Úvod do kombinatorické optimalizace

Zdeněk Hanzálek, Přemysl Šůcha, Jan Kelbel, Jiří Trdlička
Michal Kutil, Zdeněk Baumelt, Roman Čapek
hanzalek@fel.cvut.cz

ČVUT FEL Katedra řídicí techniky

14. února 2012

Obsah přednášky

Obsah přednášek

1. Příklady aplikací a základní pojmy. Test 0.
2. Celočíselné lineární programování - algoritmy.
3. Formulace problémů pomocí celočíselného lineárního programování.
4. Nejkratší cesty. Test I.
5. Toky a řezy v sítích. Párování v bipartitních grafech.
6. Multi-komoditní toky.
7. Problém batohu, pseudo-polynomiální a aproximační algoritmy.
8. Úloha obchodního cestujícího. Test II.
9. Rozvrhování na jednom procesoru.
10. Paralelní procesory.
11. Rozvrhování projektu s časovými omezeními.
12. Programování s omezujícími podmínkami.

1 - motivace

4,5,6 - většinou polynomiální složitost

7,8,9,10,11 - klasické NP-obtížné problémy

2,3,12 - obecné metody deklarativního programování

Obsah cvičení

1. Seznámení s experimentálním prostředím a knihovnou pro optimalizaci
2. Celočíselné lineární programování
3. Aplikace celočíselného lineárního programování
4. Samostatná úloha I - zadání a kategorizace
5. Nejkratší cesty
6. Samostatná úloha II - rešerše literatury a prezentace řešení
7. Aplikace toků a řezů v sítích
8. Samostatná úloha III
9. Rozvrhování
10. Test III
11. Samostatná úloha IV - odevzdání algoritmu a písemné zprávy
12. Zápočet

1,2,3,5,7,9 - úloha 1-6

4,6,8,11 - samostatná úloha - odevzdávání a konzultace

10 - test III - naprogramování úlohy v daném čase

Organizace předmětu

A4M35KO naleznete na https://moodle.dce.fel.cvut.cz

- Přednášky
- Cvičení
- Samostatná práce
- Hodnocení
- Literatura

Co je Kombinatorická optimalizace?

Optimalizace je souhrnné označení pro disciplíny zabývající se minimalizací/maximalizací zvolené cílové funkce při dodržení daných omezení, respektive rozhodnutím zdali je množina řešení prázdná.

Kombinatorika je matematika pro diskrétně strukturované problémy.

Kombinatorická optimalizace je optimalizace zahrnující diskrétní proměnné.

Významně se překrývá s pojmem **operační výzkum** (termín používaný zejména ekonomy, vznikl v průběhu 2. světové války pro logistické vojenské operace).

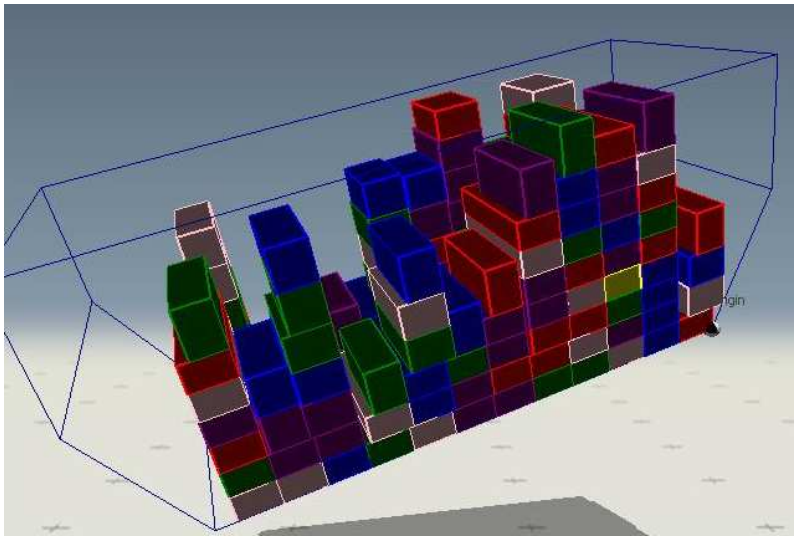
Co s tím můžeme řešit?

Řada problémů reálného života může být formulována jako úloha kombinatorické optimalizace.

Typické oblasti použití:

- výroba (zrychlení výroby, snížení nákladů, efektivní využití zdrojů ...)
- doprava (úspora pohonných hmot, zkrácení doby doručení zásilky ...)
- rozvrhování lidí (redukce počtu potřebných zaměstnanců ...)
- návrh hardware (zrychlování výpočtů ...)
- návrh komunikační sítě (zkrácení doby doručení ...)
- ...

Nakládání kontejneru



Krabice umístěné v kontejneru.

Cíl:

- naskládat do kontejneru co nejvíce nákladu

Omezení:

- rozměry kontejneru
- rozměry krabic, beden ... s nákladem
- způsob nakládání (odkud je do kontejneru přístup, možnosti nakládacího zařízení ...)
- stabilita nákladu, možnost otočení krabic ...
- požadavky na pořadí při vykládání nákladu

Formalizujeme jako **3-D batoh**.

Rozvrhování lidských zdrojů

Přidělení služeb zaměstnancům

Cíl:

- vytvořit přípustný rozdělovník směn pokrývající naplánované směny

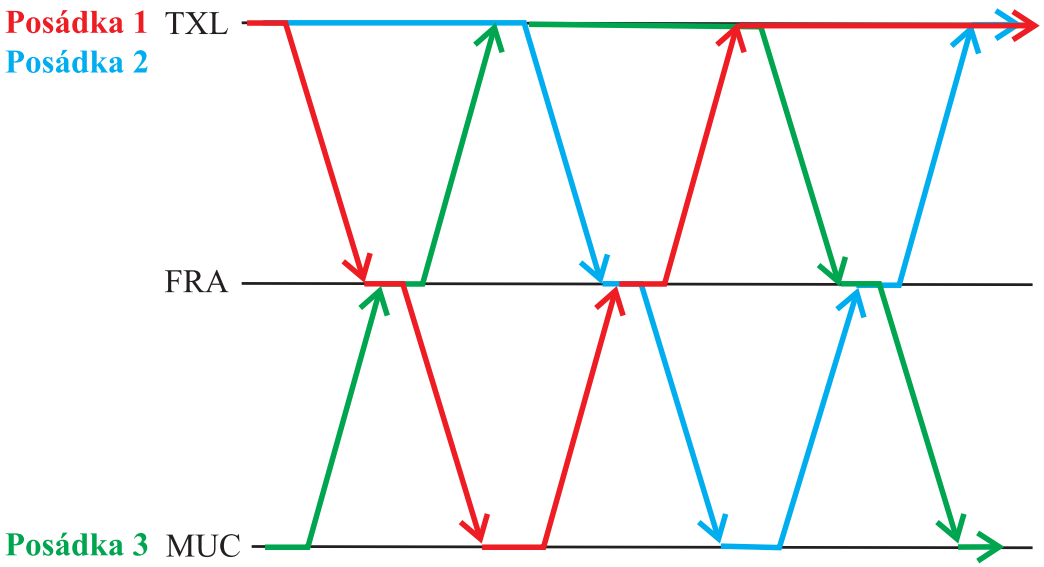
Omezení:

- kvalifikace zaměstnanců
- omezení daná zákoníkem práce (např. odpočinek po dobu alespoň 12 hodin během 24 hodin)
- omezení daná kolektivní smlouvou (např. maximální počet nočních směn po sobě)
- požadavky zaměstnanců (např. návštěva lékaře)
- spravedlivost přiřazení směn (např. stejný počet víkendových směn)
- vstřícný přístup k zaměstnanci (např. shlukovat volné dny do bloků)

Rozvrhování lidských zdrojů

Rozvrhování lidských zdrojů často formalizujeme jako
Párování v bipartitním grafu

Problém je složitější, pokud je třeba brát v úvahu geografickou polohu zaměstnanců (např. letušky a piloti u letecké společnosti).



Regálový zakladač v automatizovaném skladišti

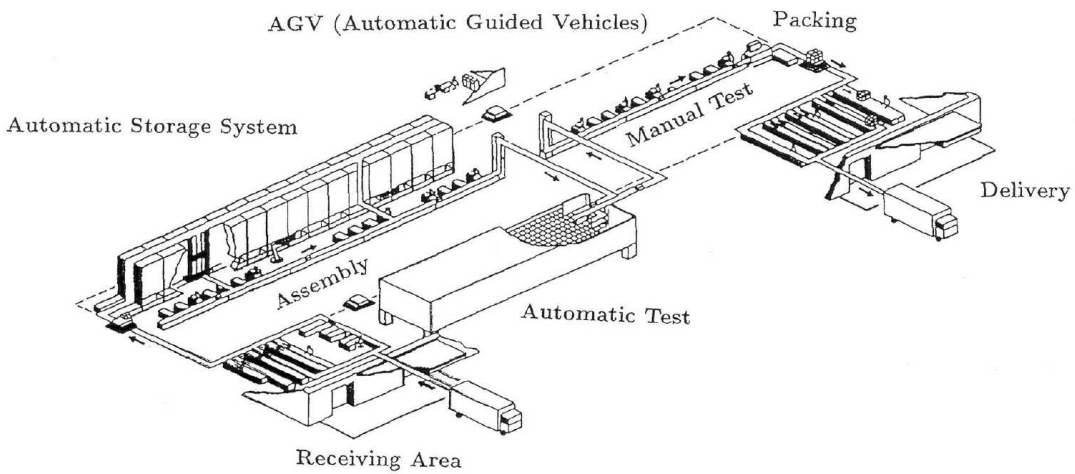
Automatizované skladiště je použito jako mezisklad propojující jednotlivé části továrny. Při větším zatížení se stává omezujícím prvkem.

Cíl:

- minimalizace cest nenaloženého zakladače

Omezení:

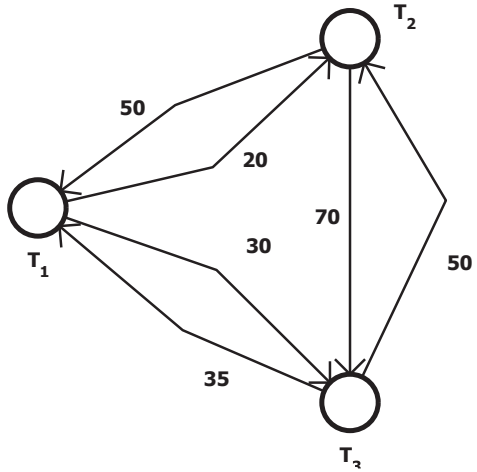
- zadané transportní úlohy
- parametry skladiště
- zakladač má kapacitu 1 *kontejner*
- úlohy přibývají on-line



Regálový zakladač v automatizovaném skladišti

Transportní úloha je zadaná startovací a cílovou pozicí.
Reprezentace problému orientovaným grafem.
Vrcholy grafu představují transportní úlohy.
Or. hrana (i, j) znamená možnost provedení úlohy j ihned po úloze i.
Cena hrany odpovídá ceně cesty zakladače mezi i a j.

Formulace jako **Asymetrickou úlohu obchodního cestujícího**.



Rozvoz zboží

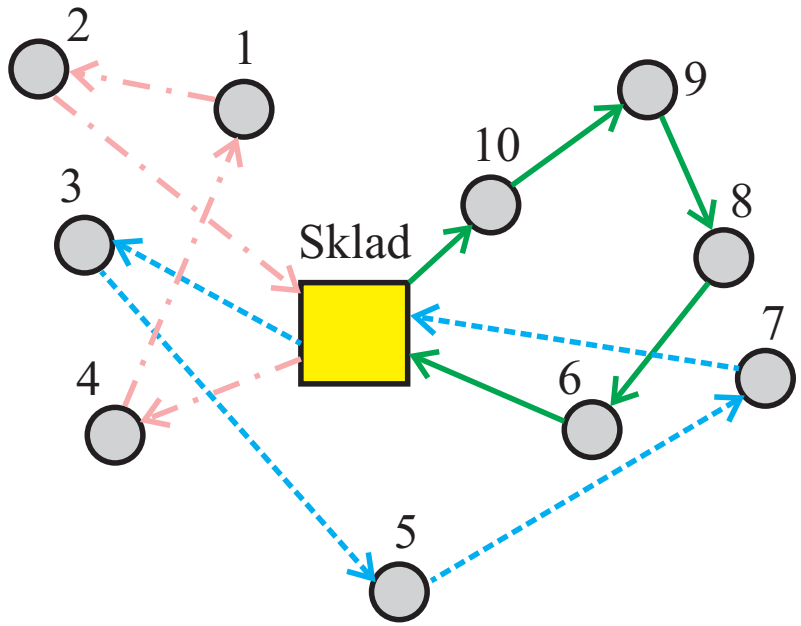
Zboží, zákazníci a flotila automobilů.

