni ti nejvhodnější. Vzhledem k tomu, že je algoritmus do určité částí řízen náhodou, tak nelze ale vyloučit, že se novou generaci budou vytvářet slabší jedinci.

Tvorba nové generace ze slabších jedinců není ve skutečnosti negativním jevem. Silný jedinec může obsahovat hodně pozitivních genů, díky kterým má vysokou fitness funkci, ale nemusí obsahovat pozitivní gen slabšího jedince. Pokud by byl slabý jedinec vyřazen z procesu tvorby nové generace, tak by se tato vlastnost ztratila.

Ve skutečnosti by to znamenalo, že by jedinec uvízl v lokálním extrému. Proto existují způsoby, které dávají šanci i těm slabším jedincům. Různé způsoby selekce mají různé výhody a nevýhody.

Existuje například selekce pomocí rulety. Zde je fitness funkce přímo úměrná ploše na ruletě. V tomto případě je velmi vysoká pravděpodobnost, že jedinec s vysokou fitness funkcí (2) se neztratí a jedinec s nízkou fitness funkcí (3) se ztratí. Nevýhodou je, že slabý jedinec (3) může obsahovat důležitý gen, který nemusí obsahovat jiný jedinec z populace.

Ruletu je možné pozměnit, aby plocha nebyla přímo úměrná velikosti fitness funkce (ohodnocení), ale pořadí. Potom bude ruleta vypadat takto. Zde má slabší jedinec vyšší šanci na podílení se tvorby nové generace. Zároveň jsou silnější jedinci stále zvýhodněni díky jejich vlastnostem.

Dalším způsobem selekce je selekce pomocí turnaje. Selekce pomocí turnaje funguje tak, že jsou náhodně vybráni 2 dvojice jedinců a mezi z těchto dvojic jsou vybráni 2 vítězové, kteří se budou podílet na tvorbě nové generace. Pokud tedy vybereme dvojici 2 slabších a 2 silnějších jedinců, tak rodiči se stane 1 silný a 1 slabý jedinec. Nevýhodou je, že některý silný jedinec nemusí být vybrán do turnaje právě kvůli náhodě.

* 1. Křížení

Křížení je operace, která vezme 2 jedince a z jejich genů vytvoří nového jedince. Tento nový jedinec bude kombinací jeho 2 rodičů z předešlé generace. Křížení je možné realizovat mnoha způsoby. Toto je hlavní způsob, kterým jsou tvoření noví jedinci. Po křížení se může výsledný potomek podstatně lišit od jeho rodičů.

Může být ke křížení použita vždy konkrétní část jedinců. Například první polovina z prvního jedince a druhá polovina z druhého jedince. Tyto poloviny mohou být vybírány náhodně. Je možné každý gen vybírat náhodným způsobem z jedinců.

Zároveň je důležité, aby nebylo křížení zvoleno nerozumně. Například by neměl být křížen jedinec v poměru 1:9, kde rodič 1 by předal potomku 10% svých genů a rodič 2 by předal 90% svých genů. Z by mohlo pravidelně docházet ke ztrátě genů.

* 1. Mutace

Mutace je operace, kdy se jeden nebo více genů náhodně změní. Touto náhodnou změnou může vzniknout gen, který dříve neexistoval. Mutace je způsob, kterým je algoritmus chráněn od lokálního extrému.

K mutaci na rozdíl od křížení by mělo docházet velmi zřídka. Pokud by byla pravděpodobnost mutace příliš vysoká, tak by byl algoritmus příliš náhodný a nikdy by nemusel dojít do konečného řešení.

