11 Kombinatorické optimalizační problémy

Velmi mnoho známých kombinatorických úloh má přirozené formulace v lineární či celočíselné optimalizaci. Předpokládáme, že čtenáři, zvláště ti mající informatický základ, znají různé běžné kombinatorické problémy na grafech, nebo třeba problémy splnitelnosti formulí.

Ukázkami jejich formulace zároveň čtenáři ukážeme některé obecné principy formulace úloh IP.

Stručný přehled lekce

- Formulace problémů splnitelnosti logických formulí SAT.
- Formulace běžných grafových problémů – párování, nezávislá a dominující množina, barevnost.
- Problém obchodního cestujícího.

11.1 Formulace problému SAT

Začneme s IP formulací jednoho ze základních obtížných algoritmických problémů a jeho optimalizačního rozšíření.

Definice: Logická formule Φ s proměnnými $x_1, x_2, \ldots, x_n$ je v *konjunktivním normálním tvaru* pokud je zapsaná jako

$$\Phi = c_1 \wedge c_2 \wedge \ldots \wedge c_k ,$$

kde každé c_i se nazývá *klauzule* a představuje zkratku pro

$$c_i = (\ell_{i1} \vee \ell_{i2} \vee \ldots \vee \ell_{im_i})$$

a kde ℓ_{ij} je *literál* mající jednu ze dvou podob pro některé $p \in \{1, \ldots, n\}$

$$\ell_{ij} \equiv x_p \ \text{ nebo } \ \ell_{ij} \equiv \neg x_p .$$

Příklady logických formulí v konjunktivním normálním tvaru jsou:

$$\Phi_1 = (x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_1 \vee x_3 \vee x_4) \wedge (\neg x_2 \vee \neg x_3 \vee \neg x_4)$$
$$\Phi_2 = (x_1 \vee x_2) \wedge (\neg x_1 \vee x_3 \vee x_4) \wedge (\neg x_2 \vee \neg x_3 \vee x_4) \wedge (x_1 \vee x_4 \vee \neg x_5) \wedge (x_2 \vee x_5)$$

U takovýchto formulí si budeme všímat, zda některé logické ohodnocení vstupních proměnných má za výsledek hodnotu T (pravda) celé formule.

Definice: Problém *splnitelnosti SAT* se pro danou formuli v konjunktivním normálním tvaru ptá, zda je pro nějaké ohodnocení proměnných tato formule pravdivá.

Problém *MAX-SAT* se pro danou formuli v konjunktivním normálním tvaru ptá, kolik nejvíce klauzulí v ní lze splnit vhodným ohodnocením proměnných.

Věta 11.1. *Problém MAX-SAT lze formulovat v IP s polynomiálním počtem nerovností a proměnných.*

Důkaz: Použijeme binární proměnné z_i pro logické proměnné x_i dané formule Φ a další binární proměnné t_l pro klauzule c_l ve Φ. Účelovou funkcí bude

$$\max t_1 + t_2 + \ldots + t_k$$

za podmínek, ve kterých pro každou klauzuli $c_l = x_i \vee \neg x_j \vee \ldots$ píšeme

$$t_l \leq z_i + (1 - z_j) + \ldots .$$

Snadno vidíme, že hodnota $t_l = 1$ pro klauzuli c_l je přípustná právě tehdy, pokud aspoň jedna pozitivní proměnná v c_l má hodnotu 1 nebo pokud aspoň jedna negovaná proměnná v c_l má hodnotu 0. To jest právě když je klauzule c_l splněná v ohodnocení. $\square$

Pro výše uvedený příklad formule

$$\Phi_1 = (x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_1 \vee x_3 \vee x_4) \wedge (\neg x_2 \vee \neg x_3 \vee \neg x_4)$$

důkaz Věty 11.1 vyprodukuje úlohu IP ve tvaru

$$\begin{aligned} \max\ t_1 + t_2 + t_3 \quad &: \\ t_1 \quad &\leq \quad z_1 + 1 - z_2 + z_4 \\ t_2 \quad &\leq \quad 1 - z_1 + z_3 + z_4 \\ t_3 \quad &\leq \quad 1 - z_2 + 1 - z_3 + 1 - z_4\,. \end{aligned}$$

Důsledek 11.2. *Problém SAT, a tudíž všechny problémy ve třídě NP, lze efektivně vyjádřit jako problém existence přípustného řešení pro úlohu IP.*

Důsledek 11.3. *Otázka existence přípustného řešení úlohy IP je NP-úplná a řešení úloh IP je NP-těžké.*

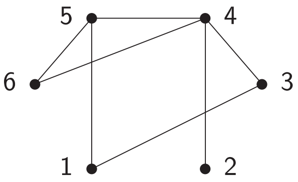
Uvědomte si, jak silné jsou uvedené důsledky. Nejen, že víme, že řešení úloh IP je v obecnosti efektivně nezvládnutelné, ale víme i, že každý problém ve třídě NP je nějakým způsobem zformulovatelný jako problém přípustnosti IP. Něco takového u mnoha příkladů vůbec není zřejmé, jak sami z vlastní zkušenosti poznáte. . .