Cíl:

- splnit požadavky zákazníků
- minimalizovat náklady na dopravu

Omezení:

- kapacita aut
- časová okna
- dopravní omezení
- pracovní doba, přestávky



Osazování plošných spojů

Osazovací stroje jsou z důvodu vysokých pořizovacích nákladů limitním zdrojem při SMT montáži.

Cíl:

- zrychlení výrobního taktu linky

Omezení:

- konfigurace osazovací linky
- popis vyráběného PCB

Problém rozdělen na dva podproblémy.

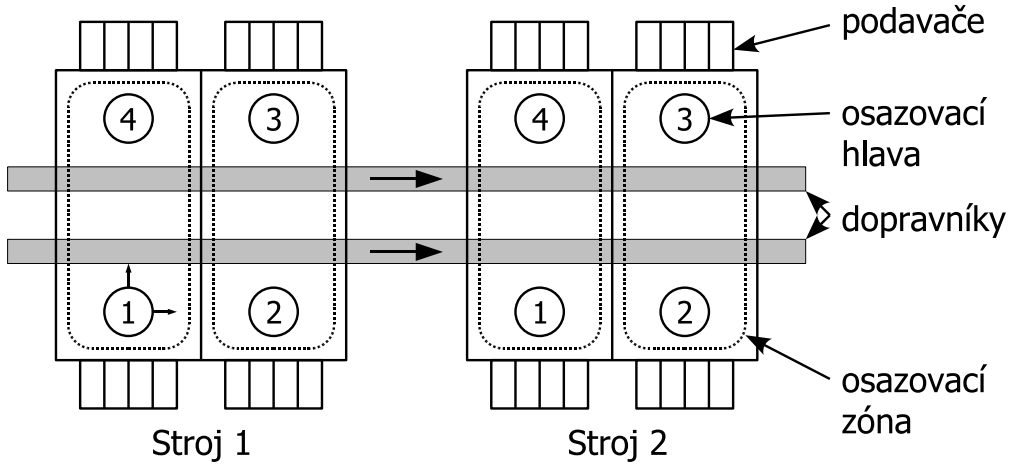
Osazování plošných spojů

A) Přiřazování komponent na osazovací hlavy

Formalizováno jako **Problém dělení kořisti**

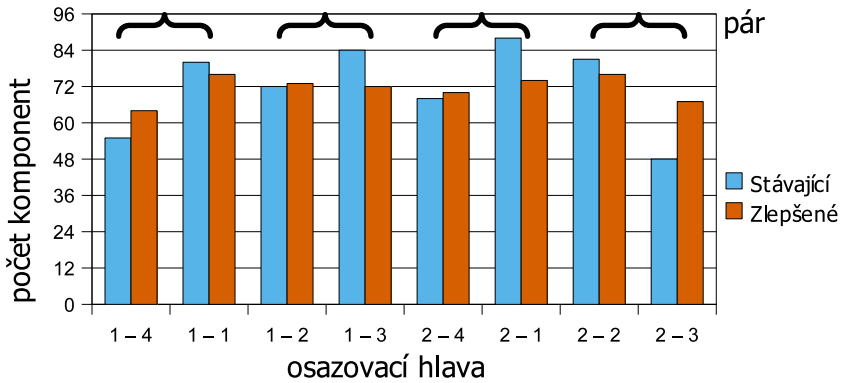
Vstup:

- typy SMT komponent
- množství komponent pro každý typ
- precedence osazení pro určité typy komponent
- charakteristika osazovacích strojů



Výstup:

- přiřazení typu komponenty na osazovací hlavu

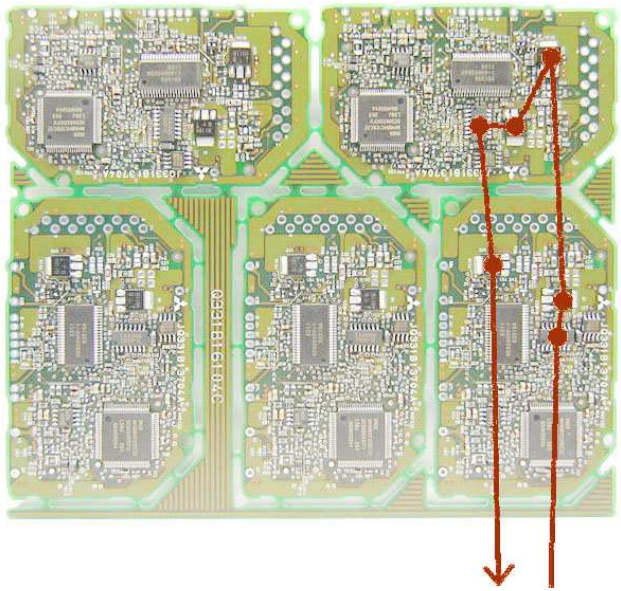


Osazování plošných spojů

B) Určení sekvence osazování
Formalizováno jako (capacitated multi-trip)
Problém obchodního cestujícího

Vstup:

- přiřazení typu komponenty na osazovací hlavu
- pozice komponent na PCB



Výstup:

- osazovací sekvence
- odhad výrobního taktu

Dělení ocelových odlitků

Ocelárna vyrábí odlitky, z nichž dělením a dalším zpracováním produkuje finální výrobky pro uspokojení zakázek.

Cíl:

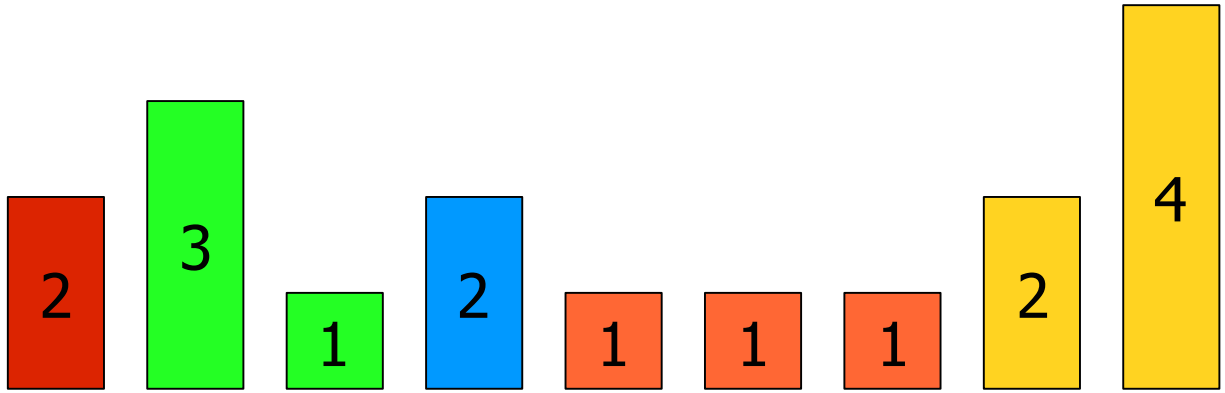
- vyrobit všechny zakázky při minimalizaci potřebného množství oceli

Omezení:

- n možných velikostí odlitků
- velikosti zakázek
- barva zakázky určuje zpracování odlitku
- z jednoho odlitku je možné vyrobit nejvíce p různých barev zakázky

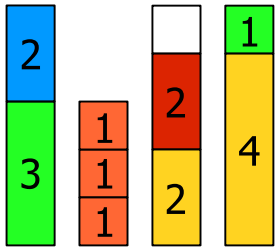
Dělení ocelových odlitků

Příklad



- Rozměry odlitků: $\{3, 5\}$, $n = 2$
- Zakázky $1 \ldots 9$
- Barvy $1 \ldots 5$
- Počet možných barev na odlitku $p = 2$

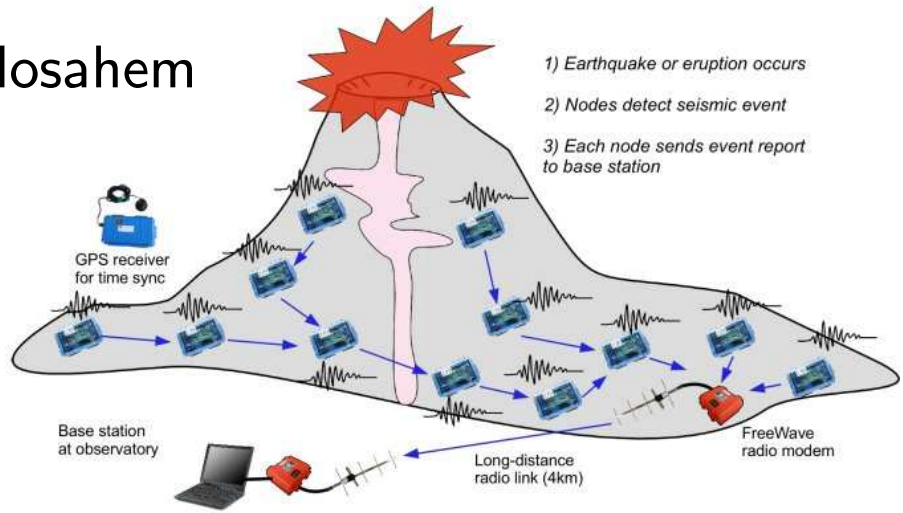
Řešení:



Směrování v senzorové síti - specifikace

Monitorování rozsáhlého prostoru pomocí autonomních zařízení vybavených

- vlastním napájením
- bezdrátovou komunikací s krátkým dosahem
- senzory teploty



Cíl:

- vytvořit směrovací tabulky
- minimalizovat spotřebu energie

Omezení:

- kapacita komunikačních linek
- omezený výkon vysílače
- maximální dovolené časové zpoždění během komunikace
- kapacita paměti jednotlivých zařízení

Směrování v senzorové síti - formalizace

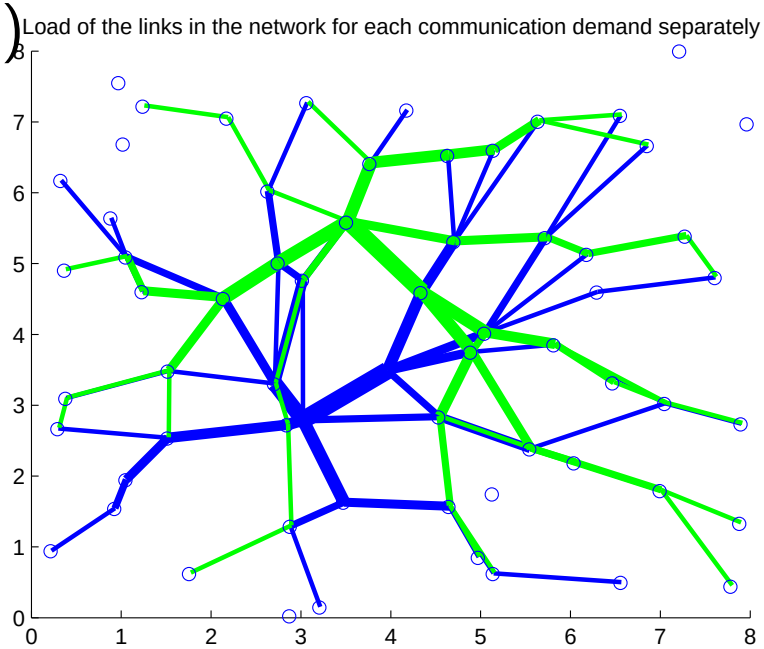
Formalizujeme jako **Multi-Commodity Network Flows**

Reprezentace sítě grafem (zařízení = vrchol, komunikační linka = hrana)

Konstantní komunikační zpoždění pro všechny linky (perioda TDMA, ...)

Komunikační požadavky = toky dané komodity:

- zdrojová (zdrojový vrchol) a cílová (cílový vrchol) zařízení
- velikost požadavku (množství dat za čas)
- deadline (maximální počet hopů)



Komunikační linky:

- kapacita (množství dat za čas)
  - souvisí s počtem slotů v TDMA
- cena (energie na přenesení jedn. dat)

Varianty rozšíření problému

- různé komunikační zpoždění linek
- nedělitelné toky
- maximalizovat životnost sítě (minimalizovat spotřebu energie)
- distribuovaná verze

Směrování v senzorové síti - distribuovaný problém

U této aplikace (senzorové sítě) je centralizovaný přístup nepraktický:

- vstupy (kapacity linek,...) i výstupy (množství toků po linkách) optimalizačního algoritmu jsou lokálního charakteru
- přidání/ubrání zařízení v centralizovaném algoritmu vyžaduje:
  - komunikaci vstupů
  - centralizovaný výpočet
  - komunikaci výstupů
  - přechod mezi módy sítě

Směrování v senzorové síti - distribuovaný problém

U této aplikace (senzorové sítě) je centralizovaný přístup nepraktický:

- vstupy (kapacity linek,...) i výstupy (množství toků po linkách) optimalizačního algoritmu jsou lokálního charakteru
- přidání/ubrání zařízení v centralizovaném algoritmu vyžaduje:
  - komunikaci vstupů
  - centralizovaný výpočet
  - komunikaci výstupů
  - přechod mezi módy sítě

Distribuovaný algoritmus - totožný kód v každém zařízení:

Vstup: kapacity a ceny incidentních linek, zdroje a cíle požadavků ...