1. Návrh algoritmu
   1. Úloha, problematik
   2. Rozbor zadání
      1. Vnitřní jedinec
      2. Vnější jedinec
      3. Křížení
      4. Mutace
   3. Predpoklady
2. Řešení vlastní práce
   1. Krok Výpočet nejkratší vzdálenosti
      1. Matice nejkratších vzdalenosti
      2. Pythagorova veta
      3. Realna vzdalenost vs vzdusna cara
   2. Krok Programovani vnitřního jedince
      1. Vhodna reprezentace a populace
      2. Zlepseni populace
      3. Fitness jedince
      4. Zobrazeni jedince
   3. Krok Programovani Vnejsiho jedince
      1. List<string> - Orders | InvidualOrders | PopulationOrder
      2. SuperInvidual
   4. Pomocne tridy
      1. Functions.showXXX
      2. InputOutput
      3. Cities
      4. Controller
3. Statistika
   1. Stare reseni full (Ok výsledky)
   2. Stare reseni bez opravy (Spatny výsledky)
   3. Nove reseni full (Top výsledky)
   4. Nove reseni vnejsi jedinec 50+ vnitrni 20-
   5. Nove reseni vnejsi jedince 20- vnejsi 50+
   6. Pouze vnitrni jedinec
4. Zaver
5. Seznam literatury:
6. BARCUCH, Libor. *TRANSPORTNÍ ZAŘÍZENÍ PRO DOPRAVU KUSOVÝCH PŘEDMĚTŮ* [online]. Brno, 2017 [cit. 2019-04-22]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/100909> Vysoké učení technické Brno. Vedoucí práce Ing. Jiří Malášek
7. MICHAEL, Konečný. *Návrh robotického ramene* [online]. Brno, 2016/207 [cit. 2019-04-22]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/100537> . Vedoucí práce Tomáš Marada
8. BENDA, Tomáš. *Konstrukce a řízení manipulátoru* [online]. Pardubice, 2015 [cit. 2019-04-22]. Dostupné z: <https://portal.upce.cz/StagPortletsJSR168/PagesDispatcherServlet?pp_destElement=%23ssSouboryStudentuDivId_717&pp_locale=cs&pp_reqType=render&pp_portlet=souboryStudentuPagesPortlet&pp_page=souboryStudentuDownloadPage&pp_nameSpace=G3593&soubidno=16098> . Diplomová. Vedoucí práce Havlíček Libor
9. PROKOP, Michal. *Model třídicí linky v systému Tecnomatix Process Simulate* [online]. Brno, 2017 [cit. 2019-04-22]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/102661>. Vedoucí práce Václav Kaczmarczyk
10. VICAN, Pavel. *Třídící linka* [online]. Česká lípa, 2017 [cit. 2019-04-22]. Dostupné z: http://www1.fs.cvut.cz/stretech/2017/sbornik\_2017/pdf/16.pdf. Maturitní práce. Střední průmyslová škola, Česká Lípa. Vedoucí práce Josef Bašta.¨
11. KOUMARIS, Nick. C# Application Arduino Communication. *Educ8s.tv* [online]. [cit. 2019-04-22]. Dostupné z: <https://educ8s.tv/c-application-arduino-communication/>
12. 375668457. C# Serial Port Communication Arduino. *Instructables* [online]. [cit. 2019-04-22]. Dostupné z: <https://www.instructables.com/id/C-Serial-Communication-With-Arduino/>
13. *Historický vývoj automatizace? Poznejte 12 zásadních dat* [online]. [cit. 2019-04-21]. Dostupné z: <https://factoryautomation.cz/historicky-vyvoj-automatizace-poznejte-12-zasadnich-dat/>
14. [online]. [cit. 2019-04-21]. Dostupné z: <http://nd04.jxs.cz/419/332/3ce8c88974_74039634_o2.jpg>
15. [online]. [cit. 2019-04-21]. Dostupné z: <http://fyzweb.cz/materialy/sily/paka/kolec1.gif>
16. [online]. [cit. 2019-04-21]. Dostupné z: <http://www.j4.cz/uploads/tx_odphotogallery/thumbs/e4c6364f0a0aa08538edd5177649f77b.jpg>
17. [online]. [cit. 2019-04-21]. Dostupné z: https://www.zenavaute.cz/wp-content/uploads/vyroba.jpg
18. [online]. [cit. 2019-04-21]. Dostupné z: <http://www.newsreck.com/wp-content/uploads/2019/03/Global-Wafer-And-Integrated-Circuits-Market.jpg>
