

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta strojního inženýrství

Ústav automatizace a informatiky

Ing. Petr Majer

MODERNÍ METODY ROZVRHOVÁNÍ VÝROBY

MODERN METHODS OF MANUFACTURING SCHEDULING

ZKRÁCENÁ VERZE PH.D. THESIS

Obor: Technická kybernetika

Školitel: RNDr. Jiří Dvořák, CSc.

Oponenti: prof. Ing. Miloš Konečný, DrSc. FP VUT v Brně
doc. RNDr. Josef Zapletal, CSc. FEKT VUT v Brně
doc. RNDr. Miloslav Mikulík, CSc. ESF MU Brno

Datum obhajoby: 28. listopadu 2003

KLÍČOVÁ SLOVA

rozvrhování zakázkové výroby, rozvrhování proudové výroby, disjunktivní graf, heuristické metody, dopravní dávky, montážní a distribuční operace, volitelnost strojů, fuzzy termíny dokončení, fuzzy doby zpracování

KEY WORDS

job shop scheduling, flow shop scheduling, disjunctive graph, heuristic methods, transfer batches, assembly and distribution operations, interchangeability of machines, fuzzy due dates, fuzzy processing times

ORIGINÁL DISERTAČNÍ PRÁCE JE ULOŽEN NA ADRESE

Ústav automatizace a informatiky

FSI VUT v Brně

Technická 2

616 69 Brno

Tel. +420 541 143 342

© 2003 Petr Majer

ISBN 80-214-2530-X

ISSN 1213-4198

OBSAH

1	Úvod	5
1.1	Současný stav	5
1.2	Cíle disertační práce	6
2	Kombinatorické problémy a rozvrhování výroby	6
2.1	Časová složitost	7
2.2	Rozvrhování výroby	7
2.2.1	Rozvrhování proudové výroby (flow shop scheduling)	7
2.2.2	Rozvrhování zakázkové výroby (job shop scheduling)	8
3	Metody řešení	9
3.1	Přesné metody	10
3.2	Heuristické metody	10
4	Rozvrhování zakázkové výroby	12
4.1	Reprezentace dat	12
4.2	Kriteria rozvrhu	14
4.3	Metody řešení založené na kódování disjunktivním grafem	14
4.4	Výsledky výpočtů	16
5	Problémy rozvrhování v reálných výrobních podmínkách	18
5.1	Montážní a distribuční operace	18
5.2	Časy přípravy, technologické a dopravní časy, termíny spuštění a termíny dokončení úloh	19
5.3	Dopravní dávky	19
5.4	Aplikace stochastických heuristických metod	20
5.5	Volitelnost strojů	20
5.6	Další aspekty reálné výroby	21
5.7	Návrh softwarového vybavení pro rozvrhování zakázkové výroby	21
6	Rozvrhování s neurčitými údaji	22
6.1	Fuzzy čísla	23
6.2	Fuzzy doby trvání operací	24
6.3	Kombinace s termíny dokončení úloh	24
6.4	Výsledky výpočtů	25
7	Závěr	26
	Literatura	27
	Shrnutí	30
	Summary	30
	Curriculum vitae autora	31
	Seznam publikací autora	31

1 ÚVOD

Všude kolem sebe vidíme snahu o zvyšování produktivity práce. Při výrobě je na operativní úrovni důležitým úkolem tvorba výrobních rozvrhů. Obecně je problém rozvrhování dán konečnou množinou výrobků, které je potřeba vyrobit a omezeným počtem výrobních strojů, které jsou k dispozici. Přitom je pro každý výrobek znám technologický postup, který určuje operace ve stanoveném pořadí i jejich doby trvání a každé operaci přiděluje určitý druh stroje.

O rozvrhování zakázkové výroby (*job shop scheduling*) mluvíme v případě, kdy pořadí strojů smí být pro každý výrobek různé. Ve speciálním případě, kdy je pořadí strojů pro všechny výrobky stejné, mluvíme o rozvrhování proudové výroby (*flow shop scheduling*). Úkolem rozvrhování je najít nejlepší rozvrh (*schedule*), tj. optimální pořadí operací na jednotlivých strojích. Jako kritérium rozvrhu je možné použít celkovou dobu trvání výroby, délky prostojů strojů, ztráty spojené s nesplněním prací v požadovaných termínech, stupeň rozpracovanosti apod. Uvedená kritéria jsou minimalizačního charakteru.

V praxi se setkáváme s řadou modifikací problému rozvrhování (zahrnutí časů přípravy operací, dopravních a technologických časů, zohlednění dopravních dávek, rozvrhování montážních operací, volitelnost strojů, omezená kapacita mezikladů, neurčitý charakter zadaných údajů apod.). Disertační práce se zabývá vytvářením a řešením takových modelů rozvrhování výroby, které reálné výrobní podmínky reflektují lépe než základní modely rozvrhování.

1.1 SOUČASNÝ STAV

Úlohy rozvrhování lze formulovat jako úlohy celočíselného programování a řešit klasickými optimalizačními metodami (zde se obvykle používá metoda větví a mezí nebo dynamické programování). Problémy rozvrhování patří mezi NP-těžké problémy, což znamená že zmíněné přesné optimalizační metody je možné použít pouze pro úlohy omezeného rozsahu. Optimální řešení rozsáhlejších úloh nelze těmito metodami získat v reálném čase a proto je nutné používat různé heuristické metody. Praxe se zatím omezuje převážně na jednoduché techniky nebo problémově specifické heuristické metody šité na míru konkrétní situaci. Ukazuje se, že lepších výsledků je možno dosáhnout s využitím stochastických heuristických metod založených na principech genetických algoritmů (*genetic algorithms*), simulovaného žihání (*simulated annealing*) a zakázaného prohledávání (*tabu search*) nebo jejich kombinací.

V českém průmyslu se při rozvrhování výroby používají různé programové systémy (např. iBaan, mySAP, Movex, ESO9, OR-Systém). Jejich posláním je však spíše než vytváření a optimalizace rozvrhů, pouze shromažďování a organizace dat z technologických postupů v různých databázích. Tyto systémy sice umějí pěkně prezentovat rozvrhy v různých tabulkách a Ganttových diagramech. Rozvrh si v nich ale uživatel musí vytvořit buď ručně sám, nebo jsou použity pouze primitivní techniky pro určení pořadí operací na strojích (FCFS – *first come, first served*, SPT – *shortest processing time*, apod.).

Většina současných manažerů se spokojí s poskytnutím jakéhokoliv proveditelného rozvrhu, u kterého jednoduchou metodou kritické cesty určí kritické činnosti a ty se pak snaží dokončit v termínech. Důsledky nedobře rozvržené výroby jsou dlouhé časové prostoje mezi jednotlivými operacemi při výrobě, nevytíženost kapacit, dlouhé průběžné doby provádění zakázek a tím pádem jejich neúměrné prodražování, nízká produktivita práce a konkurenceschopnost firmy. Naproti tomu v rozvinutých ekonomikách jsou matematické modely a metody běžnou součástí průmyslové praxe. Podniky, které si systémy rozvrhování výroby samy vyvíjejí, většinou nejsou ochotny je poskytnout jiným firmám. Na softwarovém trhu zatím chybí komplexní programový systém na vytváření a optimalizaci rozvrhů výroby, který by zahrnoval všechny aspekty reálné výroby a dokázal pružně reagovat na změny vznikající při výrobě.

V českém prostředí navíc existuje propast mezi akademickou teorií a praxí. Na akademické půdě se řeší většinou základní modely problémů a propracovávají se algoritmy schopné nejrychleji

najít optimální řešení, ale podniky požadují zejména zohlednění skutečných výrobních podmínek v těchto modelech a poskytnutí těchto algoritmů v podobě umožňující snadné nasazení v praxi.

1.2 CÍLE DISERTAČNÍ PRÁCE

Cílem disertační práce je na základě dosavadního stavu poznání ve zmíněné oblasti vyvinout takové modely rozvrhování, které lépe reflektují reálné výrobní podmínky a metody jejich řešení. Práce postupovaly v těchto krocích:

- 1) Podrobná analýza problémů rozvrhování proudové výroby a rozvrhování zakázkové výroby, popis použitelných modelů reprezentací těchto problémů a popis použitelných metod řešení. Cílem je vytvoření přehledu ve studované problematice.
- 2) Studium možnosti použití přesných metod. Jedním z cílů je určení mezí velikosti rozvrhovací úlohy, pro kterou jsou tyto metody efektivně použitelné.
- 3) Tvorba a popis matematických modelů problému rozvrhování, které zahrnují hlediska reálných výrobních podmínek, jako je zohlednění dopravních dávek, rozvrhování montážních operací, apod. Pro řešení vytvářených modelů jsme se snažili o nasazení stochastických heuristických metod.
- 4) Implementace metod na počítači a ověření na praktických aplikacích. Mimo jiné jsme se pokusili navrhnout a realizovat prototyp softwarového vybavení pro rozvrhování zakázkové výroby zahrnující aspekty reálných výrobních podmínek.
- 5) Vytvoření modelu problému rozvrhování zakázkové výroby, který dokáže pracovat s neurčitými vstupními údaji. K modelování neurčitostí jsme využili teorii fuzzy množin. Zejména nám šlo o navržení vhodných účelových funkcí a o vzájemné srovnání optimalizačních kritérií.

Poděkování

Práce je součástí výzkumného záměru CEZ: J22/98: 261100009 *Netradiční metody studia komplexních a neurčitých systémů*.

Elektronická verze disertační práce

Úplná verze disertační práce v rozsahu 90-ti stran, je k dispozici také v elektronické podobě na Internetové adrese <http://www.majer.czweb.org/scheduling>.

2 KOMBINATORICKÉ PROBLÉMY A ROZVRHOVÁNÍ VÝROBY

Úlohy rozvrhování řadíme do třídy kombinatorických problémů. To jsou problémy, ve kterých vystupují diskrétní proměnné a jejichž množina přípustných řešení je konečná. V každém z těchto problémů se skrývá možnost kombinatorické exploze.

2.1 ČASOVÁ SLOŽITOST

Časová složitost $O(g(n))$ algoritmu řešícího problém o rozsahu n udává horní mez počtu $g(n)$ elementárních operací (na hypotetickém počítači), které jsou pro vyřešení problému tímto algoritmem zapotřebí.

Řekneme, že algoritmus je *polynomiální*, jestliže jeho časová složitost je $O(n^k)$, pro nějakou konstantu k . Naproti tomu řekneme, že algoritmus má *exponenciální* časovou složitost, jestliže jeho časová složitost není polynomiální. Příkladem jsou algoritmy s časovou složitostí $O(n!)$, $O(2^n)$ atd.

Obecná teorie algoritmů [12, 34] rozlišuje dvě třídy problémů: polynomiální P (*polynomial*) a nedeterministicky polynomiální NP (*non-deterministic polynomial*). Problémy, pro jejichž řešení jsou známy polynomiální algoritmy patří do třídy P. Ostatní problémy s exponenciální složitostí patří do třídy NP. Třída problémů NP je v podstatě množina problémů, pro které polynomiální algoritmy mají být teprve nalezeny. Samozřejmě i problém patřící do třídy P můžeme neefektivně řešit algoritmem s nepolynomiální časovou složitostí, jestliže nám nevadí, že to zabere vzhledem k rozsahu problému exponenciálně dlouhou dobu. Výjimečně může být problém ze třídy NP přeřazen do třídy P, pokud se někomu podaří objevit polynomiální algoritmus k jeho řešení. Bohužel existují problémy, o kterých mnoho moderních matematiků předpokládá, že pro jejich řešení nikdy nebudou polynomiální algoritmy nalezeny, protože tyto problémy jsou příliš složité. Do této kategorie patří i problémy rozvrhování.

Klasickými metodami (např. dynamickým programováním nebo metodou větví a mezí) lze řešit problémy rozvrhování pouze do omezeného rozsahu. Na rozsáhlejší úlohy je nutné použít heuristických metod. Nejznámějšími heuristickými metodami jsou simulované žíhání, zakázané prohledávání a genetické algoritmy [24, 36].

2.2 ROZVRHOVÁNÍ VÝROBY

Základními pojmy teorie rozvrhů jsou *stroj*, *operace* a *úloha*.

Operace (*operation*) je základní technologický úkon, který už dále není dělitelný na částečné technologické úkony.

Úloha (*job*) je posloupnost operací, které je potřeba vykonat v rámci jedné zakázky. V české literatuře se někdy ve významu úlohy vyskytuje termín práce.

Stroj (*machine*) je zařízení schopné vykonávat jednu nebo několik operací. V některé literatuře se používá též termín procesor.

2.2.1 Rozvrhování proudové výroby (flow shop scheduling)

Problém rozvrhování proudové výroby (*flow shop scheduling*) [1, 40] je charakterizován množinou úloh $J = \{J_1, \dots, J_n\}$ a množinou strojů $M = \{M_1, \dots, M_m\}$. Všechny n úloh musí být vykonáno na m strojích při dodržení následujících podmínek: (i) Každá úloha je složena z m operací, které musejí být vykonány na m různých strojích. (ii) Všechny úlohy provádějí své operace na strojích ve stejném pořadí strojů (stroje jsou řazeny sériově). (iii) Na každém stroji může být v jednom okamžiku vykonávána nejvýše jedna operace jedné úlohy. (iv) Operace nemohou být přerušeny. (v) Mezi operacemi různých úloh není precedenční omezení. Obvykle se k uvedeným pěti podmínkám přidává šestá: (vi) Pořadí úloh je na každém stroji stejné. V takovém případě hovoříme o tzv. permutačním rozvrhování proudové výroby (*permutation flow shop scheduling*).

