

ŠKODA AUTO VYSOKÁ ŠKOLA, O.P.S.

Studijní program: N6208 Ekonomika a management

Studijní obor: 6208T088 Podniková ekonomika a management provozu

VYUŽITÍ LINEÁRNÍHO PROGRAMOVÁNÍ PŘI VÝBĚRU SVAŘOVACÍCH KLEŠTÍ V RÁMCI SVAŘOVNY ŠKODA AUTO a.s.

Bc. Filip KUŘÍK

Vedoucí práce: doc. Ing. Jan Fábry, Ph.D.

Tento list vyjměte a nahrad'te zadáním diplomové práce

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a veškeré použité prameny a informace uvádím v seznamu použité literatury.

Souhlasím, aby moje práce byla zveřejněna v souladu s §47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a v souladu s vnitřním předpisem ŠKODA AUTO VYSOKÉ ŠKOLY o.p.s. o zveřejňování závěrečných prací Směrnice Vypracování závěrečné práce.

Jsem si vědom, že se na tuto práci plně vztahuje zákon č. 121/2000 Sb., autorský zákon, ve znění pozdějších předpisů, zejména pak § 35 odst. 3, tzn., že ŠAVŠ nezasahuje do mých práv v případě využití této práce pro vnitřní potřebu a §60 – školní dílo. Beru na vědomí, že ŠAVŠ má právo na uzavření licenční smlouvy k této práci za obvyklých podmínek. Užiju-li tuto práci, nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti ŠAVŠ. V tomto případě má ŠAVŠ právo ode mne požadovat příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to až do jejich skutečné výše.

V Mladé Boleslavi dne 17. května 2019

Děkuji doc. Ing. Janu Fábrymu, Ph.D. za odborné vedení diplomové práce, poskytování rad a informačních podkladů. Dále bych chtěl poděkovat svým kolegům z oddělení plánování svařoven za vstřícný přístup a poskytnutí podkladů důležitých pro vypracování této diplomové práce.

Obsah

Seznam použitých zkratk a symbolů.....	6
Úvod.....	7
1 Úlohy lineárního programování.....	8
1.1 Řešení pomocí lineárního programování	9
1.1.1 Celočíselné programování.....	9
1.1.2 Aplikace úlohy o pokrytí.....	11
1.1.3 Aplikace úlohy o dělení.....	16
1.1.4 Aplikace úlohy o rozkladu	16
1.2 Řešení pomocí heuristických metod.....	17
1.3 Visual Basic for Applications	18
1.4 Software MPL for Windows 5.0	19
1.4.1 Charakteristika částí modelovacího jazyka	20
2 Plánování svařoven ve ŠKODA Auto a.s.....	23
2.1 Digitální továrna v plánování svařoven ve ŠKODA AUTO a.s.....	23
2.1.1 Charakteristika činností plánování svařoven v digitální továrně.....	24
3 Výběr svařovacích kleští.....	28
3.1 Konstrukční provedení svařovacích kleští	28
3.2 Tvorba vstupní matice a analýza dat pro výběr svařovacích kleští.....	29
3.2.1 Úprava vstupních dat v MS Excel	32
4 Navrhovaná řešení pro výběr svařovacích kleští	34
4.1 Optimální výběr svařovacích kleští pomocí matematického modelu	34
4.2 Výběr svařovacích kleští pomocí heuristické metody	36
4.3 Porovnání výsledků navrhovaných řešení.....	38
4.3.1 Ekonomické zhodnocení navrhovaných řešení.....	41
4.3.2 Doporučení	43
Závěr	44
Seznam literatury	46
Seznam obrázků a tabulek.....	48
Seznam příloh	49

Seznam použitých zkratk a symbolů

MPL	Mathematical Programming Language
MS	Microsoft
SCP	Set Covering problem
ŠA	ŠKODA AUTO
VBA	Visual Basic for Windows

Úvod

Tato diplomová práce se zabývá výběrem svařovacích kleští ve svařovnách ŠKODA AUTO a.s. pomocí lineárního programování. V dnešním světě IT technologií lze řešit mnoho složitých úloh pomocí lineárního programování, ať už v oblasti veřejných služeb, tak v oblasti výroby. Častým cílem těchto úloh, je najít takové řešení, které minimalizuje náklady. V oblasti automobilového průmyslu, konkrétně v oblasti svařoven ŠKODA AUTO a.s. lze minimalizovat počet druhů svařovacích kleští, a s tím souvisejících nákladů ve výrobních linkách, na což bude tato práce zaměřena.

Tématem diplomové práce jsou rozhodovací procesy ve svařovně ŠKODA AUTO a.s. týkající se výběru svařovacích kleští s cílem minimalizovat počet jejich druhů a s tím souvisejících pořizovacích nákladů. Teoretickým východiskem budou modely a metody matematického programování, především matematický model úlohy o pokrytí. Získané výsledky budou návrhem pro reálné plánování v úseku svařovny.

Teoretická část práce se bude zabývat řešením reálných příkladů úloh lineárního programování se zaměřením na úlohu o pokrytí, jež je stěžejní pro praktickou část práce. Dále bude popsán alternativní způsob řešení úlohy o pokrytí pomocí heuristiky a představen programovací jazyk Visual Basic for Windows. Následně bude popsán programovací software MPL for Windows 5.0, který je nutný pro řešení úlohy o pokrytí pomocí lineárního programování a budou charakterizovány jednotlivé části modelovacího jazyka. V další kapitole bude představeno oddělení plánování svařoven ve ŠKODA AUTO a.s., se zaměřením na digitální továrnu a činnosti s tím spojené, zejména na výběr svařovacích kleští.

V praktické části práce budou analyzována vstupní data pro výběr svařovacích kleští a vytvořena vstupní matice dat v MS Excel. Matice dat pak bude použita v rámci navrhovaných řešení, která budou podrobně popsána společně se zápisem syntaxe v daném programovacím jazyce. Následně bude provedeno porovnání výsledků z obou navrhovaných řešení se současným stavem analyzované výrobní linky svařovny. Na závěr praktické části budou představeny přínosy navrhovaných řešení, včetně ekonomického zhodnocení a jejich doporučení.

1 Úlohy lineárního programování

Set-Covering, *Set-Packing* a *Set-Partitioning problems* neboli problémy úloh o pokrytí, dělení a o rozkladu jsou klasickými problémy v operačním výzkumu, kombinatorice a programování. Jedná se o problémy, které vedly k vývoji základních technologií v oblasti aproximačních algoritmů. Tyto úlohy lze aplikovat v celé řadě průmyslových odvětví, včetně plánování, výroby, logistiky a služeb. *Set-Covering*, *Set-Packing* a *Set-Partitioning* problémy patří mezi nejstarší a nejvíce studované optimalizační problémy v literatuře o matematickém programování (Hindawi, 2014).

Jako příklad zde budou uvedeny výňatky toho nejdůležitějšího z několika článků, které se zabývají tímto tématem, především řešením *Set-Covering* problému pomocí lineárního programování a heuristických metod.

V článku od autorů Luttera a kol. (2017) je *Set-Covering problem* (SCP) uveden jako jeden z klasických kombinatoristických optimalizačních problémů a je součástí mnoha praktických aplikací. Jednou z těchto aplikací je umístění záchranných stanic, kde cílem je, aby dojezdové časy k místu zásahu v daných lokalitách byly co nejkratší, aby byly pokryty všechny lokality a aby toto rozmístění poskytovalo dostatečně vysokou kvalitu služeb. V mnoha reálných problémech nejsou vyžadované statistické údaje zcela známy, (počet vozidel záchranné služby, počet volání o pomoc, počasí, provoz, denní doba) a tím pádem získaná řešení mohou odhalit neadekvátní provedení dané optimalizace. V článku se autoři zabývají robustní formulací pravděpodobných statistických údajů SCP, kde kombinují koncept robustní a pravděpodobnostní optimalizace. Tuto robustní metodu lze stanovit kompaktním smíšeným modelem celočíselného lineárního programování. K tomu, aby byli poskytovatelé služeb schopni podat adekvátní výsledky, je tedy z praktického hlediska důležité zahrnout tyto pravděpodobné a nejisté události. V tomto článku se autoři dále odkazují na formulaci celočíselného programování SCP, kterou představili Toregas a kol. (1971), kteří do své formulace pravděpodobné a nejisté situace nezahrnuli.

Dále se SCP objevuje v člancích, které poskytují přehledy optimalizačních modelů rozmístění záchranných stanic od autorů Brotcora, Laporteho a Semeta (2003), Lia a kol. (2011) či ReVellea (1989). Nebo jako jeden z problémů v disciplíně

graph coloring v teorii grafů od autorů Hansena, Labbého a Schindla (2009). Dalšími autory, kteří se SCP zabývali, byli Ceria, Nobili a Sassano (1998), kteří ve svém článku řešili rozvrhování pracovních směn nebo Owen a Daskin (1998) či Snyder (2006), kteří ve svých pracích uvádějí přehledy o plánování umístění různých zařízení s ohledem na nejisté a pravděpodobné situace (Lutter, 2017).

Studie Kuo a Leunga (2016) o polyhedrální struktuře *Set-Covering, Packing a Partitioning* problému je tématem, které sice nepřitahuje velkou pozornost v literatuře, ale přesto má opět mnoho praktických aplikací (Kuo, 2016).

Posledním zmíněným článkem bude článek od Gomeze a kol. (2006), kteří vedli kontrolovanou empirickou studii aproximačního algoritmu pro SCP. Mnoho dalších autorů již dříve navrhlo aproximační algoritmus pro tento problém, ale cílem zmíněných autorů bylo především lépe porozumět slabinám, silným stránkám a jednotlivým operacím algoritmů. Z jejich práce vyplývá, že nalezení přibližného řešení je teoreticky často odlišné od experimentálního provedení (Gomes, 2006).

1.1 Řešení pomocí lineárního programování

Pro řešení výše zmíněných problémů pomocí lineárního programování je na místě definovat základní myšlenku lineárního programování.

„Lineární programování je disciplínou operačního výzkumu, která se zabývá řešením rozhodovacích problémů, ve kterých jde o určení intenzit realizace procesů, které probíhají (nebo mohou probíhat) v daném systému“ (Jablonský, 1998, str. 17).

Jinými slovy, lineární programování je disciplína operačního výzkumu, která se zabývá řešením rozhodovacích problémů. Musí se zde respektovat určité podmínky a najít řešení, při kterém je cíl splněn co nejlépe. Programování je synonymem pro plánování programů a slovo lineární pak vyjadřuje to, že všechny vazby v modelech jsou lineární.

1.1.1 Celočíselné programování

Celočíselné programování je cestou k řešení optimalizačních problémů za pomoci diskrétních proměnných. Proměnné jsou používány k modelování nedělitelnosti a „0/1“ proměnné jsou používány k reprezentaci „ano/ne“ rozhodnutí v rámci nákupu, investic, nájmu. Tyto problémy se vyskytují v různých reálných situacích, ať už se

jedná o vytvoření letových nebo jízdnic řádů, plánování pracovní náplně údržbového týmu či plánování týdenní produkce elektrické energie.

V posledních 30 letech významně pokročila schopnost řešit složité praktické celočíselné úlohy, a to díky kombinaci:

- zdokonaleného modelování,
- rychlejších počítačů,
- pokročilých lineárních programovacích softwarů,
- nových teorií a algoritmů,
- nových heuristických metod.

Dříve mnoho uživatelů v oblasti průmyslu používalo celočíselné programovací modely, ale skončili u prvního možného řešení, které jim jejich software poskytl. I když byl problém velmi jednoduchý, řešení poskytnutá softwarem se často výrazně lišila od optimálního řešení, což vedlo k velkým finančním ztrátám. V dnešní době lze již poskytnout řešení, která jsou optimální nebo se od optimálního řešení jen minimálně liší, a to za rozumný čas. Avšak k tomu je zapotřebí mít k dispozici novější komerční softwary, speciálně navržené algoritmy a lepší celočíselné programovací modely (Wolsey, 1998).

Celočíselné lineární programování

Celočíselné lineární programování se od obecného lineárního programování liší pouze tím, že všechny proměnné smějí nabývat pouze celočíselných hodnot. Úlohy celočíselného lineárního programování jsou obecně velmi obtížné.

V celočíselných úlohách se obvykle objevují dva základní typy proměnných:

Celočíselné, které mohou nabývat pouze celočíselných hodnot.

Bivalentní (též nula-jedničkové, popřípadě binární), které mohou nabývat pouze hodnot 0 nebo 1. Striktně vzato jde o kombinaci podmínky celočíselnosti a podmínky omezení na interval $\langle 0,1 \rangle$, tedy o zvláštní případ výše popsaného, přesto jde o případ velmi důležitý, jak z hlediska praxe, tak i z hlediska metod výpočtu.

V úlohách smíšeně-celočíselných se vyskytují proměnné **spojité**, které mohou nabývat všech reálných hodnot z nějakého intervalu. Množina hodnot proměnné je

omezena pomocí běžné podmínky pro jednotlivou proměnnou, zpravidla jde o podmínku nezápornosti (Demel, 2011, str. 55).

Následující část práce bude zaměřena na formulování problému úlohy o pokrytí jako problému celočíselného lineárního programování (Eiselt a Sandblom, 2000 str. 155).

Nechť M je množinou a $\{M_j \subseteq M, j \in J\}$ souborem podmnožin M . Tento soubor je úlohou o pokrytí množiny M , když $\bigcup_{j \in J} M_j = M$, úlohou o dělení množiny M , když $M_j \cap M_k = \emptyset \forall j, k \in J$ a $j \neq k$, a úlohou o rozkladu množiny M , když platí úloha o pokrytí i dělení množiny M .

Například, když $M = \{1, 2, 3, 4, 5\}$, tak $\{1, 2, 3\}, \{2\}, \{3, 4, 5\}$ je úlohou o pokrytí, ale není úlohou o dělení množiny M ; $\{1, 4\}, \{2, 5\}$ je úlohou o dělení, ale není úlohou o pokrytí množiny M ; $\{1, 2, 5\}, \{3, 4\}$ je úlohou o rozkladu množiny M . Pokud bychom definovali logickou proměnnou y_j jako $y_j = 1$ když $j \in J$, jinak 0, pak můžeme odkázat na omezující podmínky:

$$\sum_{j \in J} y_j \geq 1, \quad (1)$$

jež je omezením úlohy o pokrytí,

$$\sum_{j \in J} y_j \leq 1, \quad (2)$$

jež je omezením úlohy o dělení,

$$\sum_{j \in J} y_j = 1, \quad (3)$$

jež je omezením úlohy o rozkladu (Eiselt a Sandblom, 2000 str. 155).