Výstup: množství toků po incidentních linkách, ...

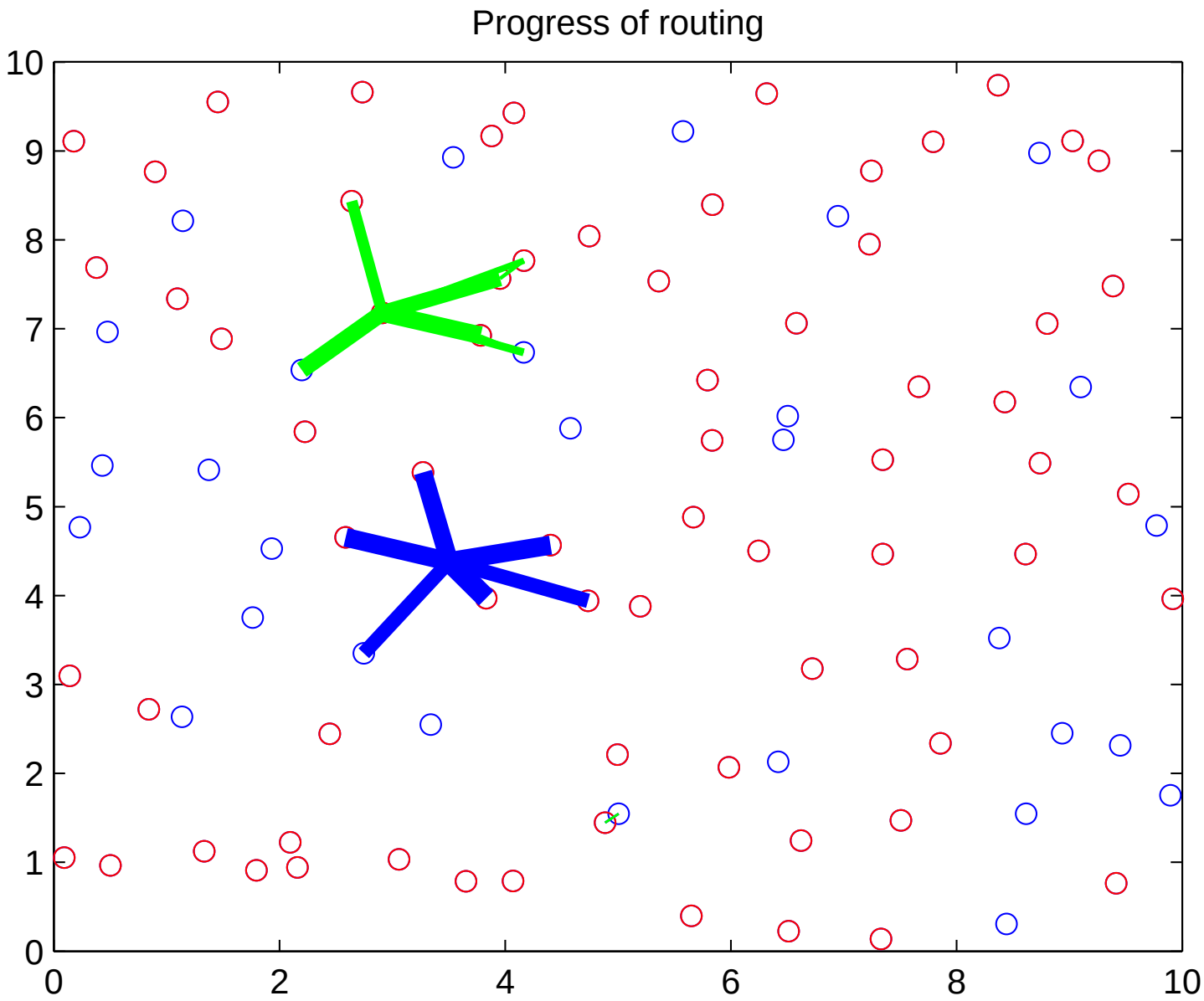
while *consensus_not_reached* **do**

    proveď lokální optimalizaci;
    komunikuj hraniční proměnné se sousedy;

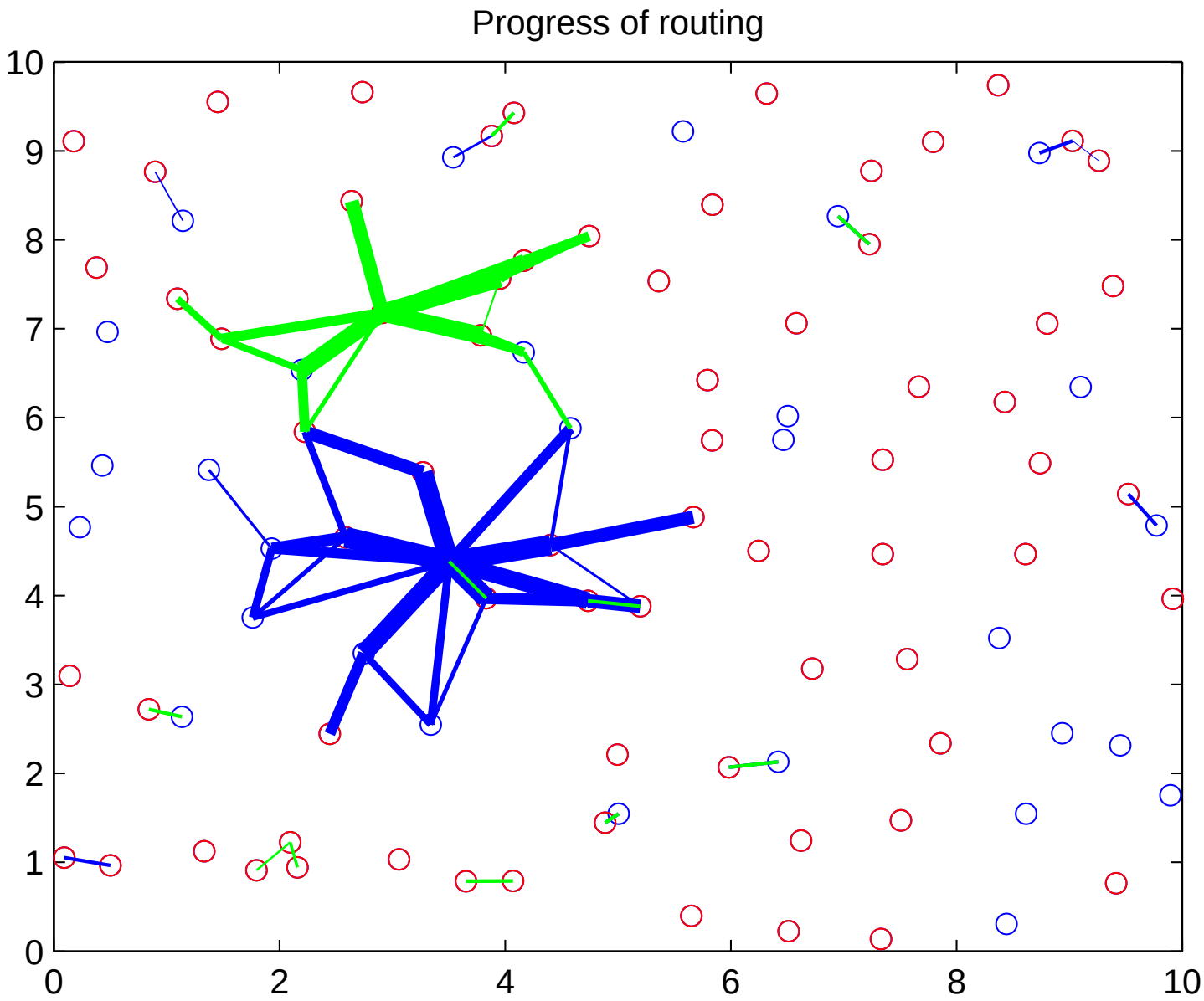
end

Typické obtíže distribuovaných algoritmů - konvergence a ukončení

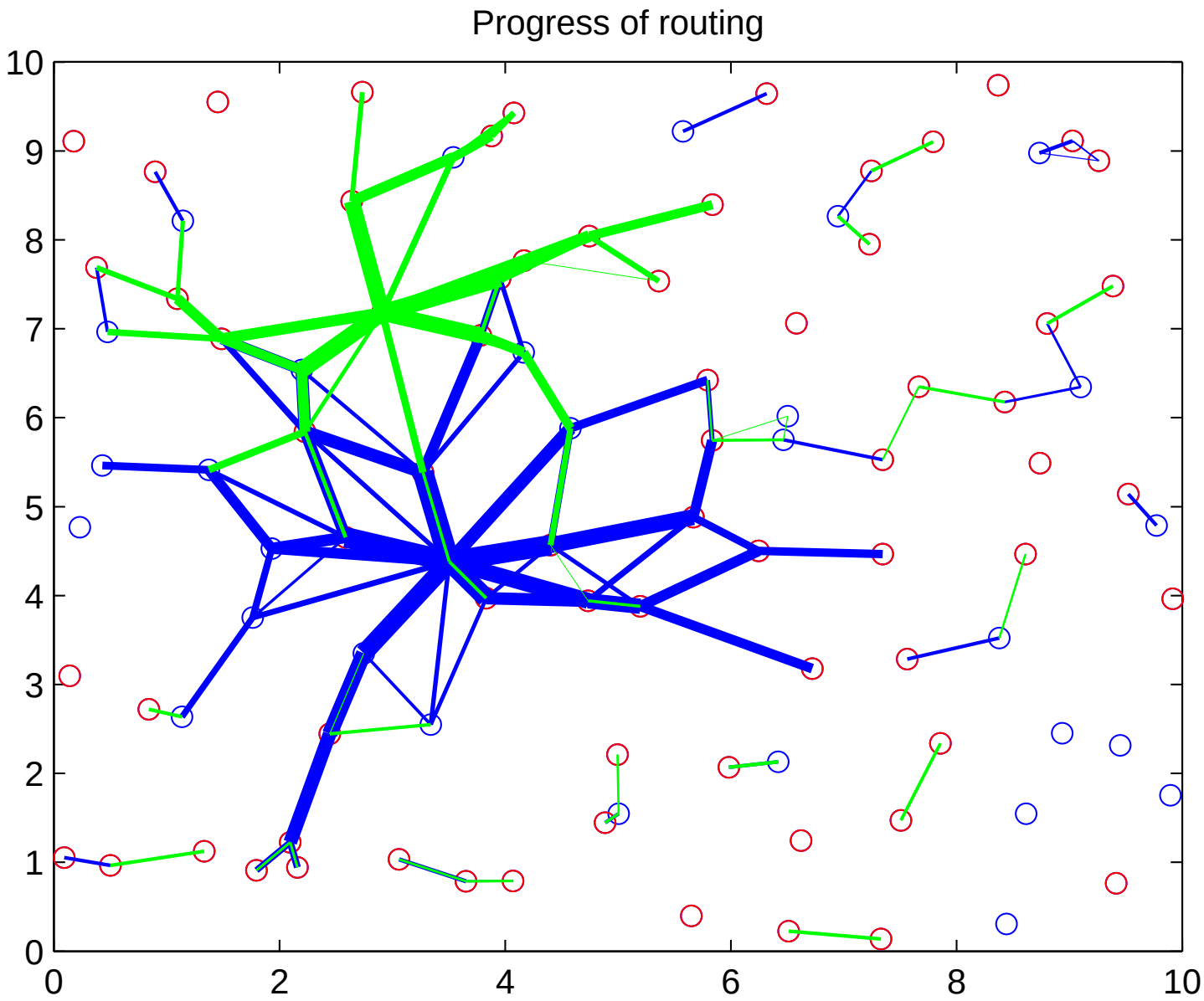
Směrování v senzorové síti - běh distribuovaného algoritmu