11.2 Některé grafové problémy

Pro ukázku se podíváme na IP formulace několika známých grafových problémů.

Příklad 11.4. *Zformulujme problém párování v grafu jako problém IP.*

Množina hran M je *párováním*, pokud žádné dvě z nich nesdílí vrchol. Pro ilustraci, následující graf má největší párování velikosti 3.

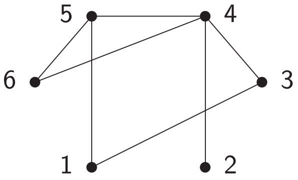


Řešení: V příkladech jako tento, kdy hledáme podmnožinu jistých vlastností, je přirozenou volbou přiřadit této podmnožině její charakteristický vektor (složený z binárních proměnných). V tomto případě tedy hraně $\{i,j\}$ přiřadíme binární proměnnou $z_{i,j}$ s významem, že $z_{i,j} = 1$ právě když $\{i,j\} \in M$. Aby vybraná podmnožina M byla párováním, může každý vrchol grafu náležet nejvýše jedné hraně M. Pro vrchol i se sousedy $j_1, \ldots, j_k$ tedy zformulujeme podmínku

$$z_{i,j_1} + z_{i,j_2} + \ldots + z_{i,j_k} \leq 1\,.$$

Účelovou funkcí – velikostí párování, je potom součet všech proměnných $z_{i,j}$ pro hrany grafu, případně i vážený součet.

Například pro graf na obrázku zformulujeme úlohu

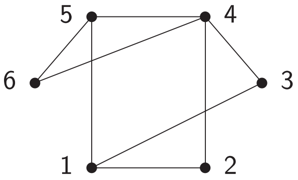


$$
\begin{array}{rcl}
\max\ z_{13}+z_{15}+z_{24}+z_{34}+z_{45}+z_{56}+z_{46} & : & \\
z_{13}+z_{15} & \leq & 1 \\
z_{13}+z_{34} & \leq & 1 \\
z_{24}+z_{34}+z_{45}+z_{46} & \leq & 1 \\
z_{15}+z_{45}+z_{56} & \leq & 1 \\
z_{46}+z_{56} & \leq & 1 \\
z_{13}, z_{15}, z_{24}, z_{34}, z_{45}, z_{56}, z_{46} & \in & \{0,1\}\,.
\end{array}
$$

□

Příklad 11.5. *Zformulujme problém nezávislé množiny v grafu jako problém IP.*

Množina vrcholů I je *nezávislá*, pokud žádné dva z nich nejsou spojené hranou. Pro ilustraci, následující graf má největší nezávislou množinu velikosti 3. Dokážete ji najít?



Řešení: V tomto příkladě hledáme podmnožinu vrcholů, takže vrcholu i přiřadíme binární proměnnou z_i s významem, že $z_i = 1$ právě když $i \in I$. Podmínka nezávislosti se pak formuluje jako nerovnost $z_i + z_j \leq 1$ pro každou hranu $\{i, j\}$ daného grafu.

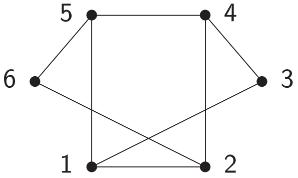
Například pro graf na obrázku zformulujeme úlohu

$$
\begin{aligned}
\max\ z_1 + z_2 + z_3 + z_4 + z_5 + z_6 &:\\
z_1 + z_2 \leq 1,\quad z_1 + z_3 &\leq 1\\
z_1 + z_5 \leq 1,\quad z_2 + z_4 &\leq 1\\
z_3 + z_4 \leq 1,\quad z_6 + z_4 &\leq 1\\
z_5 + z_4 \leq 1,\quad z_6 + z_5 &\leq 1\\
z_1, z_2, z_3, z_4, z_5, z_6 &\in \{0, 1\}\,.
\end{aligned}
$$

□

Příklad 11.6. *Zformulujme problém dominující množiny v grafu jako problém IP.*

Dominující množina D v grafu G je taková $D \subseteq V(G)$, že každý vrchol G mimo D je spojený hranou s některým vrcholem v D. Najdete v následujícím grafu dominující množinu velikosti 2?