1.1.2 Aplikace úlohy o pokrytí

V této kapitole budou popsány konkrétní příklady aplikace úlohy o pokrytí, ve kterých bude přiblížena daná problematika a rozsah použití.

Vezměme například projekty $P = \{P_1, P_2, \dots, P_m\}$, $M = \{1, 2, \dots, m\}$, oddělení O_1, O_2, \dots, O_n , přičemž každé oddělení zajišťuje určitou podmnožinu ze všech projektů, a to za cenu c_j , což znamená, že každý projekt zajišťuje oddělení za

stejnou cenu (Fábry, 2017). V matici A máme informace o tom, zda je oddělení j schopno zajistit projekt i : $a_{ij} = 1$ tehdy, pokud je oddělení j schopno zajistit projekt i , jinak 0.

$$a_{ij} = \begin{cases} 1 & \text{jestli je oddělení } j \text{ schopno zajistit projekt } i, \\ 0 & \text{jinak,} \end{cases} \quad (4)$$

Cílem pokrývacího problému je vybrat oddělení tak, aby byly co nejlevněji pokryty všechny projekty.

- **Proměnná**

V matematickém modelu použijeme binární proměnnou:

$$x_j = \begin{cases} 1 & \text{pokud je vybráno oddělení } j \\ 0 & \text{jinak,} \end{cases} \quad j = 1, 2, \dots, n. \quad (5)$$

- **Účelová funkce**

Minimalizujeme celkové náklady (6) na zajištění projektů:

$$z = \sum_{j=1}^n c_j x_j \rightarrow \min. \quad (6)$$

- **Omezující podmínky**

Omezující podmínky (7) říkají, že každý projekt musí zajišťovat alespoň jedno oddělení a sčítáme jen přes ta oddělení, která jsou schopna zajistit i -tý projekt, protože jinak je $a_{ij} = 0$.

$$\sum_{j=1}^n a_{ij} x_j \geq 1, \quad i = 1, 2, \dots, m. \quad (7)$$

Příklad 1

Výše uvedené lze aplikovat například u výstavby závodních jídelen ve ŠKODA AUTO a.s. v různých lokalitách (analogie oddělení), které mají obsluhovat určité úseky závodu (analogie projektů). V matici A je prvek $a_{ij} = 1$ tehdy, pokud je jídelna v j -té lokalitě ve stanovené vzdálenosti od i -tého úseku závodu.

$$a_{ij} = \begin{cases} 1 & \text{jestli je jídelna v lokalitě } j \text{ schopna obsloužit úsek závodu } i, \\ 0 & \text{jinak,} \end{cases} \quad (8)$$

c_j pak značí například náklady na provoz jídelny v j -té lokalitě.

- **Proměnná**

$$x_j = \begin{cases} 1 & \text{pokud bude v lokalitě } j \text{ postavena jídelna} \\ 0 & \text{jinak,} \end{cases} \quad j = 1, 2, \dots, n. \quad (9)$$

- **Účelová funkce**

Cílem účelové funkce je minimalizovat celkové náklady (10) na provoz závodní jídelny:

$$z = \sum_{j=1}^n c_j x_j \rightarrow \min. \quad (10)$$

- **Omezující podmínky**

Následující omezující podmínky (11) říkají, že každý úsek závodu musí obsluhovat alespoň jedna jídelna:

$$\sum_{j=1}^n a_{ij} x_j \geq 1, \quad i = 1, 2, \dots, m. \quad (11)$$

Příklad 2

Další typickou aplikací úlohy o pokrytí je situace, kdy máme n obvodů O_1, O_2, \dots, O_n , ve kterých chceme vybudovat celkem K hasičských stanic, přičemž $n > K$ (Jablonský, 1998). Dále je třeba stanovit, které obvody budou obsluhovány, kterou hasičskou stanicí. Proto abychom mohli rozhodnout, ve kterých obvodech postavit hasičské stanice, a které obvody stanicím přiřadit tak, aby doba zásahu byla minimální, potřebujeme znát informace o čase a dojezdové vzdálenosti z hasičské stanice postavené v obvodu O_i do obvodu O_j a průměrnou frekvenci zásahů v každém obvodu. A musí platit, že jeden prvek množiny, tedy jedna hasičská stanice, by měla obsluhovat více prvků, obvodů. Náklady (čas či vzdálenost) na dojezd ze stanice v i -tém obvodě do j -tého obvodu označíme c_{ij} . Průměrnou frekvenci zásahů v j -tém obvodě označíme f_j .

- **Proměnné**

V matematickém modelu zavedeme dvě binární proměnné:

$$x_{ij} = \begin{cases} 1 & \text{pokud stanice v obvodu } O_i \\ & \text{obsluhuje stanici } O_j \\ 0 & \text{jinak,} \end{cases} \quad \begin{matrix} i = 1, 2, \dots, m, \\ j = 1, 2, \dots, n. \end{matrix} \quad (12)$$

$$y_i = \begin{cases} 1 & \text{pokud je v obvodě } O_i \text{ zřízena stanice} \\ 0 & \text{jinak,} \end{cases} \quad i = 1, 2, \dots, m. \quad (13)$$

- **Účelová funkce**

Cílem účelové funkce je minimalizovat čas a dojezdovou vzdálenost (14) násobenou frekvencí zásahů:

$$z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} f_j \rightarrow \min. \quad (14)$$

- **Omezující podmínky**

Omezující podmínky (15) říkají, že když bude v i -tém obvodě stanice, bude obsluhovat nejvýše n obvodů:

$$\sum_{j=1}^n x_{ij} \leq n y_i, \quad i = 1, 2, \dots, m. \quad (15)$$

Omezující podmínky (16) říkají, že každý obvod patří právě jedné stanici:

$$\sum_{i=1}^m x_{ij} = 1, \quad j = 1, 2, \dots, n. \quad (16)$$

Omezující podmínka (17) říká, že celkem musí být zřízeno K stanic:

$$\sum_{i=1}^m y_i = K. \quad (17)$$

Příklad 3

V určité zemi je publikováno pět denních tisků, přičemž každý tisk pokrývá některé z devíti regionů v zemi (viz Tab. 1).

Tab. 1 Vstupní data pro příklad 3

Denní tisk	Pokryté regiony	Náklady na reklamu	Zisk z reklamy
1	1, 2, 4	3	12
2	2, 3, 6	4	10
3	4, 5, 6	3	14
4	5, 7, 8	7	19
5	6, 8, 9	5	16

Zdroj: Eiselt a Sandbom (2000, str. 156)

Nechť $M = \{1, 2, \dots, 9\}$ je množinou všech regionů a $M_j, j = 1, \dots, 5$ je množinou pokrytých regionů j -tým denním tiskem. Nechť c_j značí náklady na reklamu v j -tém denním tisku a definuje proměnnou x_j , která předpokládá hodnotu 1, pokud je reklama umístěna v denním tisku j , jinak 0. Úloha o pokrytí vyřeší problém s nalezením minimálních celkových nákladů při výběru denního tisku pro reklamu v rámci celé země. Obecně lze model formulovat takto:

- **Proměnná**

$$x_j = \begin{cases} 1 & \text{pokud je reklama umístěna v denním tisku } j \\ 0 & \text{jinak,} \end{cases} \quad j = 1, 2, \dots, 5. \quad (18)$$

- **Účelová funkce**

Cílem účelové funkce je minimalizovat celkové náklady (19) při výběru denního tisku pro reklamu v rámci celé země:

$$z = \sum_j c_j x_j \rightarrow \min. \quad (19)$$

- **Omezující podmínky**

$$\sum_{j:i \in M_j} x_j \geq 1 \quad \forall i \in M \quad (20)$$

1.1.3 Aplikace úlohy o dělení

V této kapitole bude popsán příklad aplikace úlohy o dělení.

Modifikovaný příklad 3

Struktura problému úlohy o dělení je velmi podobná struktuře problému úlohy o pokrytí. Pokud definujeme d_j jako zisk z umístění reklamy v denním tisku j a pokud budeme vyžadovat, že žádná z reklam nebude dostupná ve více jak jednom regionu, pak můžeme problém úlohy o dělení formulovat jako:

- **Proměnná**

$$x_j = \begin{cases} 1 & \text{pokud je reklama umístěna v denním tisku } j \\ 0 & \text{jinak,} \end{cases} \quad j = 1, 2, \dots, 5. \quad (21)$$

- **Účelová funkce**

Cílem účelové funkce je maximalizovat zisk z reklamy (22) v daném regionu:

$$z = \sum_j d_j x_j \rightarrow \max. \quad (22)$$

- **Omezující podmínky**

Omezující podmínky (23) říkají, že žádná z reklam nebude dostupná ve více jak jednom regionu:

$$\sum_{j: i \in M_j} x_j \leq 1 \quad \forall i \in M \quad (23)$$

1.1.4 Aplikace úlohy o rozkladu

Modifikovaný příklad 3

Formulace problému úlohy o rozkladu je identická formulaci problému úlohy o dělení, až na to, že nerovnostní podmínky jsou nahrazeny rovnicí. V našem příkladu o denním tisku (Eiselt a Sandblom, 2000 str. 157), omezení vyžaduje, aby reklama byla v každém regionu v zemi právě v jednom denním tisku. Obecně lze model formulovat takto:

- **Proměnná**

$$x_j = \begin{cases} 1 & \text{pokud je reklama umístěna v denním tisku } j \\ 0 & \text{jinak,} \end{cases} \quad j = 1, 2, \dots, 5. \quad (24)$$

- **Účelová funkce**

Cílem účelové funkce je maximalizovat zisk z reklamy (25) v daném regionu:

$$z = \sum_j d_j x_j \rightarrow \max. \quad (25)$$

- **Omezující podmínky**

Omezující podmínky (26) požadují, aby reklama byla v každém regionu v zemi právě v jednom denním tisku:

$$\sum_{j:i \in M_j} x_j = 1 \quad \forall i \in M. \quad (26)$$

1.2 Řešení pomocí heuristických metod

Set Covering problem neboli úlohu o pokrytí, lze řešit pomocí aproximačních metod, mezi které patří heuristiky. Pro pochopení základní myšlenky heuristických metod lze použít následující definici (Maňák a Švec, 2003, str. 113):

„Heuristika (z řec. heuréka = objevil jsem, našel jsem) je věda zkoumající tvůrčí myšlení, také heuristická činnost, tj. způsob řešení problémů.“

Jinými slovy, heuristické metody obecně neposkytují optimální řešení úlohy, ale měly by poskytnout dostatečné řešení v rozumném čase.

Začněme s formulováním úlohy o pokrytí. Nechť M je množinou všech řádků, M_j je množinou řádků pokrytých sloupcem j ; N je množinou všech sloupců, N_i je množinou sloupců pokrývajících řádek i . Dále nechť R je množinou nepokrytých řádků a S množinou sloupců, které jsou obsaženy v řešení ($S = \{j \in N \mid x_j = 1\}$).

Jednotlivé kroky heuristiky Vasko a kol. (2016) založené na práci (Vasko a Wilson, 1984) budou popsány níže.

Krok 1

- Množina $R = M$, $S = \emptyset$, $t = 1$, kde $M = \{1, 2, \dots, m\}$. Přejděme ke kroku 2.

Krok 2

- Pokud $R = \emptyset$, postoupíme ke kroku 3. Jinak, nechť $K_j = |M_j \cap R|$ pro všechna $j \in N$, kde $N = \{1, 2, \dots, n\}$, a vyberme $j^* = \arg \min \{f(c_j, K_j) \mid K_j > 0\}$, kde $f(c_j, K_j)$ je definováno níže. Množina $R = R \setminus M_{j^*}$, $S = S \cup \{j^*\}$, $t = t + 1$, $M_j = \{i \in M \mid a_{ij} = 1\}$ a K_j označuje číslo kladného koeficientu x_j v řádcích, které ještě nebyly pokryty současnou omezující množinou. Opakujme krok 2.

Krok 3

- Roztřídíme prvky z množiny S na základě klesajících c_j hodnot. Berme prvky $i \in S$ v pořadí a pokud je proveditelné pokrytí $S \setminus \{i\}$, tak množina $S = S \setminus \{i\}$. A když všechna $i \in S$ byla vzata v potaz, množina S definuje primární pokrytí (pokrytí bez žádných zbylých sloupců).

Vasko a Wilson vytvořili následujících sedm sloupcových funkcí $f(c_j, K_j)$: (1) c_j , (2) c_j/K_j (3) $c_j / (\log_2 K_j)$, (4) $c_j / (K_j \log_2 K_j)$ (5) $c_j / (K_j \ln K_j)$, (6) c_j / K_j^2 , (7) $(c_j)^{1/2} / K_j^2$.

Použitím každé ze sedmi sloupcových funkcí může být zkonstruováno sedm řešení pokrývacího problému, které nejsou nezbytně unikátní. Krok 3 je použit k velmi efektivnímu odstranění nežádoucích (přebytečných) sloupců, které mohou vstoupit na začátku do řešení, ale nejsou dále potřeba v konečném řešení.

1.3 Visual Basic for Applications

Visual Basic for Applications ve zkratce VBA je programovací jazyk vyvinutý společností Microsoft (Walkenbach, 2013, str. 12). Microsoft (MS) Excel spolu s ostatními aplikacemi Microsoft Office obsahují VBA, jež je nástrojem pro vývoj programů. VBA for Excel je v podstatě prostředník, který zná všechno o Excelu, umí číst instrukce a dokáže pracovat velmi rychle. Pokud chceme, aby byla provedena nějaká instrukce, je nutné ji zapsat pomocí speciálního kódu.

Programovací jazyk VBA lidé využívají k nejrůznějším úkonům, ať už složitějším, tak jednodušším a umožňuje tyto úkony řešit automaticky a rychle. Například se může jednat o analýzu dat, hledání přípustného řešení, vytváření faktur a nejrůznějších prognóz. Většina lidí využívá VBA for Excel k automatizaci rutinních úkolů v MS Excel, a i v této práci bude použit VBA for Excel.

