O dynamickém programování

ŠKOLA MLADÝCH MATEMATIKŮ

O DYNAMICKÉM PROGRAMOVÁNÍ

33

Vydal ÚV Matematické olympiády a ÚV ČSM v nakladatelství Mladá fronta

ŠKOLA MLADÝCH MATEMATIKŮ

JAROSLAV MORÁVEK

o dynamickém programování

PRAHA
VYDAL ÚV MATEMATICKÉ OLYMPIÁDY
V NAKLADATELSTVÍ MLADÁ FRONTA

PŘEDMLUVA

Motivem k napsání této knížky byla moje týdenní přednáška na letním soustředění řešitelů MO v červnu 1970 v Martině. Je určena převážně řešitelům MO a zájemcům o hlubší studium matematiky a jejích aplikací z řad středoškolských studentů, avšak nevylučuje se ani širší okruh čtenářů, kteří chtějí získat první představu o dynamickém programování, a přitom nemají chuť začínat s obšírnými publikacemi.

Na rozmanitých úlohách na nalezení maxima nebo minima (převážně pro číselné funkce definované na konečných množinách) se čtenář seznámí s ideou dynamického programování, záležející v jistém rozkladu původní úlohy na několik úloh jednodušších.

Vývoj metod dynamického programování byl ve značné míře ovlivněn potřebami matematiky v jiných vědních a technických oborech, jako např. při optimálním řízení ekonomických a technologických procesů, při výpočtu drah raket, družic aj. Tato skutečnost určila i výběr látky pro naši knížku: pro většinu uvažovaných úloh na určení maxima nebo minima jsou uvedeny konkrétní, i když samozřejmě zjednodušené, praktické aplikace.

Autor

1. kapitola

POJEM EXTREMÁLNÍHO PROBLÉMU

Cílem této knížky je ukázat jednu elementární metodu pro řešení jisté třídy extremálních problémů. S pojmem *extremálního problému* (tj. *úlohy na určení maxima nebo minima*) se čtenář asi setkal již na střední škole. Uveďme několik příkladů extremálních úloh.

1. Máme nalézt nejmenší hodnotu kvadratického trojčlenu $x^2 + x + 1$, kde x je libovolné reálné číslo.

2. Z množiny všech obdélníků dané délky obvodu máme nalézt obdélníky s největším obsahem.

3. Z množiny všech obdélníků daného obsahu máme nalézt obdélníky s nejmenší délkou obvodu.

4. V prostoru jsou dány dvě mimoběžky p a q. Máme určit všechny dvojice bodů **P** a **Q** takové, že **P** leží na p, **Q** leží na q a vzdálenost mezi **P** a **Q** je minimální.

5. Na šachovnici máme umístit nejmenší počet královen tak, aby každé pole bylo „ohroženo".

Všechny uvedené příklady lze zahrnout do obecného schématu extremálního problému. V extremálním problému jde o tuto situaci: každému prvku nějaké pevně zvolené množiny je přiřazeno nějaké reálné číslo; úkolem je nalézt nejmenší (popříp. největší) z takto přiřazených čísel a příslušné prvky, kterým je toto nejmenší (popříp. největší) číslo přiřazeno.

Při vyšetřování extremálních problémů budeme pracovat s pojmem množiny a s pojmem funkce na této množině definované. *Množinou* rozumíme soubor (souhrn) všech objektů, charakterizovaných nějakou vlastností, popříp. několika vlastnostmi. Množiny označujeme obvykle velkými písmeny, např. $\mathbf{A}$, $\mathbf{A}_1$, $\mathbf{B}$, $\mathbf{M}$ apod. Fakt, že $\mathbf{A}$ je množina všech objektů množiny $\mathbf{B}$, majících vlastnost V, budeme symbolicky zapisovat

$$\mathbf{A} = \{x \in \mathbf{B} \mid x \text{ má vlastnost } V\}$$

(V případě, že množina $\mathbf{B}$ bude známa z kontextu, budeme používat též stručnějšího zápisu $\mathbf{A} = \{x|x \text{ má vlastnost } V\}$.)

Objekty náležející dané množině se nazývají jejími prvky, zápis $x \in \mathbf{A}$ znamená „*x je prvkem* $\mathbf{A}$“, zápis $x \notin \mathbf{A}$ — „*x není prvkem* $\mathbf{A}$“. Říkáme, že množiny $\mathbf{A}$ a $\mathbf{B}$ *se sobě rovnají*, což zapisujeme symbolem $\mathbf{A} = \mathbf{B}$, jestliže každý prvek množiny $\mathbf{A}$ je prvkem množiny $\mathbf{B}$, a obráceně, každý prvek množiny $\mathbf{B}$ je prvkem množiny $\mathbf{A}$. Negaci vztahu $\mathbf{A} = \mathbf{B}$ zapisujeme pomocí $\mathbf{A} \neq \mathbf{B}$ a čteme „$\mathbf{A}$ *je různé od* $\mathbf{B}$“, nebo „$\mathbf{A}$ *se nerovná* $\mathbf{B}$“. Říkáme, že množina $\mathbf{A}$ je *částí* množiny $\mathbf{B}$ (též $\mathbf{A}$ je *podmnožinou* $\mathbf{B}$), jestliže každý prvek $\mathbf{A}$ je prvkem $\mathbf{B}$.

Množina se nazývá *konečná*, jestliže obsahuje konečně mnoho prvků. Množina, která není konečná, se nazývá *nekonečná*. Příkladem nekonečné množiny je množina všech přirozených čísel, nebo množina všech reálných čísel. Poslední množinu budeme označovat v celé knížce symbolem $\mathbf{R}$. Přitom místo výrazu „reálné číslo“ budeme používat většinou stručnějšího „číslo“.

Mezi konečné množiny zahrnujeme i *prázdnou* množinu, definovanou tím, že neobsahuje žádný prvek. Prázdnou množinu označujeme symbolem $\emptyset$. Množina se nazývá *neprázdná*, obsahuje-li alespoň jeden prvek. Konečnou neprázdnou množinu obsahující prvky x_1, x_2, $\ldots$, x_n, kde n označuje počet jejích prvků, označujeme symbolem $\{x_1, x_2, \ldots, x_n\}$. V posledním zápisu nezáleží na pořadí prvků $x_1, x_2, \ldots, x_n$. Všimněme si však, že v uvažovaném zápisu jsou všechny prvky $x_1, \ldots, x_n$ navzájem různé.

Pro množiny se definuje řada operací, jako např. sjednocení, průnik, rozdíl, kartézský součin aj. Pro náš výklad budeme potřebovat pojem sjednocení množin. Nejprve si připomeneme definici *sjednocení* dvou množin, známou asi už ze školy. Nechť jsou dány množiny $\mathbf{A}$, $\mathbf{B}$. *Sjednocením* množin $\mathbf{A}$, $\mathbf{B}$ budeme rozumět množinu označovanou $\mathbf{A} \cup \mathbf{B}$ a definovanou takto: $x \in \mathbf{A} \cup \mathbf{B}$ právě tehdy, jestliže je splněna alespoň jedna z podmínek: $x \in \mathbf{A}$ nebo $x \in \mathbf{B}$. Pojem sjednocení rozšíříme dále na případ libovolného konečného počtu množin. Nechť jsou dány množiny $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_n$. *Sjednocením* množin $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_n$ budeme rozumět množinu $\mathbf{C}$ definovanou takto: vztah $x \in \mathbf{C}$ platí právě tehdy, jestliže existuje j ($j = 1, 2, \ldots, n$) tak, že $x \in \mathbf{A}_j$. Sjednocení množin $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_n$ označujeme symbolem $\mathbf{A}_1 \cup \mathbf{A}_2 \cup \ldots \cup \mathbf{A}_n$ nebo $\bigcup\limits_{j=1}^{n} \mathbf{A}_j$ nebo $\bigcup\limits_{j=1,\ldots,n} \mathbf{A}_j$. (Speciálně tedy dostáváme $\bigcup\limits_{j=1} \mathbf{A}_j = \bigcup\limits_{j=1}^{1} \mathbf{A}_j = \mathbf{A}_1$ a $\bigcup\limits_{j=1,2} \mathbf{A}_j = \bigcup\limits_{j=1}^{2} \mathbf{A}_j = \mathbf{A}_1 \cup \mathbf{A}_2$.)

Dále se budeme zabývat pojmem *funkce*. Ve škole jste se asi setkali s pojmem funkce, jakožto zobrazení z $\mathbf{R}$ do $\mathbf{R}$*). Pro náš výklad pojem funkce poněkud rozšíříme. Nechť je dána libovolná neprázdná množina $\mathbf{A}$. *Funkcí definovanou na množině* $\mathbf{A}$ budeme rozumět jakýkoliv předpis, podle kterého je každému $x \in \mathbf{A}$ přiřazeno jediné reálné číslo. Funkce budeme označovat

*) Připomeňme si, že $\mathbf{R}$ označuje množinu všech reálných čísel.

malými tučnými typy, např. $\boldsymbol{f}$, $\boldsymbol{g}$, $\boldsymbol{h}$ apod. Nechť $\boldsymbol{f}$ je funkce definovaná na $\mathbf{A}$ a nechť $x \in \mathbf{A}$. Symbolem $\boldsymbol{f}(x)$ označíme číslo přiřazené prvku x funkcí $\boldsymbol{f}$, ve smyslu vyslovené definice. Číslo $\boldsymbol{f}(x)$ budeme nazývat *hodnotou funkce* $\boldsymbol{f}$ v x, nebo stručněji *hodnotou* $\boldsymbol{f}(x)$. Množinu $\mathbf{A}$ nazýváme definičním oborem funkce $\boldsymbol{f}$. Ve smyslu naší definice je tedy každá funkce určena dvěma „věcmi": 1) svým definičním oborem $\mathbf{A}$, 2) předpisem, podle kterého je každému $x \in \mathbf{A}$ přiřazeno jediné $\boldsymbol{f}(x)$.