Úkolem je najít takové pořadí (permutaci) úloh, které minimalizuje trvání výrobního procesu, případně které zajistí dokončení úloh v požadovaných termínech, jsou-li zadány. Nyní předpokládejme následující označení: π - permutace úloh, $\pi(i)$ - i -tá úloha v permutaci úloh, p_{ik} -

doba trvání operace i -té úlohy na k -tém stroji, C_{ik} - termín dokončení operace i -té úlohy na k -tém stroji. Jestliže je dána permutace úloh π , pak jsou termíny dokončení $C_{\pi(i),k}$ určeny následovně [40]:

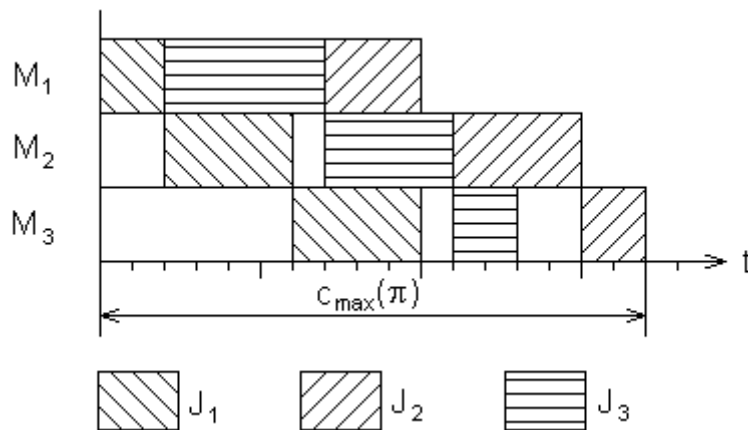
$$C_{\pi(1),1} = p_{\pi(1),1} \quad (2.1)$$

$$C_{\pi(i),1} = C_{\pi(i-1),1} + p_{\pi(i),1} \quad i = 2, \dots, n \quad (2.2)$$

$$C_{\pi(1),k} = C_{\pi(1),k-1} + p_{\pi(1),k} \quad k = 2, \dots, m \quad (2.3)$$

$$C_{\pi(i),k} = \max \{C_{\pi(i-1),k}, C_{\pi(i),k-1}\} + p_{\pi(i),k} \quad i = 2, \dots, n; k = 2, \dots, m \quad (2.4)$$

Ukázka rozvrhu pro $n = 3$ úlohy a $m = 3$ stroje je na obrázku 2.1.



Obrázek 2.1. Rozvrh v proudové výrobě

V permutačním problému rozvrhování proudové výroby je řešení reprezentováno pouze pořadím π (permutací) všech úloh. V tomto případě je každé řešení proveditelné a nazývá se rozvrh (*schedule*). Cílem je nalezení takového rozvrhu π , pro který je hodnota účelové funkce optimální. Nejčastěji se používá kritérium C_{\max} (označované anglickým termínem *makespan*) definované jako termín dokončení poslední úlohy (resp. jako doba potřebná k provedení všech úloh):

$$\text{Minimalizovat } C_{\max} = C_{\pi(n),m} \quad (2.5)$$

2.2.2 Rozvrhování zakázkové výroby (job shop scheduling)

V klasické podobě byl problém rozvrhování zakázkové výroby (*job shop scheduling*) [8, 30, 42] popsán v [3].

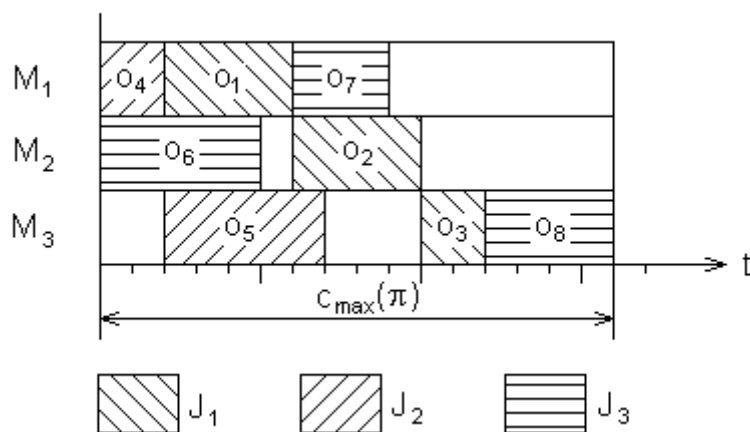
Jsou zadány tři konečné množiny: množina úloh $J = \{J_1, \dots, J_n\}$, množina strojů $M = \{M_1, \dots, M_m\}$ a množina operací $O = \{o_1, \dots, o_N\}$. Každá úloha se skládá z určitého, obecně nestejného, počtu operací. Množina O je množina všech operací všech úloh. Doba trvání k -té operace je p_k . Je potřeba vykonat všechny úlohy při splnění následujících omezení: (i) Pro provedení každé operace je zapotřebí jednoznačně přiřazený stroj z množiny M . (ii) Pořadí operací u každé úlohy je dáno technologickým postupem a nelze je měnit. (iii) Na jednom stroji lze současně provádět nejvýše jednu operaci jedné úlohy. (iv) Provádění operace nelze přerušit. (v) Mezi operacemi různých úloh není precedenční omezení, to znamená, že pro žádnou dvojici operací z různých úloh není předepsáno pořadí provádění.

Vykonání každé úlohy tedy znamená provedení všech jejích operací v zadaném pořadí na daných strojích. Jakékoli proveditelné pořadí π operací na všech strojích z M se nazývá rozvrh (*schedule*). Úkolem rozvrhování je nalézt nejvhodnější pořadí operací na daných strojích. Označme

symbolem C_j okamžik dokončení j -té operace. Často se jako kritérium používá celková doba trvání rozvrhu (*makespan*):

$$\text{Minimalizovat } C_{\max}(\pi) = \max_{1 < k < N} \{C_k\} \quad (2.6)$$

Příkladem rozvrhu pro problém o osmi operacích na třech strojích může být $\pi = \{(4,1,7), (6,2), (5,3,8)\}$, kde jednotlivé seznamy operací v kulatých závorkách představují pořadí operací na jednotlivých strojích. Tento rozvrh vidíme zakreslený pomocí Ganttova diagramu na obrázku 2.2.



Obrázek 2.2. Rozvrh v zakázkové výrobě

Při rozvrhování ve skutečné výrobě mohou být sledovány i jiné cíle, např. minimalizace ztrát spojených s nesplněním prací v požadovaných termínech, minimalizace prostojů, minimalizace rozpracované výroby, atd. Pro optimalizaci se používají většinou heuristické metody ve spojení s vhodnou reprezentací dat. Různými reprezentacemi dat v problému rozvrhování zakázkové výroby a kritérii rozvrhů se budeme zabývat v kapitole 4.

3 METODY ŘEŠENÍ

Součástí problémů rozvrhování je vždy požadavek na optimalizaci nějaké účelové funkce, která závisí na sekvencích jednotlivých operací na strojích. Rozsah prostoru řešení je pro problémy rozvrhování obrovský. Jedná se o NP-těžké úlohy [12, 34]. K jejich řešení je možné použít přesné metody (např. metodu větví a mezí), ale tyto metody přestávají být od určitého rozsahu problému efektivní, proto je nutné používat heuristické metody (např. simulované žíhání, tabu search nebo genetické algoritmy). Obecně jsou všechny metody popsány například ve [24, 32, 36] a na deterministický problém rozvrhování výroby jsou aplikovány např. ve [22, 30, 42]. Přehled používaných metod k řešení problémů rozvrhování je uveden ve [18].

Popis následujících metod vztahujeme na řešení následujícího problému: Je dána konečná množina X přípustných řešení a kritérium $f : X \rightarrow R$. Hledáme takové řešení $x^* \in X$, pro které je hodnota účelové funkce minimální $f(x^*) = \min_{x \in X} \{f(x)\}$.

3.1 PŘESNÉ METODY

Historie použití přesných metod pro problémy rozvrhování je poměrně dlouhá. Již Johnson [19] se v roce 1954 zabýval řešením rozvrhování proudové výroby libovolného počtu úloh na dvou strojích. Vycházel přitom z teoremu, že má-li permutace n úloh tu vlastnost, že platí $\min\{p_{i1}, p_{j1}\} \leq \min\{p_{i2}, p_{j2}\}$ pro $1 \leq i, j \leq n$, pak je pro tuto permutaci hodnota C_{\max} optimální. Johnson dále popsal metody řešící rozvrhování i na třech strojích. Přehled přístupů k jednoduchým problémům rozvrhování můžeme najít ve [1].

Za nejúspěšnější exaktní metodu považuje mnoho autorů [6] metodu větví a mezí. Zde se snažíme najít celkové nejlepší řešení x^* pomocí rozdělování původní množiny řešení X na stále menší podmnožiny a stanovením dolní meze hodnoty účelové funkce v každé podmnožině. Tyto podmnožiny chápeme jako množiny řešení odpovídající podproblémům původního problému. Pokud v některé podmnožině je stanovena dolní mez vyšší než je dosud nalezené nejlepší řešení, je možné tuto podmnožinu eliminovat. Obvykle se používá jedna ze dvou prohledávacích strategií: prohledávání do šířky a prohledávání do hloubky.

Naproti tomu v dynamickém programování je problém rozdělen na několik stupňů a v každém stupni je potřeba učinit rozhodnutí, jež mají vliv na rozhodnutí, která budou učiněna v pozdějších stupních. Na základě Bellmanova principu optimality je sestavena rekurzivní rovnice, která popisuje optimální hodnotu účelové funkce v daném stupni jako funkci hodnoty, získané ve stupni předchozím.

Přesné metody nelze použít na rozsáhlejší problémy, protože jejich časová náročnost roste exponenciálně s lineárním růstem rozsahu problému.

3.2 HEURISTICKÉ METODY

Heuristické metody sice nezaručují nalezení globálně optimálního řešení, ale jsou schopny v přiměřené době poskytnout řešení, jenž je uspokojivé. Heuristické optimalizační metody se používají pro optimalizaci mnohparametrových funkcí s divokým průběhem, tj. s mnoha extrémy nebo neznámým gradientem. Takovými funkcemi je i většina účelových funkcí problémů rozvrhování výroby.

Proces prohledávání prostoru řešení vyžaduje rovnováhu dvou cílů: 1. Co nejrychleji najít nejbližší (většinou lokální) optimum v okolí výchozího bodu. 2. Co nejlépe prohledat prostor všech řešení. Nejčastěji používanými heuristickými metodami jsou: zakázané prohledávání, simulované žíhání a genetické algoritmy.

3.2.1 Relace sousedství a účelová funkce

Pro použití většiny stochastických heuristických metod potřebujeme definovat určitou relaci sousedství, která pro každé přípustné řešení x umožňuje pomocí množiny transformací $S(x)$ stanovit jeho jisté okolí $U(x)$ jako množinu přípustných řešení sousedících s x . Při stanovení množiny sousedních řešení je důležité splnění podmínky dosažitelnosti, která požaduje, aby každé řešení bylo dosažitelné z libovolného jiného řešení postupnou aplikací vztahu sousednosti.

V případě problému rozvrhování výroby je přípustným řešením jistá množina proveditelných sekvencí operací na jednotlivých strojích (rozvrh). Pro každé přípustné řešení lze vypočítat hodnotu účelové funkce, která je kritériem kvality každého přípustného řešení. Cílem optimalizace pomocí heuristických metod je najít takové přípustné řešení, které má nejlepší hodnotu účelové funkce.

Pro permutační problém je přípustným řešením např. sekvence úloh ABCDEF. Relaci sousedství můžeme jednoduše definovat např. jako prohození pořadí dvou sousedních úloh. Potom by naše řešení ABCDEF mělo pět sousedních řešení: BACDEF, ACBDEF, ABDCEF, ABCEDF, ABCDFE. Způsobů, jak definovat relaci sousedství lze vymyslet mnoho (výměna libovolných dvou,

prohození tří, otočení pořadí části řetězce, atd). Experimenty ukázaly že pro permutační problém je nejvýhodnější použití relace sousedství pomocí tzv. dvoubodové shift operace. Při volbě nejvhodnějšího způsobu generování sousedních řešení pro konkrétní heuristickou metodu vycházíme obvykle z praktických zkušeností.

3.2.2 Metoda simulovaného žíhání

Simulované žíhání (*simulated annealing*) je variantou horolezeckého algoritmu, v němž heuristické kroky směřující k horšímu řešení jsou řízeny určitou pravděpodobností. Přístup simulovaného žíhání je založen na simulování fyzikálních procesů probíhajících při odstraňování defektů krystalické mřížky. Existuje určitá analogie tohoto přírodního procesu s procesem řešení optimalizačních problémů.

V aplikaci na řešení minimalizačního problému je krystal reprezentován nějakým přípustným řešením x tohoto problému. Každému přípustnému řešení můžeme přiřadit „energii krystalu“ - funkční hodnotu $f(x)$. Důležitý je rovněž parametr T , který je formální analogií teploty krystalu.