Výhody VBA for Excel

VBA for Excel umožňuje automatizovat téměř cokoli v aplikaci Excel. Stačí pouze napsat instrukce pomocí speciálního kódu, které aplikace Excel provede. Automatizace úkolů pomocí VBA for Excel má několik výhod:

- Excel vždy provádí úkol přesně stejným způsobem.
- Excel provádí úkoly mnohem rychleji, než bychom to zvládli sami (ručně).
- Pokud jste dobrý programátor maker, Excel vždy provede úkol bez chyb.
- Pokud je vše správně nastaveno, tak i začátečník v MS Excel, může provést daný úkol.

Nevýhody VBA for Excel

Vedle výhod je třeba uvést i nevýhody spojené s VBA for Excel:

- Nutná znalost psaní programů ve VBA.
- Ti, kteří chtějí používat programy VBA, musejí mít svoji vlastní kopii MS Excel.
- Nelze se vždy spolehnout na správné fungování programu ve VBA.

1.4 Software MPL for Windows 5.0

V následující kapitole bude stručně představen optimalizační software MPL (Mathematical Programming Language) for Windows 5.0, který je určen pro formulaci optimalizačních modelů. V rámci této kapitoly bude popsán teoretický základ samotného softwarového nástroje a vysvětlena syntaxe pro zadávání modelů (MAXIMAL SOFTWARE, 2019).

Software MPL for Windows (dále jen MPL) je pokročilý modelovací nástroj, který umožňuje modelovat i složité a rozsáhlé optimalizační modely jasným a efektivním způsobem. Modely vytvořené v tomto nástroji lze řešit některým z mnoha komerčních řešitelů dostupných na současném trhu, například pomocí: CPLEX, GUROBI, Lindo, LPSolve, Ipopt, Conopt nebo Knitro. Základním řešitelem je předvybrán CPLEX 300, který umožňuje řešení úloh s max. 300 proměnnými. MPL v sobě zahrnuje algebraický modelovací jazyk, který umožňuje vytvářet modely pomocí algebraických rovnic. Samotný model je pak základem pro generování

matematické matice. Generování matematických matic probíhá na pozadí, což znamená, že se stačí zaměřit pouze na formulaci modelu.

1.4.1 Charakteristika částí modelovacího jazyka

Modelovací jazyk v softwaru MPL využívá pro vytvoření obecné formulace modelu 11 částí, které však nejsou všechny povinné. V rámci této kapitoly budou popsány jednotlivé části modelovacího jazyku, které jsou použity v praktické části práce (Jablonský, 2007).

TITLE

Tato část modelu obsahuje pouze název modelu, který má však jasné specifikace. Nesmí obsahovat mezery, které je možné nahradit například podtržítkem a neměla by být použita ani diakritika. Zápis jména modelu lze zapsat následujícím způsobem:

```
TITLE model_v1;
```

OPTIONS

Část OPTIONS se využívá pro propojení pomocného souboru v tabulkovém kalkulátoru MS Excel. Pomocný soubor musí být pojmenovaný, uložený a otevřený při řešení modelu. Pro příklad bude soubor uložený a otevřený v MS Excel pojmenován jako *soubor_pro_mpl*. Soubor uložený v MS Excel bude mít příponu *.xlsx*. V MPL bude zápis vypadat následujícím způsobem:

```
ExcelWorkBook="soubor_pro_mpl.xlsx";
```

INDEX

V této části modelu je zapotřebí definovat jakých hodnot mají nabývat indexy použité v modelu. Lze použít jak numericky, tak textově definované indexy. Pro numericky definované indexy lze využít například odkaz na definovaný název určité oblasti v pomocném *.xlsx* souboru. Zápis může vypadat takto:

```
i:=EXCELRange("zarizeni");  
j:=EXCELRange("vyrobni_stanice");
```

DATA

Po části INDEX následuje část DATA, kde lze zadat, jak data definovaná přímo v modelu, tak i data z externích datových souborů, tedy například data z již zmíněného pomocného souboru *soubor_pro_mpl.xlsx*. Zde existuje více možností zápisu:

```
a[i, j] :=EXCELRANGE("realizuje");  
M=1000;
```

VARIABLES

V části VARIABLES jsou definovány proměnné modelu. Vždy záleží na typu definované proměnné, které se mohou jmenovat: `BINARY`, `INTEGER`, `DECISION` nebo `FREE VARIABLES`. Zápis při definování proměnných musí obsahovat příslušné indexy v hranatých závorkách. Pro příklad lze uvést následující zápis:

```
INTEGER VARIABLES
```

```
x[i] EXPORT TO EXCELRANGE("pocet");
```

```
BINARY VARIABLES
```

```
y[i] EXPORT TO EXCELRANGE("koupit");
```

`EXPORT TO EXCELRANGE` pak umožňuje export hodnot proměnné zpět do předem definované a pojmenované oblasti souboru **.xlsx*.

MODEL

Modelová část obsahuje aktuální formulaci problémů. Uspořádání modelu je jednoduché. Prvním krokem je účelová funkce. Používá se klíčové slovo `MAX` nebo klíčové slovo `MIN` v závislosti na tom, zda se jedná o problém maximalizace nebo minimalizace. V lineárních optimalizačních úlohách jsou často obsaženy sumace, proto se používá sčítací funkce `SUM`. Zápis může vypadat takto:

```
MIN z1 EXPORT TO EXCELRANGE("PocetTypu") = sum(y);
```

Po cílové funkci je dalším krokem seznam omezení. Klíčové slovo `SUBJECT TO` se používá jako oddělovač mezi objektivní funkcí a omezeními. Příslušná omezení mohou být buď jednoduchá nebo indexovaná. V případě indexovaných proměnných

je třeba přiřadit jméno. Při určování vztahu se používají kombinace znaků větší, menší a rovná se. Pro příklad je uveden zápis indexované podmínky:

```
body[j]: sum(i:a[i,j]*y[i]) >= 1;
```

END

Klíčové slovo `END` je volitelné a označuje konec modelu.

2 Plánování svařoven ve ŠKODA Auto a.s.

V dnešním světě je kladen čím dál větší důraz na automatizaci a digitalizaci, automobilový průmysl nevyjímaje. V oddělení plánování svařoven ŠKODA AUTO a.s. se touto problematikou zabývají každý den, jelikož výrobní linky svařoven jsou až z 90% automatizované a neustále se hledají další potenciální místa k automatizaci. Nepřetržitě dochází k integraci nových zařízení do stávajících výrobních linek a k výstavbě zcela nových linek z důvodu změny produktu či navýšení kapacit. Tyto změny je nutné nejprve důkladně naplánovat a realizovat tak, aby byla výrobě předána odladěná výrobní linka, která bude schopna plnit požadavky dalších zákazníků. Oddělení plánování svařoven má na starost plánování všech svařoven, které spadají pod ŠKODA AUTO a.s., tudíž i ty, které jsou v zahraničí. Například závod v Jižním Novgorodu nebo závod v Pune v Indii. V rámci ČR se ve ŠKODA AUTO a.s. nacházejí čtyři svařovny, z nichž dvě jsou v Mladé Boleslavi a dvě v Kvasinách.

Pro samotné plánování veškerých změn ve svařovnách ŠKODA AUTO a.s. se využívá digitální továrna, která odráží reálný stav výrobních linek a umožňuje vytvořit koncepty nových linek ve virtuálním prostředí. Následující kapitola bude zaměřena na digitální továrnu v oddělení plánování svařoven.

2.1 Digitální továrna v plánování svařoven ve ŠKODA AUTO a.s.

Jak již bylo zmíněno výše, oddělení plánování svařoven ve ŠKODA AUTO a.s. využívá ke své práci digitální továrnu, která má jednu z hlavních rolí. Jak z názvu vyplývá, jedná se o digitální prostředí, ve kterém je možné vytvářet koncepty výrobních linek, zapracovávat změny do stávajících linek a vyměňovat data s dodavateli. Nástroj, který umožňuje všechny tyto funkce je od německé společnosti Siemens a jedná se o software Tecnomatix, konkrétně Process Designer a Process Simulate, které zajišťují komplexní portfolio digitálních výrobních řešení, která pomáhají digitalizovat výrobu a procesy s tím spojené. Software Tecnomatix zajišťuje synchronizaci mezi produktovým inženýrstvím, výrobním inženýrstvím, výrobními a servisními operacemi, a to s cílem maximalizovat efektivitu výroby a realizovat inovace.

2.1.1 Charakteristika činností plánování svařoven v digitální továrně

Než je možné začít se sériovou výrobou nového modelu auta, je zapotřebí naplánovat, jaké výrobní zdroje budou použity. Zjistit, zda je možné integrovat výrobu nového modelu auta do stávající výrobní linky a prověřit, jestli budou stačit výrobní kapacity a současný prostor v hale. Na výše zmíněné otázky umožňuje odpovědět digitální továrna a činnosti s ní spojené. Tato kapitola se bude věnovat hlavním činnostem, které se týkají digitální továrny v oddělení plánování svařoven ŠKODA AUTO a.s. a dále výběru svařovacích kleští v rámci výrobních linek.

Import a export dat

Import a export dat je v rámci digitální továrny nedílnou součástí práce s daty. Umožňuje pravidelnou výměnu dat s dodavateli, kteří vytvářejí nové koncepty výrobních linek nebo zapracovávají změny do současných linek. Import a export dat se provádí v pravidelných intervalech, přičemž po importu dat je zapotřebí jejich aktualizace ze strany plánování svařoven.

Správa knihoven

Součástí digitální továrny jsou knihovny, které obsahují seznamy výrobních prostředků s 3D reprezentací, například knihovna robotů nebo svařovacích kleští. Existují standardní knihovny poskytované koncernem Volkswagen, které by se měly primárně používat. V případě, že nelze použít standardní knihovny výrobních prostředků, lze využít rozšířených knihoven. Tyto rozšířené knihovny obsahují ty výrobní prostředky, které bylo nutné přidat do výroby. Před použitím určitého výrobního prostředku z knihovny je vhodné prověřit, zda je nadefinovaný podle standardu. Pokud tomu tak není, je nutné tuto chybu opravit podle stanovených standardů. Například se může jednat o chybně nadefinovanou kinematiku svařovacích kleští, kterou je zapotřebí ručně opravit v softwaru digitální továrny.

Správa produktových dat

Další součástí digitální továrny jsou produktová data a spojovací body. Produktová data zajišťuje technický vývoj, který je vždy musí dodat do stanoveného termínu na oddělení plánování svařoven. Produktová data jsou součástí tzv. kusovníku, který tvoří skupiny a podskupiny dílů v předem stanovené struktuře. V rámci digitální továrny je zapotřebí zkontrolovat strukturu kusovníku a zda obsahuje všechny díly.

Společně s produktovými daty jsou dodávány spojovací body. Například svařovací body, navařovací čepy, lepidla a laserové spoje. V digitální továrně jsou tyto spoje ve speciální knihovně, kde jsou rozděleny do jednotlivých podskupin.

Rovnění 3D dat podle reálného skenu výrobní linky

Rovnění 3D dat podle reálného skenu výrobní linky je stěžejní pro ostatní činnosti prováděné ve virtuálním modelu. V současnosti je situace taková, že není možné zajistit shodnost reálného prostředí s virtuálním 3D modelem. Je to zejména v důsledku velkého množství dodavatelů a lidí, kteří jsou zahrnuti do procesu výstavby svařovacích linek. Není tedy zatím možné zajistit, aby byla daná zařízení přesně umístěna na pozici, která byla naplánována ve virtuálním prostředí, tedy v digitální továrně. Vzhledem k těmto okolnostem je zapotřebí provádět skenování všech stávajících svařovacích linek a získat tak *point cloud* (mračno bodů), které je následně vloženo do virtuálního prostředí. V případě že jsou 3D modely výrobních zařízení ve virtuálním prostředí umístěny mimo pozice v reálném prostředí, jež je reprezentováno mračnem bodů, je nutné každé výrobní zařízení přemístit do polohy odpovídající realitě. V příloze č. 1 lze vidět 3D model výrobní linky a mračno bodů, které představuje realitu.

Tato činnost je důležitá z hlediska dalšího využití 3D dat. Zejména pak při off-line programování robotů, ověřování dosažitelnosti výrobních zařízení, kolizní analýze, vyrobitelnosti produktu a v neposlední řadě při integraci do stávajících výrobních linek.

Kolizní analýza

Pomocí funkce v softwaru Process Simulate lze provádět kolizní analýzu. Kolizní analýza se používá zejména při integraci nového zařízení do stávající linky nebo když se provádí průjezd linkou nového modelu auta. Analýza spočívá ve vytvoření drah, po kterých se pohybuje daný robot s nářadím nebo se může jednat o dráhu, po které se pohybuje karoserie. Poté se určuje skupina výrobních zdrojů nebo produktových dat, které se mají kontrolovat v rámci kolizní analýzy. Posledním krokem je samotné spuštění simulace pohybu. V případě, že software Process Simulate zaznamená kolizi, dojde k vizuálnímu upozornění. Na základě této analýzy se upravují dráhy robotů tak, aby nedocházelo ke kolizi nebo se nahrazují výrobní prostředky (svařovací kleště za jiné svařovací kleště).

Kontrola simulací

Tato kontrola se provádí zhruba jednou za dva měsíce, a to v době, kdy dodavatelé předávají data oddělení plánování svařoven, aby se zkontroloval stav 3D dat a jejich simulace. V rámci simulací se ověřuje: struktura operací, kinematika zařízení, připnutí kabeláže, definice náradí a typ výrobních operací. Tato činnost má za cíl zabezpečit správnost dat a s tím související simulace, které lze následně použít při integraci do již používaných výrobních linek.

Analýza *Geometric Gun Search*

Analýza *Geometric Gun Search* je speciální funkcí pro výběr svařovacích kleští v softwaru Process Simulate. Tato funkce vyžaduje příslušnou knihovnu svařovacích kleští nebo jeden konkrétní typ svařovacích kleští a produktová data, tedy kusovník. Pro ověření dostupnosti kleští je nutné vytvořit svařovací operaci, pod kterou se vloží dané svařovací body. Nástroj *Geometric Gun Search* poté všechny vybrané kleště postupně nasazuje kolmo na plech v místě, kde se vyskytuje svařovací bod. Výstupem je Obr. 1 s názvem *Gun search Report*, který lze exportovat do Ms Excel a následně analyzovat a dále s ním pracovat.

