

EFFECTIVE PRODUCTION PLANNING AND SCHEDULING

Michal Křen

Master Degree Programme (2), FIT BUT

E-mail: xkrenm00@stud.fit.vutbr.cz

Supervised by: Martin Hrubý

E-mail: hrubym@fit.vutbr.cz

Abstract: This paper presents optimization and simulation tools for production planning. A schedule of production created by those tools can use company resources very effectively. Production planning is formally described as the well-known resource-constrained project scheduling problem (RCPSP). The optimizer uses a genetic algorithm for solving the RCPSP and the simulator is based on the discrete event simulation (DES).

Keywords: planning, scheduling, optimization, simulation, RCPSP, genetic algorithm, MES

1 ÚVOD

V současné době se ve výrobních firmách střetávají dva protichůdné požadavky. Vedoucí výroby požadují dopředu znát přesně požadavky výroby, aby dokázali efektivně naplánovat jednotlivé operace a údržbu s ohledem na společné strojní vybavení ostatní výroby. Naopak vedoucí logistiky musí neustále měnit požadavky výroby s ohledem na aktuální potřeby zákazníka. Tento konflikt lze vyřešit nasazením SW nástroje, který dokáže vytvořit výrobní plán pro celou výrobu a efektivně tak využít zdroje. Tímto plánovacím problémem, který lze formálně reprezentovat jako „Resource-Constrained Project Scheduling Problem“ (RCPSP), se zabývá tato práce.

2 RESOURCE-CONSTRAINED PROJECT SCHEDULING PROBLEM

RCPSP jakožto zobecnění „Job-Shop“ je NP těžký problém a lze jej formálně popsat jako [3]:

- $\mathcal{J} = \{0, 1, \dots, n, n+1\}$ označující množinu všech operací, přičemž operace s číslem 0 odpovídá začátku projektu a operace s číslem $n+1$ odpovídá konci projektu.
- p_j vyjadřující dobu trvání nepřerušitelné operace $j \in \mathcal{J}$. Platí, že $p_0 = p_{n+1} = 0$.
- \mathcal{P}_j označuje množinu všech předchůdců operace j .
- $\mathcal{K} = \{1, \dots, K\}$ reprezentující množinu všech obnovitelných zdrojů.
- R_k určující kapacitu obnovitelného zdroje $k \in \mathcal{K}$, která je stejná po celou dobu projektu.
- $r_{j,k}$ určující požadavky úlohy j na obnovitelný zdroj k . Platí, že $r_{0,k} = r_{n+1,k} = 0$ pro $\forall k \in \mathcal{K}$.

Zavedeme F_j značící čas ukončení provádění operace j . Cílem RCPSP je nalezení proveditelného časového přiřazení operací na zdroje (rozvrhu) s minimálním F_{n+1} . Rozvrh je proveditelný, pokud dodržuje omezení na kapacitu zdrojů (2) a omezení na předchůdce operací (1). Množina všech úloh, které jsou vykonávány v čase t , je popsána jako $\mathcal{A}(t) = \{j \in \mathcal{J} \mid F_j - p_j \leq t < F_j\}$.

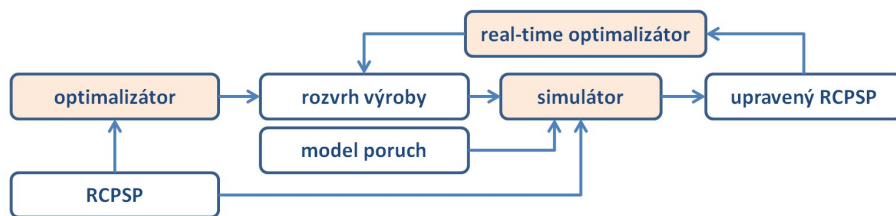
$$F_h \leq F_j - p_j \quad j = 1, \dots, n+1; h \in \mathcal{P}_j \quad (1)$$

$$\sum_{j \in \mathcal{A}(t)} r_{j,k} \leq R_k \quad k \in \mathcal{K}; t \geq 0 \quad (2)$$

K řešení RCPSP lze přistupovat dvěma způsoby. Zaprvé, hledat optimální řešení přesnými metodami typu celočíselné programování, CSP nebo SAT. Druhý způsob je využít stochastických optimalizačních metod a spokojit se pouze se sub-optimálním řešením, které nemusí být optimální. Prohledávaný stavový prostor je obrovský díky stům až tisícům operací, a tudíž přesné metody zde narážejí na svá výpočetní omezení, a proto se s výhodou využívají stochastické metody. V této práci jsou použity genetické algoritmy, protože v porovnání s ostatními metodami vykazují příznivé výsledky [1].

3 NÁSTROJE NA PLÁNOVÁNÍ VÝROBY

Koncepce tohoto projektu se skládá ze třech klíčových nástrojů, jak lze vidět na obrázku 1. Na začátku procesu (týdne) se dle požadovaných zakázek určí nutné operace a vytvoří se RCPSP. Poté optimalizátor sestaví sub-optimální plán výroby, který je následně možné otestovat v simulátoru a získat nové znalosti o daném plánu. Simulátor také umožňuje zastavit svůj běh při narušení původního plánu, vytvořit upravený RCPSP a požádat o přeplánování. Real-time optimalizátor dokáže upravit rozvrh výroby s ohledem na aktuální situaci ve výrobě.



Obrázek 1: Základní koncepce celého systému.

3.1 OPTIMALIZÁTOR ŘEŠENÝ GENETICKÝMI ALGORITMY

Optimalizátor po načtení RCPSP spustí výpočet genetického iteračního algoritmu. Jedinec (plán) je reprezentován jako posloupnost operací $\lambda = (j_1, \dots, j_J)$, přičemž pro každou operaci platí, že se její předcházející operace objeví v seznamu dříve než ona sama (3).