19. Maixner, L.; et al. Optoelektrické senzory binární. *Mechatronika*, 1st ed.; Computer Press: Brno, 2006; Chapter 3.2.1.6., p 49.
20. Maixner, L.; et al. Clonicí senzor rozměrů. *Mechatronika*, 1st ed.; Computer Press: Brno, 2006; Chapter 3.2.1.7, p 52.
21. Maixner, L.; et al. Clonicí senzor rozměrů. *Mechatronika*, 1st ed.; Computer Press: Brno, 2006; Chapter 3.2.1.7, p 53.
22. [online]. [cit. 2019-04-21]. Dostupné z: <http://www.riko.cz/image.php?nid=12547&oid=3490858&width=640&height=480>
23. [online]. [cit. 2019-04-21]. Dostupné z: <http://www.zakhodonin.cz/src/img/produkty/dopravniky/dopravniky/dopravnik_1.jpg>
24. [online]. [cit. 2019-04-21]. Dostupné z: <http://www.tmt.cz/cz/CMS/foto/1902.jpg>
25. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.logsys.cz/files/th_8b1a7f5cccad127c10a8fd1f45573c4e.jpg>
26. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.logsys.cz/files/c042868367d48ab0b6a02dfccc0a580a.jpg>
27. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.feifer.cz/obrazky/1678acz.jpg>
28. [online]. [cit. 2019-04-21]. Dostupné z: <https://automatizace.hw.cz//files/styles/full/public/story_automat/11806/vyhybka_5.jpg?itok=2RpPhCv2>
29. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.exasoft.cz/buddy-toys-bcr-10-robotic-arm-kit_ie561785.jpg>
30. SKALICKÝ, Tomáš. *Návrh a realizace robotického manipulátoru* [online]. Pardubice, 2015 [cit. 2017-10-15]. Dostupné z: <https://portal.upce.cz/StagPortletsJSR168/PagesDispatcherServlet?pp_destElement=%23ssSouboryStudentuDivId_745&pp_locale=cs&pp_reqType=render&pp_portlet=souboryStudentuPagesPortlet&pp_page=souboryStudentuDownloadPage&pp_nameSpace=G3593&soubidno=16238> . Vedoucí práce Havlíček Libor
31. [online]. [cit. 2019-04-21]. Dostupné z: <https://t3.ftcdn.net/jpg/01/10/03/22/240_F_110032210_m8lKqMJawvwCrlj2sD1kjB6s8P22klXN.jpg>
32. [online]. [cit. 2019-04-21]. Dostupné z: https://www.souepl.cz/wp-content/ucitele/valecka/at\_mega\_32\_popis%20(Opraveno)\_soubory/image004.jpg
33. [online]. [cit. 2019-04-21]. Dostupné z: <http://ww1.microchip.com/downloads/en/devicedoc/doc2503.pdf>
34. [online]. [cit. 2019-04-21]. Dostupné z: https://i.ebayimg.com/00/s/NDIzWDU4Mw==/z/q10AAOSwBRVae8zd/$\_35.JPG?set\_id=8800005007
35. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.researchgate.net/profile/Shyam_Akashe/publication/257799438/figure/fig1/AS:341731565424653@1458486562832/MUX-graphical-symbol-a-truth-table-b.png>
36. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.researchgate.net/figure/MUX-graphical-symbol-a-truth-table-b_fig1_257799438>
37. [online]. [cit. 2019-04-21]. Dostupné z: <https://assets.nexperia.com/documents/data-sheet/74HC_HCT4067.pdf>
38. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.arduino.cc/en/tutorial/ShiftIn>
39. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.arduino.cc/en/uploads/Tutorial/shftin_cd4021_pins.png>
40. [online]. [cit. 2019-04-21]. Dostupné z: <https://create.arduino.cc/projecthub/SAnwandter1/programming-8x8-led-matrix-23475a>
41. [online]. [cit. 2019-04-21]. Dostupné z: <http://www.lucky-light.com/Dot%20Matrix%20LED%20Displays/5x7_Dot_Matrix/KWM-50571.pdf>
42. [online]. [cit. 2019-04-30]. Dostupné z: <https://create.arduino.cc/projecthub/SAnwandter1/programming-8x8-led-matrix-23475a>