Pro funkci $\boldsymbol{f}$ definovanou na množině $\mathbf{A}$, zavedeme dále toto označení: Symbolem $\boldsymbol{f}(\mathbf{A})$ označíme množinu všech hodnot $\boldsymbol{f}(x)$, kde $x \in \mathbf{A}$. V souhlasu s dříve zavedeným zápisem množiny pomocí charakteristické vlastnosti pro její prvky můžeme psát

$$\boldsymbol{f}(\mathbf{A}) = \{\boldsymbol{f}(x) \mid x \in \mathbf{A}\}$$

Množinu $\boldsymbol{f}(\mathbf{A})$ nazýváme *oborem hodnot* funkce $\boldsymbol{f}$.

Zmíníme se dále o jednom speciálním případu, kdy prvek x definičního oboru nějaké funkce $\boldsymbol{f}$ má sám složitou strukturu v tom smyslu, že je *uspořádanou n-ticí nějakých objektů* $x_1, x_2, \ldots, x_n$, což zapisujeme $x = (x_1, x_2, \ldots, x_n)$*). V tomto případě nazýváme příslušnou funkci $\boldsymbol{f}$ *funkcí n proměnných* (jestliže $n > 1$, mluvíme též o *funkci více proměnných*) a její hodnotu v $x = (x_1, x_2, \ldots, x_n)$ označujeme $\boldsymbol{f}(x_1, x_2, \ldots, x_n)$. V této souvislosti poznamenejme, že budeme mluvit výhradně o uspořádaných n-ticích, takže přívlastek „uspořádaný" budeme většinou vynechávat.

K označování funkcí se kromě stručného způsobu $\boldsymbol{f}$, $\boldsymbol{g}$, $\boldsymbol{h}$ apod., používá též obšírnějšího zápisu $y = \boldsymbol{f}(x)$,

---

*) Srovnej se zápisem $\{x_1, x_2, \ldots, x_n\}$ konečné neprázdné množiny, ve kterém na pořadí $x_1, x_2, \ldots, x_n$ nezáleží.

$y = \boldsymbol{g}(x)$, $y = \boldsymbol{h}(x)$, apod. Obdobný způsob zápisu budeme používat v případě, že funkce bude definována pomocí nějakého „vzorce". Např. symbolem $y = x^2 + x + 1$ budeme v souhlasu s běžnou zvyklostí rozumět funkci, jejímž definičním oborem je $\mathbf{R}$, popřípadě libovolná neprázdná podmnožina $\mathbf{R}$, a která každému číslu x z definičního oboru přiřazuje číslo $x^2 + x + 1$. Předností tohoto způsobu zápisu konkrétní funkce je jednak jeho názornost, jednak to, že odpadá nutnost zavádět nový symbol. Podobně zápisem $y = x_1 + x_2 + \ldots + x_n$ budeme označovat funkci, která uspořádaným n-ticím $(x_1, x_2, \ldots, x_n)$ reálných čísel přiřazuje jejich součet $x_1 + x_2 + \ldots + x_n$. (Definičním oborem takové funkce je množina všech uspořádaných n-tic čísel, nebo její libovolná neprázdná podmnožina.)

Nyní zavedeme pojmy maxima a minima množiny čísel a maxima a minima funkce. Nechť je dána neprázdná množina čísel $\mathbf{M}$. Číslo a_0 nazveme *minimálním prvkem (minimem)* množiny $\mathbf{M}$, jestliže platí

1. $a_0 \in \mathbf{M}$, 2. $a_0 \leqq a$ pro všechna $a \in \mathbf{M}$

Číslo a_1 nazveme *maximálním prvkem (maximem)* množiny $\mathbf{M}$, jestliže platí

1. $a_1 \in \mathbf{M}$, 2. $a_1 \geqq a$ pro všechna $a \in \mathbf{M}$

Poznamenejme, že ani minimum ani maximum množiny čísel nemusí obecně existovat; některé příklady ukážeme v příští kapitole.

Pojmy minima a maxima funkce se definují pomocí pojmů minima a maxima množiny čísel. Nechť $\boldsymbol{f}$ je funkce definovaná na neprázdné množině $\mathbf{A}$. Obor hodnot $\boldsymbol{f}(\mathbf{A})$ je rovněž neprázdná množina. Minimum množiny $\boldsymbol{f}(\mathbf{A})$ potom nazýváme *minimem* (nebo *minimální hodnotou*) funkce $\boldsymbol{f}$ *na* množině $\mathbf{A}$ a maximum množiny $f(\mathbf{A})$ nazýváme *maximem* (nebo *maximální hodnotou*) funkce f *na* množině $\mathbf{A}$. V případě, že definiční obor $\mathbf{A}$ je znám z kontextu, mluvíme stručně pouze o minimu (minimální hodnotě), resp. maximu (maximální hodnotě) funkce f.

Z uvedené definice vyplývá bezprostředně, že minimum funkce f na $\mathbf{A}$ je takové číslo f_0, pro které

1. existuje $x_0 \in \mathbf{A}$ tak, že $f_0 = f(x_0)$
2. pro všechna $x \in \mathbf{A}$ platí $f(x) \geqq f(x_0)$

Analogicky platí, že maximum funkce f na $\mathbf{A}$ je takové číslo f_1, pro které

1. existuje $x_1 \in \mathbf{A}$ tak, že $f_1 = f(x_1)$
2. pro všechna $x \in \mathbf{A}$ platí $f(x) \leqq f(x_1)$

Přitom o číslech x_0 resp. x_1 říkáme, že f *nabývá v* x_0 (nebo *pro* x_0) svého minima, a f *nabývá v* x_1 (nebo *pro* x_1) svého maxima.

Podobně jako v případě množin čísel, nemusí obecně existovat ani minimum ani maximum funkce. Minimum funkce f na $\mathbf{A}$ budeme označovat symbolem

$$\min_{x \in \mathbf{A}} f(x)$$

a maximum f na $\mathbf{A}$

$$\max_{x \in \mathbf{A}} f(x)$$

V případě, že minimum (resp. maximum) funkce f na $\mathbf{A}$ existuje, platí podle našich definic vztahy

$$\min_{x \in \mathbf{A}} f(x) = \min f(\mathbf{A}) = \min \{f(x) \mid x \in \mathbf{A}\}$$

resp.

$$\max_{x \in \mathbf{A}} f(x) = \max f(\mathbf{A}) = \max \{f(x) \mid x \in \mathbf{A}\}$$

Minimum a maximum funkce se souhrnně nazývají jejími *extrémy*.

Nyní, když jsme zavedli pojem extrému funkce a řadu pojmů s ním souvisejících, budeme schopni přesněji než na začátku kapitoly zformulovat, co rozumíme extremálním problémem.

Nechť je dána funkce f definovaná na neprázdné množině $\mathbf{A}$. *Úlohou určení minima* funkce f se obvykle rozumí problém určení min $\{f(x)|x \in \mathbf{A}\}$ a dále alespoň jednoho prvku x_0, pro který nabývá f svého minima (popř. všech prvků, pro něž f nabývá svého minima). Analogicky *úlohou určení maxima* funkce f se rozumí problém určení max $\{f(x)|x \in \mathbf{A}\}$, a dále alespoň jednoho prvku x_1, pro který nabývá f svého maxima (popř. všech prvků, pro něž nabývá f svého maxima). V tomto ne zcela jednoznačně vymezeném smyslu budeme chápat úlohu určení minima i úlohu určení maxima též v naší knížce. Přitom nám zpravidla půjde o nalezení pouze jednoho prvku (libovolného), pro který funkce nabývá svého extrému, pokud ovšem nebude výslovně řečeno něco jiného.

Obě zformulované úlohy zahrnujeme společným názvem — *extremální problém (extremální úloha)*. Přitom prvek, ve kterém funkce nabývá svého extrému, se nazývá *řešením* extremálního problému.

Příští kapitola má rovněž ještě pomocný charakter. Uvádíme v ní některé elementární vlastnosti číselných množin a funkcí, které budeme při výkladu metody dynamického programování potřebovat.