Guns	svar1	svar2	svar3	svar4	svar5	svar6	svar7	svar8	svar9	svar10	svar11	svar12
gun 1	+	+	-	+	+	-	-	-	+	-	+	-
gun 2	+	+	-	+	+	-	-	-	+	-	+	-
gun 3	-	-	-	-	-	-	-	-	-	-	-	-
gun 4	+	+	+	+	+	-	-	+	+	-	+	-
gun 5	+	+	-	+	+	-	-	-	+	-	+	-
gun 6	-	-	-	-	-	-	-	-	+	-	-	+
gun 7	-	-	-	-	-	-	-	-	+	-	+	-

Zdroj: Interní dokumentace ŠKODA AUTO a.s. (2019)

Obr. 1 Gun search Report

Ověření výrobitelnosti

Software Tecnomatix konkrétně Process Simulate umožňuje provádět nejrůznější analýzy, mezi které patří i prověření výrobitelnosti. Základem pro tuto činnost jsou produktová data, která poskytuje oddělení technického vývoje. Mezi produktová data v rámci svařoven patří kompletní okovaná karoserie, jejíž součástí jsou spojovací body, mezi které patří svařovací body, lepení a čepy. Pro ověření výrobitelnosti jsou právě spojovací body stěžejní. Dále je zapotřebí mít 3D data výrobních zařízení, mezi které v rámci svařoven patří: roboty, graifery, lepicí pistole, svařovací zařízení (kleště, laserové zařízení, mig mag) a zařízení na rolnové lemování (viz Příloha č. 2).

Nejčastěji se provádí ověření výrobitelnosti pomocí svařovacích kleští. Nejprve je nutné vytvořit studii, do které se přidá konkrétní model úseku linky, která v sobě zahrnuje jednotlivé stanice. Dále je zapotřebí do studie přidat produktová data a svařovací body, které se mají svařovat v jednotlivých stanicích. Jakmile jsou data připravena, lze pomocí nástrojů v softwaru Process Simulate ověřit výrobitelnost, v tomto případě svařitelnost konkrétních bodů na karoserii. Při ověření svařitelnosti ověřujeme, zda lze svařovací kleště nasadit kolmo na svařovací bod. To znamená, že kleště nebudou v kolizi s okolními plechy nebo výrobními zařízeními. Ověřuje se, jak otevřený, tak zavřený stav svařovacích kleští (viz Příloha č. 3). V Příloze č. 3 lze vidět ověření svařitelnosti čtyř svařovacích bodů v oblasti zadního světlometu karoserie současného modelu krátké Fabie. V případě, že dané kleště nelze použít na daný svařovací bod, je nezbytné vybrat jiné svařovací kleště, které budou schopny bezkolizně svařit daný svařovací bod. K tomu slouží funkce v Process Simulate, která byla popsána v předešlé kapitole. Pokud není nalezen jiný vhodný druh svařovacích kleští, je nutné poslat produktová data dodavateli svařovacích kleští, aby zkonstruoval nový speciální druh kleští. Tato varianta je však ojedinělá a ve většině případů lze použít kleště buď ze standardní knihovny svařovacích kleští nebo z rozšířené knihovny svařovacích kleští.

Součástí ověření výrobitelnosti je dále ověření dosažitelnosti robota, na kterém jsou namontovány dané svařovací kleště. Ověřujeme, zda robot s kleštěmi dosáhne souřadným systémem kleští do svařovací pozice. Souřadný systém k tomu určený je podle standardu označen TCP.

3 Výběr svařovacích kleští

Svařovací kleště samotné jsou jedním z hlavních nástrojů ve svařovnách ŠKODA AUTO a.s. a je tedy nutné brát v potaz mnoho okolností. V současnosti je ve ŠA používáno více než 100 druhů svařovacích kleští, které se liší způsobem ovládání, velikostí, konstrukcí a v neposlední řadě pořizovací cenou.

3.1 Konstrukční provedení svařovacích kleští

Svařovací kleště lze rozdělit,

a) podle způsobu ovládání na:

- Pneumatické

Tento typ kleští se vyznačuje tím, že je otevírání a zavírání kleští prováděno pneumatickým válcem. Kleště mají dvě základní polohy: poloha plného otevření a přivření. V porovnání se servokleštěmi je rychlost otevírání a zavírání o něco nižší. Výhodou je však nižší cena (FESTO AG & CO. KG, 2010).

- Servokleště

Servokleště lze rozdělit na servopneumatické a servoelektrické, vzhledem k použitému servomechanismu. Mezi hlavní výhody těchto svařovacích kleští patří rychlost otevírání a zavírání kleští a možnost nastavení síly sevření, zvláště pro jednotlivé svary. Díky svým vlastnostem, servokleště v současné době nahrazují kleště pneumatické. (NIMAK GmbH, 2007).

b) dle způsobu upevnění na:

- Pohyblivé kleště

Pohyblivé znamená, že jsou svařovací kleště namontovány například na robota a svařovaný díl je upnutý v nepohyblivém přípravku. S kleštěmi pohybujeme do pozice sváření.

- Stacionární kleště

Stacionární kleště jsou pevné a do pozice svaru manipulujeme se svařovaným dílem. To znamená, že kleště jsou pevně připěvněny k podlaze a svařovaný díl je upnutý v přípravku, který je nesený manipulujícím robotem.

c) dle konstrukce čelistí na:

- X kleště

Svařovací kleště do tvaru X (viz Příloha č. 4) nebo také nůžková konstrukce kleští se vyznačuje dobrým dosahem svarů v rovinách kolmých ke svářeči. V případě ručního svařování je možné při vhodném natáčení součásti a změně pozice obsluhy svařovat většinu svarů. Hlavní nevýhodou těchto kleští je menší rychlost otevírání a zavírání kleští v porovnání s C kleštěmi. U kleští tvaru X je třeba dbát větší opatrnosti, protože jedno z ramen kleští je pohyblivé a při otevření kleští dochází ke zvětšení rozměru kleští.

- C kleště

Konstrukce C kleští připomíná stolní svěrku (viz Příloha č. 5). Tento typ svařovacích kleští je využíván především pro svary v rovině kolineární s rovinou svářeče. Disponují vysokou rychlostí otevírání a zavírání kleští a vyšší bezpečností při manipulaci.

3.2 Tvorba vstupní matice a analýza dat pro výběr svařovacích kleští

V předchozí kapitole byly popsány svařovací kleště z hlediska způsobu ovládnání a konstrukce čelistí. A nyní bude popsána tvorba vstupní matice pro výběr svařovacích kleští a analýza dat.

K vytvoření vstupní matice pro výběr svařovacích kleští je nutné připravit data v digitální továrně. Nejprve se zvolí část výrobní linky, která bude podrobena analýze. Například se může jednat o část linky v Kvasinách s názvem Aufbau 1 (viz Příloha č. 6).

Dále se v softwaru Process Designer vytvoří studie, ve které budou navázány všechny svařovací operace, které se v dané svařovací lince provádějí. Poté je nutné zkontrolovat, zda jsou na svařovací operace navázány všechny svařovací body. Spustíme připravenou studii v softwaru process Simulate a zvolíme funkci *Geometric Gun Search*, kde ponecháme standardní nastavení a vybereme knihovnu kleští. Nejprve volíme standardní knihovnu svařovacích kleští, a v případě, že nebudou pokryty všechny svařovací body, zvolíme rozšířenou knihovnu svařovacích kleští. Samotná analýza je automatický proces, který trvá určitou dobu,

a to v závislosti na počtu svařovacích bodů a počtu svařovacích kleští. Výstupem je matice, která je ve zjednodušené podobě znázorněna na Obr. 8. Kompletní a reálná výstupní matice je přiložena v elektronické podobě.

	Svar1	Svar2	Svar3	Svar4	Svar5	Svar6
Svařovací kleště 1	✓	✗	✗	✓	✗	✓
Svařovací kleště 2	✗	✓	✓	✗	✗	✗
Svařovací kleště 3	✗	✓	✗	✗	✗	✗
Svařovací kleště 4	✗	✓	✗	✓	✗	✗
Svařovací kleště 5	✗	✗	✓	✗	✗	✗

Zdroj: Vlastní zdroj (2019)

Obr. 2 Výstupní zjednodušená matice z analýzy Geometric Gun Search

Výstupní matice ze softwaru Process Simulate je tvořena svařovacími kleštěmi v řádcích a svařovacími body ve sloupcích. Pakliže jsou svařovací kleště schopny svařit daný svařovací bod, objeví se v tabulce na Obr. 8 znak zelené barvy ve tvaru fajfky. V případě, že tomu tak není, objeví se v tabulce na Obr. 8 červený křížek. Z tabulky na Obr. 8 lze tedy rychle zjistit, které svařovací body jsou svařitelné, kterými svařovacími kleštěmi. Vzhledem k množství svařovacích bodů a svařovacích kleští je vhodné tabulku exportovat do softwaru MS Excel a dále s ní pracovat v tomto programu.

V MS Excel lze jednoduše analyzovat data pro potřebu výběru svařovacích kleští. Pro příklad budou použity všechny svařovací body z linky Aufbau 1 v Kvasinách, kde se svařují SUV vozy značky ŠKODA a SEAT, konkrétně ŠKODA Karoq, Kodiaq a SEAT Ateca. V Tab. 2 budou uvedeny základní informace o reálných datech, které byly použity pro tuto práci.

Tab. 2 Základní informace o reálných datech

Celkový počet druhů svařovacích kleští ve svařovně v Kvasinách	107
Počet druhů standardních svařovacích kleští	48
Počet druhů svařovacích kleští z rozšířené knihovny	59
Počet svařovacích bodů v lince Aufbau 1 v Kvasinách	729
Současný počet používaných druhů standardních svařovacích kleští v lince Aufba 1 v Kvasinách	6
Současný počet druhů používaných svařovacích kleští z rozšířené knihovny svařovacích kleští v lince Aufbau 1 v Kvasinách	8

Zdroj: Interní dokumentace ŠKODA AUTO a.s. (2019)

V první analýze *Geometric Gun Search* byla použita standardní knihovna kleští a získána výstupní matice. Dále bylo nutné tuto matici vyexportovat do MS Excel a zjistit, zda byly pokryty všechny svařovací body standardními kleštěmi. V každém sloupci bylo zjištěno maximum, čímž jsme získali „0/1“, kde 0 znamená, že bod nelze svařit žádnými standardními kleštěmi a 1 znamená, že bod je svařitelný alespoň jedním druhem standardních kleští. Všechny červené křížky byly nahrazeny za nuly a všechny zelené fajfky za jedničky (viz Tab. 3). Kompletní a reálná data z analýzy jsou opět pro svoji rozsáhlost přiložena v elektronické podobě.

Tab. 3 Upravená zjednodušená výstupní matice z analýzy *geometric gun search*

	Svar1	Svar2	Svar3	Svar4	Svar5	Svar6	
Svařovací kleště 1	1	0	0	1	0	1	
Svařovací kleště 2	0	1	1	0	0	0	
Svařovací kleště 3	0	1	0	0	0	0	
Svařovací kleště 4	0	1	0	1	0	0	
Svařovací kleště 5	0	0	1	0	0	0	
Funkce max	1	1	1	1	0	1	5

Zdroj: Vlastní zdroj (2019)

Po použití funkce *max* v MS Excel bylo zjištěno, že ve zjednodušené výstupní matici bylo pokryto pouze 5 svařovacích bodů z celkových 6 svařovacích bodů. Stejný postup byl aplikován na reálná data a bylo zjištěno, že standardní svařovací kleště pokryly 686 svařovacích bodů z celkových 729 bodů. Zbýlých 43 svařovacích bodů nebylo standardními svařovacími kleštěmi pokryto, a proto bylo nutné provést analýzu *Geometric Gun Search* pro těchto 43 svařovacích bodů znovu, s použitím rozšířené knihovny svařovacích kleští. Po této druhé analýze bylo zjištěno, že zbýlých 43 svařovacích bodů lze za pomoci rozšířené knihovny svařovacích kleští svařit.

V současné době je ve svařovací lince Aufbau 1 v Kvasinách svařováno 729 svařovacích bodů pomocí 6 druhů standardních svařovacích kleští a 8 druhů svařovacích kleští z rozšířené knihovny (viz Tab. 2).

Cílem práce je zredukovat počet druhů svařovacích kleští použitých ve svařovací lince Aufbau 1, a to jak ze standardní, tak z rozšířené knihovny svařovacích kleští. Tomuto cíli budou věnovány následující kapitoly práce.

3.2.1 Úprava vstupních dat v MS Excel

Pro aplikaci konkrétního matematického modelu je nutné upravit vstupní data v MS Excel. První úprava výstupní matice z analýzy *Geometric Gun Search* spočívá v nahrazení znaků červený křížek a znaků zelená fajfka za nuly a jedničky, jak bylo popsáno v předešlé kapitole. Dalším krokem je přidat na začátek tabulky sloupec s pořadím jednotlivých kleští a řádek s pořadím jednotlivých svařovacích bodů. A na konec tabulky přidat sloupce s náklady, počtem kleští a potenciálním nákupem („koupit“). Dále je nutné definovat oblasti, na které se bude odkazovat matematický model v softwaru MPL for Windows (viz Tab. 4). V Tab. 4 nebude v levém sloupci použita diakritika, protože nemůže být použita ani v MPL for Windows, aby bylo zajištěno propojení s definovanými oblastmi v softwaru MS Excel.

Tab. 4 Definované oblasti pro matematický model

Název oblasti	Vysvětlení oblasti
kleste	Pořadí jednotlivých svařovacích kleští v tabulce
body	Pořadí jednotlivých svařovacích bodů v tabulce
umi	Matice s 0 a 1, kde 0 = neumí svařit, 1 = umí svařit
pocet	Celočíselná proměnná x
koupit	Binární proměnná y s 0 nebo 1, kde 0 = nekoupit, 1 = koupit
PocetTypu	Suma vybraných druhů svařovacích kleští
Naklady	Cena jednotlivých kleští
cena	Suma vybraných kleští podle ceny

Zdroj: Vlastní zdroj (2019)

Posledním krokem úpravy vstupních dat je pojmenování souboru bez diakritiky, například *minimalizace_pocty_typu.xlsx*.