Aktuální řešení x je přeměněno náhodnou transformací $t \in S$ na nové řešení x' z okolí $U(x)$. Původní řešení se nahradí novým x' v následném procesu simulovaného žíhání s pravděpodobností dle Metropolisova vzorce [28]:

$$\text{Pravděpodobnost}(x \rightarrow x') = \min \left\{ 1, e^{-\frac{f(x')-f(x)}{T}} \right\}. \quad (3.1)$$

Jestliže funkční hodnota nového řešení x' je stejná nebo lepší než funkční hodnota původního řešení x , $f(x') \leq f(x)$, pravděpodobnost nahrazení je rovna jedné. V tomto případě je nové řešení automaticky akceptováno do dalšího procesu simulovaného žíhání. V případě, že funkční hodnota nového řešení x' není lepší než funkční hodnota původního řešení x , $f(x') > f(x)$, pravděpodobnost akceptování je menší než jednotková, ale i v tomto případě má nové řešení šanci pokračovat v simulovaném žíhání. Algoritmus simulovaného žíhání má následující jednoduchou formu:

```
x:=náhodně vygenerované řešení;
T:=Tmax; x*:=x; k:=1;
while (T>Tmin and k>0) do
begin t:=0;k:=0;
  while(t<tmax and k<kmax) do
  begin t:=t+1;
    x':=t(x);
    if f(x')≤f(x) then Pr:=1 else Pr:= exp(e, f(x')-f(x)/T);
    if random<Pr then
    begin x:=x'; k:=k+1;
      if f(x)<f(x*) then x*:=x;
    end;
  end;
end;
T:=α *T;
end;
```

Teplota T je ohraničena minimální a maximální hodnotou $T_{min} \leq T \leq T_{max}$, snižování teploty je realizováno po každém provedení vnějšího cyklu. Způsob snižování teploty určuje tzv. plán chlazení. Nejjednodušeji je možné snižovat teplotu pomocí vynásobení koeficientem α . Zkušenosti ukazují, že nejlepší hodnoty koeficientu α jsou mezi 0.8 a 0.99. V sofistikovanějších implementacích algoritmu může být rychlost snižování teploty závislá na procentu úspěšných pokusů o překlopení do jiného stavu při aktuální teplotě.

Celočíselné proměnné t a k jsou počítadla pro vnější resp. vnitřní while-cyklus. Proměnná t zaznamenává celkový počet „pokusů“ simulovaného žhání pro danou teplotu T , zatímco proměnná k zaznamenává počet úspěšných pokusů, které byly akceptovány Metropolisovým vzorcem. Pro volbu maximálních hodnot t_{max} a k_{max} neexistuje všeobecný předpis, t_{max} a k_{max} jsou obvykle ve vztahu k velikosti množiny sousedů, hodnota k_{max} je volena od několika set do několika tisíc a $t_{max} = 10 * k_{max}$. Volba jednotlivých parametrů metody je obvykle uskutečněna na základě zkušeností po mnoha experimentech.

Podmínka ukončení algoritmu je splněna buď dosažením minimální teploty nebo tzv. zmrazením krystalu, tj. neuskutečněním při dané teplotě ani jednoho úspěšného pokusu z celkového počtu t_{max} pokusů o „vykmitnutí z pevné pozice v krystalické mřížce“.

Reálná proměnná *random* je náhodně generované číslo z intervalu (0,1). Proměnná x^* zaznamenává nejlepší řešení v průběhu provádění celého algoritmu. Ve všeobecnosti proměnná x nemusí po skončení simulovaného žhání obsahovat nejlepší řešení.

V literatuře [31] je popsána podrobná teorie simulovaného žhání. Byly dokonce dokázány existenční teorémy, za jakých podmínek simulované žhání poskytuje globální minimum funkce $f(x)$ v definičním oboru x . Často se používá rozšíření simulovaného žhání směrem ke genetickému algoritmu. Místo jednoho řešení se současně optimalizuje simulovaným žháním malá populace řešení, které si vždy po určitém počtu kroků s malou pravděpodobností vymění informaci operací totožnou s křížením z genetického algoritmu.

V disertační práci jsou uvedeny podrobné popisy všech používaných heuristických optimalizačních metod. V těchto tezí je z důvodu omezeného prostoru neuvádíme.

4 ROZVRHOVÁNÍ ZAKÁZKOVÉ VÝROBY

V této kapitole ukážeme různé způsoby reprezentace problému rozvrhování zakázkové výroby. Zvláštní pozornost budeme věnovat zřejmě nejvýhodnější reprezentaci pomocí disjunktivního grafu. Budou zde popsány související algoritmy a způsoby konstruování sousedství. Dále se budeme zabývat množinou použitelných kritérií, podle nichž se optimalizuje rozvrh.

4.1 REPREZENTACE DAT

Pod pojmem reprezentace dat problému rozvrhování rozumíme určitý způsob zápisu posloupností operací na strojích do určité struktury, která nám umožní provádění výpočtů na počítačích a hledání optimálního řešení.

Při kódování a dekódování řešení do zvolené struktury musíme vzít v úvahu tři zásadní otázky:

1. Přípustnost struktury – řešení dekódované ze struktury musí ležet v oblasti přípustných řešení daného problému.
2. Legálnost struktury – struktura reprezentuje řešení daného problému.
3. Jedinečnost zobrazení – každá vytvořená struktura by měla odpovídat jedinečnému rozvrhu.

V průběhu posledních několika let bylo navrženo několik způsobů reprezentací problému rozvrhování zakázkové výroby, které mohou být rozděleny podle kódovacího postupu na reprezentace s přímým přístupem a reprezentace s nepřímým přístupem.

V přímém přístupu je rozvrh zakódován do seznamu jednotlivých operací a heuristické metody pro tyto struktury vytváří přesně definovaná sousedství, pomocí kterých probíhá vlastní

hledání. Do této skupiny patří reprezentace založená na operacích, reprezentace založená na pracích, reprezentace založená na vzájemném vztahu mezi pracemi, reprezentace založená na časech dokončení a reprezentace náhodných klíčů.

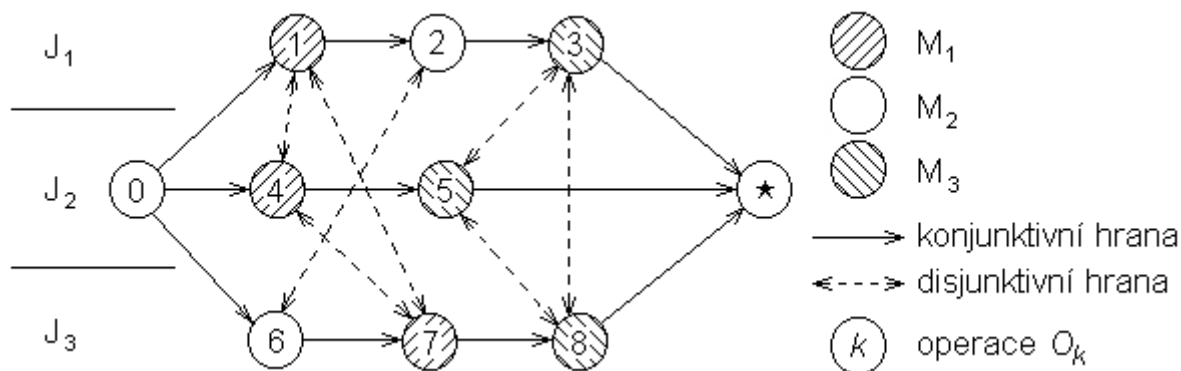
V nepřímém přístupu se místo rozvrhů kódují sekvence takzvaných dispečerských pravidel, pomocí kterých posléze uspořádáme operace. Cílem hledání je tedy nalezení co nejlepší sekvence dispečerských pravidel. Rozvrh je nakonec sestaven z této nejlepší sekvence dispečerských pravidel postupným dosazováním operací. Do této skupiny patří reprezentace založená na preferenčním seznamu, reprezentace založená na prioritních pravidlech, reprezentace založená na disjunktivním grafu a reprezentace založená na strojích.

Všechny uvedené reprezentace jsou zhruba popsány v [8]. Nejvýhodnější z hlediska použití heuristických metod se jeví reprezentace pomocí disjunktivního grafu.

4.1.1 Reprezentace disjunktivním grafem

Disjunktivní graf G [4, 5, 18, 27] je definován množinou uzlů V , množinou konjunktivních hran C a množinou disjunktivních hran D , $G = (V, C \cup D)$. Uzly reprezentují operace všech úloh. Množina uzlů V obsahuje navíc dva speciální uzly, počáteční 0 (nula) a koncový uzel * (hvězdička). Tyto speciální uzly jsou ohodnoceny nulami, ostatní uzly jsou ohodnoceny dobami trvání odpovídajících operací. Orientované konjunktivní hrany vyjadřují zadané pořadí operací v rámci jednotlivých úloh. Dále jsou zde hrany vycházející z počátečního uzlu 0 a směřující do uzlů příslušných prvním operacím úloh a hrany vycházející z uzlů příslušných posledním operacím úloh a směřující do koncového uzlu *. Disjunktivní hrany spojují operace, které musejí být provedeny na stejném stroji. Na začátku algoritmu jsou disjunktivní hrany neorientované. Graf G může být rozložen na Graf $G_0 = (V, C)$ a na m úplných podgrafů $G_k = (O^k, D_k)$, kde O^k je množina operací prováděných na k -tém stroji a $D^k = O^k \times O^k$.

Množina disjunktivních hran D tvoří tedy m úplných podgrafů, z nichž každý přísluší jednomu stroji, kde m je počet strojů. Příklad reprezentace problému rozvrhování zakázkové výroby disjunktivním grafem představuje obrázek 4.1.



Obrázek 4.1. Disjunktivní graf

Stanovit výrobní rozvrh znamená rozhodnout o pořadí operací na jednotlivých strojích, tedy stanovit orientaci disjunktivních hran. Množina S všech orientovaných disjunktivních hran se nazývá *úplná selekce*. Úplná selekce určuje přípustný rozvrh π pouze v případě, že výsledný graf $G(\pi) = (V, C \cup S)$ je acyklický. V tom případě se S nazývá *acyklická úplná selekce*.

Příkladem rozvrhu pro problém znázorněný disjunktivním grafem na obrázku 4.1 může být $\pi = \{(4,1,7), (2,6), (5,3,8)\}$, kde jednotlivé seznamy operací v kulatých závorkách představují pořadí operací na jednotlivých strojích. Celková doba zpracování všech úloh je pak dána délkou nejdelší cesty z počátečního uzlu do koncového uzlu grafu. Tuto cestu nazýváme *kritickou cestou*.

Máme-li vytvořen acyklický orientovaný graf, můžeme pomocí algoritmu nalezení kritické cesty CPM (*critical path method*) [21, 34] určit nejdříve možné časy dokončení operací C_k . Než k tomu přistoupíme, můžeme ještě graf zjednodušit. Protože libovolná operace má na stejném stroji nejvýše jednu bezprostředně předcházející operaci a nejvýše jednu bezprostředně následující operaci, můžeme z grafu vypustit ty disjunktivní hrany, které nespojují uzly odpovídající bezprostředně po sobě následujícím operacím na stejném stroji. Vedlejším efektem aplikace metody CPM na proveditelný rozvrh je, že zjistíme všechny nejdříve možné časy zahájení, nejpozději přípustné časy dokončení a časové rezervy operací.

4.2 KRITERIA ROZVRHU

Jako kritérium kvality rozvrhu se nejčastěji volí celková doba zpracování všech úloh (*makespan*). Úkolem je najít takový rozvrh π , který minimalizuje čas dokončení poslední zpracovávané operace:

$$C_{\max}(\pi) = \max_{1 \leq j \leq n} \{C_j\}. \quad (4.1)$$

Kritérium (4.1) použijeme např. v případě, kdy odběratel zadá jistou množinu zakázek J a hotové zakázky odebere a zaplatí až po ukončení poslední z nich.

Často je pro každou zakázku zadán termín jejího dokončení d_i . V takovém případě je naším úkolem nalezení rozvrhu, který minimalizuje ztráty spojené s nedodržením termínů. Pak můžeme pro vytvoření účelové funkce použít následujících parametrů:

$$L_i = C_{Last(i)} - d_i \text{ časová odchylka od plánovaného ukončení úlohy } J_i,$$

$$T_i = \max\{0, L_i\} \text{ zpoždění úlohy } J_i,$$

$$E_i = \max\{0, -L_i\} \text{ předstih úlohy } J_i,$$

přičemž $Last(i)$ je funkce, jejíž hodnotou je index poslední operace v i -té úloze.

Občas se vyskytne požadavek minimalizace rozpracované výroby, protože v systému jsou například ve formě materiálu či ve formě vyplacených mezd vázány prostředky, se kterými není možné disponovat dříve, než je hotová zakázka odevzdána. Důležitým parametrem v tomto případě je doba pobytu úlohy J_i v systému:

$$F_i = C_{Last(i)} - C_{First(i)} - p_{First(i)},$$

přičemž $First(i)$ je index první prováděné operace v i -té úloze.

Další kritéria rozvrhu jsou uvedena v úplné verzi disertační práce.

4.3 METODY ŘEŠENÍ ZALOŽENÉ NA KÓDOVÁNÍ DISJUNKTIVNÍM GRAFEM

Množina všech operací O může být přirozeně rozdělena do m podmnožin O^k , $1 \leq k \leq m$, kde každá z nich odpovídá množině operací prováděných na k -tém stroji. Pořadí operací na k -tém stroji může být definováno jako permutace π_k , která se vytvoří pouze z operací příslušné množiny O^k . Výrobní rozvrh π je tedy definován také jako množina permutací $\pi = \{\pi_k \mid k \in \{1, \dots, m\}\}$ operací na strojích $1, \dots, m$, pro kterou je odpovídající graf $G(S)$ acyklický. Problém rozvrhování zakázkové výroby můžeme znovu formulovat jako hledání proveditelného výrobního rozvrhu π , který minimalizuje

účelovou funkcí. Často se používá účelová funkce (4.1). Hodnota této funkce se rovná délce kritické cesty v grafu $G(\pi)$.

Víme, že kritická cesta $u(\pi)$ v grafu $G(\pi)$ je posloupnost kritických operací $u(\pi) = (u_1, \dots, u_v)$, kde v je počet operací na kritické cestě. Kritická cesta je rozdělena do podsekvencí nazývaných *kritické bloky* B_h . Kritické bloky jsou maximální podskupiny operací kritické cesty, které obsahují operace prováděné na stejném stroji. Kritická cesta $u(\pi)$ je tedy posloupnost kritických bloků $u(\pi) = (B_h \mid h \in (1, \dots, g))$, kde g je počet kritických bloků na kritické cestě $u(\pi)$. Definice kritických bloků je důležitá, protože pomocí ní budeme moci definovat operátory heuristických metod.