2. kapitola

VLASTNOSTI MINIM A MAXIM

V první kapitole jsme se zmiňovali o tom, že neprázdná množina čísel nemusí mít ani minimum ani maximum. Vezměme např. tyto množiny: $\mathbf{M}_1 = \mathbf{R}$, $\mathbf{M}_2 = \{x \in \mathbf{R} | 0 < x < 1\}$, $\mathbf{M}_3 = \{x \in \mathbf{R} | x$ je přirozené$\}$, (tj. $\mathbf{M}_3$ je množina všech přirozených čísel), $\mathbf{M}_4 = \{x \in \mathbf{R} | -\sqrt{2} \leqq x \leqq \sqrt{2}$ a x je racionální$\}$, $\mathbf{M}_5 = \{x \in \mathbf{R} | 0 < x \leqq 1\}$. Snadno lze ověřit následující jednoduchá fakta:

a) množiny $\mathbf{M}_1$, $\mathbf{M}_2$ a $\mathbf{M}_4$ nemají ani minimální ani maximální prvek

b) $\mathbf{M}_3$ má minimální, ale nemá maximální prvek

c) $\mathbf{M}_5$ má maximální, ale nemá minimální prvek

Jako příklad ověříme tvrzení c). Skutečně, číslo 1 je maximálním prvkem $\mathbf{M}_5$, protože $1 \in \mathbf{M}_5$ a pro všechna $x \in \mathbf{M}_5$ platí $x \leqq 1$. Na druhé straně připusťme, že $\mathbf{M}_5$ má minimální prvek x_0. Potom však musí platit $1 \geqq x_0 > 0$ (protože $x_0 \in \mathbf{M}_5$), a tedy rovněž platí $1 \geqq \frac{x_0}{2} > 0$, odkud přicházíme k závěru, že $\frac{x_0}{2}$ rovněž náleží množině $\mathbf{M}_5$. Z předpokladu $x_0 = \min \mathbf{M}_5$ však nyní vyplývá, že musí platit $x_0 \leqq x_0/2$, což odporuje $x_0 > 0$. Důkaz tvrzení c) je dokončen. Doporučujeme čtenáři aby samostatně dokázal tvrzení a) a b).

Existuje-li však minimum (resp. maximum) neprázdné množiny čísel, existuje právě jedno, jak vyplývá z další věty.

Věta 1: Neprázdná množina čísel má nejvýše jedno minimum a nejvýše jedno maximum.

Důkaz: Označíme uvažovanou množinu $\mathbf{M}$ a dokážeme pouze první část tvrzení, pojednávající o minimu, neboť druhá část se dokazuje zcela analogicky. Důkaz provedeme sporem: Připusťme, že $\mathbf{M}$ má dvě navzájem různá minima a_0 a b_0. Z toho, že a_0 je minimum $\mathbf{M}$ a b_0 prvkem $\mathbf{M}$ vyplývá $a_0 \leqq b_0$, a obráceně, z toho, že b_0 je minimum $\mathbf{M}$ a a_0 prvkem $\mathbf{M}$ vyplývá $b_0 \leqq a_0$. Dostáváme tedy nakonec $a_0 = b_0$, což dokazuje naše tvrzení.

Následující věta ukazuje jeden systém*) množin, majících vždycky minimum a maximum.

Věta 2: Nechť $\mathbf{M}$ je konečná neprázdná množina čísel. Potom existuje $\min \mathbf{M}$ a $\max \mathbf{M}$.

Důkaz této věty budeme provádět metodou úplné (ve škole se říká též „matematické“) indukce. Nebude na škodu připomenout si ji. Nechť $V(n)$ označuje nějaké tvrzení o přirozeném čísle n a nechť n_0 je nějaké pevně zvolené přirozené číslo. Naším cílem je ověřit, zdali tvrzení $V(n)$ platí pro všechna přirozená čísla $n \geqq n_0$. Poslední tvrzení se často dokazuje podle tohoto schématu:

(i) nejprve se ověří platnost $V(n_0)$
(ii) dále se ověří, že pro libovolné $n \geqq n_0$ z platnosti $V(n)$ vyplývá platnost $V(n+1)$

---

*) Místo „množina množin“ se často z jazykových důvodů používá hezčího slovního spojení „systém množin“.

Uvedené schéma důkazu se nazývá *úplnou* (nebo *matematickou*) *indukcí*.*) Krok (i) se přitom nazývá *bází indukce*, krok (ii) *indukčním krokem*, n se nazývá *parametrem indukce*.

Přistoupíme k důkazu věty.

Důkaz věty 2: Množinu $\mathbf{M}$ můžeme zapsat ve tvaru

$$\mathbf{M} = \{x_1, x_2, \ldots, x_n\}$$

kde $x_1, x_2, \ldots, x_n$ jsou její prvky. Větu dokážeme úplnou indukcí vzhledem k n, jakožto parametru indukce, a pro $n_0 = 1$. Přitom se omezíme pouze na důkaz existence minima; důkaz existence maxima se provádí zcela analogicky a doporučujeme jej čtenáři jako lehké cvičení.

(i) V případě $n = 1$ máme jednoprvkovou množinu $\mathbf{M} = \{x_1\}$. Protože $x_1 \in \mathbf{M}$ a pro žádné $y \in \mathbf{M}$ neplatí $y < x_1$, je $x_1 = \min \mathbf{M}$, takže tvrzení je v tomto případě dokázáno.

(ii) Předpokládejme, že tvrzení je dokázáno pro přirozené číslo n a uvažujme libovolnou $(n + 1)$-prvkovou množinu

$$\mathbf{M} = \{x_1, x_2, \ldots, x_n, x_{n+1}\}$$

Množina $\mathbf{M}_0 = \{x_1, x_2, \ldots, x_n\}$ je n-prvková, a tedy podle indukčního předpokladu existuje $\min \mathbf{M}_0 = x_{i_0}$, kde $1 \leqq i_0 \leqq n$. Rozlišíme nyní dva případy:

$$(\alpha)\ x_{i_0} < x_{n+1} \qquad (\beta)\ x_{i_0} > x_{n+1}$$

(Případ $x_{i_0} = x_{n+1}$ nemůže nastat, neboť při zápisu množiny ve tvaru $\{\ldots\}$ zapisujeme pouze její navzájem různé prvky.)

*) Jde vlastně o jeden z axiómů Peanovy axiomatiky přirozených čísel.

V případě (α) dostáváme jednak $x_{i_0} \in \mathbf{M}_0$, a tedy $x_{i_0} \in \mathbf{M}$, jednak $x_{i_0} \leqq x$ pro všechna $x \in \mathbf{M}$. Odtud přicházíme k závěru, že $x_{i_0} = \min \mathbf{M}$.

V případě (β) dostáváme obdobně $x_{n+1} \in \mathbf{M}$ a $x_{n+1} \leqq \leqq x$ pro všechna $x \in \mathbf{M}$ a tedy $x_{n+1} = \min \mathbf{M}$.

V obou případech (α), (β) tedy existuje min $\mathbf{M}$, což dokončuje důkaz indukcí.

Poznámka: Vyloženého důkazu lze použít i jako metody pro skutečné nalezení minima konečné neprázdné množiny čísel.

Důležitým důkazovým nástrojem v naší knížce bude další věta, jejíž odvození provede čtenář lehce sám s použitím metody úplné indukce.

Věta 3: Nechť $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n$ jsou dané neprázdné množiny čísel a položme

$$\mathbf{M} = \mathbf{M}_1 \cup \mathbf{M}_2 \cup \ldots \cup \mathbf{M}_n$$

a) Jestliže existuje min $\mathbf{M}_j$ pro $j = 1, 2, \ldots, n$, potom existuje rovněž min $\mathbf{M}$ a platí

$$\min \mathbf{M} = \min \{\min \mathbf{M}_j | j = 1, 2, \ldots, n\}$$

b) Jestliže existuje max $\mathbf{M}_j$ pro $j = 1, 2, \ldots, n$, potom existuje rovněž max $\mathbf{M}$ a platí

$$\max \mathbf{M} = \max \{\max \mathbf{M}_j | j = 1, 2, \ldots, n\}$$

Problém nalezení minima množiny čísel lze převést na problém nalezení maxima, a obráceně. Platí totiž následující věta, jejíž snadný důkaz rovněž přenecháváme čtenáři.

Věta 4: Nechť $\mathbf{M}$ je neprázdná množina čísel. Položme

$$\mathbf{M}_1 = \{x \in \mathbf{R} | -x \in \mathbf{M}\}$$

a) Jestliže existuje $\min \mathbf{M}_1$, existuje rovněž $\max \mathbf{M}$ a platí $\max \mathbf{M} = -\min \mathbf{M}_1$.
b) Jestliže existuje $\max \mathbf{M}_1$, existuje rovněž $\min \mathbf{M}$ a platí $\min \mathbf{M} = -\max \mathbf{M}_1$.

Je-li například $\mathbf{M}$ konečná a neprázdná, můžeme psát $\mathbf{M} = \{m^{(1)}, m^{(2)}, \ldots, m^{(r)}\}$, kde $m^{(1)}, m^{(2)}, \ldots, m^{(r)}$ jsou všechny prvky $\mathbf{M}$, a z věty 4 dostáváme

$$\max \{m^{(1)}, \ldots, m^{(r)}\} = -\min \{-m^{(1)}, \ldots, -m^{(r)}\}$$

a

$$\min \{m^{(1)}, \ldots, m^{(r)}\} = -\max \{-m^{(1)}, \ldots, -m^{(r)}\}$$

Věty 1—4 se týkaly minim a maxim množin čísel. Uvedeme nyní analogické věty pro extrémy funkcí.

Minimum (resp. maximum) funkce bylo definováno jako minimum (resp. maximum) jejího oboru hodnot. Odtud vyplývají bezprostředně následující tvrzení.

Věta 5: Každá funkce má nejvýše jedno minimum a nejvýše jedno maximum.

Věta 6: Nechť funkce f je definována na konečné neprázdné množině $\mathbf{A}$. Potom existuje $\min\limits_{x \in \mathbf{A}} f(x)$ a $\max\limits_{x \in \mathbf{A}} f(x)$.