4 Navrhovaná řešení pro výběr svařovacích kleští

Jak již bylo zmíněno v předešlých kapitolách, výběr svařovacích kleští je stěžejním úkolem v rámci plánování svařoven. V této kapitole budou navržena dvě možná řešení, kterými lze zredukovat počet druhů svařovacích kleští. První metodou řešení bude výběr druhů svařovacích kleští pomocí optimalizačního matematického modelu s použitím softwaru MPL for Windows a druhou metodou bude výběr pomocí heuristických metod za použití programovacího jazyka VBA for Excel. První způsob řešení představuje exaktní metodu, která umožňuje získat optimální výsledky z hlediska počtu druhů svařovacích kleští. Naopak druhý způsob řešení představuje aproximační metodu, která poskytuje přípustná řešení, ale nezaručuje získání optimálního řešení při výběru počtu druhů svařovacích kleští.

4.1 Optimální výběr svařovacích kleští pomocí matematického modelu

Výběr svařovacích kleští pomocí lineárního programování umožňuje získat optimální řešení z hlediska počtu druhů svařovacích kleští. Při řešení tohoto problému bude aplikována úloha o pokrytí, jejíž teoretický základ je popsán v první kapitole. Úlohu o pokrytí lze definovat několika možnými způsoby. Konkrétně může jít o pokrytí svařovacích bodů určitým druhem svařovacích kleští. Z praxe vyplývá, že nelze svařit určitou množinu svařovacích bodů jedním typem kleští v rámci jedné linky, ale je zapotřebí použít více druhů kleští, aby byla pokryta množina všech svařovacích bodů.

Formulace: Necht' $M = \{1, 2, \dots, m\}$ je konečná množina typů svařovacích kleští a $N = \{1, 2, \dots, n\}$ konečná množina svařovacích bodů. Je dána matice A s hodnotami $a_{ij} = 1$, jestliže svařovací kleště i jsou schopny svařit svařovací bod j , $a_{ij} = 0$ jinak:

$$a_{ij} = \begin{cases} 1 & \text{jestliže } i\text{-tý typ kleští umí svařit} \\ & j\text{-tý svařovací bod,} \\ 0 & \text{jinak,} \end{cases} \quad \begin{matrix} i = 1, 2, \dots, m. \\ j = 1, 2, \dots, n. \end{matrix} \quad (27)$$

Cílem je pokrýt všechny svařovací body s minimálním počtem typů svařovacích kleští. K nalezení optimálního řešení provedeme obecnou formulaci matematického modelu.

Proměnné: V obecném matematickém modelu definujeme dva typy proměnných:

$$x_i - \text{počet použitých kleští } i\text{-tého typu,} \quad i = 1, 2, \dots, m, \quad (28)$$

$$y_i = \begin{cases} 1 & \text{jestliže se použije alespoň 1 kus kleští } i\text{-tého typu,} \\ 0 & \text{jinak,} \end{cases} \quad i = 1, 2, \dots, m. \quad (29)$$

Účelová funkce (30) představuje minimální počet druhů svařovacích kleští:

$$z = \sum_{j=1}^n y_j \rightarrow \min. \quad (30)$$

Omezující podmínky: Omezující podmínky (31) definují, že musí být pokryty všechny svařovací body:

$$\sum_{i=1}^m a_{ij} y_i \geq 1, \quad j = 1, 2, \dots, n \quad (31)$$

Omezující podmínky (32) mají dvojí význam. Pokud je proměnná y_i nulová, tj. nejsou vybrány kleště i -tého druhu, pak jejich počet je také nulový. Naopak, pokud jsou vybrány alespoň jedny kleště i -tého druhu, tj. proměnná $x_i > 0$, pak proměnná y_i musí být rovna 1. Symbol M představuje vysokou konstantu:

$$x_i \leq M y_i, \quad i = 1, 2, \dots, m \quad (32)$$

Nerovnice (33) definují, že pokud $y_i = 1$, tak x_i je alespoň 1.

$$y_i \leq x_i, \quad i = 1, 2, \dots, m \quad (33)$$

V případě, že bychom chtěli brát v potaz náklady na pořízení i -tého typu svařovacích kleští, je zapotřebí zadat parametr c_i , který představuje náklady na 1 kus svařovacích kleští i -tého typu. Účelová funkce pak bude vypadat následovně:

$$z = \sum_{j=1}^n c_j x_j \rightarrow \min. \quad (34)$$

Následně je nutné daný matematický model napsat v přesné syntaxi v programovacím softwaru MPL for Windows, jež byla popsána v části 1.4.1. Tato přesná syntaxe pro řešení problému pro případ výběru standardních svařovacích kleští bude uvedena v Příloze č. 7 a pro případ výběru speciálních svařovacích kleští v Příloze č. 8. Jediná odlišnost mezi zápisy syntaxe bude v odkazování na jiný zdrojový soubor.

Následně bylo získáno optimální řešení pomocí matematického modelu a bylo zjištěno, že pro svaření všech 729 svařovacích bodů v rámci linky Aufbau1 je zapotřebí pouze 5 druhů standardních svařovacích kleští a 3 druhy speciálních svařovacích kleští. Konkrétní druhy vybraných kleští lze nalézt v elektronické příloze s názvem "minimalizace_pocty_typu_standard_kleste.xlsx" a „minimalizace_pocty_typu_special_kleste.xlsx“.

Daný matematický model umožňuje minimalizovat počet druhů svařovacích kleští nebo minimalizovat celkové náklady. Je však možné, že se bude lišit počet druhů vybraných svařovacích kleští, v závislosti na použitém modelu. Z důvodu neověřitelnosti důvěrných informací o pořizovacích cenách svařovacích kleští nebudou uvedeny výsledky.

4.2 Výběr svařovacích kleští pomocí heuristické metody

Výběr svařovacích kleští pomocí heuristické metody umožňuje získat přípustné řešení, které může být optimální. Následující heuristická metoda je inspirována heuristikou od Vasko a kol. (2016) popsané v kapitole 1.2. Níže je uveden obecný matematický postup použité heuristiky.

Množina $N = \{1, 2, \dots, n\}$ je množina všech svařovacích bodů.

Množina $U \subseteq N$ je množina nepokrytých bodů.

Množina $M = \{1, 2, \dots, m\}$ je množina všech typů kleští.

Množina $V \subseteq M$ je množina typů kleští, které lze vybrat.

Krok 1.

Definujeme $U = N$ a $V = M$.

Krok 2.

Pro každý typ kleští z množiny V vypočteme hodnotu představující počet dosud nepokrytých bodů, které je možné tímto typem svařit:

$$d_i = \sum_{j \in U} a_{ij}, \quad i \in V. \quad (35)$$

Krok 3.

Najdeme maximum z hodnot vypočtených v kroku 2:

$$d_k = \max_{i \in V} d_i. \quad (36)$$

Krok 4.

Nechť množina $K \subseteq U$ je množina dosud nepokrytých bodů, které lze svařit typem kleští k určených v kroku 3. Upravíme množiny U a V :

$$U = U \setminus K. \quad (37)$$

$$V = V \setminus \{k\} \quad (38)$$

Pokud $U = \emptyset$, pak pokračujeme krokem 5, jinak pokračujeme krokem 2.

Krok 5.

Konec. Množina $M \setminus V$ obsahuje vybrané typy kleští.

K výpočtu přípustného řešení byl využit programovací jazyk VBA for Excel (viz kapitola 1.3). Konkrétní zápis syntaxe pro řešení daného problému pro případ výběru standardních svařovacích kleští bude uveden v Příloze č. 9 a pro případ výběru speciálních svařovacích kleští v Příloze č. 10. Odlišnostmi mezi zápisy syntaxí bude rozdílný počet druhů svařovacích kleští a počet svařovacích bodů.

Pomocí programovacího jazyka VBA for Excel bylo získáno přípustné řešení a bylo zjištěno, že pro svaření všech 729 svařovacích bodů v rámci linky Aufbau1 je potřeba 5 druhů standardních svařovacích kleští a 4 druhy speciálních svařovacích kleští. Konkrétní druhy vybraných kleští lze nalézt v elektronické příloze s názvem „minimalizace_pocty_tupu_standard_kleste.xlsx“ a „minimalizace_pocty_tupu_special_kleste.xlsx“.

Klady a zápory navrhovaných řešení

V Tab. 5 jsou uvedeny klady a zápory řešení pomocí matematického modelování a pomocí heuristické metody.

Tab. 5 Klady a zápory navrhovaných řešení

Matematický model	Heuristika
Klady:	Klady:
<ul style="list-style-type: none"> • Získání optimálního řešení • Jednoduchá syntaxe v MPL • Univerzální použití modelu 	<ul style="list-style-type: none"> • Rychlost výpočtu • Univerzální použití metody • Neplacený software, součástí MS Office
Zápory:	Zápory:
<ul style="list-style-type: none"> • Placený software • Nutnost znalosti matematického modelování 	<ul style="list-style-type: none"> • Složitější zápis syntaxe • Získání pouze přípustného řešení • Nutnost znalosti programovacího jazyka VBA

Zdroj: Vlastní zdroj (2019)

4.3 Porovnání výsledků navrhovaných řešení

Výsledky, které byly získány z navrhovaných řešení, budou porovnány se současnou situací na lince Aufbau 1 v Tab. 6.

Tab. 6 Porovnání výsledků navrhovaných řešení se současnou situací na lince Aufbau 1

Řešení	Počet druhů standardních svařovacích kleští	Počet druhů speciálních svařovacích kleští
Současné	8	6
Pomocí matematického modelu	5	3
Pomocí heuristiky	5	4

Zdroj: Vlastní zdroj (2019)

Cílem obou řešení bylo minimalizovat počet druhů svařovacích kleští tak, aby bylo pokryto všech 729 svařovacích bodů v lince Aufbau 1. Z Tab. 6 vyplývá, že nejlepší

a zároveň optimální řešení nabízí použití matematického modelu, které zredukovalo počet druhů standardních svařovacích kleští o 3 druhy kleští a speciální svařovací kleště také o 3 druhy kleští. Řešení pomocí heuristiky pak nabízí přípustné řešení blízké optimálnímu řešení. U obou navrhovaných řešení lze vidět podstatné zlepšení, co se týče počtu druhů svařovacích kleští. Dle standardu je nutné, aby každý druh svařovacích kleští měl dle počtu kusů na lince určitý počet náhradních svařovacích kleští, aby bylo možné rychle nahradit nefunkční kleště. V Tab. 7 je znázorněn počet náhradních svařovacích kleští vzhledem k počtu kusů použitých kleští od jednoho druhu podle standardu.

Tab. 7 Počet náhradních svařovacích kleští podle počtu druhů

Počet použitých kusů svařovacích kleští od jednoho druhu	Počet kusů náhradních svařovacích kleští
1 až 3	1
4 až 6	2
7 až 10	3

Zdroj: Interní dokumentace ŠKODA AUTO a.s. (2019)

Ve výrobní svařovací lince Aufbau 1, která byla analyzována v rámci této práce, bylo zjištěno, že je použito 6 druhů standardních svařovacích kleští a 8 druhů svařovacích kleští z rozšířené knihovny. V Tab. 8 je znázorněno, kolik kusů kleští od jednoho druhu je použito v lince Aufbau 1.

Tab. 8 Počet náhradních svařovacích kleští v lince Aufbau 1

Druhy svařovacích kleští	Počet použitých kleští (kus)	Počet náhradních kleští (kus)
Standardní knihovna kleští		
Kleště č. 13	2	1
Kleště č. 4	2	1
Kleště č. 6	2	1
Kleště č. 3	3	1
Kleště č. 1	4	2
Kleště č. 27	1	1
Kleště z rozšířené knihovny kleští		
Kleště č. 37	2	1
Kleště č. 36	3	1
Kleště č. 13	1	1
Kleště č. 14	1	1
Kleště č. 56	1	1
Kleště č. 57	1	1
Kleště č. 28	2	1
Kleště č. 44	1	1
Celkový počet	26	15

Zdroj: Interní dokumentace ŠKODA AUTO a.s. (2019)

V Tab. 8 jsou uvedeny konkrétní druhy vybraných svařovacích kleští společně s jejich potřebným množstvím v lince Aufbau 1 a tomu odpovídající množství náhradních kleští. V Tab. 9 jsou uvedeny výsledky po optimalizaci za použití matematického modelu a heuristiky.

Tab. 9 Počet náhradních svařovacích kleští v lince Aufbau 1 po optimalizaci

Druhy svařovacích kleští	Počet použitých kleští (kus)		Počet náhradních kleští (kus)	
	Matematický model	Heuristika	Matematický model	Heuristika
Standardní knihovna kleští				
Kleště č. 7	3	3	1	1
Kleště č. 17	3	3	1	1
Kleště č. 23	3	3	1	1
Kleště č. 35	3	3	1	1
Kleště č. 47	2	2	1	1
Kleště z rozšířené knihovny				
Kleště č. 4	3		1	
Kleště č. 22	6	3	2	1
Kleště č. 58	3	3	1	1
Kleště č. 14		2		1
Kleště č. 17		3		1
Celkový počet	26	26	9	9

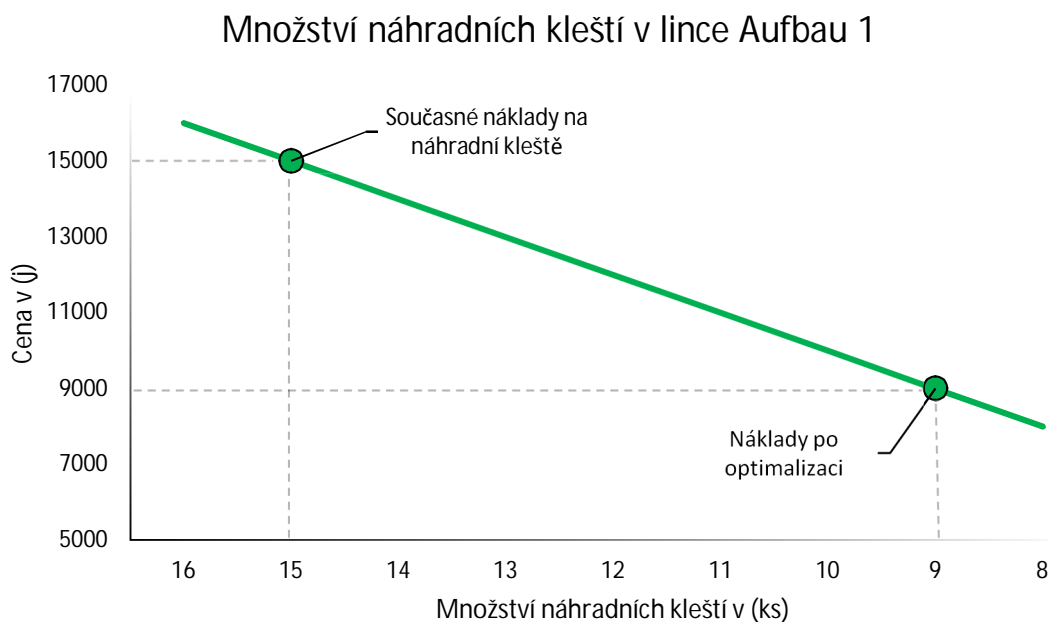
Zdroj: Vlastní zdroj (2019)

Obě navrhovaná řešení nabízí dostatečný důkaz o tom, že lze zredukovat počet druhů svařovacích kleští ve výrobní svařovací lince Aufbau 1, a s tím i související počet náhradních kleští tak, aby bylo pokryto všech 729 svařovacích bodů. Jak řešení pomocí lineárního programování, tak řešení pomocí heuristiky umožňuje zredukovat počet náhradních svařovacích kleští o 6 kusů. Z výsledků vyplývá, že počet druhů svařovacích kleští má přímý vliv na počet náhradních svařovacích kleští v lince.

