

Semestrální práce
Kombinatorické optimalizace

Rozvrh vedoucích pracovníků při výstavbě elektrárenského zařízení.

1. Zadání a kategorizace

Na výstavbě elektrárenského zařízení se podílejí tři typy zaměstnanců: vedoucí směny, vedoucí pracovník a běžný zaměstnanec. Celé staveniště je rozděleno do dvou úseků. V prvním úseku se pracuje na dvě směny (6:00 – 18:00 a 18:00 – 6:00), ve druhém je zaveden třísměnný provoz (6:00 – 14:00, 14:00 – 22:00 a 22:00 – 6:00).

Aby práce na výstavbě mohla probíhat, musí být v každém úseku a při každé směně přítomen alespoň minimální počet vedoucích pracovníků. Cílem této semestrální práce bude optimální rozvrh směn (rovnoměrné zatížení všech vedoucích pracovníků, ohled na spokojenost zaměstnanců – co největší bloky směn) pro N vedoucích pracovníků na D dní práce dopředu.

Omezující podmínky kladené na rozvrháře:

1. Pokrytí všech směn na obou úsecích minimálním počtem vedoucích pracovníků. [HC]
2. Vedoucí může pracovat maximálně 40 hodin týdně. [HC]
3. Vedoucí může pracovat maximálně 12 hodin denně. [HC]
4. Maximálně tři noční směny za týden. [HC]
5. Minimálně dva volné víkendy v měsíci. [SC]
6. Plánovat co největší bloky směn. [SC]

Úlohu lze formalizovat jako Employee Timetabling Problem (ETP), tedy problém rozvrhování.

HC – Hard Constraint
SC – Soft Constraint

2. Související práce

Úloha vytváření rozvrhu směn je praktický problém, který se řeší ve velkém množství organizací a výrobních podniků. Algoritmické řešení může rozvrhnout měsíčně uspořít až desítky hodin náročné práce a to s lepšími výsledky. V odborné literatuře se problém rozvrhování nachází pod pojmy timetabling, rostering a nurse scheduling problem.

Řešení můžeme hledat pomocí úplné metody nebo pomocí nějakých heuristik. Úplné metody prochází celý stavový prostor a vždy vrací nejlepší možný výsledek. Algoritmus je tedy optimální, ale časově náročný. V článku [1] jsou porovnávány dvě úplné metody Integer Linear Programming (dále ILP) a Constraint Programming (dále CP), ze kterého vyplývá, že metoda ILP je rychlejší a při pozastavení po nějakém čase vracela lepší výsledky než CP.

Další zrychlení přináší použití Boolean Satisfiability Problem (dále SAT). V článku [2] bylo provedeno srovnání komerčního generického ILP solveru CPLEX [4] a PBS[5] solveru, který je postaven na základě SAT. PBS pracuje s konjunktivní normální formou (CNF) i s pseudo-boolean (PB) podmínkami. Ze srovnání vyplývá, že SAT lze použít na větší instanci problému.

Druhý přístup k řešení problému je nasazení nějakých heuristik, kde prohledáváme pouze část stavového prostoru. Řešení tedy nemusí být optimální, ale ve většině případů je uspokojivé. Řešení dostáváme v mnohonásobně kratším čase než při použití úplných metod. Jednou z možností jsou genetické algoritmy [3], které patří mezi evoluční algoritmy, které využívají evoluční procesy křížení, mutace a přirozený výběr. Vycházíme z nějaké počáteční populace a pomocí evolučních procesů (nějakých jejich kombinací) získáváme novou populaci. Tento krok opakujeme dokud nedosáhneme určité generace populace.

3. Popis řešení

Úlohu rozvrhu směn vedoucích pracovníků budu řešit pomocí úplné metody ILP. Při vytváření modelu vycházím do značné míry z článku [1]. Jako ILP solver jsem se rozhodl použít CPLEX[4] od firmy IBM, který umožňuje úlohu pozastavit a vrátit doposud nejlepší nalezené řešení. CPLEX je komerční produkt, proto v této úloze využiji jeho trial verzi, která je omezená na 90 dní, 500 proměnných a 500 podmínek, což je pro účel této úlohy dostačující.

3.1 Model

Model vychází do značné míry z článku [1], kde se řeší typově podobná úloha.