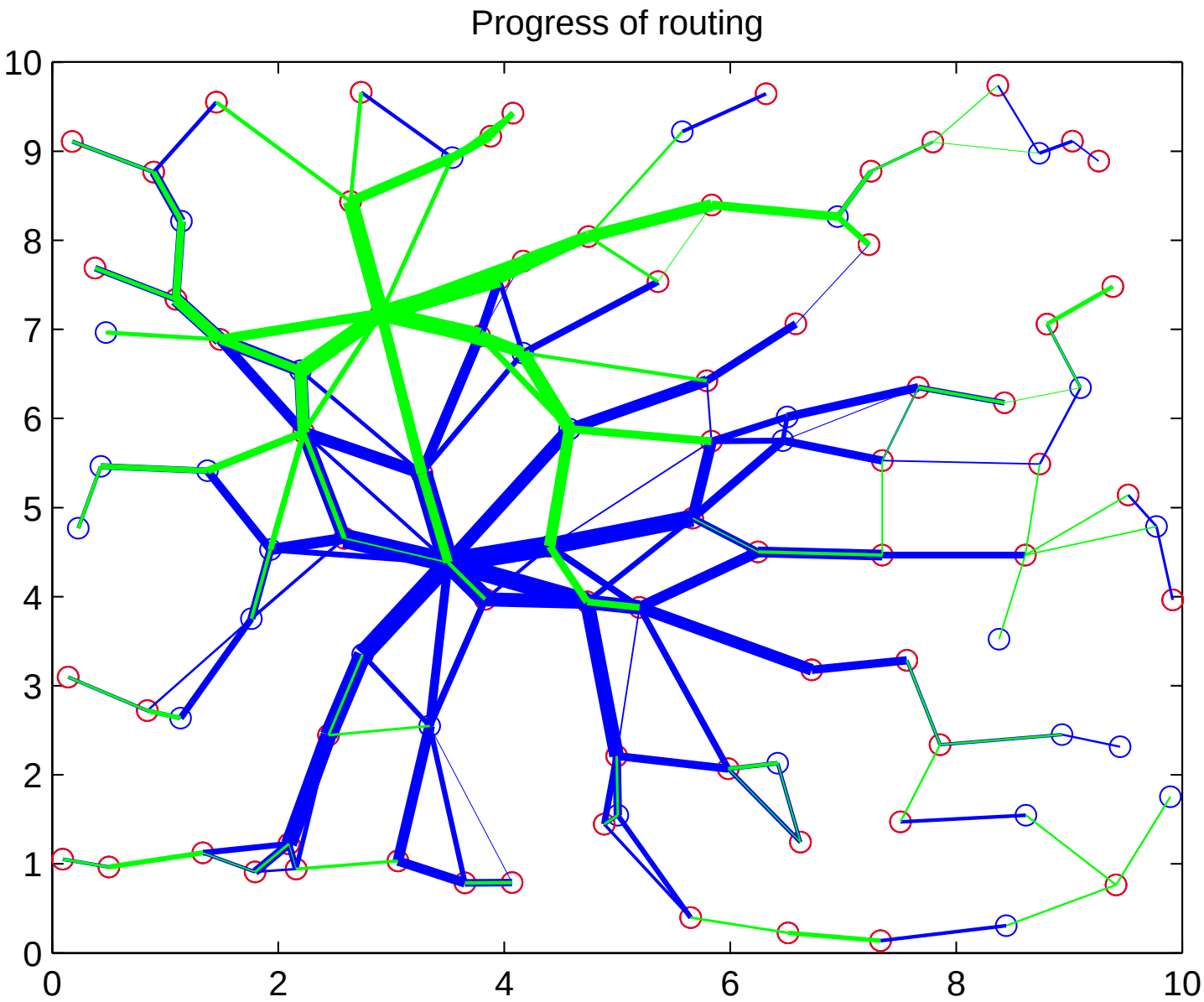
Směrování v senzorové síti - běh distribuovaného algoritmu



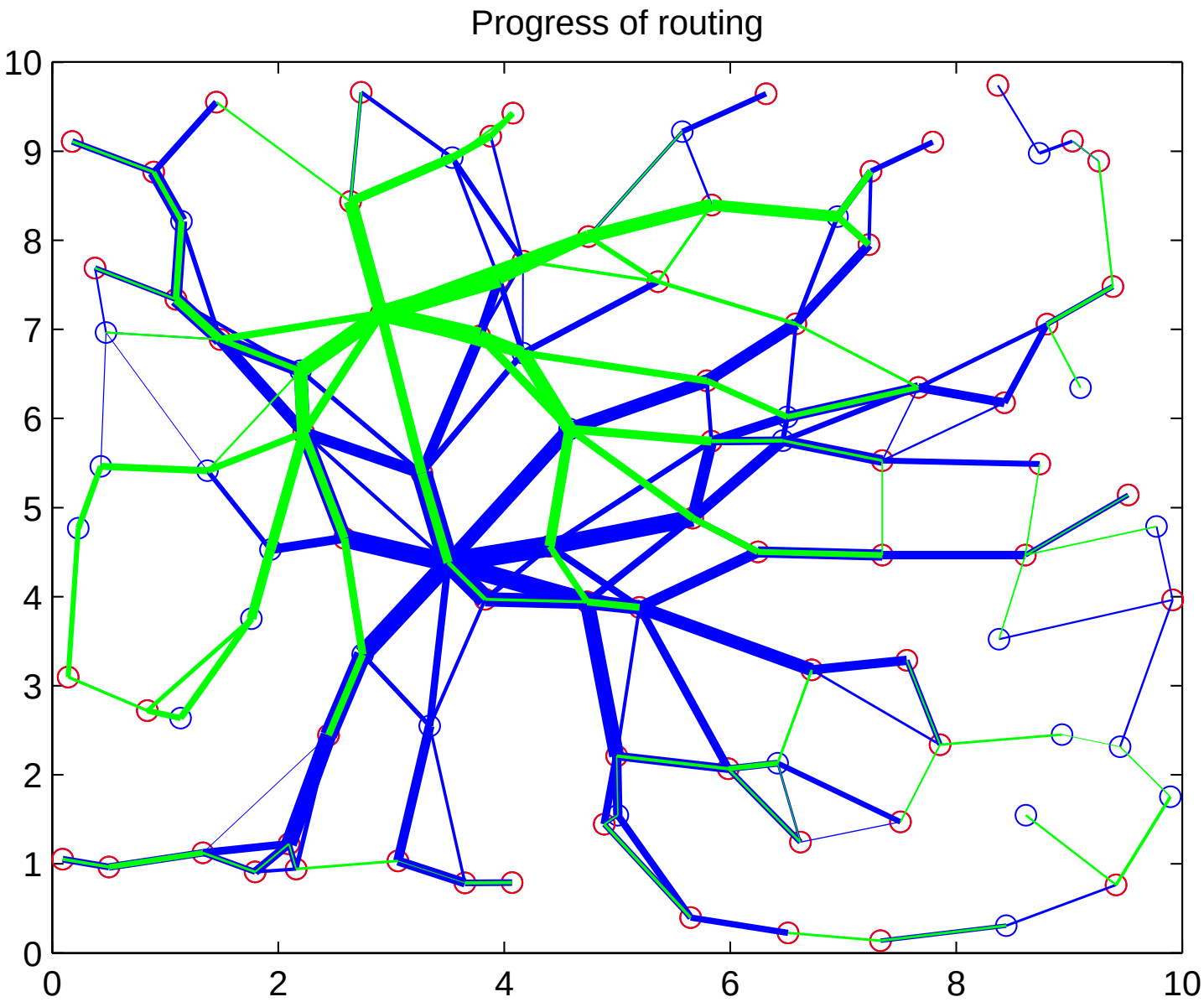
Směrování v senzorové síti - běh distribuovaného algoritmu



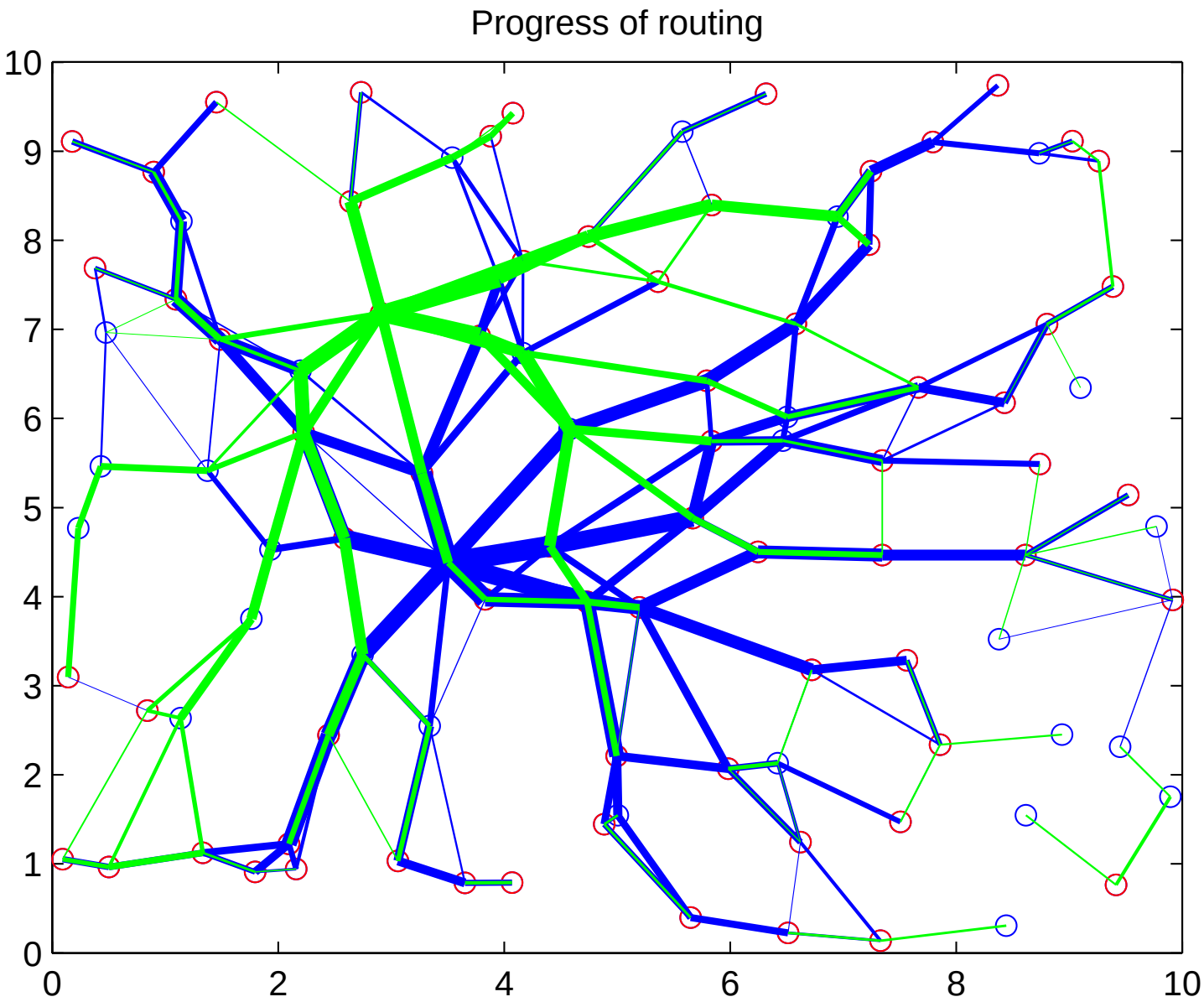
Směrování v senzorové síti - běh distribuovaného algoritmu



