

# REAL AND SYNTHETIC BENCHMARKS FOR JOB SHOP SCHEDULING IN LOGISTIC WAREHOUSES

**Lukáš Povoda**

Master Degree Programme (II), FEEC BUT

E-mail: xpovod00@stud.feec.vutbr.cz

Supervised by: Jan Karasek

E-mail: karasekj@feec.vutbr.cz

**Abstract:** This paper introduces real and synthetic benchmarks for process planing and scheduling based on genetic programming driven by context-free grammar. Main purpose of this paper is to discuss the importance of complex testing of algorithm. Genetic programming algorithm has been subjected to tests based on real traffic and syntetic tests. Optimal parameters for different sizes of problems has been investigated based on test results.

**Keywords:** benchmarking, planning scheduling, warehouse optimization

## 1 ÚVOD

Tento článok sa zaoberá technikami testovania algoritmu určeného na riadenie a rozvrhovanie práce. Riadenie a rozvrhovanie práce v logistických skladoch je v praxi veľmi časovo náročná činnosť kvôli obrovskému počtu dát, ktoré je nutné pri návrhu zohľadniť. Problémy sa prejavujú najmä pri väčšom vytížení skladu, ktoré nastáva napr. v období vianočných sviatkov.

Pri vývoji zložitejších a komplexnejších algoritmov je nutné kontrolovať chovanie algoritmu pri každej jeho zmene, ako aj jeho výpočtovú náročnosť, či schopnosť iterácie k lepším výsledkom. Takúto kontrolu možno zabezpečiť prostredníctvom testov. Algoritmus genetického programovania určený pre riadenie a rozvrhovanie práce v sklade musí byť takisto podrobený takýmto testom. Aby bola zaistená kontrola správnosti behu, ale aj rýchlosti a schopnosti nájsť lepšie výsledky, boli okrem jednoduchých testov a testov zameraných na reálne problémy ktoré v skladoch nastávajú vytvorené zložité a komplexné syntetické testy.

## 2 POPIS REÁLNYCH TESTOV

Pre testovanie reálnych prípadov a overenie korektnosti algoritmu bolo navrhnutých 60 príkladov, aby bola možnosť porovnania úspešnosti genetického algoritmu voči riešeniu navrhnutého človekom. Testy boli založené na extrahovaných dátach z reálnej prevádzky skladu. Náročnosť jednotlivých testov bola rozdelená do piatich úrovní.

Testy 1 až 10 sú jednoduché prípady, kedy sa v sklade nachádzajú len ručné vozíky a každý pracovník má priradenú jednu jednoduchú úlohu. Testy 11 až 20 sú podobné prípady ako z predošlej skupiny, sú však rozšírené o ďalšie typy vozíkov. Každý pracovník má opäť priradenú jednu jednoduchú úlohu. Testy 21 až 30 sú stredne zložité prípady, kedy pracovníci majú vykonať väčší počet úloh. Úlohy sú jednoduché, ide o nové scenáre alebo o rozšírenia scenárov z predošlých testov. Testy 31 až 40 sú najmä rozšírené testy z predošlej skupiny. V týchto testoch bolo rozvrhnutie úloh navrhnuté tak, aby nedochádzalo ku kolíziám. Testy 41 až 60 sú zložité prípady s vyšším počtom úloh, kedy sa často vyžaduje spolupráca pracovníkov. Opäť sú úlohy rozvrhnuté tak, aby dochádzalo k čo najnižšiemu počtu kolízií. V tejto skupine je možné očakávať lepšie výsledky v prospech algoritmus genetického programovania.

### 3 POPIS SYNTETICKÝCH TESTOV

Algoritmus genetického programovania je nutné otestovať aj na väčších príkladoch. Z tohto dôvodu bol navrhnutý generátor testov, ktorý dokáže na základe určitých vstupných parametrov vytvoriť náhodnú ukážku problému riadenia práce v sklade. Generátor je založený na podobnom princípe ako sada vytvorených testov, využíva rovnakú štruktúru kódu ako sada reálnych testov.