Aby bylo možné použít stochastické heuristické metody pro optimalizaci rozvrhu reprezentovaného orientovaným disjunktivním grafem, je potřeba formulovat algoritmus nalezení počátečního proveditelného rozvrhu, definovat vztah sousedství, případně určit další operátory heuristických metod (křížení, mutace) a vymežit funkci ohodnocení rozvrhu (optimalizační kritérium).

Algoritmus pro vytvoření počátečního rozvrhu navrhli Giffler a Thomson ve [13]. Tento algoritmus nám poskytne náhodný přípustný rozvrh, kterému budeme moci přiřadit hodnotu účelové funkce.

Sousedství je množina rozvrhů, které lze získat aplikováním operátoru přechodu do jiného rozvrhu. V práci [3] bylo popsáno několik způsobů, jak měnit orientace disjunktivních hran v grafu, aby se co nejefektivněji prohledal prostor řešení. Mezi nejjednodušší patří sousedství S_1 použité v práci [25]:

S_1 : Přechod ze současného řešení do nového je vytvořeno změnou orientace disjunktivní hrany (i, j) na kritické cestě současného grafu na (j, i) .

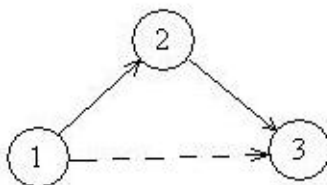
Jinak řečeno, S_1 vyjadřuje záměnu pořadí dvou bezprostředně následujících operací o_i a o_j na jednom stroji v případě, že obě operace leží na kritické cestě.

Jiné zajímavé sousedství bylo představeno v práci [14]:

S_4 : Pro každý uzel o_i v kritickém bloku posuň o_i na úplný začátek nebo konec tohoto bloku.

V literatuře [3, 30, 33] je popsána řada dalších možností, jak tvořit sousedství v rozvrzích založených na kódování disjunktivním grafem.

Všechny operátory přechodu do jiného rozvrhu jsou založeny pouze na změnách pořadí operací v rámci kritických bloků. Pro takto definované operátory lze dokázat, že pokud výchozí disjunktivní graf je acyklický, potom také každý graf vzniklý jejich aplikováním je acyklický, tedy jemu odpovídající rozvrh je proveditelný. Pokusme se o náznak tohoto důkazu. Předpokládejme, že máme problém rozvrhování zakázkové výroby reprezentovaný uzlově ohodnoceným disjunktivním grafem, kde uzly jsou ohodnoceny kladnými hodnotami. Dále předpokládejme, že žádné dvě bezprostředně po sobě následující operace jedné úlohy se neprovádějí na stejném stroji. Pro takový graf platí, že existuje-li v grafu orientovaná disjunktivní hrana, změnou jejíž orientace by vznikla v grafu smyčka, potom tato hrana nemůže ležet na kritické cestě a tudíž nemůže být vybrána k otočení.



Obrázek 4.6.

Na obrázku 4.6 vidíme 3 uzly, z uzlu 1 do uzlu 3 vede disjunktivní hrana, jejímž otočením by vznikl v grafu cyklus, protože mezi uzly 1 a 3 vede ještě jiná cesta než přímo po disjunktivní hraně (1, 3). Cesta z 1 do 3 přes uzel 2 bude vždy delší než přímá cesta po hraně (1, 3), proto tato hrana nemůže ležet na kritické cestě.

4.4 VÝSLEDKY VÝPOČTŮ

V následující části jsou prezentovány výsledky z provedených experimentů souvisejících s problémem rozvrhování zakázkové a proudové výroby. Provedli jsme tyto experimenty:

- zkoumání hranic rozsahu problému, pro který jsou efektivně použitelné přesné metody
- porovnání výsledků dosažených různými stochastickými heuristickými metodami: horolezeckým algoritmem, tabu search, simulovaným žiháním a genetickým algoritmem
- srovnání reprezentace problému preferenčním seznamem s reprezentací disjunktivním grafem

4.4.1 Použitelnost přesných metod

Metodou větví a mezí jsme řešili úlohy rozvrhování zakázkové výroby pro odstupňované rozsahy úloh. K optimalizaci jsme využili algebraický modelovací systém GAMS (*General Algebraic Modelling System*) [7], kde je implementována metoda větví a mezí (B&B).

Pro testování jsme použili etalonové příklady [2], u kterých je známá optimální hodnota účelové funkce. Vlastností všech těchto etalonových příkladů je, že každá úloha vykonává na každém stroji právě jednu operaci. V tomto experimentu jsme za minimalizační účelovou funkci zvolili celkovou dobu trvání rozvrhu C_{max} dle rovnice (2.6). Pro srovnání jsme tytéž úlohy řešili heuristickou metodou simulovaného žihání. Metoda větví a mezí, pokud skončila korektně, nám vždy poskytla optimální řešení. Simulované žihání jsme pro každý příklad spouštěli 10krát a zaznamenali jsme získanou nejlepší a průměrnou hodnotu účelové funkce. V tabulce 4.1 jsou zaznamenány všechny vypočtené hodnoty.

příklad			SA			B & B		
rozsah $n \times m$	prostor řešení $(n!)^m$	optimum C_{max}	doba výpočtu	průměr C_{max}	nejlepší C_{max}	doba výpočtu	dosažená C_{max}	status
3 × 3	216	74	1s	74	74	2s	74	optimum
3 × 6	46656	47	1s	47	47	14s	47	optimum
4 × 4	331776	96	2s	96	96	17s	96	optimum
4 × 5	7,96E+6	44	2s	44	44	31s	44	optimum
4 × 6	1,91E+8	47	2s	47	47	128s	47	optimum
5 × 3	1,72E+6	32	2s	32	32	64s	32	optimum
5 × 4	2,07E+8	35	2s	36	35	178s	35	optimum
5 × 5	2,48E+10	49	3s	52	49	1020s	49	optimum
6 × 3	3,73E+8	41	3s	41	41	661s	41	optimum
6 × 6	1,39E+17	55	4s	57	55	908s	58	nedosažitelné
10 × 5	6,29E+32	666	6s	667	666	>>1000s	-	nedosažitelné
10 × 10	3,96E+65	945	18s	987	959	>>1000s	-	nedosažitelné
20 × 10	7,27E+183	1218	93s	1315	1287	>>1000s	-	nedosažitelné
30 × 10	1,724E+324	1784	189s	1869	1819	>>1000s	-	nedosažitelné

Tabulka 4.1. Možnosti nasazení metody větví a mezí v porovnání se simulovaným žiháním

Z naměřených výsledků vyplývá, že přesnými metodami lze získat optimální řešení v reálném čase pouze pro úlohy do omezeného rozsahu. Konkrétně algoritmus větví a mezí byl za 1000 sekund schopen prozkoumat řádově průměrně 10^{10} možných řešení. To odpovídá rozsahu problému o velikosti $n = 5$ jobů, z nichž každý má $m = 5$ operací.

Při řešení úloh heuristickou metodou sice není zaručeno, že jsme získali optimální hodnotu účelové funkce, protože se neprozkoumal celý prostor řešení, ale heuristická metoda je schopna nám v přijatelně dlouhém čase poskytnout řešení, které je přijatelné.

4.4.2 Srovnání heuristických metod

Na skupině 12ti etalonových příkladů [2] rozvrhování proudové výroby jsme optimalizovali účelovou funkcí C_{max} těmito stochastickými heuristickými metodami: HC – horolezecký algoritmus, TS – tabu search, SA – simulované žíhání, GA – genetický algoritmus. Pro každý příklad jsme nechali algoritmy proběhnout 30krát, vypočetli jsme průměrnou hodnotu účelové funkce a zaznamenali nejlepší nalezené řešení. Pro všechny metody byla délka trvání výpočtu omezena počtem volání účelové funkce úměrným počtu úloh: počet volání funkce = $10 \times n^2$. Počáteční permutace úloh byly pro všechny metody generovány náhodně. Doba trvání výpočtu na počítači pro největší příklad o rozsahu 20 úloh \times 75 operací nepřesáhla 100 sekund.

Z provedených experimentů můžeme vyvodit tyto závěry: z navržených přístupů řešení rozvrhovacího problému jsme nejlepších výsledků dosahovali s použitím metody tabu search, zatímco pro rozsáhlejší příklady se osvědčily metody simulované žíhání a genetický algoritmus. Všechny tři metody (TS, SA a GA) jsou výrazně lepší v porovnání s metodou horolezeckého algoritmu.

Je nutné podotknout, že vzájemné porovnání metod je vždy relativní, výkonnost jednotlivých metod záleží na nastavení jednotlivých parametrů metod. My jsme se pokoušeli parametry metod nastavovat co nejlépe na základě zkušeností získaných po provedení mnoha experimentů s každou metodou.

4.4.3 Srovnání reprezentací

Na základě počítačového programu Plánování výroby 3.0 [5] bylo provedeno srovnání reprezentací problému rozvrhování zakázkové výroby pomocí disjunktivního grafu s reprezentací pomocí preferenčního seznamu při použití optimalizačních metod: simulované žíhání, tabu search, genetický algoritmus.

Z porovnání účinnosti způsobů kódování mezi disjunktivním grafem a preferenčním seznamem je patrné, že metody používající kódování disjunktivním grafem jsou ve všech případech účinnější. Metody založené na kódování preferenčním seznamem jsou sice schopny pro svou jednoduchost za stejnou dobu vykonat řádově více svých kroků, ale nejsou schopny dosáhnout optima i u poměrně jednoduchých příkladů. I přes svou časovou náročnost lze metody založené na kódování disjunktivním grafem upřednostnit před metodami používajícími kódování preferenčním seznamem.

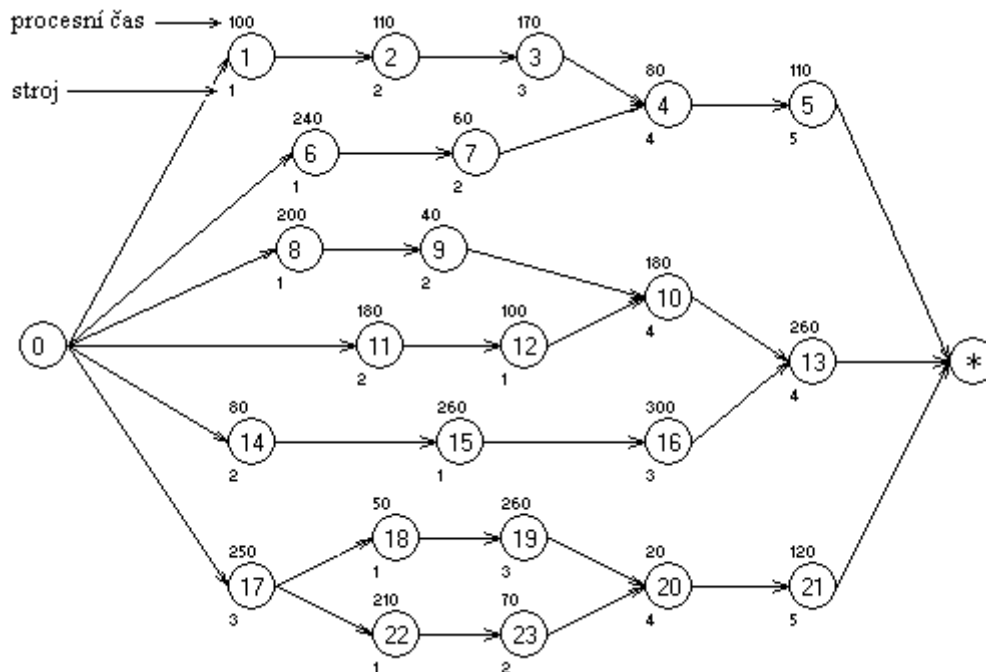
5 PROBLÉMY ROZVRHOVÁNÍ V REÁLNÝCH VÝROBNÍCH PODMÍNKÁCH

V předcházející kapitole byl popsán klasický model problému rozvrhování zakázkové výroby. V praxi je možné se setkat s řadou jeho modifikací (zohlednění dopravních dávek, rozvrhování montážních operací, různé výrobní způsoby, volitelnost strojů, omezená kapacita mezikladů apod.). V této kapitole vyjdeme z modelu prezentovaného ve [15] a budeme studovat možnosti reprezentace nejčastějších reálných podmínek pomocí tzv. rozšířeného disjunktivního grafu.

5.1 MONTÁŽNÍ A DISTRIBUČNÍ OPERACE

V technologickém postupu i méně složitých výrobků se často setkáme s potřebou kompletovat celek z více částí. S těmito celky neřídka dále provádíme další montážní operace, než vznikne konečný výrobek. Montážní celek může být určen k dalšímu zpracování nebo může být samostatným prodávaným produktem. Obrácenou operací montáže je operace typu distribuce. Často je úkolem rozvrhnout výrobu, ve které se vyskytuje současně několik distribučních operací a několik montážních operací.

Nechť $G = (V, C \cup D)$ je disjunktivní graf. V klasické podobě disjunktivního grafu má každý uzel (kromě uzlů 0 a *) v podgrafu $G_0 = (V, C)$ pouze jednoho předchůdce a pouze jednoho následníka. Problém s montážními operacemi lze jednoduše modelovat takovým způsobem že připustíme, aby uzel reprezentující montážní (resp. distribuční) operaci měl více předchůdců (resp. následníků), spojených s tímto uzlem konjunktivními hranami. Ukázku grafu $G_0 = (V, C)$ rozšířeného tímto způsobem můžeme vidět na obrázku 5.1. Operace 17 je distribuční a operace 4, 10, 13 a 20 jsou montážní.