3.1.1 Proměnné

Jako proměnné úlohy definuji X_{ijk} , pro každého vedoucího pracovníka (dále VP) přiděluje určitou směnu v určitý den. Kde $1 \leq i \leq N$ indexuje VP; $1 \leq j \leq D$ indexuje den v rámci intervalu plánování; $1 \leq k \leq S$ vyjadřuje druh směny. $S = 5$ pro oba úseky staveniště dohromady.

X_{ijk} je celočíselná proměnná, která nabývá hodnot 0 nebo 1;

$$X_{ijk} = \begin{cases} 1 & \text{pokud VP } i \text{ má službu } k \text{ v den } j \\ 0 & \text{v jiném případě} \end{cases}$$

Další proměnnou je pomocná proměnná Y_{ih} , která říká o každém VP i , zda musí v sobotu nebo v neděli v týdnu h do práce nebo má volno.

$$Y_{ih} = \begin{cases} 1 & \text{pokud VP } i \text{ má službu v sobotu nebo v neděli v týdnu } h \\ 0 & \text{v jiném případě} \end{cases}$$

V následující tabulce je uveden příklad požadavků na obsazenost směn R_{jk} na obou úsecích výstavby.

ÚSEK	SMĚNA	OZNAČENÍ	PO - PÁ	SO - NE
1	6:00 – 18:00	S12D	3	2
	18:00 – 6:00	S12N	3	2
2	6:00 – 14:00	S08R	2	1
	14:00 – 22:00	S08O	2	1
	22:00 – 6:00	S08N	2	1

Tab.1 – Obsazení směn VP

Aby bylo možné vytvořit spravedlivý rozvrh, tedy každý VP je přibližně stejně vytížen a mají přibližně stejný počet volných víkendů v měsíci, je nutné definovat penalizační funkci pomocí, které lze ohodnotit náročnost jednotlivých směn na obou úsecích pracoviště. Příklad penalizační funkce P_{jk} je uveden v tabulce Tab.2.

ÚSEK	SMĚNA	OZNAČENÍ	PO - PÁ	SO - NE
1	6:00 – 18:00	S12D	2	3
	18:00 – 6:00	S12N	2	3
2	6:00 – 14:00	S08R	1	2
	14:00 – 22:00	S08O	1	2
	22:00 – 6:00	S08N	1	2

Tab.2 – Penalizační funkce

3.1.2 Kriteriaální funkce

Kriteriaální funkci pro rovnoměrné zatížení všech VP lze formulovat jako

$$\min P_{\max} - P_{\min}$$

, kde P_{\max} je horní hranice a P_{\min} je spodní hranice celkové penalizace každého VP. Q_i je pomocná proměnná určující penalizace za pracovní víkendy s váhou 10 (viz podmínka 5).

$$P_{\min} - \left(\sum_{j=1}^D \sum_{k=1}^S X_{ijk} \cdot P_{jk} + 10 \cdot Q_i \right) \leq 0 \quad \forall i=1, \dots, N$$

$$P_{\max} - \left(\sum_{j=1}^D \sum_{k=1}^S X_{ijk} \cdot P_{jk} + 10 \cdot Q_i \right) \geq 0 \quad \forall i=1, \dots, N$$

$$Q \in \langle -2; 2 \rangle$$

3.1.3 Omezující podmínky

Podmínku číslo 1, tedy požadavek na pokrytí všech směn na obou úsecích požadovaným počtem vedoucích pracovníků, lze formulovat jako

$$\sum_{i=1}^N X_{ijk} = R_{jk} \quad \forall k=1, \dots, S \quad \forall j=1, \dots, D$$

, kde R_{jk} je počet požadovaných VP v den j na směně k .

Druhou podmínku, že každý VP může pracovat 40 hodin týdně formulují jako

$$\sum_{j=7h-6}^{7h} \sum_{k=1}^S X_{ijk} \cdot M_k \leq 40 \quad \forall i=1, \dots, N \quad \forall h=1, \dots, \frac{D}{7}$$

, kde M_k je počet hodin směny k .

Vedoucí pracovník může pracovat maximálně 12 hodin denně. Tuto podmínku lze formulovat takto:

$$\sum_{k=1}^S X_{ijk} \leq 1 \quad \forall i=1, \dots, N \quad \forall j=1, \dots, D$$

Čtvrtá podmínka říká, že VP může sloužit maximálně tři noční směny za týden. Formulace:

$$\sum_{j=7h-6}^{7h} X_{ij2} + X_{ij5} \leq 3 \quad \forall i=1, \dots, N \quad \forall h=1, \dots, \frac{D}{7}$$

, kde h indexuje týden, kde h je $1 \leq h \leq \frac{D}{7}$.