Řešení: Opět každému vrcholu i přiřadíme binární proměnnou z_i s významem, že $z_i = 1$ právě když $i \in D$. Podmínka dominující množiny se pak formuluje pro každý vrchol i se sousedy $j_1, \dots, j_k$ jako nerovnost $z_i + z_{j_1} + \ldots + z_{j_k} \geq 1$ vyjadřující, že vrchol i je v D nebo některý jeho soused je v D.

Například pro graf na obrázku zformulujeme úlohu

$$
\begin{aligned}
\min\ z_1 + z_2 + z_3 + z_4 + z_5 + z_6 &\quad : \\
z_1 + z_2 + z_3 + z_5 &\quad \geq \quad 1 \\
z_2 + z_1 + z_4 + z_6 &\quad \geq \quad 1 \\
z_3 + z_1 + z_4 &\quad \geq \quad 1 \\
z_4 + z_2 + z_3 + z_5 &\quad \geq \quad 1
\end{aligned}
$$

$$\begin{array}{rcl} z_5 + z_1 + z_4 + z_6 & \geq & 1 \\ z_6 + z_2 + z_5 & \geq & 1 \\ z_1, z_2, z_3, z_4, z_5, z_6 & \in & \{0, 1\}\,. \end{array}$$

□

Předchozí příklady byly docela jednoduché a přímočaré, že? Co ale třeba s podobně jednoduše vypadajícím problémem barvení grafu? *Obarvením* grafu rozumíme přiřazení přirozených čísel (*barev*) vrcholům grafu tak, aby sousední vrcholy dostaly různé barvy. *Barevností grafu* pak je nejmenší počet barev, kterým lze daný graf obarvit.

Příklad 11.7. *Zformulujme problém barevnosti grafu jako problém IP.*

Přirozenou první myšlenkou většiny čtenářů asi bude použít celočíselné proměnné pro barvy jednotlivých vrcholů. (Ponechme stranou otázku omezenosti proměnných, neboť počet barev lze vždy omezit počtem vrcholů.) Jak ale vyjádříme, že sousední vrcholy mají různé barvy? Rámec formulace úloh IP nám totiž nedává možnost přímo napsat $z_i \neq z_j$.

Pokud bychom pro hranu $\{i, j\}$ chtěli například napsat $z_i \geq z_j + 1$, museli bychom připustit i možnost $z_i \leq z_j - 1$, ale my neumíme přímo vyjádřit disjunkci podmínek. (Blíže viz Oddíl 12.2.) Využijeme raději podobnost problému barvení s nezávislou množinou – vrcholy stejné barvy musí být nezávislé. Obarvení grafu k barvami tudíž znamená nalezení rozkladu množiny vrcholů na k nezávislých podmnožin.

Řešení: Pro každou potenciální barvu c (je třeba počítat s počtem barev až rovným počtu vrcholů n) zavedeme sadu binárních proměnných $z_{1,c}, \dots, z_{n,c}$, kde $z_{i,c} = 1$ znamená, že vrchol i může být obarven barvou c. Pro každé c pak přidáme na $z_{1,c}, \dots, z_{n,c}$ podmínky vyjadřující nezávislou množinu, jako v Příkladě 11.5

$$\forall c = 1, 2, \dots, n, \ \forall \{i, j\} \in E(G) : \quad z_{i,c} + z_{j,c} \leq 1 \,.$$

Dále musíme vyjádřit požadavek, že každý vrchol dostane přidělenu jednu barvu

$$\forall k = 1, 2, \dots, n : \quad z_{k,1} + z_{k,2} + \ldots + z_{k,n} = 1 \,.$$

Posledním poněkud trikovým bodem je vyjádření počtu barev, které celkem použijeme. Pro to musíme zavést další binární proměnné $t_1, \dots, t_n$ indikující ty z barev, které byly skutečně použity, a vyjádříme účel

$$\begin{array}{rrcl} & \min \ t_1 + \ldots + t_n & & \\ \forall c = 1, 2, \dots, n : & z_{1,c} + z_{2,c} + \ldots + z_{n,c} & \leq & n \cdot t_c \,. \end{array}$$

Všimněte si dobře, jakým způsobem jsme v posledních vztazích "spočítali", kolik barev c je vlastně při barvení našeho grafu použito: Pokud $t_c = 0$, barva c nemůže být použita vůbec, pokud naopak $t_c = 1$, lze c použít až na všech n vrcholech. Jedná se o docela užitečný trik k zapamatování.

□

11.3 Problém obchodního cestujícího

Klasickou dobře známou úlohou diskrétní optimalizace je tzv. problém obchodního cestujícího (TSP), motivovaný následovně: Úkolem cestujícího je obejít daných n měst (v libovolném pořadí, s návratem do výchozího) tak, aby byla celková délka či cena cesty minimální. Tento tradiční problém neopomineme ani v našem textu.