Obrázek 5.1. Rozšíření disjunktivního grafu o distribuční a montážní operace

Jako jednu úlohu v tomto případě chápeme množinu operací reprezentovaných uzly, které jsou navzájem pospojovány konjunktivními hranami (kromě hran spojených s uzly 0 a *). Na obrázku 5.1 jsou tedy 3 úlohy: první úloha je tvořená operacemi 1 až 7, druhá úloha operacemi 8 až 16 a třetí úloha operacemi 17 až 23.

5.2 ČASY PŘÍPRAVY, TECHNOLOGICKÉ A DOPRAVNÍ ČASY, TERMÍNY SPUŠTĚNÍ A TERMÍNY DOKONČENÍ ÚLOH

Nyní se podrobně podíváme na jednu (k -tou) operaci. Klasický model rozvrhování zakázkové výroby počítá pouze s jedním výrobním časem p_k operace (*process time*). Ale v praxi doba trvání k -té operace zahrnuje ještě několik dalších časů se kterými musíme počítat: čas přípravy, technologický čas a dopravní čas. Tyto časy mohou být reprezentovány pomocí ohodnocení hran v disjunktivním grafu.

Čas přípravy operace (*setup time*) je doba potřebná např. k seřízení stroje. Časem přípravy chápeme dobu přípravy stroje, po kterou ještě není nutné, aby byla dokončena operace stejné úlohy na předchozím stroji, protože v opačném případě by tento čas bylo možné jednoduše zahrnout do výrobního času. Necht' každá disjunktivní hrana $\{i, j\}$ je reprezentována dvěma opačnými orientovanými hranami (i, j) a (j, i) . Úkolem optimalizace bude vybrat pouze jednu z této dvojice hran. Jsou-li časy přípravy na pořadí operací nezávislé, potom každé disjunktivní hraně (i, j) přiřadíme ohodnocení s_j , což je čas přípravy operace o_j . Jsou-li časy přípravy na pořadí závislé, potom každou disjunktivní hranu (i, j) ohodnotíme hodnotou s_{ij} , což je čas přípravy operace o_j , kterou předchází na stejném stroji operace o_i .

Technologický čas τ_k (*technologic time*) je doba, po kterou musí úloha po skončení vlastní (k -té) operace čekat, než může postoupit na další stroj. Vlastní stroj ale přitom už může být použit pro zpracování další operace jiné úlohy. Dopravní čas t_k (*transport time*) je doba dopravy úlohy na následující pracoviště. Technologický a dopravní čas mají na čas začátku navazující operace stejný vliv, proto je můžeme sečíst a nahradit jednou hodnotou, kterou ohodnotíme konjunktivní hranu vedoucí k následující operaci stejné úlohy.

Nyní předpokládejme, že i -tá úloha má stanoven nejdříve možný čas zahájení ρ_i a termín dokončení d_i . V disertační práci je ukázáno, že problém minimalizace maximálního zpoždění úlohy může být modelován disjunktivním grafem pomocí ohodnocení hran spojujících uzly 0 a *.

Nakonec orientací disjunktivních hran pomocí optimalizační metody získáme orientovaný graf, v němž jsou ohodnoceny uzly hodnotami p_k a hrany hodnotami h_{ij} . Metodou nalezení kritické cesty snadno získáme dobu trvání rozvrhu, nejdříve možné termíny začátků operací, nejpozději přípustné termíny ukončení operací a časové rezervy operací.

5.3 DOPRAVNÍ DÁVKY

Požadovaný objem výroby každé úlohy je její *výrobní dávka*. Výrobní dávka může být předávána z i -tého na bezprostředně následující j -tý stroj jedním následujících tří způsobů postupný, souběžný a smíšený způsob [39, 41].

Při postupném výrobním způsobu předává pracoviště celou výrobní dávku na j -té pracoviště až po jejím úplném dokončení na i -tém pracovišti. Minimální průběžná doba výrobní dávky je při postupném způsobu v porovnání s ostatními způsoby vysoká.

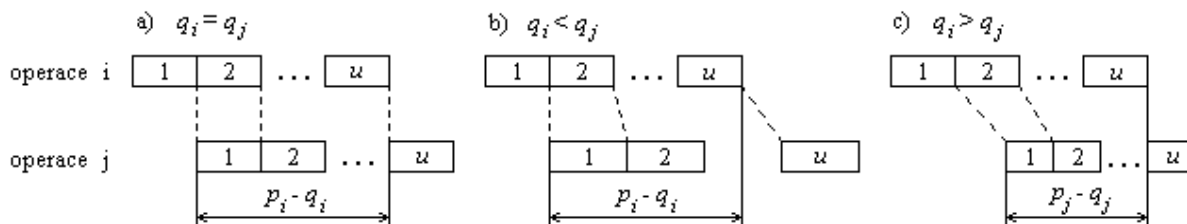
Naproti tomu v souběžném výrobním způsobu je výrobní dávka rozdělena na u dopravních dávek, které se předávají postupně ze stroje přiděleného i -té operaci na bezprostředně následující stroj přidělený j -té operaci. Zpracování dopravní dávky začne na následujícím pracovišti ihned, jakmile dorazí dopravní dávka z předcházejícího pracoviště. Při čekání na další dopravní dávku se však mohou na strojích vyskytovat zbytečné prostoje.

Při smíšeném výrobním způsobu je stanovena podmínka, že zpracování výrobní dávky nesmí být na žádné operaci přerušeno. Když je stroj přiřazený j -té operaci volný, není nutné čekat na dokončení celé výrobní dávky na předcházejícím pracovišti. Provedení j -té operace může začít dříve, ale tak, aby zpracování výrobní dávky na j -té operaci nebylo přerušeno. Čas zpracování i -té operace

jedné dopravní dávky je $q_i = p_i / u$, kde p_i je čas zpracování i -té operace celé výrobní dávky. Z obrázku 5.2. lze snadno odvodit velikost *největšího časového překrytí* v_{ij} dvou po sobě následujících operací o_i a o_j :

$$v_{ij} = \min\{p_i - q_i, p_j - q_j\} \quad (5.1)$$

Pokud jsou bezprostředně po sobě následujícím operacím i a j přiděleny různé stroje a výrobní dávka je rozdělena na několik dopravních dávek, zmenšíme ohodnocení konjunktivní hrany (i, j) v disjunktivním grafu o hodnotu v_{ij} . Při započítání technologických a dopravních časů bude tedy ohodnocení konjunktivní hrany (i, j) rovno $h_{ij} = \tau_i + t_{ij} - v_{ij}$.



Obrázek 5.2. Časové překrytí operací při souběžném výrobním způsobu

5.4 APLIKACE STOCHASTICKÝCH HEURISTICKÝCH METOD

Rozšířeným disjunktivním grafem rozumíme graf, ve kterém se mohou vyskytovat uzly odpovídající distribučním a montážním operacím a kde jsou ohodnoceny uzly hodnotami p_k a hrany hodnotami h_{ij} . Pro vytvoření počátečního rozvrhu jsme upravili algoritmus z [13] do podoby, která je použitelná pro rozšířený disjunktivní graf. Tento algoritmus, jenž je publikován v disertační práci, nám poskytne přípustný rozvrh $\pi = \{\pi_1, \dots, \pi_m\}$, kterému můžeme přiřadit ohodnocení (například využijeme CPM). Vytvořený rozvrh použijeme jako počáteční řešení pro aplikování některé z heuristických optimalizačních metod.

Operátor přechodu do jiného řešení je možné definovat stejnými způsoby jako bylo popsáno v kapitole 4.3. V rozšířeném disjunktivním grafu připouštíme existenci precedenčních omezení daných konjunktivními hranami mezi operacemi na jednom stroji a také ohodnocení disjunktivních hran kladnými hodnotami. Pro přípustný graf tohoto druhu ovšem právě proto, že připouštíme i ohodnocení hran, není splněna podmínka, že aplikováním operátorů přechodu do jiného rozvrhu vznikne vždy přípustný graf (bez smyček). Proto musíme kontrolovat proveditelnost vygenerovaného rozvrhu v každém kroku našeho algoritmu.

5.5 VOLITELNOST STROJŮ

V problému volitelnosti strojů [10], nemusí být pro některou operaci pevně určen jeden konkrétní stroj, ale seznam strojů, na jednom z nich se má daná operace volitelně provést. Doba trvání stejné operace i její cena přitom může být na různých strojích různá. Zadání problému obsahuje kromě seznamu ohodnocených uzlů odpovídajících operacím a seznamu ohodnocených konjunktivních hran odpovídajících technologickým návaznostem operací, též pro některé operace seznam alternativních strojů s časy trvání. Úkolem heuristické metody, pro niž se pokusíme navrhnout reprezentaci, bude stanovit nejen nejlepší pořadí operací na strojích, ale také pro operaci vybrat z jejího seznamu stroj.

Rozvrh π je představován dvěma řetězcí parametrů $\pi = \{X, Y\}$. Řetězec Y popisuje pořadí provádění operací na jednotlivých strojích (např. pomocí reprezentace založené na operacích) při známém přiřazení operací ke strojům. Řetězec X obsahuje tolik prvků, kolik je v problému operací,

pro které má být vybrán stroj z více možných strojů. Každý prvek v řetězci X je vyjádřen indexem použitého stroje.

Pro takto reprezentovaný problém jsme navrhli tři speciální genetické algoritmy, které jsou popsány v disertační práci. V těchto tezích naznačíme pouze postup prvního z nich:

5.5.1 Genetický algoritmus pro řešení problému s nesourodými parametry

Uvažujeme jedince, který je reprezentován dvěma částmi řešení X a Y . Každému jedinci lze přiřadit hodnotu funkce fitness. V každé části řešení X , Y můžeme odděleně snadno definovat operátory křížení a mutace. Počáteční populace se vygeneruje náhodně. V každé iteraci se z populace náhodně vybere její podmnožina, z níž dva nejlepší jedinci se stanou rodiči. Náhoda rozhodne, jestli křížení rodičů proběhne v části X nebo v části Y . Křížením vzniknou dva potomci. Každý potomek se s malou pravděpodobností podrobí mutaci. Opět náhoda rozhodne, jestli mutace proběhne v části X nebo v části Y . Nakonec nově vzniklí potomci nahradí dva nejhorší jedince z náhodně vybrané podmnožiny populace. Algoritmus končí provedením daného počtu iterací nebo vypršením časového limitu. V průběhu algoritmu se zaznamenává jedinec s nejlepší hodnotou funkce fitness.

5.6 DALŠÍ ASPEKTY REÁLNÉ VÝROBY

V reálné zakázkové výrobě se při rozvrhování můžeme setkat s množstvím dalších specifických požadavků. Disertační práce se zaměřila jen na řešení nejdůležitějších z nich. Ostatním aspektům se věnujeme jen přehledově.

Často před zahájením výroby neznáme všechny zakázky, které máme zpracovávat, nýbrž zákazníci své objednávky činí postupně nahodile v průběhu doby, kdy jsou naše stroje vytíženy prací na předchozích zakázkách. Je vhodné zpracování nové objednávky do našeho rozvrhu on-line a v reálném čase. Při obdržení nové zakázky přerozvrhujeme výrobu (*rescheduling*). Proces rozvrhování je pak kontinuální a uplatní se v něm kritérium nejdelší průběžné doby zakázky.

Další omezení dostaneme v případech, když je třeba rozvrh přizpůsobit pracovní době, když kapacita mezioperačních skladů je omezená, apod.

Možnost *přerušeni operace* v určitém okamžiku je možné modelovat v disjunktivním grafu tak, že operaci rozdělíme na dvě operace, mezi které zařadíme uzel představující hypotetickou operaci s nulovou délkou trvání.

V neposlední řadě se ve výrobě často vyskytuje spousta neurčitých vstupních údajů. Tématu rozvrhování s neurčitými údaji se věnuje 6. kapitola disertační práce.