4.3.1 Ekonomické zhodnocení navrhovaných řešení

Tato část bude zaměřena na finanční zhodnocení obou navrhovaných řešení. Z důvodu důvěrných informací, nebudou použity reálné ceny daných druhů svařovacích kleští, ale bude stanovena jednotná pořizovací cena 1000 j za 1 kus kleští. Dále je nutné zahrnout pořizovací cenu za software MPL for Windows

v případě lineárního programování, avšak v porovnání s reálnou finanční úsporou, kterou navrhované řešení přinese, je tato cena zanedbatelná. Na Obr. 9 je znázorněn graf, kde lze vidět výrazný pokles pořizovacích nákladů za náhradní svařovací kleště v důsledku redukce počtu druhů kleští.



Zdroj: Vlastní zdroj (2019)

Obr. 3 Pokles pořizovacích nákladů za náhradní kleště

V současnosti je zároveň kladen velký důraz na úsporu místa ve výrobních halách, a to v důsledku neustálého rozšiřování linek a navyšování kapacit výrobních zařízení. Díky redukci počtu kusů náhradních kleští, lze za každý kus ušetřit 1 m². V tomto případě se jedná o úsporu 6 m², které lze využít k jiným účelům.

Největším přínosem obou navrhovaných řešení je možnost aplikace na další výrobní linky ve svařovnách ŠKODA AUTO a.s. V rámci jedné svařovny je celkem 19 linek, které na sebe navazují. Každá z nich se liší počtem svařovacích kleští, počtem svařovacích bodů a zejména velikostí. Obě navrhovaná řešení lze aplikovat na všech 19 linek bez ohledu na velikost, počet svařovacích bodů a počet svařovacích kleští. Jde tedy o univerzální nástroje, které umožňují získat optimální řešení v případě lineárního programování a přípustné řešení v případě heuristiky.

Výše zmíněné výsledky poslouží jako podklady při rozhodovacích procesech v oddělení plánování svařoven ve ŠKODA AUTO a.s.

4.3.2 Doporučení

Obě navrhovaná řešení poskytují dostatečný důkaz o tom, že lze zredukovat počet druhů svařovacích kleští ve výrobní lince svařovny, a tím zredukovat i počet náhradních kleští. I když obě řešení poskytují dosti podobné výsledky, je nutné zmínit, že tomu tak nemusí být v další analýze jiné výrobní linky. Řešení pomocí heuristiky nabízí přípustné řešení, nikoliv však optimální, i když je optimálnímu řešení velmi blízko, jako tomu bylo v této práci. Naproti tomu řešení pomocí matematického modelu nabízí jednodušší univerzální model, který umožňuje získání optimálního řešení. Lineární programování v softwaru MPL for Windows nabízí komplexní řešení a je vhodné i pro jiné typy úloh, než je úloha o pokrytí.

Na základě výše zmíněných informací lze doporučit řešení úlohy o pokrytí pomocí matematického modelu s využitím placeného softwaru MPL for Windows, za účelem minimalizovat počet druhů svařovacích kleští ve výrobních linkách svařoven ŠKODA AUTO a.s. Nicméně je nutné zvážit přínos a výdaje za placený software MPL a placeného řešitele CPLEX.

Závěr

Tématem práce byly rozhodovací procesy ve svařovně ŠKODA AUTO a.s., týkající se výběru svařovacích kleští s cílem minimalizovat počet jejich druhů, a s tím související pořizovací náklady. Teoretickým východiskem byly modely a metody matematického programování, především matematický model úlohy o pokrytí. Dále se vycházelo z interní dokumentace a z vlastních zkušeností autora získaných ze současné pracovní pozice v oddělení plánování svařoven ŠKODA AUTO a.s. V praktické části práce byla provedena analýza získaných dat z interní dokumentace ŠKODA AUTO a.s. a data byla upravena pro aplikaci konkrétního matematického modelu a konkrétní heuristiky.

V teoretické části práce byly nejprve popsány konkrétní příklady úlohy lineárního programování, se zaměřením na úlohu o pokrytí, jež byla předmětem mnoha prací zmíněných autorů, a která je stěžejní pro praktickou část této práce. Následně bylo popsáno řešení pomocí matematického modelu úlohy o pokrytí. Dále se tato část práce zabývala řešením pomocí heuristické metody. V návaznosti na již zmíněné možnosti řešení byl popsán programovací software MPL for Windows a nástroj Visual Basic for Windows, který je součástí MS Office. V další kapitole bylo představeno a popsáno oddělení plánování svařoven ve ŠKODA AUTO a.s., se zaměřením na digitální továrnu a popsány jednotlivé činnosti s tím spojené, zejména výběr svařovacích kleští. Pro praktickou část práce byla použita data z výrobní svařovací linky Aufbau 1 v závodě v Kvasinách.

V praktické části práce byla nejprve analyzována výstupní data z nástroje *Geometric Gun Search* pro výběr svařovacích kleští a pomocí softwaru MS Excel vytvořena vstupní matice. Tato matice byla následně použita v rámci navrhovaných řešení pomocí matematického modelu a heuristiky. V další kapitole byl popsán konkrétní matematický model, jak pro lineární programování, tak pro heuristiku. Dále byly aplikovány oba matematické modely na připravenou vstupní matici, jež je přiložena v elektronické podobě a následně získány výsledky. Na základě získaných výsledků bylo provedeno porovnání se současnou situací na lince Aufbau 1 a popsány hlavní přínosy obou řešení.

V závěru praktické části této práce bylo provedeno finanční vyhodnocení přínosu obou řešení a uvedeny další možnosti aplikace navrhovaných řešení v rámci

svařoven ŠKODA AUTO a.s. s tím, že získané výsledky této práce poslouží jako podklady při rozhodovacích procesech v oddělení plánování svařoven. Poslední kapitola je věnována doporučení jednoho z navrhovaných řešení za účelem redukce počtu druhů svařovacích kleští ve výrobních linkách svařoven.

Seznam literatury

CRAWFORD, B. a kol. Application of the Artificial Bee Colony Algorithm for Solving the Set Covering Problem. *The Scientific World Journal* [online]. 2014. [cit. 19. 4. 2019] Dostupné z URL:<<http://www.hindawi.com/journals/tswj/2014/189164/>>

DEMEL, J. Operační výzkum [online]. Praha, 2011 [cit. 2019-04-30]. Dostupné z URL:<<https://kix.fsv.cvut.cz/~demel/ped/ov/ov110215.pdf>. ČVUT>

EISELT, H, SANDBLOM, C.-L. SANDBLOM. *Integer programming and network models*. New York: Springer, 2000. ISBN 35-406-7191-9.

FÁBRY, J. Pokročilé matematické modely a metody. [online]. 2017. [cit. 19. 4. 2019] Dostupné z URL:<<http://nb.vse.cz/~fabry/POV-prezentace.pdf>>

FESTO AG & CO. KG. Product Overview for the Automotive Industry [online]. Esslingen, 2010. [cit. 15. 4. 2019] Dostupné z URL:<http://www.festo.com/net/SupportPortal/Files/17309/AMI_ProductOverview_2012b_EN_low.pdf>

GOMES, F., C. a kol. *Experimental Analysis of Approximation Algorithms for the Vertex Cover and Set Covering Problems* [online]. 2006. [cit. 29. 4. 2019] Dostupné z URL:<<https://linkinghub.elsevier.com/retrieve/pii/S0305054805001255>>

Interní dokumentace ŠKODA AUTO a.s. 2019

JABLONSKÝ, J. *Operační výzkum*. 2. vyd. Praha: Vysoká škola ekonomická, 1998. ISBN 80-7079-597-2.

JABLONSKÝ, J. *Programy pro matematické modelování*. Praha: Oeconomica, 2007. ISBN 978-80-245-1178-8.

KUO, Y. -H., LEUNG, J., M., Y. On the mixed set covering, packing and partitioning polytope. *Discrete Optimization* [online]. 2016. [cit. 29. 4. 2019] Dostupné z URL:<<http://linkinghub.elsevier.com/retrieve/pii/S1572528616300366>>

LUTTER, P. a kol. Improved handling of uncertainty and robustness in set covering problems. *European Journal of Operational Research* [online]. 2017. [cit. 29. 4. 2019] Dostupné z URL:<<https://linkinghub.elsevier.com/retrieve/pii/S0377221717303910>>

MAŇÁK, J., ŠVEC, V. *Výukové metody*. Brno: Paido, 2003. ISBN isbn8073150395.

MAXIMAL SOFTWARE Maximal Software-Optimization Modeling [online]. ©2019 [cit. 16. 4. 2019] Dostupné z URL:<<http://www.maximalsoftware.com/maximal/>>

NIMAK GmbH. NIMAK Servo Technology: for effortless welding [online]. Wissen, 2007 [cit. 20. 4. 2019] Dostupné z URL:<http://www.nimak.de/Bilder/NIMAK_Servo_e.pdf>

VASKO, F., J., LU, Y., ZYMA, K. What is the best greedy-like heuristic for the weighted set covering problem?. Operations Research Letters [online]. 2016. [cit. 23. 4. 2019]. Dostupné z URL:<<https://linkinghub.elsevier.com/retrieve/pii/S016763771600047X>>

WALKENBACH, J. Excel VBA programming for dummies. 3rd ed. Chichester: John Wiley [distributor], 2013. ISBN 978-1-118-49037-2.

WOLSEY, L. *Integer Programming*. U.S.: Wiley-Interscience, 1998. 264 s. ISBN 978-0-471-28366-9.

Seznam obrázků a tabulek

Seznam obrázků

Obr. 1 Gun search Report	26
Obr. 2 Výstupní zjednodušená matice z analýzy Geometric Gun Search	30
Obr. 3 Pokles pořizovacích nákladů za náhradní kleště	42

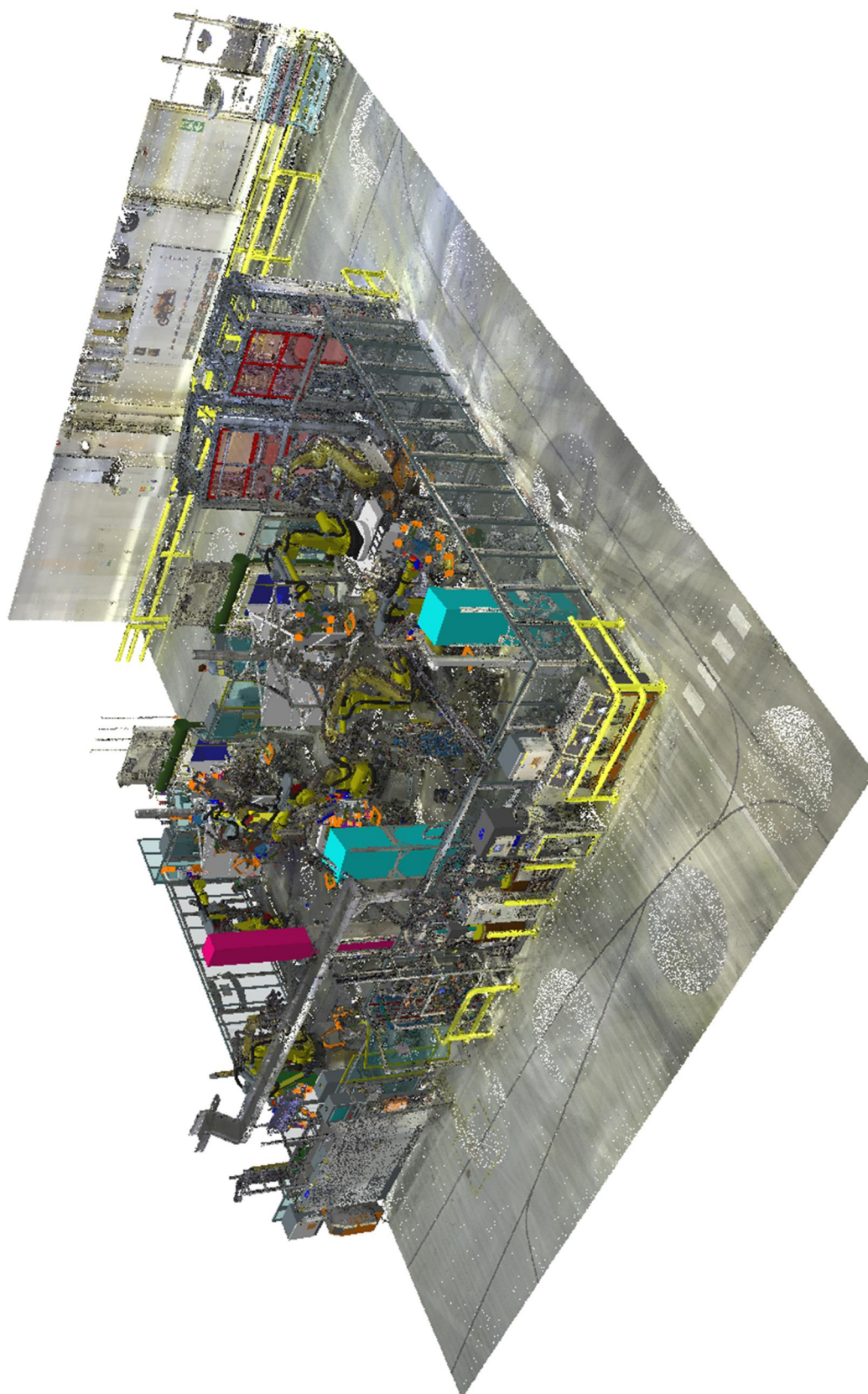
Seznam tabulek

Tab. 1 Vstupní data pro příklad 3	15
Tab. 2 Základní informace o reálných datech.....	31
Tab. 3 Upravená zjednodušená výstupní matice z analýzy geometric gun search 31	
Tab. 4 Definované oblasti pro matematický model.....	33
Tab. 5 Klady a zápory navrhovaných řešení	38
Tab. 6 Porovnání výsledků navrhovaných řešení se současnou situací na lince Aufbau 1	38
Tab. 7 Počet náhradních svařovacích kleští podle počtu druhů	39
Tab. 8 Počet náhradních svařovacích kleští v lince Aufbau 1	40
Tab. 9 Počet náhradních svařovacích kleští v lince Aufbau 1 po optimalizaci.....	41