Ukážeme nyní jeden postup, pomocí kterého lze skutečně nalézt extrémy funkce definované na konečné neprázdné množině. Místo slova „postup“ budeme používat, jak je v současné matematice běžné, výrazu „algoritmus“. Pod algoritmem rozumíme — názorně, ale dosti nepřesně řečeno — postup při konkrétním řešení nějakého problému, který je možno přesně popsat ukázáním pořadí jednotlivých kroků při řešení a který po konečně mnoha krocích končí. Na základě algoritmu může úlohu řešit výpočtář, kterému stačí se algoritmu „naučit“, aniž by rozuměl hlubším matematickým důvodům, proč algoritmus úlohu řeší. (Algoritmus lze naprogramovat i na samočinný počítač.) Uvedeme několik příkladů algoritmů většinou dobře známých ze školy, i když se tam ne vždy tímto názvem označují.

1. Eukleidův algoritmus pro určení největšího společného dělitele dvou celých čísel.
2. Známé tradiční postupy pro provádění základních aritmetických operací nad čísly zapsanými v desítkové soustavě.
3. Eratosthenovo síto pro nalezení všech prvočísel nepřesahujících dané přirozené n.
4. Postup při konstrukci trojúhelníku ze tří stran.
5. Známý postup při zjišťování dělitelnosti třemi celého čísla zapsaného v desítkové soustavě.

Nyní popíšeme jednoduchý algoritmus pro nalezení minima a maxima funkce definované na konečné neprázdné množině. Přitom se opět omezíme pouze na hledání minima a analogickou otázku hledání maxima ponecháváme čtenáři. Nechť f je funkce definovaná na konečné neprázdné množině A. Prvky množiny A uspořádáme v nějakém pevně zvoleném pořadí a_1, a_2, $\ldots$, a_r, přičemž na volbě pořadí nezáleží, a pro zjednodušení zápisu položíme $f_j = f(a_j)$ $(j = 1, 2, \ldots, r)$. (Písmenem r jsme označili počet prvků množiny A.) Nalezení minima funkce f na A je nyní totéž, jako nalezení $\min \{f_j | j = 1, \ldots, r\}$, protože $\min \{f(x) | x \in \mathsf{A}\} = \min \{f_j | j = 1, \ldots, r\}$. Zavedeme si dále toto označení: Minimem r-tice $(f_1, f_2, \ldots, f_r)$ budeme rozumět číslo $\min \{f_j | j = 1, \ldots, r\}$ a budeme je označovat $\min (f_1, f_2, \ldots, f_r)$. Všimněme si dobře rozdílu mezi označením právě definovaným a označením $\min \{f_1, \ldots, f_r\}$, které má smysl pouze v tom případě, že čísla $f_1, f_2, \ldots, f_r$ jsou navzájem různá, a tvoří tedy jistou množinu čísel. Naproti tomu v označení $\min(f_1, f_2, \ldots, f_r)$ nemusí být čísla $f_1, f_2, \ldots, f_r$ navzájem různá, neboť $(f_1, f_2, \ldots, f_r)$ neoznačuje množinu, ale uspořádanou r-tici čísel.

Algoritmus, který popíšeme, bude hledat $f^* = \min(f_1, f_2, \ldots, f_r)$ a nejmenší hodnotu indexu j, pro kterou $f^* = f_j$. Algoritmus konstruuje rekurentně dvě r-tice $(b_1, b_2, \ldots, b_r)$ a $(j_1, j_2, \ldots, j_r)$ podle těchto rekurentních pravidel:

(i) Položíme $b_1 = f_1$ a $j_1 = 1$.

(ii) Nechť je již určeno b_k a j_k, kde $1 \leqq k < r$. Položíme

$$b_{k+1} = \min(b_k, f_{k+1})$$

a

$$j_{k+1} = \begin{cases} j_k & \text{jestliže } b_{k+1} = b_k \\ k+1 & \text{jestliže } b_{k+1} < b_k \end{cases}$$

(iii) Určením dvojice čísel b_r a j_r je práce algoritmu skončena.

Nyní platí následující tvrzení, jehož snadný důkaz přenecháváme čtenáři.

Věta 7: O číslech b_r a j_r platí:

$b_r = \min(f_1, f_2, \ldots, f_r)$ a j_r je nejmenší hodnota indexu j, pro který platí $f_j = \min(f_1, f_2, \ldots, f_r)$.

Použití vyloženého algoritmu budeme ilustrovat na následujícím příkladu.

Příklad 1: Chceme určit minimální číslo v 5-tici $(f_1, f_2, \ldots, f_5) = (1; 0{,}9; 3; 0{,}9; 1{,}2)$. Pomocí vyloženého algoritmu dostáváme postupně

$$\begin{array}{llll} b_1 = 1 & & & j_1 = 1 \\ b_2 = \min(b_1, f_2) = \min(1; 0{,}9) & = 0{,}9 & & j_2 = 2 \\ b_3 = \min(b_2, f_3) = \min(0{,}9; 3) & = 0{,}9 & & j_3 = 2 \end{array}$$

$$b_4 = \min(b_3, f_4) = \min(0{,}9; 0{,}9) = 0{,}9 \qquad j_4 = 2$$
$$b_5 = \min(b_4, f_5) = \min(0{,}9; 1{,}2) = 0{,}9 \qquad j_5 = 2$$

Docházíme k závěru, že minimální číslo v uvažované 5-tici je 0,9 a jeho první výskyt zleva v 5-tici je na druhém místě. Číslo 0,9 se však rovněž vyskytuje i na čtvrtém místě zleva.

Poznámka 1: Čtenáři by se uvedený algoritmus mohl zdát zbytečný, protože úlohu určení nejmenšího (největšího) z několika daných čísel řeší běžně bez přemýšlení, „jedním pohledem". Je nutné si však uvědomit, že v případě „velkých" souborů čísel nelze všechna „přehlédnout" zrakem a naši inteligenčně-vizuální schopnost musíme nahradit systematickým algoritmickým postupem. Tato okolnost ještě více vynikne v případě použití samočinných počítačů, které se „neumějí dívat".

Poznámka 2: Všimněme si dále, že popsaný algoritmus pro určení $\min(f_1, f_2, \ldots, f_r)$ vyžaduje $r - 1$ operací srovnání čísel podle velikosti; jde o srovnání čísel b_k a f_{k+1} pro $k = 1, 2, \ldots, r - 1$.

Pokročíme dále ve zkoumání vlastností extrémů funkcí. Nechť je dána funkce f na množině $\mathbf{A}$ a nechť $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_s$ jsou neprázdné podmnožiny $\mathbf{A}$, pro něž

$$\mathbf{A}_1 \cup \mathbf{A}_2 \cup \ldots \cup \mathbf{A}_s = \mathbf{A}$$

Definujme funkci f_j na množině $\mathbf{A}_j$ pomocí funkce f takto: Položíme

$$f_j(x) = f(x) \quad \text{pro} \quad x \in \mathbf{A}_j{}^{*)}$$
$$(j = 1, 2, \ldots, s)$$

*) Říkáme, že funkce f_j vznikla z funkce f parcializací (zúžením) na množinu $\mathbf{A}_j$.

Bezprostředním použitím věty 3 dostáváme nyní následující tvrzení, jehož podrobný důkaz přenecháváme rovněž čtenáři.

Věta 8: Nechť existuje $\min\limits_{x \in A_j} f_j(x)$ [resp. $\max\limits_{x \in A_j} f_j(x)$] pro $j = 1, 2, \ldots, s$. Potom existuje rovněž $\min\limits_{x \in A} f(x)$ [resp. $\max\limits_{x \in A} f(x)$] a platí

$$\min_{x \in A} f(x) = \min \left\{ \min_{x \in A_j} f_j(x) \,|\, j = 1, 2, \ldots, s \right\}$$

$$\left(\text{resp. } \max_{x \in A} f(x) = \max \left\{ \max_{x \in A_j} f_j(x) \,|\, j = 1, 2, \ldots, s \right\}\right)$$

Pro ilustraci poslední věty uvedeme jednoduchý příklad.

Příklad 2: Chceme nalézt nejmenší číslo v tabulce 1.

5	4	—3	2	1
—1	3	0	—1	0
—4	5	10	—4	1

Tabulka 1

Tabulku lze přirozeným způsobem chápat jako funkci definovanou na množině všech jejích políček, tj. na množině všech uspořádaných dvojic (r, s), kde $r = 1, 2, 3$ (index řádku), a $s = 1, 2, \ldots, 5$ (index sloupce). Při řešení úlohy lze tabulku zkoumat „vcelku“, ale ve smyslu věty 8 lze též nalézt „částečná“ minima v jednotlivých řádcích (nebo sloupcích) a z těchto „částečných“ minim určit nejmenší číslo.

O extrémech funkcí platí dále tvrzení analogické větě 4, pomocí které je též lze snadno dokázat.

Věta 9: Nechť f je funkce definovaná na množině $\mathbf{A}$. Definujme funkci f_1 na $\mathbf{A}$ takto: $f_1(x) = -f(x)$ pro $x \in \mathbf{A}$.

a) Jestliže existuje $\min\limits_{x\in \mathbf{A}} f_1(x)$, potom existuje $\max\limits_{x\in \mathbf{A}} f(x)$ a platí $\max\limits_{x\in \mathbf{A}} f(x) = -\min\limits_{x\in \mathbf{A}} f_1(x)$.

b) Jestliže existuje $\max\limits_{x\in \mathbf{A}} f_1(x)$, potom existuje $\min\limits_{x\in \mathbf{A}} f(x)$ a platí $\min\limits_{x\in \mathbf{A}} f(x) = -\max\limits_{x\in \mathbf{A}} f_1(x)$.