Generátor pozostáva z troch častí, pričom v prvej sa definuje počet zamestnancov v sklade, ktorým budú nasledovne náhodne priradené vozíky, a pravdepodobnosti priradenia jednotlivých typov vozíkov (ručný vozík, malý rýchly vozík, vysokozdvíhový vozík určený pre najvyššie police regálu). V tejto časti sa vygeneruje presne daný počet pracovníkov. Pracovníkovi musí byť daná štartovacia pozícia (súradnice  $x$  a  $y$ ), pričom súradnicu  $x$  volíme podľa uličky. Tá sa vyberá pomocou vzorca

$$x = 2 \cdot \left( i \bmod \frac{w+1}{2} \right),$$

kde  $i$  je počet pracovníkov znížený o počet už vygenerovaných pracovníkov a  $w$  je celková šírka skladu. Súradnica  $y$  je volená náhodne v rozsahu uličky. Všetky pozície sa ukladajú do zoznamu `close`, aby nenastala situácia, kedy by na jednu pozíciu boli priradení dvaja pracovníci. Pri generovaní pracovníka sa mu ihneď priradí vozík. Typ vozíka je vybraný vzhľadom na vstupné parametre.

V druhej časti algoritmu sa generujú palety s tovarom. Keďže tento objekt nemá definovanú počiatočnú pozíciu, vytvárajú sa ihneď aj úlohy uskladnenia, v ktorých sú uložené počiatočné a konečné pozície. Požadujeme teda parametre ako je celkový počet palet tovaru a pravdepodobnosť, že paleta čaká na nákladnej bráne, respektíve pravdepodobnosť, že už je umiestnená na prekladovom mieste.

Jednotlivé palety majú kvôli rozlíšeniu definované len číslo. Počiatočné pozície sa generujú rovnako ako v predošlom kroku u generovania pracovníkov, pokiaľ teda ide o paletu ktorá je už pripravená na prekladovom mieste. Pokiaľ sa jedná o paletu, ktorá má čakať na nákladnej bráne, vyberá sa náhodne pozícia jednej z nákladných brán. Konečné pozície jednotlivých úloh sú náhodne vybrané pozície regálov, pričom súradnica  $x$  je vždy nepárne číslo z rozsahu šírky skladu a súradnica  $y$  je celé číslo z rozsahu  $\langle 1, h - 2 \rangle$ , kde  $h$  je výška skladu (teda veľkosť v súradnici  $y$ , v prípade 2D modelu).

### 4 OPTIMALIZAČNÝ ALGORITMUS

K riešeniu problému bol použitý algoritmus genetického programovania založený na stromových dátových štruktúrach, ktorý je vhodný pre riešenie optimalizačných úloh. Využívajú sa najmä v prípadoch kedy nemusíme nájsť optimálne riešenie daného problému ale stačí nám aj niektoré suboptimálne riešenie, ktoré je svojou kvalitou blízke tomu optimálnemu. [1] Testy boli vytvorené pre algoritmus genetického programovania, ktorý obsahoval nasledujúce genetické operátory:

**P** – operátor **zmena cesty**, ktorý zabezpečí zníženie počtu kolízií v sklade.

**TO** – pokiaľ dôjde k **zmene poradia úloh** v pracovnom pláne a zároveň v tom nebráni žiadne obmedzenie (napr. návaznosť úloh), môžeme sa prepracovať k lepším výsledkom. Dobrým príkladom je závislosť počiatočnej pozície úlohy od aktuálnej polohy pracovníka, teda nutný presun pracovníka.

**ST** – **výmena úloh medzi pracovnými plánmi** – Aby sme zabezpečili určitú voľnosť v rozhodovaní úloh, musíme zabezpečiť aj výmenu úloh medzi pracovnými plánmi. Tento operátor mnohonásobne zväčšuje prehl'adávací priestor.

**SW** – jednoduchá **výmena pracovných plánov** medzi dvomi pracovníkmi. V tomto prípade však musíme skontrolovať, či je možné skutočne vymeniť celé pracovné plány medzi pracovníkmi – niektoré úlohy môžu mať obmedzenia, ktoré tejto výmene môžu zabrániť.

**SP** – mutácia **rozdeľuje úlohu** na dve podúlohy, pričom každá bude vykonávaná iným pracovníkom. Táto mutácia pomáha najmä v prípadoch, kedy potrebujeme uskladniť tovar do vyšších poschodí.