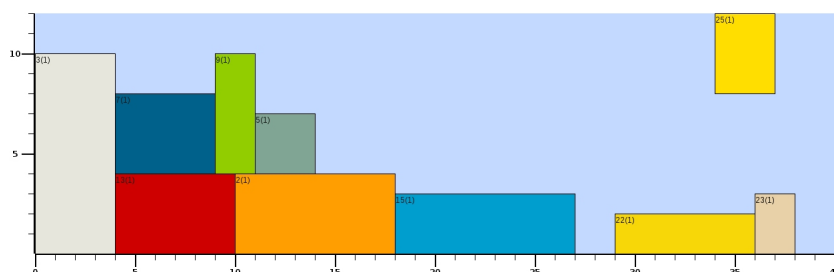
$$\mathcal{P}_i \subseteq \{0, j_1, \dots, j_{i-1}\} \text{ pro } i = 1, \dots, J. \quad (3)$$

Nyní je nutné vysvětlit mechanismus, který dokáže takto popsaného jedince převést na proveditelný rozvrh. Slouží k tomu algoritmus „Serial schedule generation schemes“ (SSGS), který je jádrem většiny heuristických funkcí při řešení RCPSP. SSGS v každém kroku svého výpočtu naplánuje jednu operaci na nejdříve možný čas s ohledem na splnění omezujících podmínek (1) a (2). Existuje ještě paralelní verze algoritmu, která postupuje po čase, ale nevykazuje tak dobré výsledky. SSGS je podrobně popsán v [2].

Počáteční populace je vytvořena pomocí SSGS, přičemž v každém kroku je z operací připravených k naplánování vybrána jedna náhodně. Zcela náhodný výběr neposkytuje příliš dobré výsledky, proto se například upřednostňují operace s menším nejpozdějším možným koncovým časem (LFT). Tento údaj lze spočítat pomocí zpětné rekurze nad grafem předchůdců. Pro výběr rodičů lze použít běžně používané metody rulety nebo turnaje. V případě operátoru jednobodového křížení je nejprve náhodně zvoleno číslo q v rozsahu $1 \leq q < J$. Poté nově vzniklý jedinec bude obsahovat operace na pozicích $i = q + 1, \dots, J$ od matky. Zbytek operací na pozicích $i = q + 1, \dots, J$ bude přejat od otce s faktem, že již použité operace se nebudou znovu opakovat. Operátor mutace s jistou malou pravděpodobností náhodně prohodí v plánu dvě operace, pokud zůstane zachováno omezení (3).

3.2 SIMULÁTOR A MODEL PORUCH

Simulátor ve své podstatě poskytuje data, která by v reálné výrobě přicházela z monitorovacích terminálů. Jeho základem je diskretní simulace (DES) a na počátku jsou do kalendáře vloženy začátky všech operací. Rovněž je do kalendáře načten model poruch ve výrobě. Těmi mohou být například jiná doba trvání operace nebo nefunkčnost stroje. U každé poruchy je možné definovat, jakého je typu, jaké operace či stroje ovlivní a s jakou pravděpodobností nastane. Aktuální implementace simulátoru také umožňuje vizualizovat výrobní plán pomocí Ganttova diagramu, jak je zobrazeno na obrázku 2.



Obrázek 2: Vygenerovaný Ganttův diagram zobrazující využití zdroje (osa y) v čase (osa x).

3.3 REAL-TIME OPTIMALIZÁTOR

Optimalizátorem vytvořený rozvrh výroby je do určité míry pouze teoretickým rozvrhem. S vysokou pravděpodobností v průběhu provádění výroby nastane nějaká neplánovaná událost popsána v modelu poruch. Obecně vzato existují dva přístupy, jak se s tímto vypořádat. Cílem aktivního přístupu je již na začátku vytvořit takový plán, který bude co nejvíce odolný vůči případným problémům ve výrobě. Lze toho docílit například vkládáním časových prodlev do již existujícího rozvrhu. Naopak pasivnější přístup čeká až daný problém vznikne a snaží se patřičně upravit rozvrh.

4 ZÁVĚR

V rámci této práce se podařilo navrhnout koncepci SW plánování výrobních operací a také se podařilo úspěšně implementovat optimalizátor a funkční základ simulátoru. Optimalizátor byl testován na sérii 480 úloh o 30 operacích (J30) z PSPLIB. Při omezení na maximálně 1000 sestavených plánů vykazuje optimalizátor průměrnou odchylku od optimálních řešení 1,39 %, přičemž největší odchylka byla 9,86 %. Výsledky jsou srovnatelné s podobnými již prezentovanými metodami [1]. Cílem dalšího zkoumání je implementovat vhodnou metodu do real-time optimalizátoru.

V současné době také probíhají jednání s firmou Visteon-Autopal, která vyrábí chladiče do automobilů. SW nástroje popsané v této práci by měly této společnosti vyřešit problém s efektivním využíváním pájecí pece. Firma by tak mohla vyrábět více při stávajících kapacitách.

REFERENCE

- [1] Hartmann, S.: A Competitive Genetic Algorithm for Resource-Constrained Project Scheduling. *Naval Research Logistics*, ročník 45, 1998: s. 733-750.
- [2] Kolisch, R., Hartmann, S.: Heuristic algorithms for solving the resource-constrained project scheduling problem: Classification and computational analysis. *Project scheduling: Recent models, algorithms and applications*, 1999: s. 147-178.
- [3] Artigues, C., Demassey, S. a Néron, E.: *Resource-Constrained Project Scheduling: models, algorithms, extensions and applications*. Hoboken: John Wiley & Sons, 2010. ISBN 047039384X.