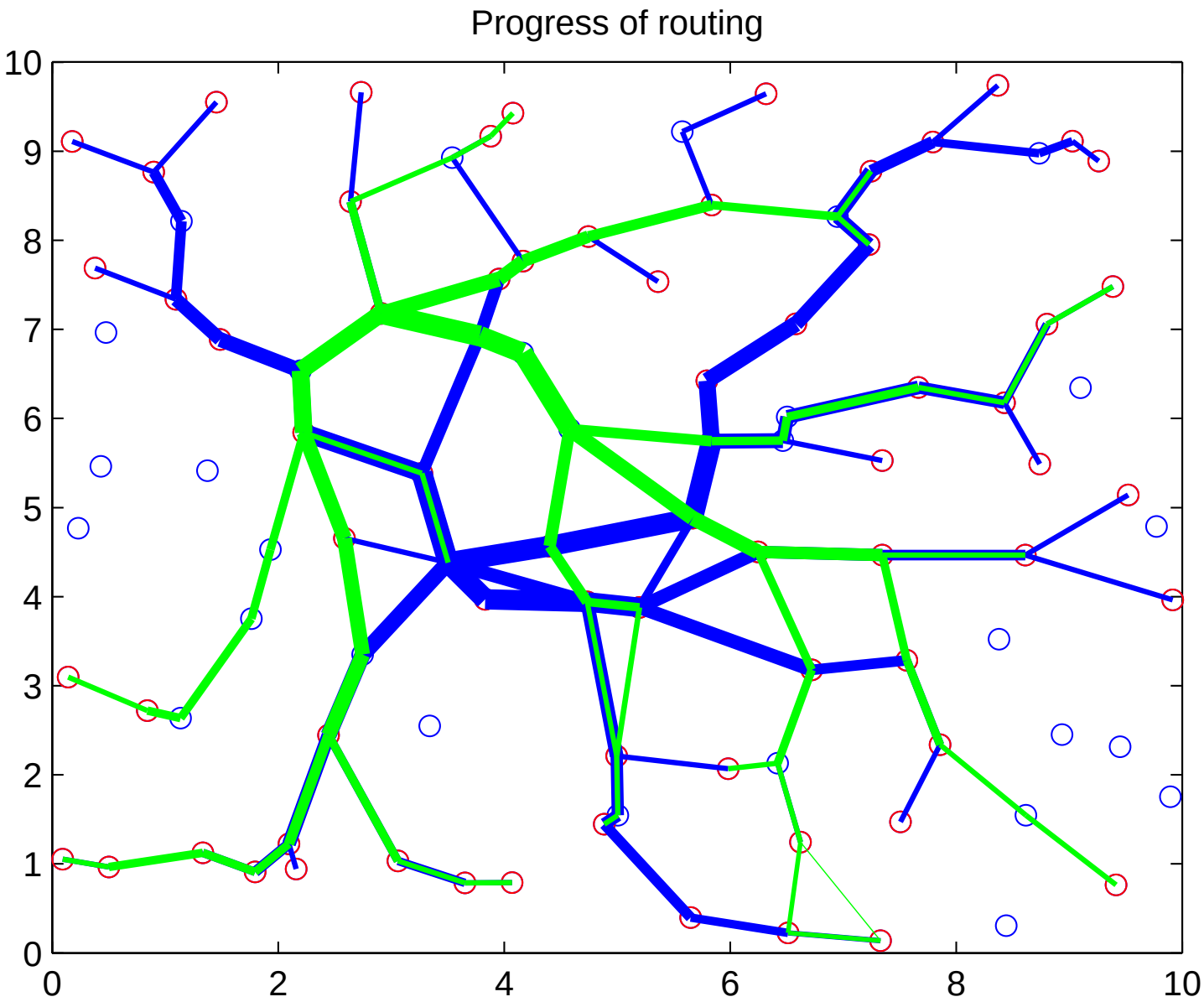
Směrování v senzorové síti - běh distribuovaného algoritmu



Směrování v senzorové síti - běh distribuovaného algoritmu



Směrování v senzorové síti - běh distribuovaného algoritmu



Návrh křižovatek ve městě - specifikace

Město dané křižovatkami a ulicemi. Hlavní dopravní tahy dané zdrojem, cílem a množstvím automobilů za čas.

Cíl:

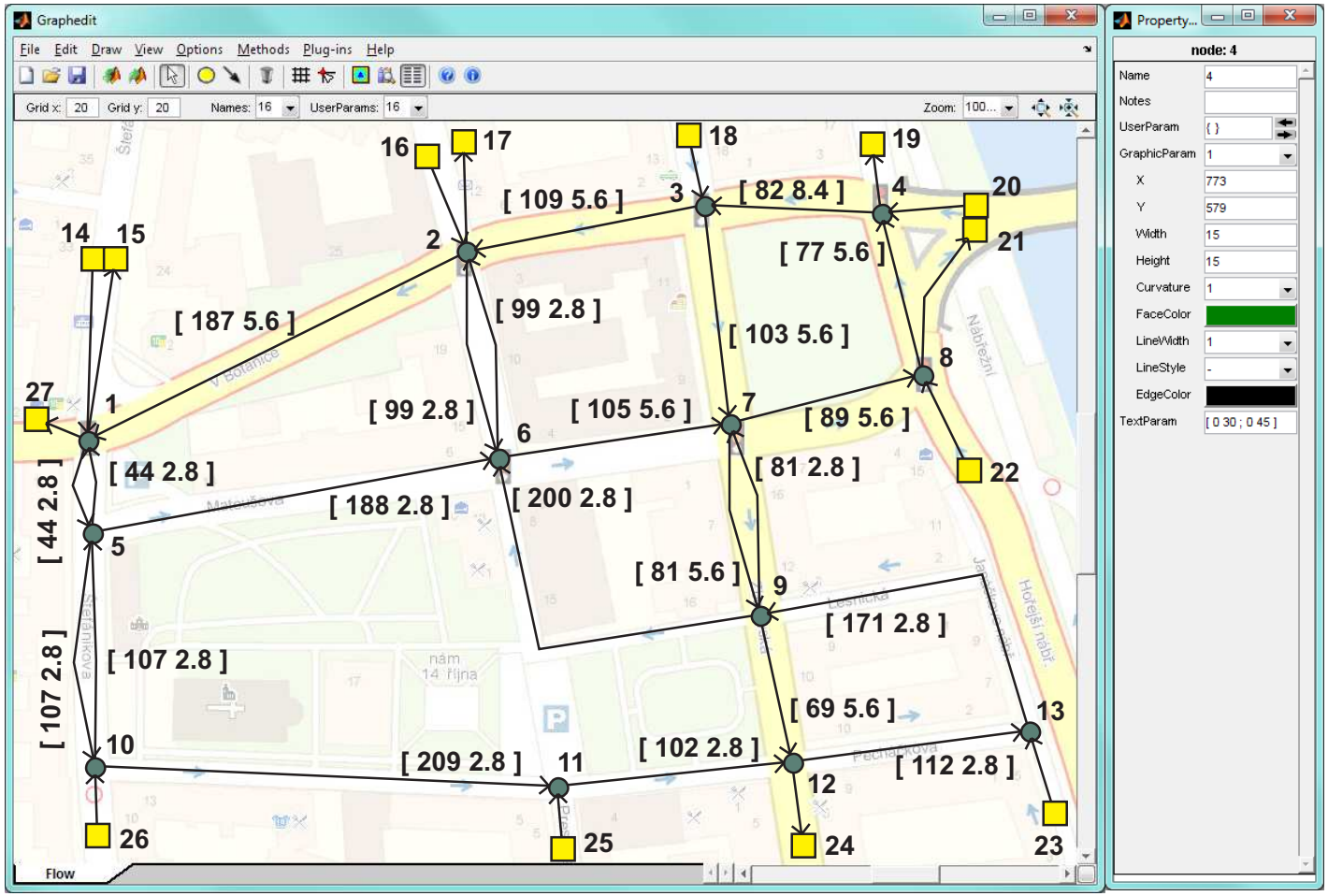
- zjistit, jaké množství aut jede z každého směru na každé z křižovatek vlevo/rovně/vpravo = směrování (na základě této informace bude stanovena kapacita odbočovacích pruhů a fáze semaforů)
- předpokládáme, že celý systém se sám chová optimálně = minimalizuje celkovou cenu přepravy (odpovídá množství najetých km nebo spotřebě pohonných hmot nebo době přepravy)

Omezení:

- kapacita ulic daná počtem pruhů a maximální povolenou rychlostí
- jednosměrné ulice

Lze formalizovat jako **Multi-Commodity Network Flows**

Návrh křižovatek ve městě - specifikace



zdroj tahu	14	14	14	14	16	16	16	16	18
cil tahu	24	17	21	27	24	21	15	27	24
automobily za min	4.9	1.9	3.1	30.1	10.1	12.1	1.2	6.6	32.7

Zakázková produkce laku

Výroba laků, kde jednotlivé zakázky jsou určeny typem laku, množstvím a termínem dodání.

Cíl:

- minimalizovat překročení dodacích termínů
- minimalizovat dobu skladování před distribucí

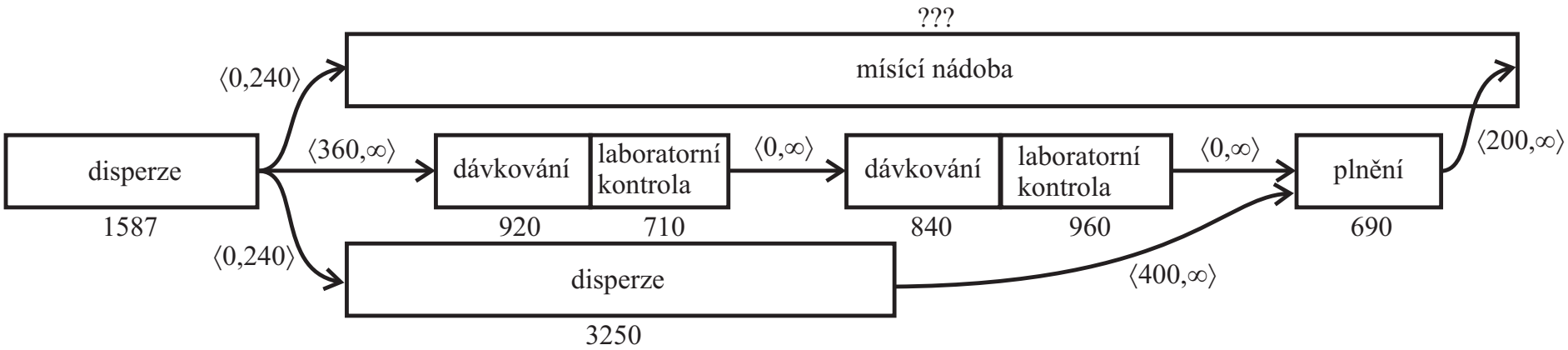
Omezení:

- dávkový proces výroby různých laků
- různé výrobní procesy/časy pro různé typy laků
- časová omezení mezi začátky a konci jednotlivých operací
- pracovní doba (některé procesní časy operací výroby překračují pracovní dobu)
- přípravné časy (*setup time*)

Lze formalizovat jako $PS\,|temp, o_{ij}, tg|\,C_{max}$

Plánování produkce laku

Operace výroby mohou být vůči sobě časově omezeny. Musíme uvažovat (1) minimální čas mezi koncem jedné operace a začátkem druhé (např. minimální čas potřebný k rozpuštění příměsi v laku), (2) maximální přípustný čas mezi dvěma operacemi (např. lak může ztvrdnout).

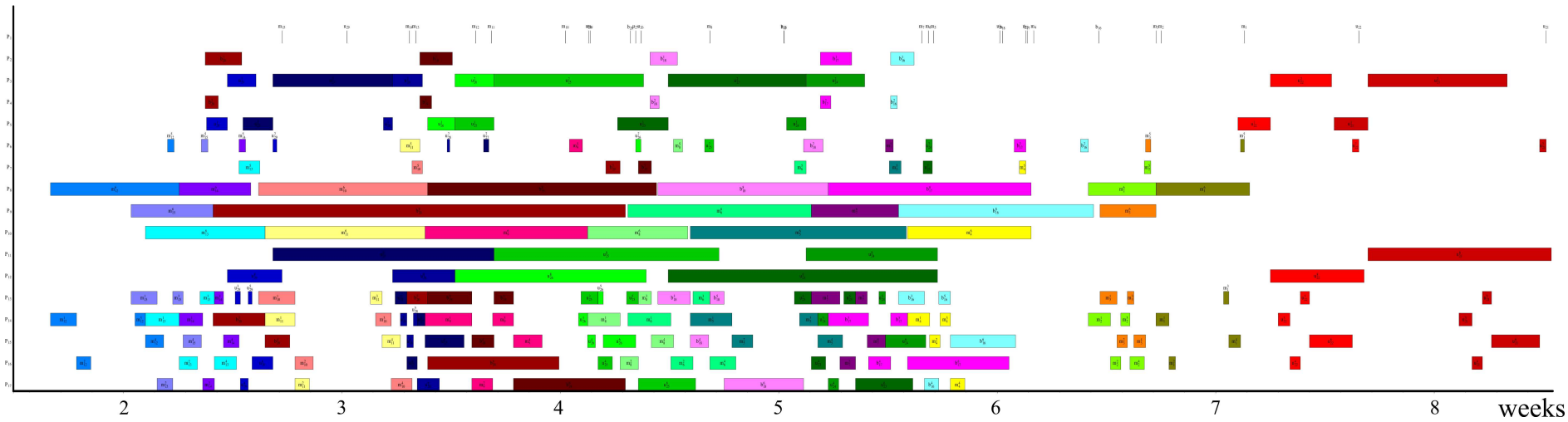


Navíc doba vykonávání některých operací může být dána začátky a konci jiných operací výroby.