43. [online]. [cit. 2019-04-30]. Dostupné z: <https://images-na.ssl-images-amazon.com/images/I/51653PgxqgL._SX425_.jpg>
44. [online]. [cit. 2019-04-21]. Dostupné z: <https://static3.tme.eu/products_pics/5/4/d/54dcf8dd74a04d30c55078ca113eb5ac/363694.jpg>
45. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.mylms.cz/text-krokovy-motor-princip/>
46. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.mylms.cz/obrazky/elektronika/krokovy-motor-14.png>
47. [online]. [cit. 2019-04-21]. Dostupné z: <https://i.ebayimg.com/images/g/BXoAAOSweBFckLAn/s-l300.jpg>
48. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.electricaleasy.com/2015/01/how-does-servo-motor-work.html>
49. [online]. [cit. 2019-04-21]. Dostupné z: <https://1.bp.blogspot.com/-WWEjibj3iX0/VK1LTqWI9NI/AAAAAAAABCs/FmiULNH04kg/s1600/servo%2Bmotor%2Bworking.png>
50. [online]. [cit. 2019-04-21]. Dostupné z: <https://i.ebayimg.com/images/g/DTQAAOSwFIRbxxlJ/s-l300.jpg>
51. [online]. [cit. 2019-04-21]. Dostupné z: <https://eluc.kr-olomoucky.cz/verejne/lekce/809>
52. [online]. [cit. 2019-04-21]. Dostupné z: <https://eluc.kr-olomoucky.cz/uploads/images/20238/obr.2.gif>
53. [online]. [cit. 2019-04-21]. Dostupné z: <https://i.ebayimg.com/images/g/AhEAAOSwUP5cLCBK/s-l640.jpg>
54. [online]. [cit. 2019-04-21]. Dostupné z: <https://eshop.merkurtoys.cz//data/product/178_352.jpg>
55. [online]. [cit. 2019-04-21]. Dostupné z: <https://images-na.ssl-images-amazon.com/images/I/61pnAVpZjuL._SY741_.jpg>
56. [online]. [cit. 2019-04-21]. Dostupné z: <https://pino-tech.eu/wp-content/uploads/2018/06/irlz44n.jpg>
57. [online]. [cit. 2019-04-21]. Dostupné z: <http://bildr.org/2012/03/rfp30n06le-arduino/>
58. [online]. [cit. 2019-04-21]. Dostupné z: <https://42bots.com/tutorials/bipolar-stepper-motor-control-with-arduino-and-an-h-bridge/>
59. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.instructables.com/id/Arduino-How-to-Control-Servo-Motor-With-Arduino/>
60. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.microchip.com/mplab/avr-support/atmel-studio-7>
61. [online]. [cit. 2019-04-21]. Dostupné z: <http://and-tech.pl/evb-4-3/>
62. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.arduino.cc/en/main/software>
63. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.arduino.cc/reference/en/language/functions/digital-io/digitalread/>
64. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.arduino.cc/reference/en/language/functions/digital-io/digitalwrite/>
65. [online]. [cit. 2019-04-21]. Dostupné z: <https://i.ebayimg.com/images/g/FAcAAOSwLfRbPg0F/s-l640.jpg>
66. [online]. [cit. 2019-04-30]. Dostupné z: <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTYE4F_YMtfyQwVEHx_Ei8wSmfKx58ZBC7T0jfNmKw3UwCVjEZjFg>
67. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.hackster.io/ianabcumming/arduino-simple-camera-slider-electronics-9e9d8b>
68. [online]. [cit. 2019-04-22]. Dostupné z: <https://www.arduino.cc/reference/en/language/structure/control-structure/while/>
69. [online]. [cit. 2019-04-22]. Dostupné z: <https://www.arduino.cc/reference/en/language/structure/control-structure/if/>
70. [online]. [cit. 2019-04-22]. Dostupné z: <https://www.arduino.cc/en/Tutorial/SerialEvent>
71. [online]. [cit. 2019-04-22]. Dostupné z: <https://www.arduino.cc/en/Reference/Loop?setlang=it>
72. [online]. [cit. 2019-04-22]. Dostupné z: <https://visualstudio.microsoft.com/downloads/?rr=https%3A%2F%2Fwww.google.cz%2F>