Kapitolu ukončíme následující větou, jejíž snadný důkaz rovněž přenecháváme čtenáři.

Věta 10: Nechť f je funkce definovaná na neprázdné množině $\mathbf{A}$, a nechť c je dané číslo. Definujme funkci g na $\mathbf{A}$ takto:
Položíme $g(x) = f(x) + c$ pro všechna $x \in \mathbf{A}$.

a) Jestliže existuje $\min\limits_{x\in \mathbf{A}} f(x)$, existuje též $\min\limits_{x\in \mathbf{A}} g(x)$ a platí

$$\min_{x\in \mathbf{A}} g(x) = c + \min_{x\in \mathbf{A}} f(x) .$$

b) Jestliže existuje $\max\limits_{x\in \mathbf{A}} f(x)$, existuje též $\max\limits_{x\in \mathbf{A}} g(x)$ a platí

$$\max_{x\in \mathbf{A}} g(x) = c + \max_{x\in \mathbf{A}} f(x) .$$

Smysl věty ukážeme na následujícím příkladě.

Příklad 3: Položme ve větě 10 $\mathbf{A} = \{x \in \mathbf{R} | 0 \leqq x \leqq \leqq 1\}$, $f(x) = x^2 (x \in \mathbf{A})$ a $c = \sqrt{2}$. V souhlasu s větou 10 platí

$$\min_{x\in A}(x^2+\sqrt{2}) = \sqrt{2}+\min_{x\in A} x^2 ;$$

$$\max_{x\in A}(x^2+\sqrt{2}) = \sqrt{2}+\max_{x\in A} x^2$$

O správnosti posledních dvou vztahů se lze přesvědčit bezprostředně.

Cvičení

Cvičení 1: Ukažte, že množiny $\mathsf{M}_1, \mathsf{M}_2$ a M_4 zavedené na začátku kapitoly nemají ani minimum ani maximum, a množina M_3 má minimum, ale nemá maximum.

Cvičení 2: Dokažte, že každá neprázdná množina čísel má nejvýše jedno maximum.

Cvičení 3: Dokažte věty 3—10.

Cvičení 4: Znázorněte si na číselné ose nějakou množinu čísel a ukažte geometrický význam věty 4.

Cvičení 5: V pravoúhlé soustavě souřadnic si nakreslete graf nějaké reálné funkce reálné proměnné a pokuste se ukázat názorný geometrický význam vět 9 a 10.

Cvičení 6: Popište a zdůvodněte algoritmus pro určování maxima funkce definované na konečné neprázdné množině čísel, analogický algoritmu pro určování minima.

Cvičení 7: Pro neprázdné množiny čísel vyslovte a dokažte větu analogickou větě 10.

3. kapitola

METODA DYNAMICKÉHO PROGRAMOVÁNÍ

Myšlenku dynamického programování vysvětlíme na jedné extremální úloze, která se již tradičně považuje za dosti typický příklad použití této metody. Nechť jsou dány funkce $g_1, g_2, \ldots, g_n$, definované na množině $\{0, 1, \ldots, a\}$*), kde a je dané celé nezáporné číslo. Pomocí funkcí $g_1, \ldots, g_n$ utvoříme nyní novou funkci

$$y = g_1(x_1) + g_2(x_2) + \ldots + g_n(x_n) \tag{1}$$

definovanou na množině

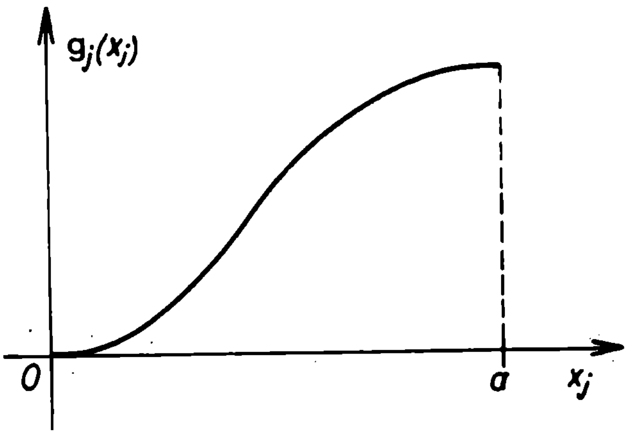
$$\left\{ (x_1, x_2, \ldots, x_n) \;\middle|\; \begin{array}{l} x_1, \ldots, x_n \text{ celá, nezáporná} \\ x_1 + \ldots + x_n = a \end{array} \right\} \tag{2}$$

[Jinými slovy: (2) je množina všech uspořádaných n-tic $(x_1, x_2, \ldots, x_n)$ celých nezáporných čísel, splňujících podmínku $x_1 + x_2 + \ldots + x_n = a$.]

V této kapitole se budeme zabývat úlohou nalezení maxima funkce (1) na množině (2). Všimněme si nejprve, že maximum funkce (1) na množině (2) skutečně existuje, jak vyplývá z věty 6. Metoda, kterou chceme použít pro skutečné řešení zformulované extremální úlohy, bude metoda dynamického programování.

*) Ve shodě s dříve zavedeným zápisem konečných neprázdných množin znamená „$\{0, 1, \ldots, a\}$“ množinu, jejímiž prvky jsou celá čísla $0, 1, \ldots, a$; v zápisu je zvoleno „přirozené“ pořadí čísel, podle velikosti.

Nejprve však ukážeme jedno použití zformulované extremální úlohy na *problém nejefektivnějšího rozdělení zdrojů*, vyskytující se v ekonomii. Předpokládejme, že v nějakém závodě pracuje celkem a dělníků a výrobní proces probíhá v n různých dílnách. Předpokládejme dále, že jednotlivé dílny vyrábějí produkci samostatně od začátku až do konce, tj. žádná není závislá na ostatních. Dále předpokládáme, že průměrný výtěžek z měsíční produkce j-té dílny, měřený v penězích, je známou funkcí $y = g_j(x_j)$ počtu x_j dělníků, kteří v dílně pracují. Vedení závodu chce rozdělit dělníky do jednotlivých dílen takovým způsobem, aby průměrný měsíční výtěžek z celé produkce závodu byl maximální. Je zřejmé, že zformulovaný problém vede na extremální úlohu (1), (2). V případě naší ekonomické aplikace budou mít grafy funkcí g_j pravděpodobně typický tvar, znázorněný schematicky na obr. 1.



Obr. 1

Tvar křivky odpovídá tomu, že při nedostatečném počtu pracovních sil výroba v dílně prakticky „stojí“ a při jejich nadbytku dochází k jistému efektu nasycení a v jeho důsledku k stagnaci dalšího růstu výroby (např. dělníků je více než techniky, navzájem si překážejí apod.). Poznamenejme však, že algoritmus, který nyní vyložíme, nevyžaduje splnění žádných speciálních předpokladů o funkcích g_j.

Použití metody dynamického programování záleží v tom, že se řeší celá množina extremálních problémů, do níž patří i problém (1), (2). Obecný problém této množiny je přiřazen uspořádané dvojici (j, x), kde $j \in \{1, 2, \ldots, n\}$ a $x \in \{0, 1, \ldots, a\}$, a záleží v určení maxima funkce j proměnných

$$y = g_1(x_1) + g_2(x_2) + \ldots + g_j(x_j)$$

na množině

$$\left\{(x_1, x_2, \ldots, x_j) \,\middle|\, \begin{array}{l} x_1, x_2, \ldots, x_j \text{ celá a nezáporná} \\ x_1 + x_2 + \ldots + \underset{j}{x_n} = x \end{array}\right\}$$

Tyto jednodušší extremální problémy se nyní řeší postupně pro $j = 1$; $x = 0, 1, \ldots, a$; $j = 2$; $x = 0, 1, \ldots, a$; $\ldots$ $j = n$; $x = 0, 1, \ldots, a$. Přitom přechod od j k $j + 1$ se provádí pomocí jistého rekurentního vztahu, jak ukazuje následující věta.

Věta 11: Pro $j = 1, 2, \ldots, n$ a $x = 0, 1, \ldots, a$ položme

$$f_j(x) = \max\left\{g_1(x_1) + \ldots + g_j(x_j) \,\middle|\, \begin{array}{l} x_1, \ldots, x_j \text{ celá a nezáp.} \\ x_1 + \ldots + x_j = x \end{array}\right\}$$

Potom platí

a) $f_1(x) = g_1(x)$ pro $x = 0, 1, \ldots, a$

b) $f_{j+1}(x) = \max \{f_j(x - x_{j+1}) + g_{j+1}(x_{j+1}) | x_{j+1} = 0, 1, \ldots, x\}$

pro $j = 1, 2, \ldots, n - 1$; $x = 0, 1, \ldots, a$

Důkaz: a) $f_1(x) = \max \{g_1(x_1) \mid x_1 = x\} = \max \{g_1(x)\} = g_1(x)$

b) Podle definice platí

$$f_{j+1}(x) = \max\left\{g_1(x_1) + \ldots + g_j(x_j) + g_{j+1}(x_{j+1}) \,\middle|\, \begin{array}{l} x_1, \ldots, x_j, x_{j+1} \\ \text{celá a nezáp.}; x_1 + \\ + \ldots + x_{j+1} = x \end{array}\right\}$$

tj. $f_{j+1}(x)$ je maximem funkce

$$y = g_1(x_1) + \ldots + g_j(x_j) + g_{j+1}(x_{j+1}) \tag{3}$$

na množině

$$\left\{(x_1, \ldots, x_j, x_{j+1}) \,\middle|\, \begin{array}{l} x_1, \ldots, x_j, x_{j+1} \text{ celá a nezáp.} \\ x_1 + \ldots + x_{j+1} = x \end{array}\right\} \tag{4}$$