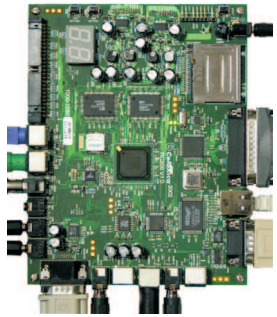
Plánování produkce laku

Příklad

- výroba 29 zakázek
- 3 typy laků
- časový horizont 9 týdnů



Návrh konfigurovatelného HW pro specifickou DSP aplikaci



Specializované výpočetní jednotky (sčítačka, násobička na FPGA).
Cyklický DSP algoritmus složený z atomických operací.

Cíl:

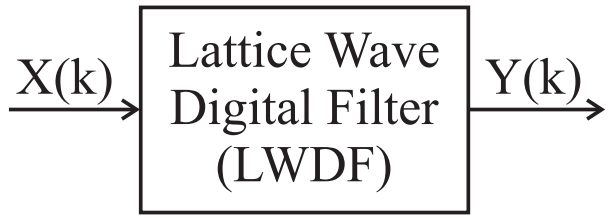
- maximalizovat rychlost výpočtu
- minimalizovat velikost použitých prostředků obvodu

Omezení:

- relace mezi atomickými operacemi (sčítání, násobení ...)
- omezený přístup do paměti obvodu
- omezená velikost paměti obvodu
- počet dostupných výpočetních jednotek na FPGA

Návrh konfigurovatelného HW pro specifickou DSP aplikaci

Příkladem aplikace je jednoduchý číslicový filtr LWDF. $X(k)$ jsou vzorky vstupního signálu, $Y(k)$ jsou vzorky výstupního signálu.



for k=1 **to** N **do**

T_1: $a(k) = X(k) - c(k-2)$

T_2: $b(k) = a(k) * \alpha$

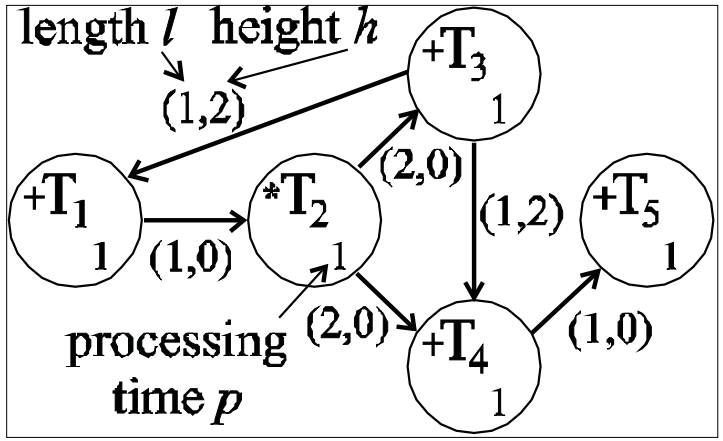
T_3: $c(k) = b(k) + X(k)$

T_4: $d(k) = b(k) + c(k-2)$

T_5: $Y(k) = X(k-1) + d(k)$

end

Algoritmus filtru LWDF



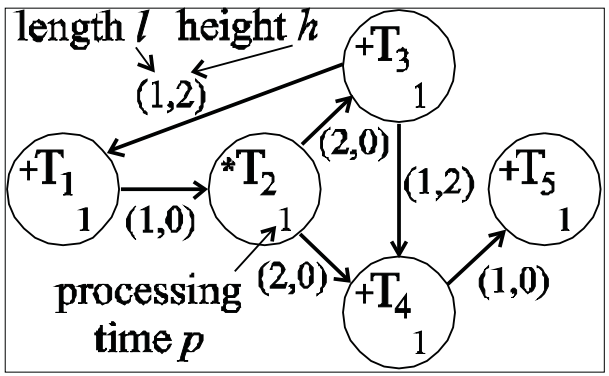
Graf závislostí atomických operací

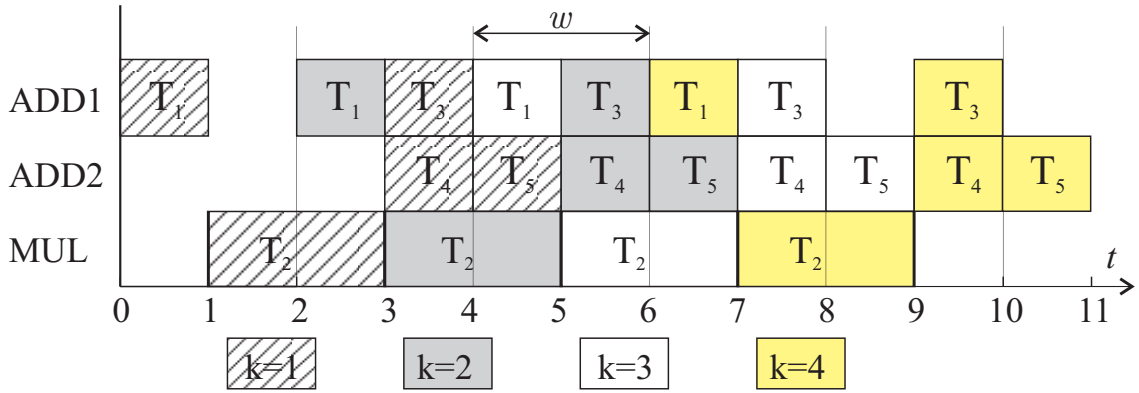
Návrh konfigurovatelného HW pro specifickou DSP aplikaci

Lze formalizovat jako cyklické rozšíření $PS\,|temp|\,C_{max}$

Parametry použitého hardware

jednotka	počet [−]	doba výpočtu [*clk*]
ADD	2	1
MUL	1	2



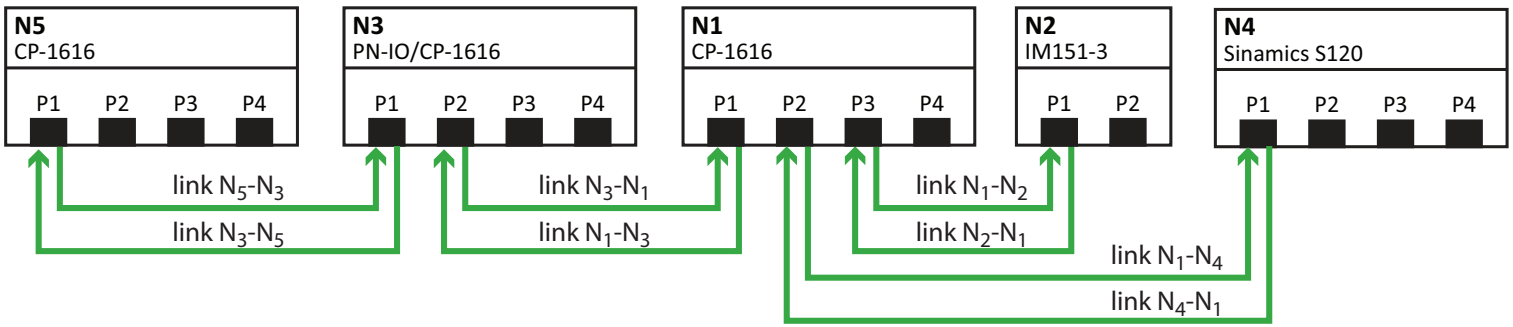


Rozvrhování zpráv pro Profinet IO IRT - specifikace

Ethernet-based hard real-time komunikační protokol.
Přesná synchronizace hodin ve všech zařízeních.
Každé zařízení obsahuje speciální switch, který v určité části periody nedodržuje standardní forwardovací pravidla, aby pro časově kritické zprávy nedocházelo ke zpoždění ve frontě.

Cíl:

- najít co nejkratší rozvrh pro časově kritické zprávy



linka	$N_1 \to N_3$	$N_1 \to N_4$	$N_1 \to N_2$	$N_2 \to N_1$	$N_3 \to N_1$	$N_4 \to N_1$	$N_3 \to N_5$	$N_5 \to N_3$
zpoždění linky [ns]	4875	5130	5862	3841	4875	4895	4875	4875

Rozvrhování zpráv pro Profinet IO IRT - specifikace

Omezení:

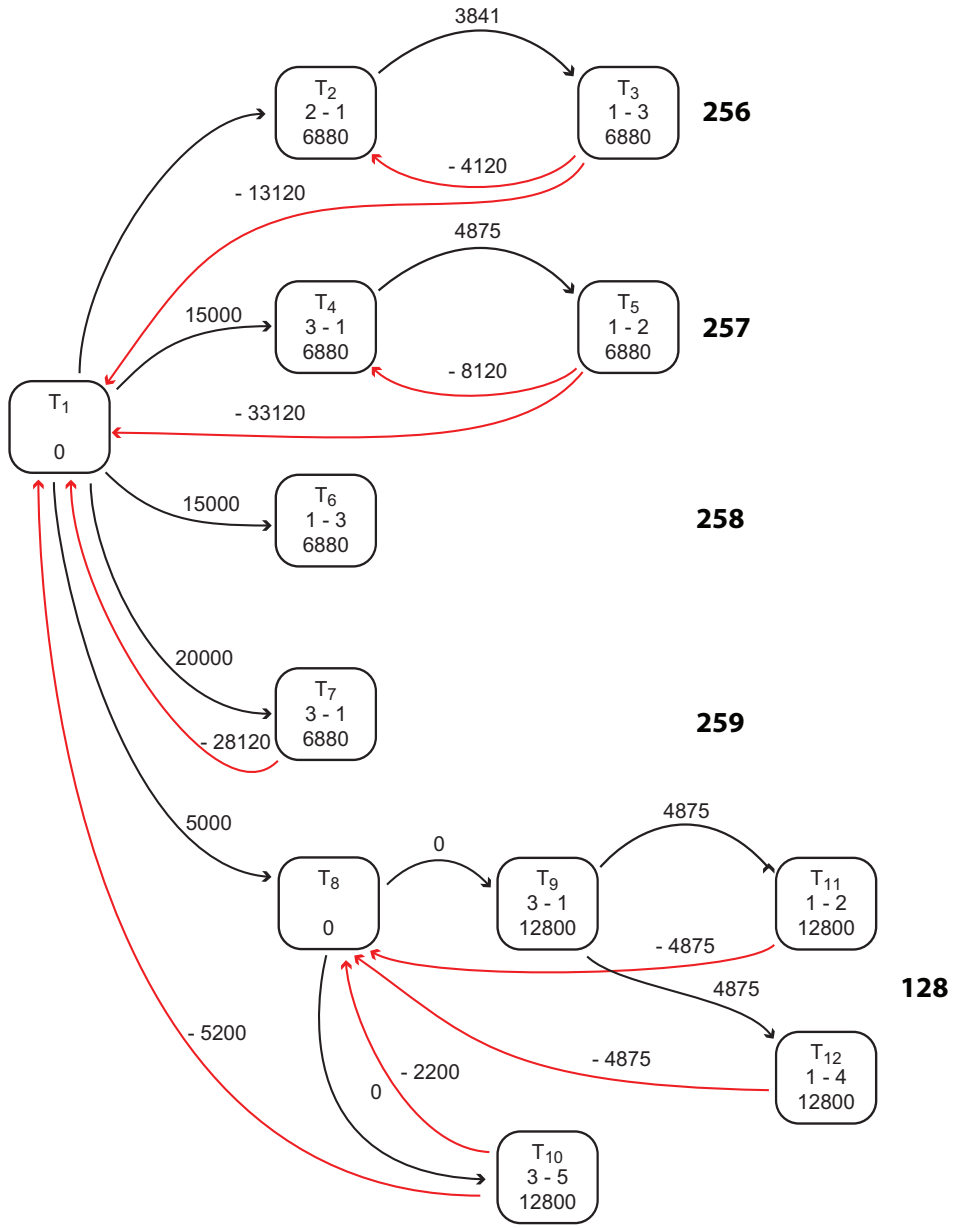
- stromová topologie $\Rightarrow$ jednoznačné směrování
- release date r, t.j. nejdřívější možné odeslání zprávy
- deadline $\widetilde{d}$, t.j. nejpozdější možné doručení zprávy
- maximální možné end2end zpoždění zprávy

ID	zdroj $\rightarrow$ cíl	délka [ns]	r [ns]	$\widetilde{d}$ [ns]	pož. end2end zpoždění [ns]
256	$N_2 \rightarrow N_3$	5760	5000	20000	11000
257	$N_3 \rightarrow N_2$	5760	15000	40000	15000
258	$N_1 \rightarrow N_3$	5760	15000	–	–
259	$N_3 \rightarrow N_1$	5760	20000	35000	–
128	$N_3 \rightarrow \{N_1,N_2,N_4,N_5\}$	11680	5000	{–,–,–,18000}	{–,17675,17675,15000}