Seznam příloh

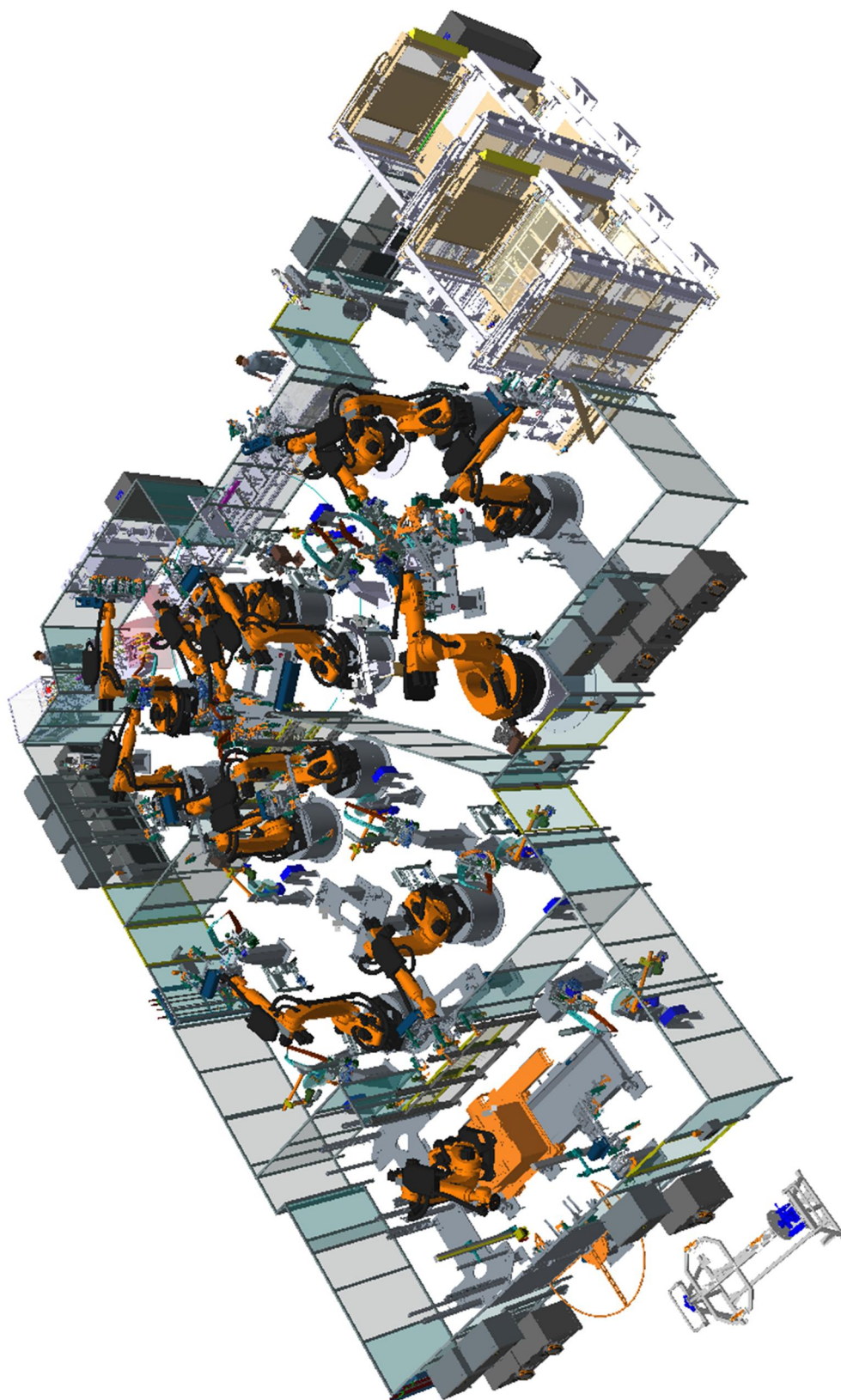
Příloha č. 1 3D model výrobní linky s mračnem bodů	50
Příloha č. 2 3D model výrobní linky ve svařovně.....	51
Příloha č. 3 Ověření svařitelnosti svařovacími kleštěmi.....	52
Příloha č. 4 X kleště	53
Příloha č. 5 C kleště	54
Příloha č. 6 Svařovací linka Aufbau 1.....	55
Příloha č. 7 Zápis pro standardní knihovnu kleští v MPL.....	56
Příloha č. 8 Zápis pro rozšířenou knihovnu kleští v MPL.....	57
Příloha č. 9 Zápis pro standardní knihovnu kleští ve VBA.....	58
Příloha č. 10 Zápis pro rozšířenou knihovnu kleští ve VBA.....	60

Příloha č. 1 3D model výrobní linky s mračnem bodů



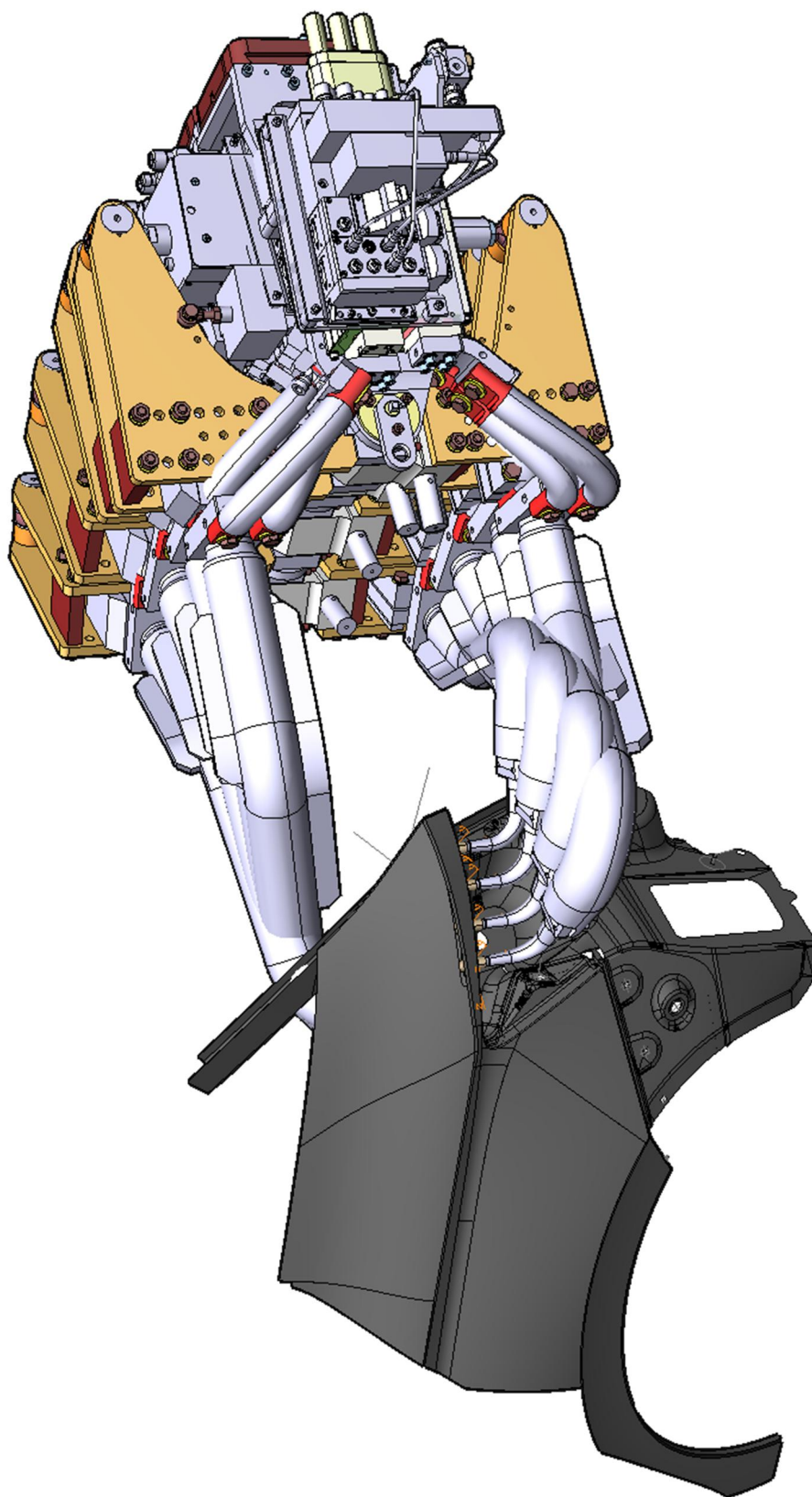
Zdroj: Interní dokumentace ŠKODA AUTO a.s. (2019)

Příloha č. 2 3D model výrobní linky ve svařovně



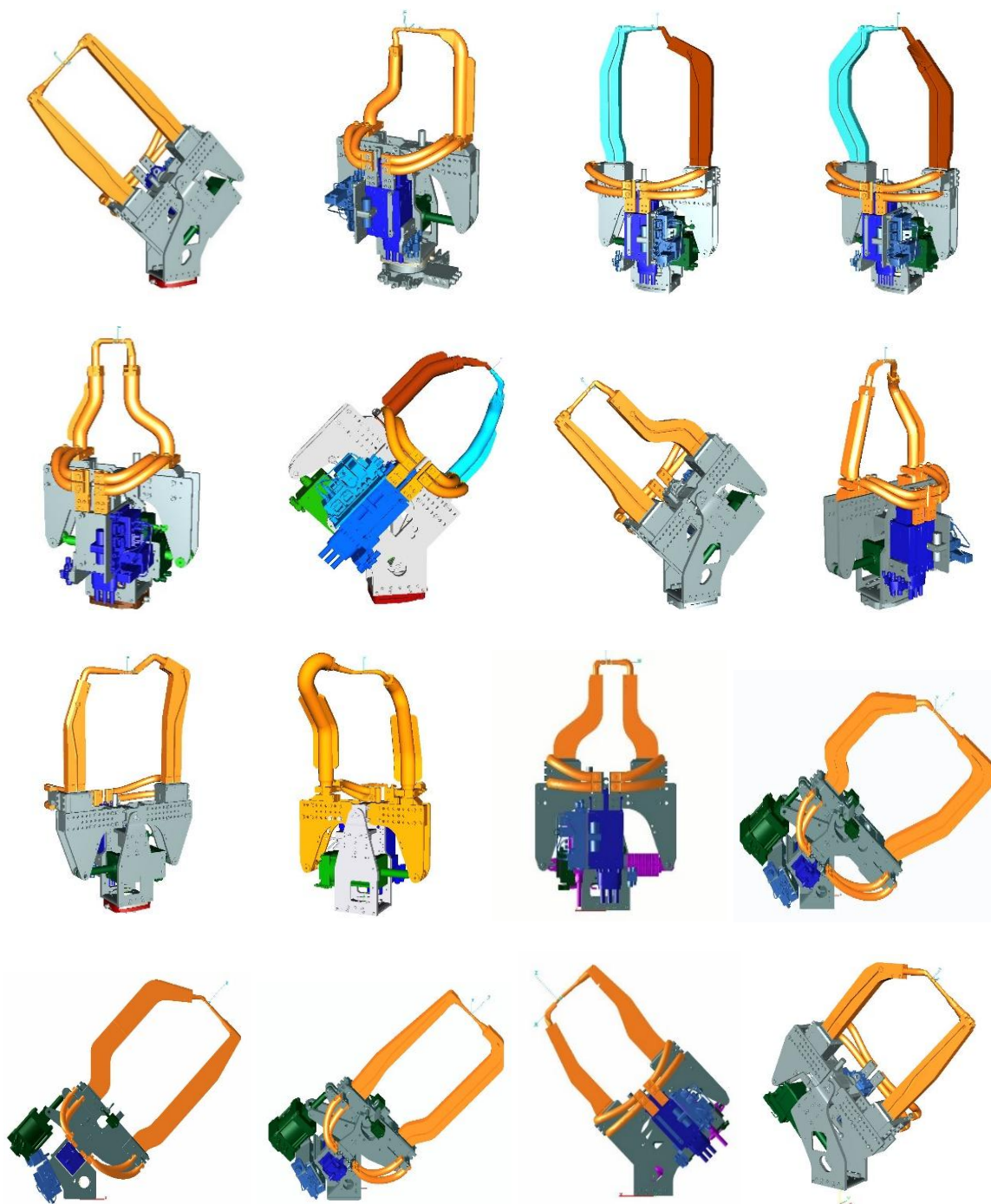
Zdroj: Interní dokumentace ŠKODA AUTO a.s. (2019)