5.7 NÁVRH SOFTWAREVÉHO VYBAVENÍ PRO ROZVRHOVÁNÍ ZAKÁZKOVÉ VÝROBY

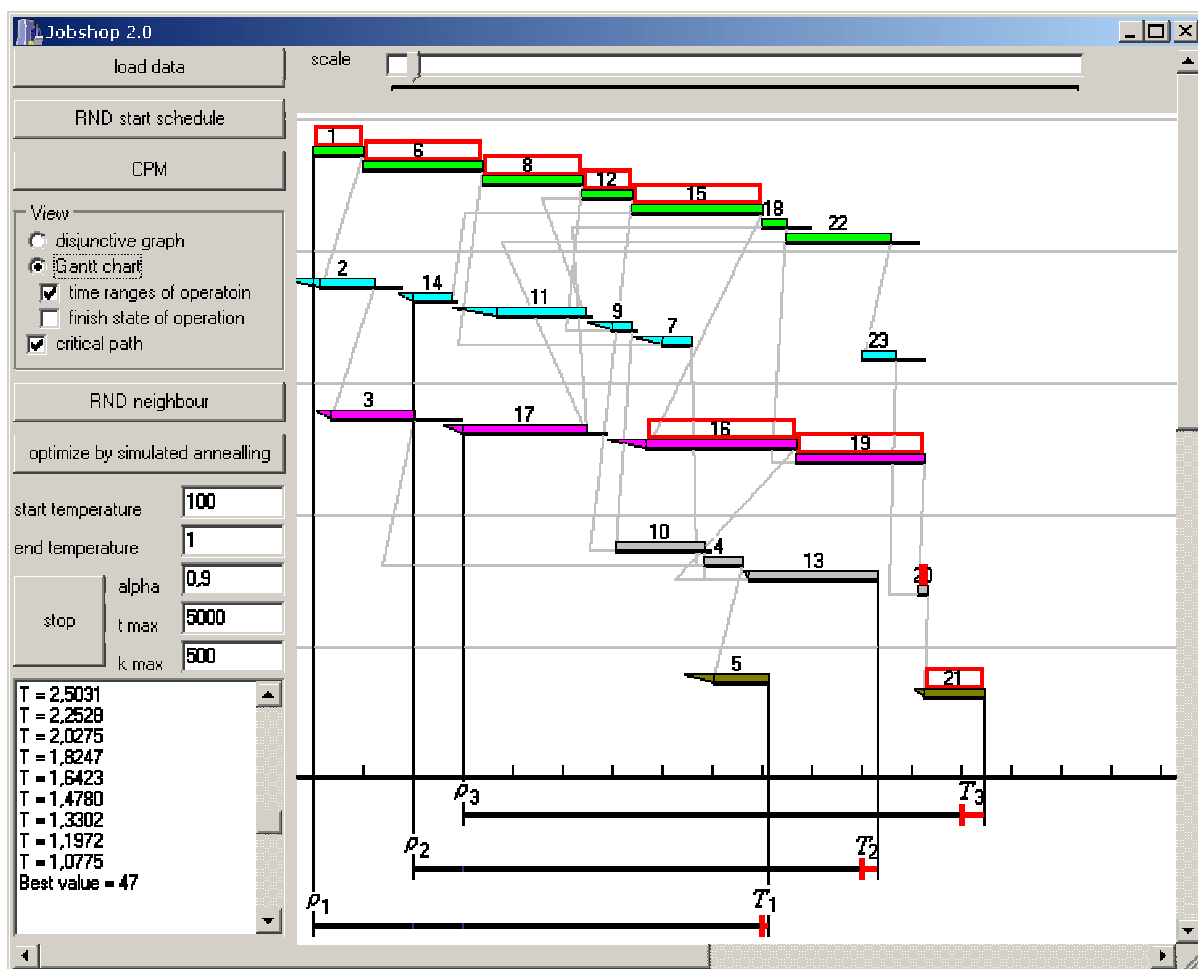
Pro řešení problémů rozvrhování zakázkové výroby se zohledněním reálných výrobních podmínek jsme navrhli a pokusili jsme se realizovat prototyp softwarového vybavení.

Předsevzali jsme si, že vytvoříme počítačový program, pomocí kterého bude možné snadno a přehledně vytvářet a optimalizovat výrobní rozvrhy na základě známých technologických postupů. Náš program pracuje s modelem rozvrhování zakázkové výroby, který zahrnuje výrobky obsahující montážní a distribuční operace, doplňkové časy i dopravní dávky. V programu jsme použili reprezentaci pomocí rozšířeného disjunktivního grafu.

Nejprve jsme navrhli relační model databáze, tj strukturu tabulek, odkud se mají data načítat a jak ukládat výsledky. V disertační práci jsou pak podrobně popsány tři základní navržené moduly programu: 1. programový modul pro přípravu dat do struktury grafu, 2. programový modul pro výpočet a 3. programový modul na prezentaci výsledků. K programování jsme použili vývojový

nástroj Borland C++ Builder verze 5 a pro uložení vstupních a výstupních dat tabulky systému Paradox.

Pro ilustraci zde uvádíme pouze jednu obrazovku našeho programu s optimalizovaným rozvrhem zobrazeným pomocí Ganttova diagramu. Vidíme, že díky optimalizaci pořadí operací na strojích pomocí simulovaného žihání lze dosáhnout minimálních časových odchylek (T_1 , T_2 , T_3) od požadovaných termínů dokončení jednotlivých úloh. V disertační práci je podrobně rozebrán i číselný příklad, který zde z důvodu nedostatku místa nelze uvést.



Obrázek 5.3. Optimalizovaný rozvrh znázorněný Ganttovým diagramem

6 ROZVRHOVÁNÍ S NEURČITÝMI ÚDAJI

Tato kapitola se zaměří na problémy rozvrhování s neurčitě zadanými údaji. Ve skutečných problémech rozvrhování se vyskytuje řada neurčitých údajů. My se budeme zabývat dvěma nejdůležitějšími z nich: 1) neurčitosti v termínech dokončení úloh a 2) neurčitosti v dobách trvání činností. Používají se dva obecné přístupy k modelování neurčitostí:

- 1) *Stochastický přístup* [11, 35], ve kterém se provádí odhad hodnot na základě dostatečně velkého souboru historických záznamů. Předpokládá se nezávislost náhodných proměnných.
- 2) Přístup pomocí *teorie fuzzy množin* [29] je obecně jednodušší, protože nevyžaduje tak velké množství vstupních dat, a díky tomu ho můžeme použít i u malosériové a kusové výroby. Na

případy rozvrhování je fuzzy přístup aplikován v literatuře [9, 16, 17] (rozvrhování proudové výroby), [23, 38] (rozvrhování zakázkové výroby).

Ve článku [20] je ukázána možnost kombinování stochastického přístupu s fuzzy přístupem. V této práci se důkladněji zaměříme na modelování neurčitostí pomocí teorie fuzzy množin.

6.1 FUZZY ČÍSLA

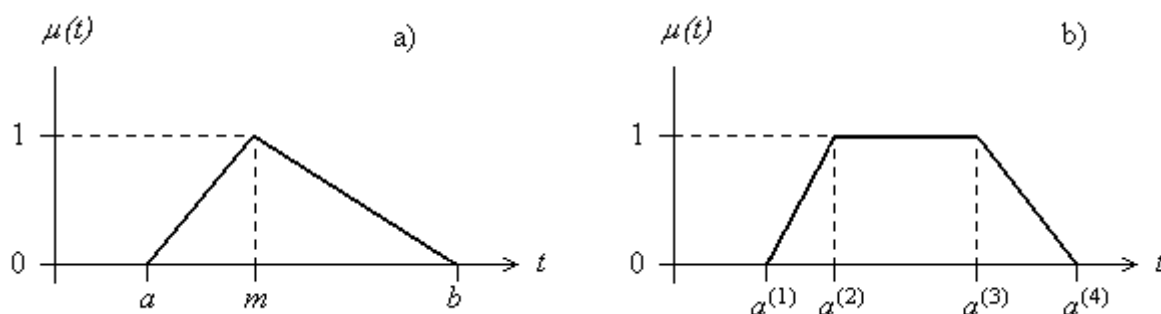
Neurčité údaje jsou často reprezentovány jako *fuzzy čísla*. Fuzzy číslo A je normální konvexní fuzzy množina na universu reálných čísel, která je určena čtveřicí bodů $(a^{(1)}, a^{(2)}, a^{(3)}, a^{(4)})$ a po částech spojitou funkcí příslušnosti s následujícími vlastnostmi:

- (i) $a^{(1)} \leq a^{(2)} \leq a^{(3)} \leq a^{(4)}$; (ii) $\mu_A(x) = 0$ pro $x \leq a^{(1)}$, $x \geq a^{(4)}$; (iii) $\mu_A(x) = 1$ pro $a^{(2)} \leq x \leq a^{(3)}$; (iv) $\mu_A(x)$ je rostoucí na $[a^{(1)}, a^{(2)}]$ a klesající na $[a^{(3)}, a^{(4)}]$.

V pracích [17, 32] je použit obecný tvar fuzzy čísla, kde funkce příslušnosti nemusejí být lineární na intervalech $[a^{(1)}, a^{(2)}]$ a $[a^{(3)}, a^{(4)}]$. Jednodušší lichoběžníková fuzzy čísla (viz obrázek 6.1b) jsou studována ve [9, 23, 37]. V tomto případě je funkce $\mu_A(t)$ definována následovně:

$$\mu_A(t) = \max \left(\min \left(\frac{t - a^{(1)}}{a^{(2)} - a^{(1)}}, \frac{t - a^{(4)}}{a^{(3)} - a^{(4)}}, 1 \right), 0 \right). \quad (6.1)$$

Někdy bývají fuzzy data v rozvrhování modelována pomocí trojúhelníkových fuzzy čísel. Trojúhelníkové fuzzy číslo (viz obrázek 6.1a) je speciální případ lichoběžníkového fuzzy čísla, ve kterém je $a^{(2)} = a^{(3)}$.



Obrázek 6.1. a) trojúhelníková a b) lichoběžníková funkce příslušnosti fuzzy čísla

Pro výpočet termínu dokončení a pro konstrukci níže popsaných optimalizačních kritérií budeme potřebovat některé operátory mezi fuzzy čísly. Necht' $X = (x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)})$, $Y = (y^{(1)}, y^{(2)}, y^{(3)}, y^{(4)})$ jsou lichoběžníková fuzzy čísla. Operace součtu fuzzy čísel je podle principu rozšíření definována následovně:

$$X + Y = (x^{(1)} + y^{(1)}, x^{(2)} + y^{(2)}, x^{(3)} + y^{(3)}, x^{(4)} + y^{(4)}), \quad (6.2)$$

Kdybychom definovali operaci maxima fuzzy čísel pomocí principu rozšíření, výsledkem by nemuselo být lichoběžníkové fuzzy číslo. Proto tuto operaci aproximujeme následovně:

$$\tilde{\max} \{X, Y\} = \left(\max \{x^{(1)}, y^{(1)}\}, \max \{x^{(2)}, y^{(2)}\}, \max \{x^{(3)}, y^{(3)}\}, \max \{x^{(4)}, y^{(4)}\} \right). \quad (6.3)$$

K optimalizaci řešení podle účelových funkcí, jejichž oborem hodnot jsou fuzzy čísla, je nutné definovat operátor seřazení fuzzy čísel. Velmi podrobně se problému seřazení fuzzy čísel věnujeme v disertační práci. Jeden ze způsobů určuje, že menší fuzzy číslo je to, které má menší střední

hodnotu m a v případě shodnosti středních hodnot je menší to s menší směrodatnou odchylkou s . Tyto veličiny jsou pro obecné fuzzy číslo A definovány takto:

$$m(A) = \frac{\int_U x \mu_A(x) dx}{\int_U \mu_A(x) dx}, \quad s(A) = \sqrt{\frac{\int_U x^2 \mu_A(x) dx}{\int_U \mu_A(x) dx} - [m(A)]^2}. \quad (6.4)$$

6.2 FUZZY DOBY TRVÁNÍ OPERACÍ

Nyní předpokládejme, že doby trvání operací na strojích nejsou dány deterministicky, ale jsou dány lichoběžníkovými fuzzy čísly $P_k = (p_k^{(1)}, p_k^{(2)}, p_k^{(3)}, p_k^{(4)})$.

Pro případ rozvrhování proudové výroby vypočítáme fuzzy časy dokončení úloh použitím rovnic (2.1 - 2.4), kde operátor součtu a operátor maxima zaměníme za příslušné operátory (6.2) a (6.3) mezi fuzzy čísla.

Podobně pro případ rozvrhování zakázkové výroby je možné za způsob reprezentace dat zvolit disjunktivní graf, ve kterém uzly představují jednotlivé operace, jejichž délky trvání jsou dány fuzzy čísly. Pro určení nejdříve možných časů dokončení operací přizpůsobíme metodu hledání kritické cesty tak, aby mohla pracovat s fuzzy čísly místo s deterministickými hodnotami. Časy dokončení jednotlivých operací C_k vyjdou také jako lichoběžníková fuzzy čísla. Místo kritéria (4.1) musíme použít kritérium pro minimalizaci fuzzy čísel.

6.3 KOMBINACE S TERMÍNY DOKONČENÍ ÚLOH

Pro případ, kdy jsou zadány termíny dokončení úloh volíme jako kritérium nespokojenost s rozdíly mezi požadovanými termíny d_i dokončení úloh a skutečnými časy $C_{Last(i)}$ dokončení posledních operací úloh. Funkce $Last(i)$ udává index poslední operace i -té úlohy. Hledáme takový rozvrh π , který minimalizuje největší nespokojenost $F_{max}(\pi)$ s dodržением termínu, případně který minimalizuje celkovou nespokojenost $F_{sum}(\pi)$ s dodržением všech termínů. Tyto problémy můžeme formulovat takto:

$$\text{Minimalizovat } F_{max}(\pi) = \max_{1 \leq i \leq n} \{F_i(C_{Last(i)})\} \quad (6.5)$$

$$\text{Minimalizovat } F_{sum}(\pi) = \sum_{i=1}^n \{F_i(C_{Last(i)})\} \quad (6.6)$$

Hodnoty dílčích účelových funkcí F_i mohou být chápány jako deterministické nebo fuzzy. Deterministické hodnoty představují stupně nespokojenosti s dokončením úloh ve stanovených termínech. V případě fuzzy hodnot je počítána celková fuzzy hodnota účelové funkce použitím příslušných operací nad fuzzy čísly a při optimalizaci se použije některý z operátorů seřazení fuzzy čísel popsanych výše.

Otázka zní, jak definovat dílčí kritéria spokojenosti F_i s dodržением jednoho termínu dokončení, když vypočtený $C_{Last(i)}$ a/nebo požadovaný termín dokončení d_i je fuzzy číslo. V disertační práci jsou studovány fuzzy modely popisující následující situace: 1) fuzzy termíny a přesné doby zpracování, 2) přesné termíny a fuzzy doby zpracování, 3) fuzzy termíny a fuzzy doby zpracování.