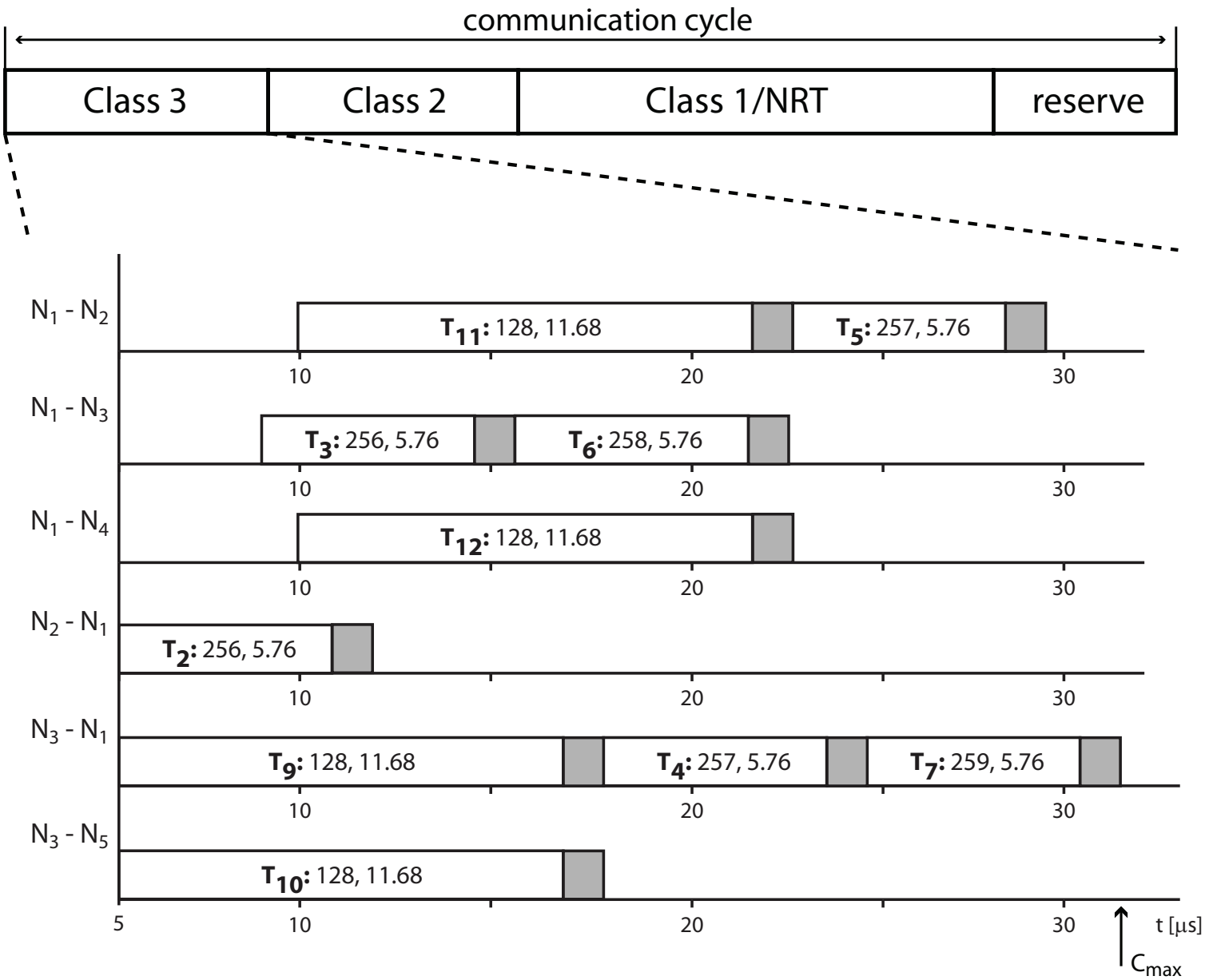
Rozvrhování zpráv pro Profinet IO IRT - formalizace

Lze formalizovat jako $PS\,|temp|\,C_{max}$

- úloha = komunikace zprávy na lince
- hrana s kladnou vahou = r, relace následností
- hrana se zápornou vahou = $\widetilde{d}$, end2end
- unicast zpráva = řetězec úloh
- multicast zpráva = kořenový strom úloh



Rozvrhování zpráv pro Profinet IO IRT - výsledek



Přesvědčivé argumenty

Vysvětlit typické cíle optimalizace:

- zvýšení produkce (zkrácení taktu výrobní linky)
- snížení nákladů (úspora pohonných hmot, méně strojů)
- omezení rizik (eliminace chyb díky automatizaci tvorby výrobního plánu)
- zeštíhlení výroby (redukce zásob a skladů, omezení následků zpoždění dodávky dílů)
- zvýšení flexibility (rychlé reakce při změně struktury nebo omezení)
- přívětivější pro uživatele (navazující MHD, vyvážené pracovní zatížení)

Jak předmět KO pokrývá užitečné dovednosti

Inženýr je zpravidla najat proto, aby systematicky vyřešil daný problém.

Dovednost	Příklad
Bussiness case	Získat zákazníka
Specifikace	Rozvrhování výroby
Formalizace	Flow-shop
Algoritmy	Johnsonův algoritmus
Prototypové řešení	Matlab OPL, ...
Implementace	C#, dB, ...

Jak předmět KO pokrývá užitečné dovednosti

Inženýr je zpravidla najat proto, aby systematicky vyřešil daný problém.

Dovednost	Příklad	Přednášky
Bussiness case	Získat zákazníka	-
Specifikace	Rozvrhování výroby	Případové studie
Formalizace	Flow-shop	Formulace opt.probl.
Algoritmy	Johnsonův algoritmus	Pseudokód iterace s daty
Prototypové řešení	Matlab OPL, ...	-
Implementace	C#, dB, ...	-

Jak předmět KO pokrývá užitečné dovednosti

Inženýr je zpravidla najat proto, aby systematicky vyřešil daný problém.

Dovednost	Příklad	Přednášky	Cvičení
Bussiness case	Získat zákazníka	-	SP?
Specifikace	Rozvrhování výroby	Případové studie	SP, úlohy
Formalizace	Flow-shop	Formulace opt.probl.	SP, úlohy
Algoritmy	Johnsonův algoritmus	Pseudokód iterace s daty	SP
Prototypové řešení	Matlab OPL, ...	-	SP, úlohy
Implementace	C#, dB, ...	-	SP?

Jak předmět KO pokrývá užitečné dovednosti

Inženýr je zpravidla najat proto, aby systematicky vyřešil daný problém.

Dovednost	Příklad	Přednášky	Cvičení	Zkoušení
Bussiness case	Získat zákazníka	-	SP?	-
Specifikace	Rozvrhování výroby	Případové studie	SP, úlohy	SP
Formalizace	Flow-shop	Formulace opt.probl.	SP, úlohy	SP zkouška
Algoritmy	Johnsonův algoritmus	Pseudokód iterace s daty	SP	testy zkouška
Prototypové řešení	Matlab OPL, ...	-	SP, úlohy	SP test III
Implementace	C#, dB, ...	-	SP?	-

Řešení problémů pomocí algoritmů

Optimalizace pomocí kombinatorických algoritmů

Problém lze zformulovat jako:

- omezení na platná řešení
- optimalizační kritérium

Najít efektivně optimální řešení není vždy snadné.
Pokud bychom prohledávali všechna řešení, pro větší instance bychom narazili na délku výpočtu.
Např. počet kombinací zakázek v úloze Flow-shop roste s $n!$ (kde n je např. počet zakázek).

Časová náročnost algoritmů

n	$100n \log n$	$10n^2$	$n^{3.5}$	$n^{\log n}$	2^n	$n!$
10	3 μs	1 μs	3 μs	2 μs	1 μs	4 ms
20	9 μs	4 μs	36 μs	420 μs	1 ms	76 years
30	15 μs	9 μs	148 μs	20 ms	1 s	$8 \cdot 10^{15}$ y.
40	21 μs	16 μs	404 μs	340 ms	1100 s	
50	28 μs	25 μs	884 μs	4 s	13 days	
60	35 μs	36 μs	2 ms	32 s	37 years	
80	50 μs	64 μs	5 ms	1075 s	$4 \cdot 10^7$ y.	
100	66 μs	100 μs	10 ms	5 hours	$4 \cdot 10^{13}$ y.	
200	153 μs	400 μs	113 ms	12 years		
500	448 μs	2.5 ms	3 s	$5 \cdot 10^5$ y.		
1000	1 ms	10 ms	32 s	$3 \cdot 10^{13}$ y.		
10^4	13 ms	1 s	28 hours			
10^5	166 ms	100 s	10 years			
10^6	2 s	3 hours	3169 y.			
10^7	23 s	12 days	10^7 y.			
10^8	266 s	3 years	$3 \cdot 10^{10}$ y.			
10^{10}	9 hours	$3 \cdot 10^4$ y.				
10^{12}	46 days	$3 \cdot 10^8$ y.				

Revize pojmů z teorie grafů

Grafy neformálně

- Graf se skládá z vrcholů a hran.
- Hrana vždy spojuje dva vrcholy a je buď orientovaná, nebo neorientovaná.
- U orientovaných hran rozlišujeme počáteční a koncový vrchol.
- Neorientované hrany chápeme jako symetrické spojení dvou vrcholů.

Orientovaný graf

Orientovaný graf je uspořádaná trojice (V, E, Ψ):

- V je konečná množina **vrcholů**
- E je konečná množina **orientovaných hran**
- Ψ je zobrazení, které každé hraně přiřazuje uspořádanou dvojici vrcholů, neboli $\Psi : E \to \{(v, w) \in V \times V : v \neq w\}$

Neorientovaný graf

Neorientovaný graf je uspořádaná trojice (V, E, Ψ):

- V je konečná množina vrcholů
- E je konečná množina **neorientovaných hran**
- Ψ je zobrazení, které každé hraně přiřazuje dvouprvkovou množinu vrcholů, neboli $\Psi : E \to \{X \subseteq V : |X| = 2\}$

Hrany grafu

Dvě hrany e, e' se nazývají **násobné**, pokud $\Psi(e) = \Psi(e')$.
Graf obsahující násobné hrany se někdy nazývá **multigraf** - většinou se jimi nebudeme zabývat.
Graf neobsahující násobné hrany se někdy nazývá **prostý graf**.
Prosté grafy zpravidla zapisujeme $G = (V(G), E(G))$, kde $V(G)$ je množina vrcholů grafu a $E(G)$ je množina (uspořádaných) dvojic vrcholů, popisujících hrany.
Říkáme, že neorientovaná hrana $e = \{v, w\}$ nebo orientovaná hrana $e = (v, w)$ spojuje v a w. Vrcholům v a w říkáme krajní vrcholy hrany e, nebo že jsou incidentní s hranou e. V případě orientované hrany říkáme, že e vede z vrcholu v do vrcholu w (leaves v and enters w).

Porovnávání grafů

Pro orientovaný graf G můžeme zavést **podkladový neorientovaný graf** G', jež má stejnou množinou vrcholů a má neorientovanou hranu $\{v, w\}$ ke každé orientované hraně (v, w) grafu G.
Graf H je **podgrafem** grafu G, vznikne-li vynecháním nějakých (nebo žádných) vrcholů a hran (spolu s každou hranou, která je v podgrafu, tam musí být i její krajní vrcholy). Dva speciální druhy podgrafů:

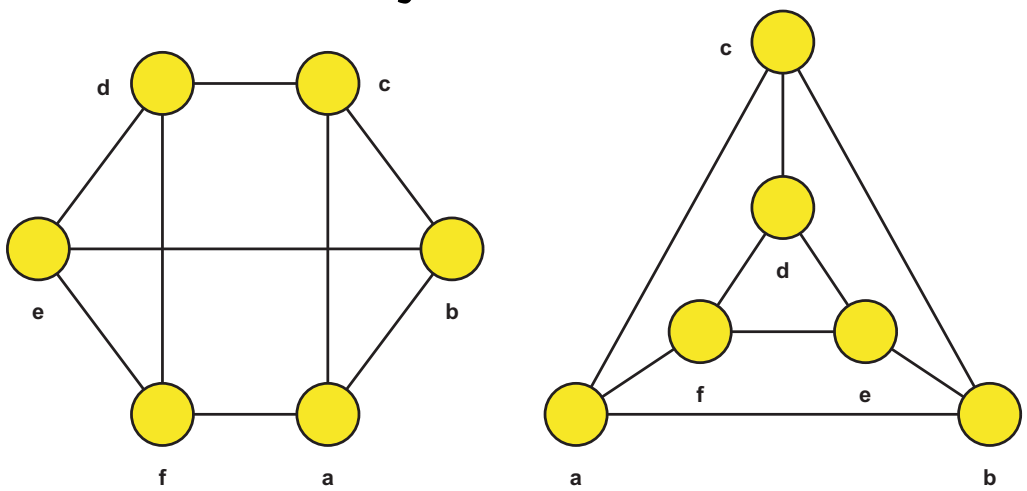
- Graf H nazýváme **faktorem** grafu G, jestliže vynecháme některé (nebo žádné) hrany, neboli platí $V(G) = V(H)$.
- Graf H nazýváme **podgrafem indukovaným množinou vrcholů** $V(H) \subseteq V(G)$, jestliže vynecháme některé (nebo žádné) vrcholy a s nimi incidentní hrany, neboli H obsahuje všechny hrany G, jejichž oba krajní vrcholy jsou v množině $V(H)$.