## 5 TESTOVANIE ALGORITMU

Sadou reálnych testov sme získali dáta pre porovnanie rozvrhovania práce človekov a algoritmom genetického programovania, tab. 1 zachytáva priemerné hodnoty z každej kategórie testov. U syntetických testoch boli výsledné časy a počty kolízií vypočítané priemerom z 5 náhodných prípadov (tab. 2). Beh algoritmu bol zastavený pri dosiahnutí 20. generácie a počet jedincov v generáciách bol

Testy	Človek	2 GO	Kvalita	3 GO	Kvalita	5 GO	Kvalita
1 – 10	7,32	8,17	90%	8,17	90%	8,17	90%
11 – 20	7,01	7,53	93%	7,53	93%	7,53	93%
21 – 30	15,79	15,15	104%	15,13	104%	15,45	102%
31 – 40	15,17	16,09	94%	16,03	95%	16,29	93%
41 – 60	23,38	14,00	167%	14,01	167%	14,18	165%

**Tabuľka 1:** Tabuľka porovnania priemerných výsledkov reálnych testov

obmedzený na 20. Vo fitness funkcií mala najvyššiu váhu hodnota obsahujúca maximálny čas potrebný na vykonanie všetkých úloh, váha kolízií bola najmenšia – zvolené preto, aby sa práca rozdelila rovnomerne medzi pracovníkov a aby pracovníci nestáli na mieste z dôvodu vyhnutiu sa kolíziám. Zvýraznené hodnoty sú najnižšie dosiahnuté, teda konkrétna kombinácia pravdepodobností použitia

Pravdepodobnosť operátora					10 úloh		50 úloh	
P	TO	ST	SW	SP	Čas	Kolízie	Čas	Kolízie
20%	20%	20%	20%	20%	<b>19,01</b>	16,80	37,08	331,80
60%	20%	20%	20%	20%	21,23	13,80	37,04	320,00
20%	20%	60%	20%	20%	<b>18,91</b>	13,80	36,95	332,40
60%	60%	60%	20%	20%	20,33	<b>9,40</b>	36,01	333,00
20%	20%	20%	60%	20%	20,43	11,80	<b>34,38</b>	323,60
60%	20%	60%	60%	20%	20,17	13,00	<b>34,35</b>	329,60
20%	60%	60%	60%	20%	19,24	17,20	35,97	361,20
60%	60%	60%	60%	20%	<b>18,70</b>	13,00	<b>34,96</b>	343,20
20%	20%	20%	60%	60%	21,06	12,40	38,66	<b>292,60</b>

**Tabuľka 2:** Porovnanie vplyvu pravdepodobnosti použitia operátorov – iba najlepšie výsledky

daného operátora zabezpečí čo najnižší čas spracovania, respektíve čo najnižší počet kolízií pre danú komplexnosť problému.

## 6 ZÁVER

Algoritmus riadenia a rozvrhovania práce bol otestovaný 60 navrhnutými testami inšpirovanými reálnymi dátami, pričom pri každom teste bola určená kvalita riešenia ako pomer času potrebného na úspešné vykonanie úloh tak ako by to navrhol človek, a času tak ako to navrhol algoritmus genetického programovania. Vysoká kvalita je dosiahnutá hlavne v ťažších prípadoch. Pri jednoduchších prípadoch je porovnateľná s človekom. Celková dosiahnutá kvalita riešení algoritmom bola 123%.

Pri podrobení testu na vygenerovaných úlohách, kedy šlo o väčšie problémy s vyšším počtom úloh, bolo výsledkom porovnanie nastavení pravdepodobností jednotlivých genetických operátorov a určenie najvhodnejších kombinácií vzhľadom na veľkosť problému.

## LITERATÚRA

- [1] MACH, M. *Evolučné algoritmy: prvky a princípy*. Elfa, Košice, 2009. 250 strán. ISBN 978-80-8086-123-0.
- [2] MOON, I.; LEE, J. Genetic algorithm application to the job shop scheduling problem with alternative routings. *Brain Korea 21 Logistics Team*, 2000.