Pro případ kombinující fuzzy termíny s fuzzy dobami trvání operací jsme ve [26] navrhli kritérium založené na plochách pod funkcemi příslušnosti fuzzy množin: Odchylnka vypočteného fuzzy času $C_{Last(i)}$ dokončení od požadovaného termínu D_i dokončení i -té úlohy může být charakterizována pomocí fuzzy množin $C_{Last(i)}^D$, $C_{Last(i)}^E$ a $C_{Last(i)}^T$, které jsou definovány následujícími funkcemi příslušnosti:

$$\mu_{C_{Last(i)}^D}(t) = \mu_{C_{Last(i)} \cap D_i}(t) = \min\{\mu_{C_{Last(i)}}(t), \mu_{D_i}(t)\} \quad (6.7)$$

$$\mu_{C_{Last(i)}^E}(t) = \begin{cases} \max\{0, \mu_{C_{Last(i)}}(t) - \mu_{D_i}(t)\} & \text{pro } t \leq d_i^{(2)} \\ 0 & \text{pro } t > d_i^{(2)} \end{cases} \quad (6.8)$$

$$\mu_{C_{Last(i)}^T}(t) = \begin{cases} 0 & \text{pro } t < d_i^{(3)} \\ \max\{0, \mu_{C_{Last(i)}}(t) - \mu_{D_i}(t)\} & \text{pro } t \geq d_i^{(3)} \end{cases} \quad (6.9)$$

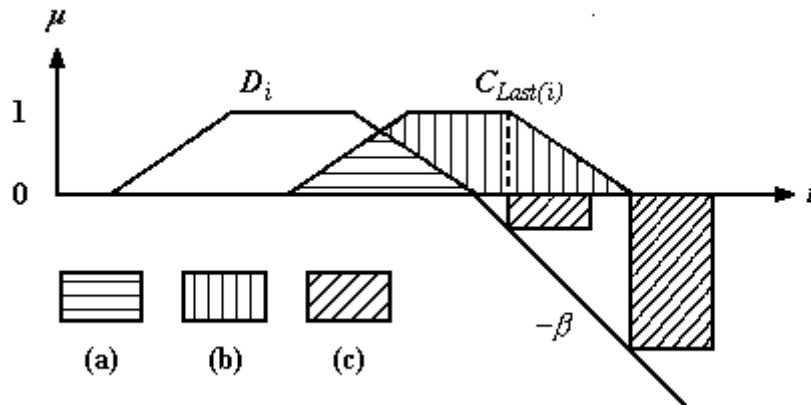
Pak můžeme definovat funkce F_i z (6.5) a (6.6) takto:

$$F_i(C_{Last(i)}) = \text{area}(C_{Last(i)}^E) + \text{area}(C_{Last(i)}^T) - \text{area}(C_{Last(i)}^D) + \alpha_i \sum_{j=1}^4 \max\{0, d_i^{(1)} - c_{Last(i)}^{(j)}\} + \beta_i \sum_{j=1}^4 \max\{0, c_{Last(i)}^{(j)} - d_i^{(4)}\} \quad (6.10)$$

kde α_i a β_i jsou nezáporné konstanty a funkce area značí velikost plochy pod funkcí příslušnosti fuzzy čísla.

Konstrukci funkce F_i ilustruje obrázek 6.2, kde velikost plochy (a) odpovídá hodnotě $\text{area}(C^D)$, velikost plochy (b) odpovídá hodnotě $\text{area}(C^T)$ a přímka se směrnici $-\beta$, určuje druhou sumu ve vzorci (6.10) penalizující zpoždění úlohy.

Při vzájemném srovnání několika kritérií na základě experimentů popsanych v disertační práci jsme nejlepších výsledků dosahovali při použití právě tohoto kritéria.



Obrázek 6.2. Plochy využití v účelové funkci $F_i^{(6)}$

6.4 VÝSLEDKY EXPERIMENTŮ

Na základě vlastních programů na počítači jsme vykonali několik experimentů. Nejprve jsme řešili otázku na základě jakých testovacích příkladů experimenty provést. Protože nebyly k dispozici žádné

standardní testovací příklady pro fuzzy rozvrhování, vytvořili jsme si některé vlastní a také jsme fuzzifikovali standardní etalonové příklady pro deterministický případ. Způsoby generování testovacích příkladů jsou popsány v disertační práci.

S pomocí nich jsme pak stanovili závislost nalezené hodnoty účelové funkce na čase u čtyř různých heuristických metod. Z testů vyplynulo, že všechny metody víceméně jistě konvergují (nejpomaleji primitivní metoda horolezeckého algoritmu a metoda zakázaného prohledávání) ke globálnímu optimu účelové funkce. Rychlost konvergence je možné ovlivnit vhodnější volbou parametrů metod.

Pak jsme provedli vzájemné srovnání několika navržených kritérií pro seřazení fuzzy čísel. Pro několik etalonových příkladů jsme pomocí metody simulovaného žíhání optimalizovali dobu dokončení poslední operace C_{max} reprezentovanou lichoběžníkovým fuzzy číslem. V metodě jsme pro seřazení fuzzy čísel použili postupně 7 různých kritérií. Jejich porovnání jsme založili na relativní chybě hodnoty účelové funkce vzhledem k nejlepší získané hodnotě této funkce ze všech provedených optimalizací pro každý příklad. Z tohoto hlediska mělo nejmenší celkovou průměrnou relativní chybu kritérium založené součtu střední hodnoty a pravé části směrodatné odchylky fuzzy čísla.

Nakonec jsme vzájemně srovnali několik navržených optimalizačních kritérií problému rozvrhování zakázkové výroby kombinujícího fuzzy doby trvání operací s fuzzy termíny dokončení úloh. Za nejlepší považujeme takové kritérium, které dává řešení úspěšná také z hlediska ostatních kritérií. Výsledky experimentů ukazují, že z tohoto hlediska má nejmenší celkovou průměrnou relativní chybu výše popsané kritérium založené na plochách pod funkcemi příslušnosti.

Všechny výsledky a úplný popis všech experimentů jsou prezentovány v úplné verzi disertační práce.

7 ZÁVĚR

Disertační práce se soustřeďuje na řešení problémů rozvrhování. Jsou zde podrobně popsány problémy rozvrhování proudové a zakázkové výroby a několik jejich reprezentací se zvláštním zaměřením na reprezentaci pomocí disjunktivního grafu. V práci jsou studovány modifikace problémů rozvrhování, které se často vyskytují v praktických aplikacích (rozvrhování výrobků se montážními a distribučními operacemi, zohlednění časů přípravy, dopravních a technologických časů, zohlednění dopravních dávek, volitelnost strojů). K optimalizaci řešení podle navržených účelových funkcí se zde používají stochastické heuristické metody (simulované žíhání, zakázané prohledávání, genetické algoritmy). V poslední části práce jsou popsány způsoby modelování neurčitostí v problémech rozvrhování pomocí teorie fuzzy množin.

Za nejdůležitější vlastní přínosy této práce je možno považovat tyto:

Byl vytvořen přehled o studované problematice. Popsali jsme použitelné metody řešení a matematické reprezentace problémů rozvrhování zakázkové a proudové výroby.

Na základě experimentů v modelovacím systému GAMS jsme stanovili meze velikosti rozvrhovací úlohy, která je efektivně řešitelná klasickými metodami.

Provedli jsme experimentální porovnání genetického algoritmu, simulovaného žíhání, zakázaného prohledávání a horolezeckého algoritmu z hlediska přesnosti, časové náročnosti a rychlosti konvergence ke globálnímu optimu.

Popsali jsme model problému rozvrhování zakázkové výroby, který zahrnuje nejdůležitější aspekty reálných výrobních podmínek (montážní a distribuční operace, dopravní dávky, doplňkové časy). K jeho řešení jsme použili reprezentaci pomocí tzv. rozšířeného disjunktivního grafu.

Navrhli jsme reprezentaci problému volitelnosti strojů pomocí chromozómů, skládajících se ze dvou nesourodých částí a navrhli jsme tři heuristické metody pro řešení problémů tohoto typu.

Navrhli a vytvořili jsme prototyp softwarového vybavení pro rozvrhování zakázkové výroby s aspekty reálné výroby. Kromě jiného jsme zobecnili algoritmus nalezení náhodného počátečního rozvrhu na problémy rozvrhování výrobků se složitou strukturou, který byl v tomto programu implementován.

V oblasti rozvrhování s neurčitými vstupními údaji jsme navrhli způsoby konstruování účelových funkcí pro problémy s neurčitými dobami trvání operací a s neurčitými termíny dokončení úloh. Navrhli jsme způsob statistického srovnání těchto kritérií založený na průměrné relativní chybě řešení, podle kterého jsme navržená kritéria mohli vzájemně srovnat.

Z provedených experimentů vyplynulo, že heuristické metody jsou vhodným nástrojem k řešení problémů rozvrhování. I když vždy nezaručují nalezení optimálního rozvrhu, jsou schopny poskytnout přijatelné řešení v reálném čase, a to i u rozsáhlých problémů, na které klasické metody nestačí. Z porovnání metod používajících kódování disjunktivním grafem a jinými klasickými způsoby (preferenčním seznamem) vyplývá, že jasně výkonnější jsou metody používající kódování disjunktivním grafem. Z kritérií pro problém rozvrhování s fuzzy dobami trvání operací a fuzzy termíny dokončení úloh bylo při předběžném srovnání s dalšími kritérii navrženými jinými autory na základě experimentů jako nejlepší vyhodnoceno kritérium navržené v této práci.

Přestože myšlenky stochastických heuristických metod byly publikovány už v 50. – 70. letech 20. století, rozšíření jejich použití pro konkrétní úlohy v praxi nebylo až do posledních desetiletí 20. století možné kvůli nedostatečnému výkonu výpočetní techniky. Rozšíření aplikace moderních metod rozvrhování v českém průmyslu je nyní podmíněno vznikem softwaru, který algoritmy rozebrané v disertační práci bude využívat. Zde je vidět zřetelná mezera na softwarovém trhu a zároveň úkol pro firmy, zabývající se vývojem systémů na podporu řízení výroby. Přínos takového komerčního programu bude vysoký, neboť nasazením heuristických metod lze podle zvoleného kritéria (například doby trvání rozvrhu) v přijatelné době zlepšit kvalitu rozvrhu, a tím podstatně snížit náklady na realizaci projektu.

Téma rozvrhování je velmi obsáhlé a je pochopitelné, že nelze vyčerpávajícím způsobem celé vyřešit v jedné disertační práci. Tato práce měla mimo jiné za cíl popsat a zpřehlednit danou problematiku a bude základem dalšího výzkumu v této oblasti.

8 LITERATURA

- [1] BAKER, K.R. A comparative study of flow shop algorithms. *European Journal of Operations Research*, 1995, vol. 23, pp. 62-73.
- [2] BEASLEY, J.E. *OR-Library*. The Management School Imperial College, London, 1996, Dostupné z: <<http://mscmga.ms.ic.ac.uk/pub/flowshop1.txt>>.
- [3] BLAZEWCZ, J. et. al. *Scheduling Computer and Manufacturing Processes*. Berlin : Springer-Verlag, 1996.
- [4] BLAZEWCZ, J.; PESCH, E.; MALGORZATA, S. The disjunctive graph machine representation of the job shop scheduling problem. *European Journal of Operations Research*, 2000, vol. 127, pp 317-331.
- [5] BURDA, J. *Rozvrhování zakázkové výroby*. Diplomová práce. Brno : VUT, FSI, 1999. 60 s.

- [6] CASEAU, Y.; LABURTHE, F. Improving Branch and Bound for Jobshop Scheduling with Constraint Propagation. *Proceedings of the 8th Franco – Japanese 4 Franco – Chinese Conference CCS'1995*.
- [7] CHARAMZA, P.; POPELA, P.; TLUSTÝ, P. a kol. *Modelovací systém GAMS*. Praha : MFF UK, 1993.
- [8] CHENG, R.; GEN, M.; TSUJIMURA, Y. A Tutorial Survey of Job-Shop Scheduling Problems Using Genetic Algorithms - I. Representation. *Computers & Industrial Engineering*, 1996, vol. 30, no. 4, pp. 983-997.
- [9] CURY, R.M.; KHATOR, S.K.; GAUTHIER, F.A.O.; BARCIA, R.M. Scheduling under Uncertainty: A Fuzzy Approach. *Proceedings of the 7th Annual Industrial Engineering Research Conference*, Banff (Canada), 1998, 11p.
- [10] DAUZERE-PERES, S.; ROUX, W.; LASSERRE, J.B. Multi-Resource Shop Scheduling with Resource Flexibility. *European Journal of Operational Research*, 1998, vol. 107, pp. 289-305.
- [11] FORST, F.G. Bicriterion Stochastic Scheduling on One or More Machines. *European Journal of Operational Research*, 1995, vol. 80, pp. 404-409.
- [12] FRENCH, S. *Sequencing and scheduling 1. Scheduling (Management)*. Ellis Horwood Limited, 1982.
- [13] GIFFLER, B.; THOMPSON, G. Algorithms for Solving Production Scheduling Problems. *European Journal of Operational Research*, 1960, vol. 8, pp. 487-503.
- [14] GRABOWSKI, Z.; NOWICKI, E; ZDRZALKA, S.S. A block approach for single machine scheduling with release dates and due dates. *European Journal of Operational Research*, 1985, vol. 26, pp. 278-285.
- [15] IVENS, P.; LAMBRECHT, M. Extending the Shifting Bottleneck Procedure to Real-Life Applications. *European Journal of Operational Research*, 1996, vol. 90, pp. 252-268.
- [16] ISHIBUCHI, H.; YAMAMOTO, N.; MURATA, T.; TANAKA, H. Genetic Algorithms and Neighborhood Search Algorithms for Fuzzy Flowshop Scheduling Problems. *Fuzzy Sets and Systems*, 1994, vol. 67, pp. 81-100.
- [17] ISHIBUCHI, H.; TANAKA, H.; MISAKI, S. Fuzzy Flow Shop Scheduling Problems by Simulated Annealing. In: Delgado, M., Kacprzyk, J., Verdegay, J.L. and Vila, M.A. (eds.), *Fuzzy Optimization*, Heidelberg : Physica-Verlag, 1994, pp. 351-363.
- [18] JAIN, A.S.; MEERAN, S. Deterministic Job Shop Scheduling: Past, Present and Future. *European Journal of Operations Research*, 1999, vol. 113, pp. 390-434.
- [19] JOHNSON, S.M. Optimal two- and three-stage production schedules with setup times included. *Naval Res. Logist*, 1954, quart. 1, pp. 61-68.
- [20] KARPÍŠEK, Z. Fuzzy Probability and its Properties. In *Proceedings of the 6th International Conference on Soft Computing MENDEL '2000*, Brno, 2000, p. 262-266.
- [21] KLAPKA, J.; DVOŘÁK, J.; POPELA, P. *Metody operačního výzkumu*. Brno : VUT, 2001.
- [22] KRISHNA, K.; GANESHAN, K.; JANAKI, R.D. Distributed Simulated Annealing Algorithms for Job Shop Scheduling. *IEEE Transactions on Systems, Man, and Cybernetics*, 1995, vol. 25, no. 7, pp. 1102-1109.
- [23] KURODA, M.; WANG, Z. Fuzzy Job Shop Scheduling. *International Journal of Production Economics*, 1996, vol. 44, pp. 45-51.
- [24] KVASNIČKA, V.; POSPÍCHAL, J.; TIŇO, P. *Evolučné algoritmy*. Bratislava : STU, 2000.