Definice: (Orientovaný) sled v grafu G je posloupnost $v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k$, kde vždy hrana e_i, $1 \leq i \leq k$ vede z vrcholu v_{i-1} do v_i.

Definice 11.8. Obecný problém TSP (Obchodního cestujícího v grafu)
V nejobecnější grafové formulaci se problém obchodního cestujícího zadá následovně:

Vstup: Orientovaný ohodn. souvislý graf G, s ohodnocením hran $w : E(G) \rightarrow \boldsymbol{R}^+$.
Výstup: Najít uzavřený orientovaný sled v G, ve kterém je součet ohodnocení (*délek*) hran nejmenší možný.

Poznámka: Uvědomme si, že optimální řešení obecného problému TSP může opakovat jeden vrchol vícekrát, například pokud graf G je stromem. Obvykle se však při formulaci problému TSP explicitně požaduje, aby se žádný vrchol ve sledu neopakoval, neboli se hledá tzv. Hamiltonovský cyklus grafu. V této "neopakovací" formulaci také graf G obvykle bývá úplný.

Příklad 11.9. *Zformulujme problém TSP na orientovaném grafu G bez povoleného opakování vrcholů ve sledu jako úlohu IP.*

Řešení: Nechť $V(G) = \{1, 2, ..., n\}$, pak každé orientované hraně $(i, j) \in E(G)$ přiřadíme proměnnou $z_{i,j} \in \{0, 1\}$. Jelikož opakování vrcholů je zakázáno, hledáme formulaci popisující Hamiltonovský cyklus v G.

- Do každého vrcholu k jednou přijdeme a jednou z něj odejdeme

  $$\sum_{i:(i,k)\in E(G)} z_{i,k} = 1, \qquad \sum_{i:(k,i)\in E(G)} z_{k,i} = 1\,.$$

  Pokud však má G dostatek vrcholů, tyto podmínky splňuje i kolekce více disjunktních cyklů, že?

- Navíc tedy musíme vyloučit případ, kdy by se náš cyklus "uzavřel" ještě před navštívením všech vrcholů. To znamená, že na každé podmnožině vrcholů $W \in V(G)$, $2 \le |W| \le n - 2$ zabráníme vytvoření *podcyklu* nerovnostmi

  $$\sum_{(i,j)\in E(G),\, i,j\in W} z_{i,j} \le |W| - 1\,, \tag{1}$$

  které zaručí pokračování cyklu mimo W.

Takto sestavených nerovností pro TSP je bohužel zhruba 2^n, takže je nelze ani rozumně zapsat do paměti. Přesto se tato formulace TSP v praxi používá tak, že nerovnosti v (1) se vygenerují "na žádost", tj. například pro jeden konkrétní podcyklus v částečném řešení přidáme příslušnou vylučující nerovnost. Ukazuje se, že nakonec je třeba využít obvykle jen rozumně malou část ze všech těchto podcyklových nerovností. □

Poznámka: V poválečné době bylo velkým matematickým úspěchem vyřešit problém TSP na 49 hlavních městech kontinentálních států USA. S dnešními vylepšenými algoritmy a výkonnými počítači jsme schopni řešit obecné problémy TSP až na tisících vrcholech. Přesto tento optimalizační problém zůstává velmi obtížný a také velmi populární.

Aproximace Euklidovského TSP

Původní motivace problému TSP nese oproti abstraktní formulaci jednu podstatnou informaci navíc – délky hran mezi městy splňují trojúhelníkovou nerovnost. Toho lze s výhodou použít při aproximaci řešení TSP.

Definice: Problém TSP se nazývá *Euklidovský*, pokud délky hran grafu splňují trojúhelníkovou nerovnost (graf musí být úplný).

Speciálně tedy pokud jsou délky dány Euklidovskou metrikou.

Příklad 11.10. *Ukážeme, jak lze problém Euklidovského neorientovaného TSP přibližně vyřešit v polynomiálním čase s cenou nejvýše dvojnásobnou od optimálního řešení.*

Řešení: Na (neorientovaném) grafu G najdeme minimální kostru $T \subseteq G$ (například hladovým algoritmem v čase $O(m \cdot \log m)$, $m = |E(G)|$). Pak "zdvojením" každé hrany T dostaneme uzavřený sled délky dvojnásobku T.

Pokud je vyloučeno opakování vrcholů, tak nakonec tento sled "narovnáme" přeskočením opakovaných vrcholů. Podle trojúhelníkové nerovnosti si narovnáním sled neprodloužíme, ale v praxi většinou zkrátíme. □