Pátou podmínkou je měkká podmínka, která vyjadřuje preference VP. Říká, že každý VP chce alespoň 2 volné víkendy v plánovacím horizontu jednoho měsíce. Podmínku formulují jako

$$\sum_{k=1}^S X_{i,(7h-1),k} + \sum_{k=1}^S X_{i,7h,k} - 10 \cdot Y_{ih} \leq 0 \quad \forall i=1, \dots, N \quad \forall h=1, \dots, \frac{D}{7}$$

$$Y_{ih} \leq \sum_{k=1}^S X_{i,(7h-1),k} + \sum_{k=1}^S X_{i,7h,k} \quad \forall i=1, \dots, N \quad \forall h=1, \dots, \frac{D}{7}$$

$$\sum_{h=1}^{\frac{D}{7}} Y_{ih} = 2 + Q_i \quad \forall i=1, \dots, N$$

$$Q_i \in \langle -2; 2 \rangle$$

, kde pomocná proměnná Y_{ih} říká o každém VP i , zda musí v sobotu nebo v neděli v týdnu h do práce $Y_{ih} = 1$ nebo má volno $Y_{ih} = 0$. Q_i je proměnná určující penalizace za pracovní víkendy. Pokud VP i pracuje přesně přes dva víkendy, tak součet Y_{ih} přes všechny týdny h je roven 2, takže penalizace Q_i pro VP i je rovna 0; pokud pracuje méně víkendů, je Q_i záporná a naopak pokud více, tak je kladná.

3. Experimenty

Jako ILP solver pro experimenty jsem se rozhodl použít CPLEX[4] od firmy IBM, který umožňuje úlohu pozastavit a vrátit doposud nejlepší nalezené řešení. CPLEX je komerční produkt, proto v této úloze využiji jeho trial verzi, která je omezená na 90 dní, 500 proměnných a 500 podmínek, což je pro účel této úlohy dostačující. Počet proměnných v úloze lze spočítat takto

$$var_{count} = P_{max} + P_{min} + N \cdot D \cdot S + N \cdot \frac{D}{7} + N$$

a počet podmínek takto

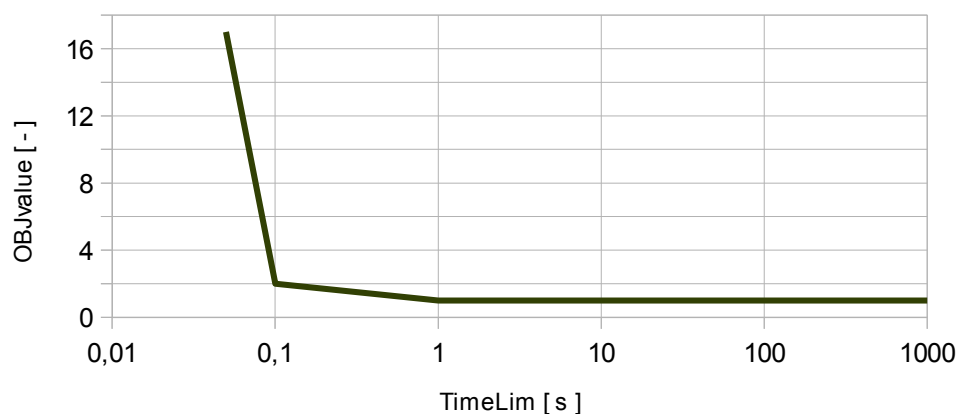
$$con_{count} = 2 \cdot N + S \cdot D + N \cdot \frac{D}{7} + N \cdot D + \frac{D}{7} \cdot N + \frac{D}{7} \cdot N + \frac{D}{7} \cdot N + N.$$

Následující tabulka uvádí pět experimentálních instancí problému rozvrhování, kde #VP je počet vedoucích pracovníků, #Dní je plánovací horizont, R_{jk} je tabulka požadavků na obsazení jednotlivých směn (S08R, S08O, ..., S12N), TiLim je čas v sekundách, kdy se případně úloha pozastaví a vrátí nejlepší dosud nalezený výsledek, Obj představuje hodnotu kritériální funkce, tedy $P_{\max} - P_{\min}$, P_{\max} je horní hranice a P_{\min} spodní hranice celkové penalizace každého VP. CPUtime udává celkový procesorový čas běhu instance a konečně Mem udává celkovou paměťovou náročnost běhu dané experimentální instance problému.