- [25] van LAARHOVEN, P.J.M.; AARTS, E.H.L.; LENSTRA, J.K. Job shop scheduling by simulated annealing, *Operational Research*, 1992, vol. 40, pp. 113-125.
- [26] MAJER, P.; DVOŘÁK, J. Fuzzy termíny dokončení a fuzzy doby zpracování úloh v rozvrhování proudové výroby. *Sborník přednášek k 5. ročníku konference Fuzzy logika, neuronové sítě a expertní systémy*, Luhačovice, 2000, s. 85-93.
- [27] MAJER, P. Reprezentace problému rozvrhování zakázkové výroby disjunktivním grafem. *Proceedings of XXVI. Seminar ASR '2001 Instrument and Control*. Ostrava, 2001, 6 s.
- [28] METROPOLIS, N. et. al. Equation for State Calculation for Fast Computing Machines. *J. Chem. Phys.*, 1963, vol. 21, pp. 1087-1092.
- [29] NOVÁK, V. *Fuzzy množiny a jejich aplikace*. Praha : SNTL, 1990.
- [30] NOWICKI, E.; SMUTNICKI, C. A Fast Taboo Search Algorithm for the Job Shop Problem. *Management Science*, 1996, vol. 42, no. 6, pp. 797-813.
- [31] OTTEN, R.H.J.M.; van GINEKEN, L.P.P.P. *The Annealing Algorithm*. Boston: Kluwer, 1989.
- [32] ÖZELKAN, E.C.; DUCKSTEIN, L. Optimal Fuzzy Counterparts of Scheduling Rules. *European Journal of Operational Research*, 1999, vol. 113, pp. 593-609.
- [33] PEZZELLA, F.; MERELLI, E. A tabu search method guided by shifting bottleneck for job shop scheduling problem. *European Journal of Operations Research*, 1998, vol. 120, pp. 297-310.
- [34] PLESNÍK, J. *Grafové algoritmy*. Bratislava : VEDA, 1983.
- [35] POPELA, P. Application of Stochastic Programming in Foundry. *Folia Mathematica*, 1988, vol. 7, pp. 117-139.
- [36] REEVES, C.R. (editor). *Modern Heuristic Techniques for Combinatorial Problems*. Oxford : Blackwell Scientific Publication, 1993.
- [37] SAKAWA, M.; MORI, T. An Efficient Genetic Algorithm for Job-Shop Scheduling Problems with Fuzzy Processing Time and Fuzzy Due date. *Computers & Industrial Engineering*, 1999, vol. 36, pp. 325-341.
- [38] SAKAWA, M.; KUBOTA, R. Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy due date through genetic algorithms. *European Journal of Operations Research*, 2000, vol. 120, pp. 393-407.
- [39] STRECK, I. *Operační a systémová analýza*. Brno : VUT , 1986.
- [40] ŠEDA, M.; DVOŘÁK, J. An Application of Heuristic Techniques to Permutation Flowshop Scheduling Problem. *Proceedings of the 4th International Conference on Genetic Algorithms, Optimization Problems, Fuzzy Logic, Neural Networks and Rough Sets MENDEL '98*, Brno, 1998, p. 132-138.
- [41] VIDECKÁ, Z. *Nový přístup k operativnímu řízení středněsériové výroby*. Disertační práce, Brno : VUT, FSI, 1999, 87 s.
- [42] YAMADA, T.; NAKANO, R. A Genetic Algorithm with Multi-Step Crossover for Job Shop Scheduling Problems. *Proceedings of the International Conference GALESIA '95*, Sheffield, 1995, p. 146–151.

V úplné verzi disertační práce je uvedeno celkem 69 titulů použité literatury.

9 SHRnutí

Tato práce se soustřeďuje na řešení problémů rozvrhování. Obecně je problém rozvrhování dán konečnou množinou výrobků, které je potřeba vyrobit a omezeným počtem výrobních strojů, které jsou k dispozici. Technologický postup určuje pro každý výrobek operace ve stanoveném pořadí. Každé operaci je přidělen určitý stroj.

O rozvrhování zakázkové výroby mluvíme v případě, kdy pořadí strojů smí být pro každý výrobek různé. Ve speciálním případě, kdy je pořadí strojů pro všechny výrobky stejné, mluvíme o rozvrhování proudové výroby. Úkolem rozvrhování je najít nejlepší rozvrh, tj. optimální pořadí operací na jednotlivých strojích. Jako kritérium rozvrhu je možné použít celkovou dobu trvání výroby, délky prostojů strojů, ztráty spojené s nesplněním prací v požadovaných termínech, stupeň rozpracovanosti apod. Uvedená kritéria jsou minimalizačního charakteru.

Problémy rozvrhování patří mezi složité kombinatorické problémy a proto se k jejich řešení používají heuristické metody. Nejčastěji používané heuristické metody jsou simulované žíhání, tabu search a genetické algoritmy. Když aplikujeme heuristické metody, je důležitá volba vhodné reprezentace dat. V práci je popsáno několik reprezentací dat problému rozvrhování zakázkové výroby se zvláštní pozorností k reprezentaci pomocí disjunktivního grafu.

Těžištěm disertační práce je studium takových modelů a metod řešení problémů rozvrhování zakázkové výroby, které reflektují reálné výrobní podmínky lépe než základní model. Jedná se o zohlednění časů přípravy, dopravních a technologických časů, zohlednění dopravních dávek, rozvrhování výrobků se složitou strukturou a problém volitelnosti strojů. K zahrnutí většiny těchto aspektů používáme model tzv. rozšířeného disjunktivního grafu. Je zde popsán námi navržený a realizovaný počítačový program pro rozvrhování zakázkové výroby se zohledněním reálných výrobních podmínek.

Poslední část práce se zabývá modely rozvrhování, které zahrnují neurčitě údaje. K modelování neurčitostí používáme teorii fuzzy množin. Termíny dokončení úloh a doby trvání operací jsou modelovány lichoběžníkovými fuzzy čísly. Je zde navrženo několik způsobů konstruování účelových funkcí pro tyto fuzzy případy.

Popsané modely i metody byly implementovány na počítači. V jednotlivých experimentech bylo provedeno srovnání navržených metod, reprezentací a kritérií, založené na použití standardních i vlastních etalonových příkladů.

10 SUMMARY

This work deals with solving of scheduling problems. The scheduling problem consists of a set of a number of different machines that process jobs from a set of jobs. Each job is composed of an ordered list of operations, each of which is determined by the machine and the processing time it requires.

The flow shop scheduling is characterized by the same order of machines for all jobs. In the job shop scheduling the machine orderings can be different for individual jobs. The aim is to find the optimal schedule, i. e. the optimal job orderings on machines. The decision criterion is usually some function of the job completion times, which is to be minimized.

Scheduling is a hard combinatorial problem and therefore real-world problems are usually solved by heuristic methods. Frequently used heuristic methods are simulated annealing, taboo search and genetic algorithms. When applying a heuristic method, the choice of problem representation is important. In this paper, several representations of job shop are described, with special focus on disjunctive graph-based representation.

Primarily, this work describes a model of job shop scheduling, which covers the most frequent conditions of real-life production: assembly and distribution operations, setup and transfer times, traffic lots and overlapping processes, interchangeability of machines. As a modeling tool, we

choose an extended disjunctive graph. We developed a computer program for real-life job shop scheduling.

The last section deals with models of scheduling under uncertainty. Our approach to modeling uncertainty is based on the use of the fuzzy set theory. Due dates of jobs and processing times of job operations are modeled by trapezoidal fuzzy numbers. We describe some approaches to the construction of decision criteria for the fuzzy cases.

Described models and methods were implemented on a computer. Proposed methods, representations and criteria were compared in individual experiments based on standard and our own benchmarks.

11 CURRICULUM VITAE AUTORA

Ing. Petr Majer

narozen: 1976 ve Strakoncích
bydliště: Voskovcova 4, Olomouc, 779 00
národnost: česká

Vzdělání

1999 – 2003 doktorské studium na Fakultě strojního inženýrství, VUT v Brně, obor technická kybernetika
1994 – 1999 studium na Fakultě strojního inženýrství, VUT v Brně, obor inženýrská informatika
1990 – 1994 SPŠ v Prostějově obor strojírenství

Pracovní aktivity

od 11/2002 Počítačová řešení SWPRO, s.r.o. Olomouc, pracovník oddělení vývoje

12 SEZNAM PUBLIKACÍ AUTORA

1. ŠEDA, M.; DVOŘÁK, J.; MAJER, P. Scheduling with Fuzzy Processing Times in a Flow Shop. *Proceedings of the 7th European Congress on Fuzzy and Intelligent Techniques & Soft Computing EUFIT '99*. Aachen (Germany), 1999, 6 pages, on CD-ROM \proceedings\fsb-12-12834_p.pdf; Abstracts of the Papers, p. 159-160.
2. DVOŘÁK, J.; ŠEDA, M.; MAJER, P. Flow Shop Scheduling with Fuzzy Due Dates and Fuzzy Processing Times. *Proceedings of the 5th International Conference on Soft Computing MENDEL '99*. Brno, 1999, p. 228-235.
3. MAJER, P. *Fuzzy rozhovávání*. Diplomová práce. Brno : VUT, FSI, 1999, 65 s.
4. MAJER, P.; DVOŘÁK, J.; ŠEDA, M. Rozhování s fuzzy termíny dokončení úloh a fuzzy trváním operací. *Sborník přednášek k 4. ročníku konference Fuzzy logika, neuronové sítě a expertní systémy*. Hrubá Skála, 1999, s. 65-72.
5. DVOŘÁK, J.; MAJER, P. Application of Genetic Algorithms to Scheduling and Lot Sizing in a Flow Shop. *Proceedings of the 6th International Conference on Soft Computing MENDEL '2000*. Brno, 2000, 7 p.

6. DVOŘÁK, J.; MAJER, P. Fuzzy Due Dates and Fuzzy Processing Times in Flow Shop Scheduling. *Proceedings of the 8th East West Zittau Fuzzy Colloquium 2000*. Zittau (Germany), 2000, 7 p.
7. MAJER, P.; DVOŘÁK, J. Fuzzy termíny dokončení a fuzzy doby zpracování úloh v rozvrhování proudové výroby. *Sborník přednášek k 5. ročníku konference Fuzzy logika, neuronové sítě a expertní systémy*. Luhačovice, 2000, s. 85-93.
8. MAJER, P. Uplatnění fuzzy přístupů v rozvrhování proudové výroby. *II. Sborník příspěvků doktorandů k Pedagogicko-vědecké konferenci u příležitosti 100. výročí založení Fakulty strojního inženýrství*. Brno : VUT, 2000, s. 181-184.
9. MAJER, P. Reprezentace problému rozvrhování zakázkové výroby disjunktivním grafem, *Proceedings of XXVI. Seminar ASR '2001 Instrument and Control*. Ostrava, 2001, 6 s.
10. MAJER, P. Modelování neurčitostí v problémech rozvrhování výroby. *Sborník přednášek k XXIII. mezinárodnímu podzimnímu kolokviu Pokroky Tvorby a využití simulačních modelů ASIS 2001*. Velké Losiny, 2001, 6 s.
11. MAJER, P. *Moderní metody rozvrhování výroby*. Pojednání o disertační práci. Brno : VUT, FSI, 2001, 25 s.
12. MAJER, P.; DVOŘÁK, J. Adaptations of the job shop scheduling problem to real-life applications. *Proceedings of the 8th International Conference on Soft Computing MENDEL '02*. Brno, 2002, p. 357-362.
13. MAJER, P.; DVOŘÁK, J. Optimalizační kritéria pro fuzzy rozvrhování. *Sborník přednášek k 11. ročníku semináře Moderní matematické metody v inženýrství*. Dolní Lomná u Jablunkova, 2002, s. 31-35.
14. DVOŘÁK, J.; MAJER, P. Fuzzy Job Shop Scheduling. *Proceedings of the 10th East West Zittau Fuzzy Colloquium 2002*. Zittau (Germany), 2002, p. 284-291.
15. DVOŘÁK, J.; MAJER, P. Scheduling With Fuzzy Processing times. *Proceedings of the 2nd International Conference APLIMAT*