Příloha č. 3 Ověření svařitelnosti svařovacími kleštěmi



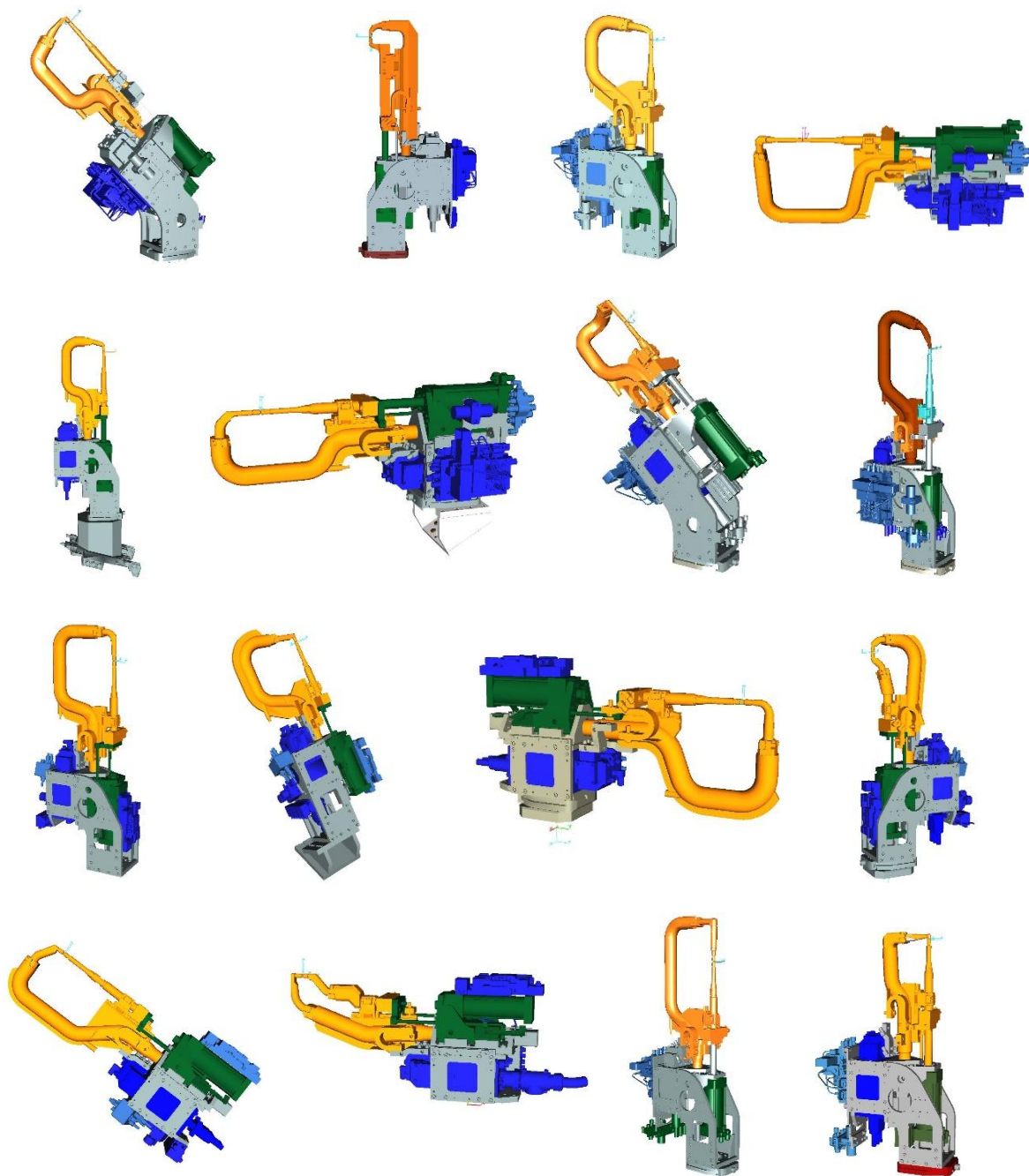
Zdroj: Interní dokumentace ŠKODA AUTO a.s. (2019)

Příloha č. 4 X kleště



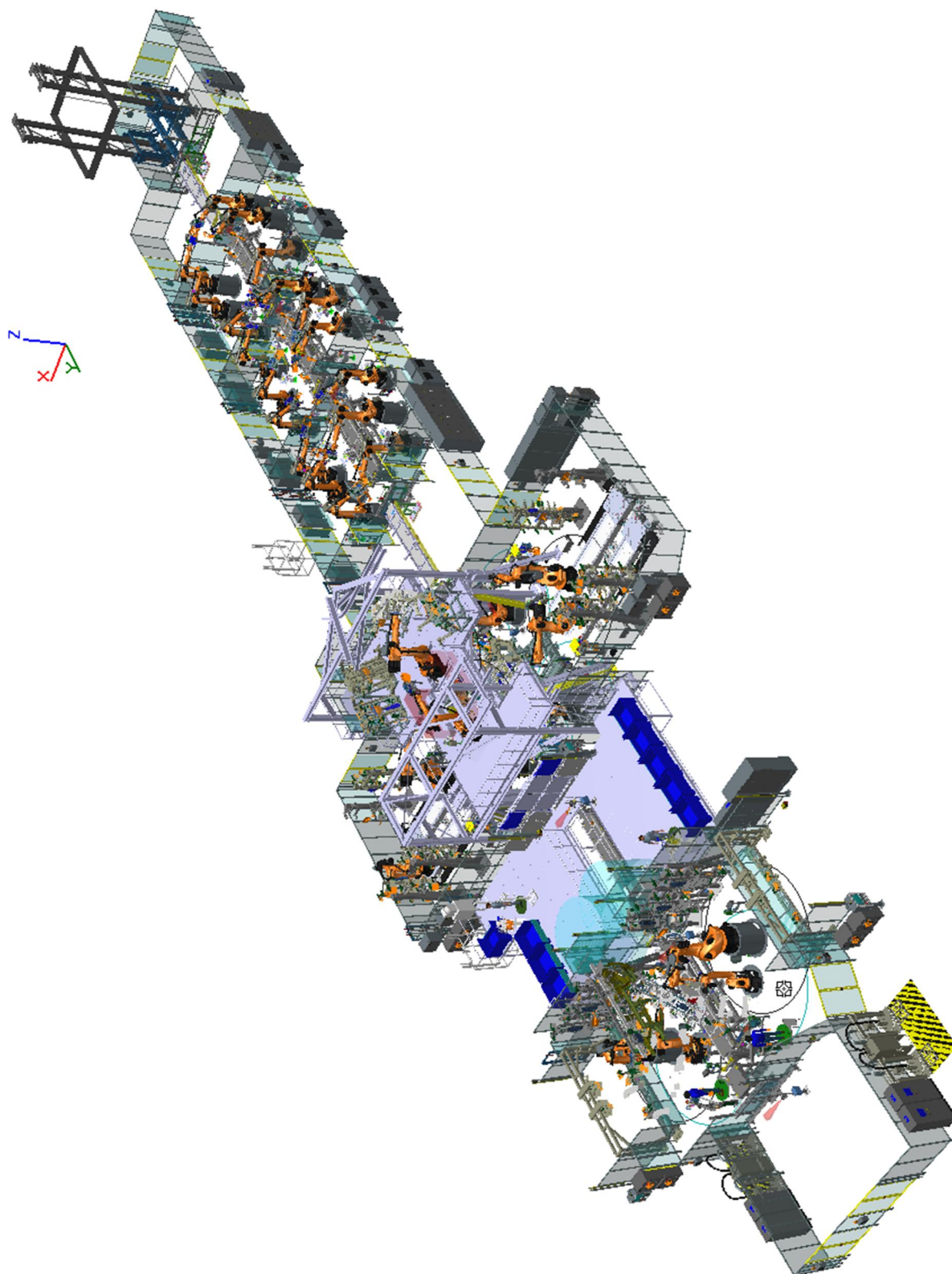
Zdroj: Interní dokumentace ŠKODA AUTO a.s. (2019)

Příloha č. 5 C kleště



Zdroj: Interní dokumentace ŠKODA AUTO a.s. (2019)

Příloha č. 6 Svařovací linka Aufbau 1



Zdroj: Interní dokumentace ŠKODA AUTO a.s. (2019)

Příloha č. 7 Zápis pro standardní knihovnu kleští v MPL

```
TITLE model_v1;
OPTIONS
ExcelWorkBook="minimalizace_pocty_typu_standard_kleste.xlsx";
INDEX
i:=EXCELRange("kleste");
j:=EXCELRange("body");
DATA
a[i,j] :=EXCELRange("umi");
c[i] :=EXCELRange("cena");
M=1000;
INTEGER VARIABLES
x[i] EXPORT TO EXCELRange("pocet");
BINARY VARIABLES
y[i] EXPORT TO EXCELRange("koupit");
MODEL
MIN z1 EXPORT TO EXCELRange("PocetTypu") = sum(y);
nebo
MIN z2 EXPORT TO EXCELRange("Naklady") = sum(c*x);
SUBJECT TO
body[j]: sum(i:a[i,j]*y[i]) >= 1;
vyber[i]: x[i] <= M*y[i];
vyber[i]: y[i] <= x[i];
END
```


Příloha č. 8 Zápis pro rozšířenou knihovnu kleští v MPL

```
TITLE model_v1;
OPTIONS

ExcelWorkBook="minimalizace_pocty_typu_special_kleste.xlsx";

INDEX

i:=EXCEL RANGE("kleste");
j:=EXCEL RANGE("body");

DATA

a[i,j] :=EXCEL RANGE("umi");
c[i] :=EXCEL RANGE("cena");

M=1000;

INTEGER VARIABLES

x[i] EXPORT TO EXCEL RANGE("pocet");

BINARY VARIABLES

y[i] EXPORT TO EXCEL RANGE("koupit");

MODEL

MIN z1 EXPORT TO EXCEL RANGE("PocetTypu") = sum(y);

nebo

MIN z2 EXPORT TO EXCEL RANGE("Naklady") = sum(c*x);

SUBJECT TO

body[j]: sum(i:a[i,j]*y[i]) >= 1;

vyber[i]: x[i] <= M*y[i];

vyber[i]: y[i] <= x[i];

END
```

Příloha č. 9 Zápis pro standardní knihovnu kleští ve VBA

```
Sub vyberkleste()  
Worksheets("vyber").Select  
pokryto = False  
zapis = 52  
Do Until pokryto  
    MaxPocetBodu = 0  
    For kleste = 1 To 48  
        PocetBodu = 0  
        For bod = 1 To 686  
            If Cells(kleste + 1, bod + 1).Value = 1 Then  
                PocetBodu = PocetBodu + 1  
            End If  
        Next bod  
        If PocetBodu > MaxPocetBodu Then  
            MaxPocetBodu = PocetBodu  
            vybranekleste = kleste  
        End If  
    Next kleste  
    If MaxPocetBodu > 0 Then  
        zapis = zapis + 1  
        Cells(zapis, 1).Value = vybranekleste  
        For bod = 1 To 686  
            If Cells(vybranekleste + 1, bod + 1).Value = 1  
Then  
                For bodi = 1 To 48
```

```
        Cells(bodi + 1, bod + 1).Value = 100
    Next bodi
End If
    Cells(vybranekleste + 1, bod + 1).Value = 100
Next bod
Else
    pokryto = True
End If
Loop
End Sub
```

Příloha č. 10 Zápis pro rozšířenou knihovnu kleští ve VBA

```
Sub vyberklesti()  
Worksheets("vyber").Select  
pokryto = False  
zapis = 63  
Do Until pokryto  
    MaxPocetBodu = 0  
    For kleste = 1 To 59  
        PocetBodu = 0  
        For bod = 1 To 43  
            If Cells(kleste + 1, bod + 1).Value = 1 Then  
                PocetBodu = PocetBodu + 1  
            End If  
        Next bod  
        If PocetBodu > MaxPocetBodu Then  
            MaxPocetBodu = PocetBodu  
            vybranekleste = kleste  
        End If  
    Next kleste  
    If MaxPocetBodu > 0 Then  
        zapis = zapis + 1  
        Cells(zapis, 1).Value = vybranekleste  
        For bod = 1 To 43  
            If Cells(vybranekleste + 1, bod + 1).Value = 1  
Then  
                For bodi = 1 To 59
```

```
        Cells(bodi + 1, bod + 1).Value = 100
    Next bodi
End If
    Cells(vybranekleste + 1, bod + 1).Value = 100
Next bod
Else
    pokryto = True
End If
Loop
End Sub
```

ANOTAČNÍ ZÁZNAM

AUTOR	Bc. Filip Kuřík		
STUDIJNÍ OBOR	6208R088 Podniková ekonomika a management provozu		
NÁZEV PRÁCE	VYUŽITÍ LINEÁRNÍHO PROGRAMOVÁNÍ PŘI VÝBĚRU SVAŘOVACÍCH KLEŠTÍ V RÁMCI SVAŘOVNY ŠKODA AUTO a.s.		
VEDOUCÍ PRÁCE	doc. Ing. Jan Fábry, Ph.D.		
KATEDRA	KLRK - Katedra logistiky a řízení kvality	ROK ODEVZDÁNÍ	2019
POČET STRAN	61		
POČET OBRÁZKŮ	3		
POČET TABULEK	9		
POČET PŘÍLOH	10		
STRUČNÝ POPIS	<p>Tato práce je zaměřena především na výběr druhů svařovacích kleští pomocí lineárního programování ve svařovně ŠKODA AUTO a.s. Cílem této práce je minimalizovat počet druhů svařovacích kleští a s tím související pořizovací náklady. Teoretickým východiskem byly modely a metody matematického programování, především matematický model úlohy o pokrytí. Následně byl aplikován matematický model bivalentního programování a model heuristiky na připravená data, která byla získána z interní dokumentace ŠA. Na základě získaných výsledků bylo provedeno porovnání obou navrhovaných řešení a na závěr provedeno ekonomické zhodnocení včetně doporučení.</p>		
KLÍČOVÁ SLOVA	Úloha o pokrytí; lineární programování; heuristika; svařovací kleště; digitální továrna		
PRÁCE OBSAHUJE UTAJENÉ ČÁSTI: Ne			

ANNOTATION

AUTHOR	Bc. Filip Kuřík		
FIELD	6208R088 Business Management and Production		
THESIS TITLE	USE OF LINEAR PROGRAMMING FOR SELECTION WELDING GUNS IN THE BODYSHOP OF ŠKODA AUTO a.s.		
SUPERVISOR	doc. Ing. Jan Fábry, Ph.D.		
DEPARTMENT	KLRK - Department of Logistics and Quality Management	YEAR	2019
NUMBER OF PAGES	61		
NUMBER OF PICTURES	3		
NUMBER OF TABLES	9		
NUMBER OF APPENDICES	10		
SUMMARY	<p>This thesis is mainly focused on selection of welding guns by mathematical programming in ŠKODA AUTO a.s. The objective is to minimize the number of kinds of welding guns and related initial costs. The theoretical basis of thesis were models and methods of mathematical programming, especially the mathematical model of Set Covering problem. The mathematical model of bivalent programming and the heuristic model were applied to the real data, which was obtained from internal documentation of ŠA. Based on the obtained results, a comparison of both proposed solutions was made and finally an economic evaluation was made including recommendations.</p>		
KEY WORDS	<p>Set Covering Problem; linear programming; heuristics; welding gun; digital factory</p>		
THESIS INCLUDES UNDISCLOSED PARTS: No			