Množinu (4) vyjádříme jako sjednocení $x + 1$ množin takto:

$$\left\{(x_1, \ldots, x_j, 0) \,\middle|\, \begin{array}{l} x_1, \ldots, x_j \text{ celá a nezáp.} \\ x_1 + \ldots + x_j = x - 0 \end{array}\right\} \cup$$

$$\cup \left\{(x_1, \ldots, x_j, 1) \,\middle|\, \begin{array}{l} x_1, \ldots, x_j \text{ celá a nezáp.} \\ x_1 + \ldots + x_j = x - 1 \end{array}\right\} \cup$$

$$\cup \left\{(x_1, \ldots, x_j, x) \,\middle|\, \begin{array}{l} x_1, \ldots, x_j \text{ celá a nezáp.} \\ x_1 + \ldots + x_j = x - x \end{array}\right\} =$$

$$\bigcup \left\{(x_1, \ldots, x_j, x_{j+1}) \,\middle|\, \begin{array}{l} x_1, \ldots, x_j \text{ celá a nezáp.} \\ x_1 + \ldots + x_j = x - x_{j+1} \end{array}\right\}$$

$x_{j+1} = 0, 1, \ldots, x$

Maximum funkce (3) na množině (4) určíme nyní pomocí věty 8 takto: Určíme maximální hodnoty funkcí definovaných předpisem (3) na množinách

$$\left\{(x_1, \ldots, x_j, x_{j+1}) \;\middle|\; \begin{matrix} x_1, \ldots, x_j \text{ celá a nezáp.} \\ x_1 + \ldots + x_j = x - x_{j+1} \end{matrix}\right\}$$

pro $x_{j+1} = 0, 1, \ldots, x$, a z takto získaných hodnot určíme maximální. Dostáváme tedy

$$\left.\begin{aligned} & \mathbf{f}_{j+1}(x) = \\ & = \max_{x_{j+1} \in \{0, \ldots, x\}} \max \Big\{ \mathbf{g}_1(x_1) + \ldots + \mathbf{g}_{j+1}(x_{j+1}) \;\Big|\; \\ & \qquad \begin{matrix} x_1, \ldots, x_j \text{ celá a nezáp.;} \\ x_1 + \ldots + x_j = x - x_{j+1} \end{matrix} \Big\} \end{aligned}\right\} \qquad (5)$$

Na druhé straně dostáváme pomocí věty 10

$$\max\left\{\mathbf{g}_1(x_1) + \ldots + \mathbf{g}_j(x_j) + \mathbf{g}_{j+1}(x_{j+1}) \;\middle|\; \begin{matrix} x_1, \ldots, x_j \text{ celá} \\ \text{a nezáp.; } x_1 + \\ + \ldots + x_j = \\ = x - x_{j+1} \end{matrix}\right\} =$$

$$= \mathbf{g}_{j+1}(x_{j+1}) + \max \Big\{ \mathbf{g}_1(x_1) + \ldots + \mathbf{g}_j(x_j) \;\Big|\; \begin{matrix} x_1, \ldots, x_j \text{ celá a nezáp.} \\ x_1 + \ldots + x_j = x - x_{j+1} \end{matrix} \Big\} =$$

$$= \mathbf{g}_{j+1}(x_{j+1}) + \mathbf{f}_j(x - x_{j+1}) \, .$$

Dosazením posledního výrazu do (5) dostáváme nakonec

$$\mathbf{f}_{j+1}(x) = \max_{x_{j+1} \in \{0, \ldots, x\}} \big(\mathbf{f}_j(x - x_{j+1}) + \mathbf{g}_{j+1}(x_{j+1}) \big) =$$

$$= \max \{ \mathbf{f}_j(x - x_{j+1}) + \mathbf{g}_{j+1}(x_{j+1}) \mid x_{j+1} = 0, 1, \ldots, x \}$$

Důkaz věty je dokončen.

Věta 11 umožňuje spočítat maximum funkce (1) na množině (2), neboť toto maximum je při našem označení rovno $\mathbf{f}_n(a)$; podrobně ukážeme za chvíli výpočetní postup na číselném příkladu. Otevřená však zatím zůstává otázka, jak nalézt n-tici $(x_1, x_2, \ldots, x_n)$ z množiny (2), pro kterou nabývá funkce (1) své maximální hodnoty (popř. nalézt všechny takové n-tice). Jinými slovy, jde o nalezení nějakého řešení (popř. všech řešení) extremálního problému (1), (2). Popíšeme nyní takový postup. Hledaná n-tice se bude konstruovat pomocí jistých funkcí $y = \mathbf{P}_1(x)$, $y = \mathbf{P}_2(x)$, $\ldots$, $y = \mathbf{P}_n(x)$ definovaných na množině $\{0, 1, \ldots, a\}$ takto:

Položíme $\mathbf{P}_1(x) = x$ pro $x = 0, 1, \ldots, a$ a dále pro $j = 1, 2, \ldots, n - 1$; $x = 0, 1, \ldots, a$ položíme $\mathbf{P}_{j+1}(x) = x^*_{j+1}$, kde x^*_{j+1} je libovolné číslo z množiny $\{0, 1, \ldots, x\}$, splňující vztah

$$\mathbf{f}_{j+1}(x) = \mathbf{f}_j(x - x^*_{j+1}) + \mathbf{g}_{j+1}(x^*_{j+1})$$

Existence alespoň jednoho takového čísla x_{j+1} vyplývá z faktu, že $\mathbf{f}_{j+1}(x)$ je maximem množiny

$$\{\mathbf{f}_j(x - x_{j+1}) + \mathbf{g}_{j+1}(x_{j+1}) \mid x_{j+1} = 0, 1, \ldots, x\}$$

Na druhé straně je nutné poznamenat, že ani x^*_{j+1} a tedy ani funkce $\mathbf{P}_{j+1}$ nejsou určeny obecně jednoznačně; poslední fakt souvisí úzce s tím, že funkce (1) může nabývat svého maxima ve více než v jedné n-tici množiny (2).

Funkce $\mathbf{P}_j$ se konstruují paralelně s výpočtem hodnot funkcí $\mathbf{f}_j (j = 1, 2, \ldots, n)$. Z následující věty vyplývá metoda nalezení řešení extremálního problému (1), (2) pomocí funkcí $\mathbf{P}_j$.

Věta 12: Položme $x_j^{(0)} = \mathbf{P}_j(x)$, $x_{j-1}^{(0)} = \mathbf{P}_{j-1}(x - x_j^{(0)})$

$x_{j-2}^{(0)} = \boldsymbol{P}_{j-2}(x - x_j^{(0)} - x_{j-1}^{(0)}), \quad \ldots, \quad x_1^{(0)} = \boldsymbol{P}_1(x - x_j^{(0)} - \ldots - x_2^{(0)})$

Potom platí

1. $x_1^{(0)}, \ldots, x_j^{(0)}$ jsou celá a nezáporná
2. $x_1^{(0)} + \ldots + x_j^{(0)} = x$
3. $\boldsymbol{g}_1(x_1^{(0)}) + \ldots + \boldsymbol{g}_j(x_j^{(0)}) = f_j(x)$

Důkaz: Tvrzení 1. a 2. jsou bezprostředním důsledkem definice funkcí $\boldsymbol{P}_1, \ldots, \boldsymbol{P}_n$. Tvrzení 3. se lehce dostane sečtením vztahů

$$\boldsymbol{f}_j(x) = \boldsymbol{f}_{j-1}(x - x_j^{(0)}) + \boldsymbol{g}_j(x_j^{(0)})$$

$$\boldsymbol{f}_{j-1}(x - x_j^{(0)}) = \boldsymbol{f}_{j-2}(x - x_j^{(0)} - x_{j-1}^{(0)}) + \boldsymbol{g}_{j-1}(x_{j-1}^{(0)})$$

$$\boldsymbol{f}_2(x - x_j^{(0)} - \ldots - x_3^{(0)}) = \boldsymbol{f}_1(x - x_j^{(0)} - \ldots - x_2^{(0)}) + \boldsymbol{g}_2(x_2^{(0)})$$

$$\boldsymbol{f}_1(x - x_j^{(0)} - \ldots - x_2^{(0)}) = \boldsymbol{g}_1(x_1^{(0)})$$

vyplývajících ze zavedení funkcí $\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_n$. Důkaz věty 12 je dokončen.

Z dokázané věty je patrné, že konstrukce řešení $(x_1^{(0)}, \ldots, x_n^{(0)})$ závisí na funkcích $\boldsymbol{P}_j$. Protože však funkce $\boldsymbol{P}_j$ lze obecně definovat různým způsobem, budeme metodou věty 12 dostávat různá řešení našeho problému. Lze však ukázat, že zkonstruujeme-li funkce $\boldsymbol{P}_j$ „všemi možnými způsoby“, dostaneme všechna řešení našeho extremálního problému (viz cvičení 6 na konci této kapitoly).

Použití vyložené teorie nyní ilustrujeme na následujícím číselném příkladu.