#VP	#Dní	Požadavky R _{jk}	S08R	S08O	S08N	S12D	S12N	TiLim	Obj	P _{max}	P _{min}	CPU _{time} [S]	Mem [MB]
13	7	PO-PA	2	2	1	2	X	100	7	13	6	100	17,8
		SO-NE	1	1	1	0							
13	7	PO-PA	2	2	1	2	2	100	Not Solved. (Rozvrh nelze sestavit)				
		SO-NE	1	1	1	2	1						
8	14	PO-PA	2	1	1	X	X	0,05	17	19	2	0,05	0,01
		SO-NE	1	1	0								
8	14	PO-PA	2	1	1	X	X	1	1	11	10	1	0,01
		SO-NE	1	1	0								
7	21	PO-PA	2	2	1	X	X	300	0	29	29	1,54	0,01
		SO-NE	1	1	1								

Tab. 3 – Experimentální instance problému

ILP je úplná metoda, prochází tedy celý stavový prostor a vždy vrací nejlepší možný výsledek. Algoritmus je tedy optimální, ale časově náročný. Na základě experimentů s instancemi 3 resp. 4 (Tab.3 řádek 3 resp. 4) bylo zjištěno, že dostatečně dobré výsledky lze získat už v relativně krátkém čase běhu výpočtu. Při pozastavení v čase jedné sekundy už získáváme OBJ rovnu jedné. Závislost kvality rozvrhu na době běhu výpočtu dané instance ilustruje následující graf.



Graf 1 - Závislost kvality rozvrhu na době běhu výpočtu.

Pro úplnost ještě uvádím příklad výstupu programu pro menší instanci problému. Ukázka rozvrhu pro šest vedoucích pracovníků s plánovacím horizontem jeden týden pro třísměnný provoz na prvním úseku staveniště.

	PO	UT	ST	CT	PA	SO	NE
VP1:	S08R	S08R	--	--	S08O	S08R	S08O
VP2:	S08N	--	S08O	S08N	S08N	--	S08O
VP3:	--	S08N	--	S08R	S08N	S08O	S08R
VP4:	S08O	--	S08N	S08O	S08R	S08O	--
VP5:	S08N	S08O	S08R	--	--	S08N	S08N
VP6:	--	S08N	S08N	S08N	--	S08R	S08R

Output 1 – Ukázka rozvrhu

4) Závěr

V této práci jsem se zabýval optimalizačním problémem rozvrhu lidských zdrojů v tří a dvousměnném provozu. V literatuře je tento problém znám jako Employee Timetabling Problem nebo také Rostering. Problém jsem řešil pomocí metody ILP, která byla obsahem tohoto kurzu. Po konzultaci bylo, pro velkou náročnost formulace v ILP, v matematickém modelu i v samotné implementaci upuštěno od šesté podmínky, tedy plánovat co největší bloky směn.

ILP je metoda úplná, prochází tedy celý stavový prostor a vždy vrací nejlepší možný výsledek. Algoritmus je optimální, ale časově náročný. Na základě experimentů bylo zjištěno, že dostatečně dobré výsledky lze získat už v relativně krátkém čase běhu výpočtu. Volně dostupný ILP solver glpk, který byl používán v rámci tohoto kurzu, neumožňuje pozastavení běhu výpočtu po uplynutí stanoveného časového limitu. Rozhodl jsem se tedy použít CPLEX[4] od firmy IBM.

CPLEX je komerční produkt, proto v této úloze využiji jeho trial verzi, která je omezená na 90 dní, 500 proměnných a 500 podmínek. Díky tomuto omezení nebylo možné úlohu testovat na rozsáhlých instancích, ale pro demonstraci správnosti řešení, jsou tato omezení dostačující.

5) Zdroje

- [1] LORRAINE TRILLING, ALAIN GUINET, DOMINIQUE LE MAGNY - *Nurse scheduling using interger linear programming and constraint programming*. 2006.
- [2] FADI ALOUL, BASKAR AL-RAWI, ANAS AL-FARRA, BASEL AL-ROH – *Solvin Employee Timetabling Problem Using Boolean Satisfiability*. 2006.
- [3] HITOSHI KANO, M. KONDO, M. SUGIMOTO – *Solving timetabling problems using genetic algorithms based on minimizing conflict heuristics*, 2002.
- [4] web: Domovská stránka projektu Cplex.
<http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/>, stav z 17.3. 2012
- [5] web: Domovská stránka projektu PBS.
<http://www.aloul.net/Tools/pbs/pbs4.html>, stav z 17.3. 2012