Porovnávání grafů – izomorfismus

Dva grafy G a H jsou **izomorfní**, jestliže existují dvě vzájemně jednoznačná zobrazení $\Phi_V : V(G) \to V(H)$ a $\Phi_E : E(G) \to E(H)$ taková, že:

- pro orientované grafy platí $\Phi_E((v, w)) = (\Phi_V(v), \Phi_V(w))$ pro všechny $(v, w) \in E(G)$
- nebo pro neorientované grafy platí $\Phi_E(\{v, w\}) = \{\Phi_V(v), \Phi_V(w)\}$ pro všechny $\{v, w\} \in E(G)$

Izomorfními grafy se v algoritmické části zpravidla nezabýváme, ale je dobré si izomorfismus uvědomit, již ve fázi modelování.



Další pojmy a značení pro orientované grafy

V orientovaném grafu G pro vrchol v definujeme:

- množina **následovníků vrcholu** v je množina vrcholů, do kterých vede hrana z v, neboli $\{w \in V : (v, w) \in \Psi(E)\}$
- množina **předchůdců vrcholu** v je množina vrcholů, ze kterých vede hrana do v, neboli $\{w \in V : (w, v) \in \Psi(E)\}$
- $\Gamma(v)$, množina **sousedů vrcholu** v je množina vrcholů spojených hranou s vrcholem v, neboli sjednocení množin následovníků a předchůdců
- $\delta^+(v)$, množina výstupních hran vrcholu v
- $\delta^-(v)$, množina vstupních hran vrcholu v
- $\delta(v)$, množina hran incidentních s vrcholem v
- $|\delta^+(v)|$, **výstupní stupeň vrcholu**
- $|\delta^-(v)|$, **vstupní stupeň vrcholu**
- $|\delta(v)|$, **stupeň vrcholu**

Všimněme si, že platí $\sum_{v \in V(G)} |\delta(v)| = 2\,|E(G)|$.
Počet vrcholů s lichým stupněm je vždy sudý.

Další pojmy a značení pro orientované grafy

Obecněji, v orientovaném grafu G pro dvě množiny $X, Y \subseteq V(G)$ definujeme:

- $E^+(X, Y)$, množina hran z množiny X do množiny Y, neboli $E^+(X, Y) = \{(x, y) \in E(G) : x \in X \setminus Y, y \in Y \setminus X\}$
- $\delta^+(X)$, množina výstupních hran množiny X, neboli $\delta^+(X) := E^+(X, V(G) \setminus X)$
- $\delta^-(X)$, množina vstupních hran množiny X, neboli $\delta^-(X) := \delta^+(V(G) \setminus X)$
- $\delta(X)$, množina "hraničních" hran množiny X, neboli $\delta(X) := \delta^+(X) \cup \delta^-(X)$
- $\Gamma(X)$, množina sousedů množiny X, neboli $\Gamma(X) := \{v \in V(G) \setminus X :$ existuje "hraniční" hrana $e \in \delta(X)$, jež je incidentní s vrcholem $v\}$

Tytéž pojmy a značení pro neorientované grafy

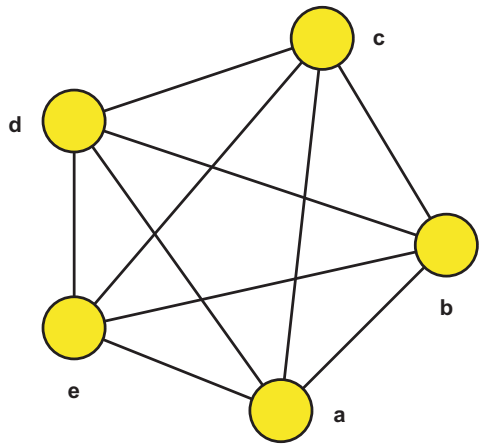
Tyto pojmy nezáleží k orientaci hran:

- $\Gamma(v)$, množina sousedů vrcholu v
- $\delta(v)$, množina hran incidentních s vrcholem v
- $|\delta(v)|$, stupeň vrcholu v
- $E(X, Y)$, množina hran mezi množinami X a Y, neboli $E(X, Y) = \{\{x, y\} \in E(G) : x \in X \setminus Y, y \in Y \setminus X\}$
- $\delta(X)$, množina "hraničních" hran množiny X, neboli $\delta(X) := E(X, V(G) \setminus X)$
- $\Gamma(X)$, množina sousedů množiny X, neboli $\Gamma(X) := \{v \in V(G) \setminus X : E(X, v) \neq \emptyset\}$

Jako základ budeme používat orientované grafy (z neorientovaného grafu uděláme orientovaný graf tak, že každou neorientovanou hranu nahradíme dvěma opačně orientovanými hranami).

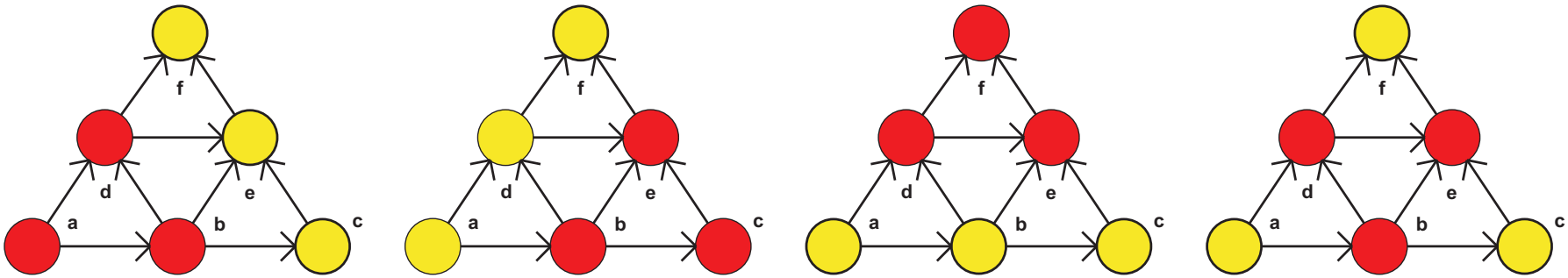
Speciální grafy

Úplný orientovaný graf je (prostý) graf $G = (V, E)$, kde E je množina všech uspořádaných dvojic různých vrcholů množiny V.
Úplný neorientovaný graf je (prostý) graf, jehož každé dva vrcholy jsou spojeny hranou. Má-li n vrcholů, značíme jej K_n.
Graf nazýváme **regulárním**, mají-li všechny vrcholy stejný stupeň. Graf, v němž všechny vrcholy mají stupeň k, nazýváme **k-regulárním**.



Speciální grafy

Doplněk prostého grafu G je prostý graf H, pro který $G + H$ je úplný graf. V doplňku jsou dva vrcholy spojeny hranou právě tehdy, když nejsou spojeny hranou v původním grafu G.
Klika je podgraf, který je úplným grafem. Počet vrcholů v největší klice nazýváme **klikovostí grafu**.



Sledy a odvozené pojmy

Sled je posloupnost vrcholů a hran $v_1, e_1, v_2, e_2, \ldots, v_k, e_k, v_{k+1}$ kde $e_i = (v_i, v_{i+1}) \in E(G)$, respektive $e_i = \{v_i, v_{i+1}\} \in E(G)$ pro všechna $i = 1, \ldots, k$.
Sled je **uzavřený** pokud $v_1 = v_{k+1}$.
Orientovaný (neorientovaný) **tah** je orientovaný (neorientovaný) sled, v němž se neopakuje žádná hrana, neboli $e_i \neq e_j$ pro všechna $1 \leq i < j \leq k$.
Orientovaná (neorientovaná) **cesta** je orientovaný (neorientovaný) tah, v němž se neopakuje žádný vrchol, neboli $v_i \neq v_j$ pro všechna $1 \leq i < j \leq k + 1$.
Cestu lze chápat jako graf $P = (\{v_1, v_2, \ldots, v_{k+1}\}, \{e_1, e_2, \ldots, e_k\})$ a nazývat ji "cesta z v_1 do v_{k+1}" nebo "v_1-v_{k+1} cesta".
Kružnice je neorientovaný tah, v němž se neopakuje žádný vrchol, kromě toho, že $v_1 = v_{k+1}$.
Cyklus je orientovaný tah, v němž se neopakuje žádný vrchol, kromě toho, že $v_1 = v_{k+1}$.

Strom

Neorientovaný graf nazýváme **souvislým**, jestliže každé jeho dva vrcholy jsou spojeny neorientovanou cestou.
Komponenta souvislosti grafu G nazýváme každý podgraf H grafu G, který je souvislý a který je maximální s touto vlastností.
Každý vrchol grafu je obsažen v právě jedné komponentě souvislosti.
Komponentu souvislosti obsahující daný vrchol x můžeme najít jako úplný podgraf určený množinou všech vrcholů, do nichž vede neorientovaná cesta z vrcholu x.
Les je takový neorientovaný graf G, který neobsahuje kružnici.
Strom je takový neorientovaný graf G, který neobsahuje kružnici a zároveň je souvislý.
Každý souvislý graf má faktor, který je stromem a nazývá se **kostra**.

Vlastnosti stromů v neorientovaných grafech

Nechť G je neorientovaný graf s n vrcholy. Potom následující tvrzení jsou vzájemně ekvivalentní:

- (a) G je strom.
- (b) G má $n-1$ hran a neobsahuje kružnici.
- (c) G má $n-1$ hran a je souvislý.
- (d) G je souvislý a odebráním kterékoli hrany přestane být souvislý.
- (e) G je minimální graf pro který platí $\delta(X) \neq \oslash$ pro všechna $\oslash \neq X \subset V(G)$.
- (f) G neobsahuje kružnici a po přidání libovolé hrany bude obsahovat přesně jednu kružnici.
- (g) Každá dvojice vrcholů je spojena přesně jednou neorientovanou cestou.

Důkaz:
(a)⇒(g): vychází z předpokladu, že sjednocením dvou cest se stejnými koncovými vrcholy vznikne kružnice.
(f)⇒(b)⇒(c): je důsledkem toho, že pro les obsahující n vrcholů, m hran a p komponent souvislosti platí vztah $n = m + p$. (Lze dokázat jednoduchou indukcí na proměnné m).
(g)⇒(e)⇒(d): viz [2] str. 17 Proposition 2.3.
(d)⇒(f): triviální.
(c)⇒(a): G má $n - 1$ hran a je souvislý. Dokud je v grafu kružnice, pak ji odstraníme odebráním odpovídající hrany. Po odstranění k kružnic získáme graf G', který je stromem (neobsahuje kružnici a je stále souvislý) a má $m = n - 1 - k$ hran. Vztah $n = m + p = n - 1 - k + 1$ implikuje $k = 0$.

Souvislost a stromy v orientovaných grafech

Orientovaný graf G je souvislý pokud jeho podkladový neorientovaný graf je souvislý.
Orientovaný graf G je **silně souvislý** pokud pro každou dvojici vrcholů x, y existuje v G orientovaná cesta z x do y a také zpět z y do x.
Silně souvislou komponentou grafu G nazýváme každý podgraf H grafu G, který je silně souvislý a je maximální s touto vlastností.
Vrchol $x \in V(G)$ je **kořen** grafu G, jestliže každý vrchol grafu G je orientovaně dostupný z vrcholu x.
Kořenový strom je takový orientovaný graf G, v němž existuje kořen. Do kořene stromu nevede žádná hrana; do každého jiného vrcholu vede přesně jedna hrana.
Binární strom je takový strom G, jehož každý vrchol má nejvýše dva následníky.
Vlastnosti stromů v neorientovaných grafech - viz [2] str. 18

Shrnutí

Grafy:

- jsou užitečný nástroj pro formalizaci řady optimalizačních problémů
- jsou velmi obecné
- snadno se reprezentují v počítači

Literatura

Jiří Demel.
Grafy a jejich aplikace.
Academia, 2002.

B. H. Korte and Jens Vygen.
Combinatorial Optimization: Theory and Algorithms.
Springer, third edition, 2006.