Příklad 4: Řešme extremální problém (1), (2) pro

x	g_1	g_2	g_3	g_4	g_5	g_6	g_7		f_1	P_1	f_2	P_2	f_3	P_3	f_4	P_4	f_5	P_5	f_6	P_6	f_7	P_7
0	1	−10	1	1	1	−12	0		1	0	−9	0	−8	0	−7	0	−6	0	−18	0	−18	0
1	−3	1	4	−2	0	−10	2		−3	1	2	1	3	0	4	0	5	0	−7	0	−7	0
2	5	1	4	2	1	0	1		5	2	2	2	6	1	7	0	8	0	−4	0	−4	0
3	7	2	2	1	0	1	4		7	3	6	1	7	0	8	0	9	0	5	2	5	0
4	8	0	1	−3	1	−5	−2		8	4	8	1	10	1	11	0	12	0	8	2	8	0
5	−2	5	1	4	2	0	1		−2	5	9	1	12	1	13	0	14	0	9	2	10	1
6	1	−2	1	0	1	4	0		1	6	9	2	13	1	14	0	15	0	12	2	12	0

Tabulka 2

$n = 7$, $a = 6$ a pro funkce $g_1, g_2, \ldots, g_7$, jejichž hodnoty jsou uvedeny v levé části tabulky 2. Hodnoty funkcí f_j a P_j, vypočtené na základě věty 11 a definice funkcí P_j, jsou zapsány v pravé části tabulky. Pravá část tabulky se zaplňuje po jednotlivých sloupcích, zleva doprava, přičemž sloupec pro f_1 vznikne opsáním sloupce pro g_1. Pro výpočet hodnot ve sloupci nadepsaném f_{j+1} se používá hodnot ležících ve sloupcích pro f_j a pro g_{j+1}. Ukažme si podrobně postup při výpočtu $f_6(3)$ a $P_6(3)$:

$$f_6(3) = \max \{f_5(3 - x_6) + g_6(x_6) \mid x_6 = 0, 1, 2, 3\} = \\ = \max (f_5(3) + g_6(0), f_5(2) + g_6(1), f_5(1) + g_6(2), f_5(0) + \\ + g_6(3)) = \max(-3, -2, 5, -5) = 5; P_6(3) = 2, \text{ neboť} \\ f_5(3 - 2) + g_6(2) = 5 = f_6(3)$$

Hledaná maximální hodnota funkce (1) na množině (2) je tedy $f_7(6) = 12$ a chceme ještě nalézt nějaké řešení našeho extremálního problému. K tomu použijeme větv 12:

$$\begin{aligned}
x_7^{(0)} &= P_7(a) = P_7(6) = 0 \\
x_6^{(0)} &= P_6(a - x_7^{(0)}) = P_6(6 - 0) = 2 \\
x_5^{(0)} &= P_5(a - x_7^{(0)} - x_6^{(0)}) = P_5(4) = 0 \\
x_4^{(0)} &= P_4(a - x_7^{(0)} - x_6^{(0)} - x_5^{(0)}) = P_4(4) = 0 \\
x_3^{(0)} &= P_3(a - x_7^{(0)} - \ldots - x_4^{(0)}) = P_3(4) = 1 \\
x_2^{(0)} &= P_2(a - x_7^{(0)} - \ldots - x_3^{(0)}) = P_2(3) = 1 \\
x_1^{(0)} &= P_1(a - x_7^{(0)} - \ldots - x_2^{(0)}) = P_1(2) = 2
\end{aligned}$$

(Řešení lze též hledat přímo pomocí tabulky, jak je ukázáno silnou čarou.) Dostáváme tedy řešení

$$(x_1^{(0)}, x_2^{(0)}, \ldots, x_7^{(0)}) = (2, 1, 1, 0, 0, 2, 0)$$

Můžeme ještě provést zkoušku správnosti dosazením; skutečně platí

$$g_1(2) + g_2(1) + g_3(1) + g_4(0) + g_5(0) + g_6(2) +$$

$$+ g_7(0) = 5 + 1 + 4 + 1 + 1 + 0 + 0 = 12 = \\ = f_7(6)\,.$$

Analogicky, jako jsme řešili problém určení maxima funkce (1) na množině (2), lze řešit i odpovídající minimalizační problém. Položme

$$h_j(x) = \min\left\{g_1(x_1) + \ldots + g_j(x_j) \;\middle|\; \begin{matrix} x_1, \ldots, x_j \text{ celá, nezáp.} \\ x_1 + \ldots + x_j = x \end{matrix}\right\}$$

pro $j = 1, 2, \ldots, n$; $x = 0, 1, \ldots, a$. Platí tvrzení analogické větě 11.

Věta 13: Platí

a) $h_1(x) = g_1(x)$

b) $h_{j+1}(x) = \min\,\{h_j(x - x_{j+1}) + g_{j+1}(x_{j+1}) \mid x_{j+1} = 0, 1, \ldots, x\}$ pro $x = 0, 1, \ldots, a$

Důkaz poslední věty, jakož i odvození metody pro konstrukci řešení minimalizačního problému přenecháváme čtenáři.

Cvičení

Cvičení 1: Dokažte větu 13 dvěma způsoby:

a) z věty 11 pomocí věty 9

b) samostatně, analogicky jako větu 11

Cvičení 2: Ukažte a odůvodněte metodu nalezení n-tice, která je řešením minimalizačního problému z věty 13.

Cvičení 3: Řešte minimalizační problém v situaci příkladu 4.

Cvičení 4: Nalezněte metodu pro určení

$$\max\left\{g_1(x_1) + \ldots + g_n(x_n) \;\middle|\; \begin{matrix} x_1, \ldots, x_n \text{ celá nezáp.} \\ a' \leqq x_1 + \ldots + x_n \leqq a'' \end{matrix}\right\}$$

kde a', a'' jsou daná celá, nezáporná čísla, $a' \leqq a''$.

Cvičení 5: Nalezněte maximum a minimum funkce

$$y = 10x_1 + x_2^2 + x_2 + 2^{x_3} - 5x_3 - 15 \sin\left(\frac{\pi}{2} x_4\right) + x_5^3 - 14x_5$$

na množině všech pětic $(x_1, x_2, \ldots, x_5)$ celých nezáporných čísel, splňujících podmínku

a) $x_1 + x_2 + \ldots + x_5 = 3$
b) $x_1 + x_2 + \ldots + x_5 = 7$
c) $2 \leqq x_1 + x_2 + \ldots + x_5 \leqq 6$

(Doporučení: Nejprve sestavte tabulku výrazů $10x_1$, $x_2^2 + x_2$, $2^{x_3} - 5x_3$, $15 \sin\left(\frac{\pi}{2} x_4\right)$, $x_5^3 - 14x_5$ pro $x_1, \ldots, x_5 = 0, 1, \ldots, 7$, analogickou levé části tabulky 2.)

Cvičení 6: Nechť $(x_1^*, x_2^*, \ldots, x_n^*)$ je libovolné řešení maximalizačního problému (1), (2). Dokažte, že funkce $P_1, P_2, \ldots, P_n$ lze zvolit tak, abychom metodou věty 12 dostali právě řešení $(x_1^*, x_2^*, \ldots, x_n^*)$. Jinými slovy: volíme-li funkce $P_1, P_2, \ldots, P_n$ ve větě 12 všemi možnými způsoby, dostaneme tak všechna řešení maximalizačního problému (1), (2).

4. kapitola

O EFEKTIVNOSTI DYNAMICKÉHO PROGRAMOVÁNÍ

Pojednáme nyní o efektivnosti algoritmu dynamického programování z předešlé kapitoly. Efektivností algoritmu zde budeme rozumět asi tolik, jako rychlost, se kterou algoritmus řeší náš problém. Pro jednoduchost se budeme zabývat pouze problémem určení maximální hodnoty

$$f^* = \max\left\{g_1(x_1) + \ldots + g_n(x_n) \,\middle|\, \begin{matrix} x_1, \ldots, x_n \text{ celá a nezáp.} \\ x_1 + \ldots + x_n = a \end{matrix}\right\}$$

a otázku nalezení řešení ponecháme stranou.

Algoritmus dynamického programování vyžaduje určení hodnot $f_j(x)$ pro $j = 1, 2, \ldots, n$ a $x = 0, 1, \ldots, a$, tj. celkem $n(a + 1)$ hodnot. Abychom získali představu o značné efektivnosti tohoto algoritmu, srovnejme jej s tak zvaným *triviálním algoritmem* pro řešení stejného problému. Triviálním algoritmem pro nalezení extrému funkce, definované na konečné neprázdné množině, budeme rozumět, v souhlasu s běžně užívanou terminologií, metodu, záležející v postupném „vyčíslení“ všech hodnot funkce (tj. pro všechny prvky jejího definičního oboru) a v určení extremální hodnoty z nich vzájemným srovnáním, tj. pomocí algoritmu popsaného v kapitole 2. Jinými slovy lze říci, že triviální algoritmus je zcela těžkopádný, mechanický postup určení extrému, nepoužívající žádné netriviální matematické myšlenky.

V našem případě bude pak triviální algoritmus záležet v tom, že se prozkoumají v nějakém libovolně zvoleném pořadí všechny n-tice $(x_1, x_2, \ldots, x_n)$ z množiny

$$\left\{(x_1, \ldots, x_n) \;\middle|\; \begin{array}{l} x_1, \ldots, x_n \text{ celá a nezáp.} \\ x_1 + \ldots + x_n = a \end{array}\right\} \tag{2}$$

pro každou z nich se určí součet $g_1(x_1) + \ldots + g_n(x_n)$ a ze všech takto určených součtů se nalezne maximální. K posouzení efektivnosti triviálního algoritmu určíme počet prvků množiny (2).

Lemma 1:*) Počet všech prvků množiny (2) je $\binom{n + a - 1}{n - 1}$**).

Důkaz: Přiřaďme každé n-tici $(x_1, x_2, \ldots, x_n)$ z množiny (2) $(n - 1)$-tici přirozených čísel $(y_1, y_2, \ldots, y_{n-1})$ takto:

$$y_1 = 1 + x_1, y_2 = 2 + x_1 + x_2, \ldots,$$
$$y_{n-1} = n - 1 + x_1 + \ldots + x_{n-1}$$

Zřejmě platí $1 \leqq y_1 < y_2 < \ldots < y_{n-1} \leqq n - 1 - a$.

Obráceně, každé $(n - 1)$-tici přirozených čísel $(y_1, y_2, \ldots, y_{n-1})$ splňujících nerovnosti

$$1 \leqq y_1 < y_2 < \ldots < y_{n-1} \leqq n - 1 + a$$

---

*) Slovo „Lemma" je starořeckého původu a označuje pomocnou větu.

**) Připomeňme si definici symbolu $\binom{m}{p}$ (čti „m nad p"):

$\binom{m}{p} = \dfrac{m(m - 1) \ldots (m - p + 1)}{1 . 2 . \ldots . p}$ pro všechny dvojice přirozených čísel m a p, kde $p \leqq m$

$\binom{m}{0} = 1$ pro všechna celá nezáporná m

odpovídá jediná n-tice $(x_1, x_2, \ldots, x_n)$ z množiny (2), splňující vztahy

$$y_1 = 1 + x_1,\ y_2 = 2 + x_1 + x_2,\ \ldots,$$
$$y_{n-1} = n - 1 + x_1 + \ldots + x_{n-1}$$

Odtud vyplývá, že počet prvků množiny (2) je roven počtu všech $(n - 1)$-tic $(y_1, y_2, \ldots, y_{n-1})$ popsaného tvaru. Avšak těchto $(n - 1)$-tic je právě tolik, kolik je všech $(n - 1)$-prvkových podmnožin množiny $\{1, 2, \ldots, n - 1 + a\}$, tj. $\binom{n - 1 + a}{n - 1}$, což dokončuje důkaz lemmatu.

Triviální algoritmus tedy vyžaduje k určení f^* vypočíst a srovnat $\binom{n - 1 + a}{n - 1}$ součtů $g_1(x_1) + \ldots + g_n(x_n)$, zatímco netriviální algoritmus dynamického programování potřebuje k dosažení stejného cíle určit tabulku obsahující $n(a + 1)$ čísel.

Ještě jasnější pohled na srovnání efektivnosti obou algoritmů získáme, určíme-li jimi požadovaný počet operací sčítání a srovnání. V případě triviálního algoritmu je to $(n - 1)\binom{n - 1 + a}{n - 1}$ sčítání a $\binom{n - 1 + a}{n - 1} - 1$ srovnání. Při výpočtu

$$f_{j+1}(x) = \max\{f_j(x - x_{j+1}) + g_{j+1}(x_{j+1}) \mid x_{j+1} = 0, \ldots, x\}$$

v netriviálním algoritmu potřebujeme $x + 1$ sčítání a x srovnání, takže výpočet všech hodnot $f_{j+1}(x)$ $(j = 1, \ldots, n - 1;\ x = 0, 1, \ldots, a)$ vyžaduje celkem

$$(n - 1)(1 + 2 + \ldots + (a + 1)) = \frac{(n - 1)(a + 1)(a + 2)}{2}$$

sčítání, a

$$(n-1)(0+1+\ldots+a) = \frac{(n-1)a(a+1)}{2} \text{srovnání}.$$

Výsledek provedeného vyšetřování shrneme v tabulce 3

algoritmus	počet sčítání	počet srovnání
netriviální	$\frac{(n-1)(a+1)(a+2)}{2}$	$\frac{(n-1)a(a+1)}{2}$
triviální	$(n-1)\binom{n-1+a}{n-1}$	$\binom{n-1+a}{n-1}-1$

Tabulka 3

Ještě konkrétnější představu o srovnání efektivnosti obou algoritmů získá čtenář, dosadí-li si za n a a několik číselných hodnot, viz např. cvičení 1.

Cvičení

Cvičení 1: Porovnejte počet operací sčítání a srovnání v obou algoritmech pro tyto hodnoty a, n:

a) $a = 5, \quad n = 5$ c) $a = 10, \quad n = 10$
b) $a = 7, \quad n = 7$ d) $a = 10, \quad n = 15$

Cvičení 2: Určete počet všech uspořádaných n-tic $(x_1, \ldots, x_n)$ celých nezáporných čísel, splňujících vztah $x_1 + \ldots + x_n \leqq a$, kde a je dané celé číslo.

Cvičení 3: Pokuste se řešit cvičení 5a) a 5c) z kapitoly 3 triviálním algoritmem.

5. kapitola

CELOČÍSELNÉ INTERVALY S MINIMÁLNÍM OHODNOCENÍM

Nechť je dána posloupnost n reálných čísel $a_1, a_2, \ldots, a_n$. Uvažujme všechny možné součty tvaru

$$a_i + a_{i+1} + \ldots + a_j \tag{6}$$

kde $i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, n$; $i \leqq j$. (Výraz (6) představuje skutečně součet $j - i + 1$ ($\geqq 2$) čísel v případě $j > i$; v případě $i = j$ je jeho hodnota a_i.)

Budeme se zabývat tímto extremálním problémem: Máme nalézt nejmenší ze součtů (6), popř. jeden z nejmenších, má-li problém více než jedno řešení.

Výrazem (6) je definována funkce, jejímž definičním oborem je množina všech uspořádaných dvojic (i, j), kde $i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, n$; $i \leqq j$, a která každé takové dvojici přiřazuje číslo $a_i + \ldots + a_j$. Tuto funkci budeme pro stručnost nazývat funkcí (6). Náš extremální problém je tedy problémem určení minima funkce (6).

Ve speciálním případě, že všechna čísla $a_1, a_2, \ldots, a_n$ jsou nezáporná, je minimální hodnotou funkce (6) zřejmě $\min \{a_j \mid j = 1, 2, \ldots, n\}$. V obecném případě je však problém obtížnější. K jeho řešení lze užít následujícího triviálního algoritmu (srov. kapitolu 4): Určí se všechny součty (6) (tj. celkem $\binom{n}{2} + n = \frac{n(n+1)}{2}$ čísel) a z množiny těchto čísel se nalezne nejmenší pomocí algoritmu z kapitoly 2. Popsaný triviální algoritmus musí tedy určit a prozkoumat celkem $\frac{n(n+1)}{2}$ čísel.

Vyložíme nyní algoritmus dynamického programování pro řešení našeho problému, který vystačí s určením a prozkoumáním pouhých n čísel. Označme symbolem f^* hodnotu nejmenšího ze všech součtů (6) a symbolem f_j ($j = 1, 2, \ldots, n$) hodnotu nejmenšího ze součtů (6) při pevném j a pro $i = 1, 2, \ldots, j$. Algoritmus je založen na rekurentním výpočtu čísel $f_1, f_2, \ldots, f_n, f^*$, vyplývajícím z další věty.

Věta 14: Čísla $f_1, f_2, \ldots, f_n$ a f^* splňují tyto vztahy

a) $f_1 = a_1$

b) $f_{j+1} = a_{j+1} + \min(0, f_j)$

c) $f^* = \min\{f_j \mid j = 1, 2, \ldots, n\}$

Důkaz: Část a) je zřejmá.

b)

$$
\begin{aligned}
f_{j+1} &= \min\{a_i + \ldots + a_j + a_{j+1} \mid i = 1, 2, \ldots, j+1\} = \\
&= \min(\min\{a_i + \ldots + a_j + a_{j+1} \mid i = j+1\}, \\
&\qquad \min\{a_i + \ldots + a_j + a_{j+1} \mid i = 1, 2, \ldots, j\}) = \\
&= \min(a_{j+1}, \min\{a_i + \ldots + a_j \mid i = 1, 2, \ldots, j\} + a_{j+1}) = \\
&= \min(a_{j+1} + 0, a_{j+1} + f_j) = a_{j+1} + \min(0, f_j)
\end{aligned}
$$

c)

$$
\begin{aligned}
f^* &= \min\{a_i + \ldots + a_j \mid i, j \text{ celá}, 1 \leqq i \leqq j \leqq n\} = \\
&= \min\{\min\{a_i + \ldots + a_j \mid i = 1, \ldots, j\} \mid j = 1, \ldots, n\} = \\
&= \min\{f_j \mid j = 1, 2, \ldots, n\}
\end{aligned}
$$

Důkaz věty 14 je dokončen.

Pro další výklad zavedeme ještě jinou, názornější interpretaci součtů (6). Množiny tvaru $\{i, i + 1, \ldots, j\}$, kde $1 \leqq i \leqq j \leqq n$, budeme nazývat *celočíselnými intervaly* (množiny $\{1, 2, \ldots, n\}$). Každý celočíselný interval je tedy množinou všech celých čísel x, splňujících nerovnosti $i \leqq x \leqq j$ (všimněte si analogie s „obyčejným" intervalem, známým ze střední školy). V dalším textu budeme pro „celočíselný interval" používat zkratky CI. Součet $a_i + a_{i+1} + \ldots + a_j$ nazveme dále *ohodnocením* CI $\{i, i + 1, \ldots, j\}$. Každému CI je tedy přiřazeno jisté reálné číslo — jeho ohodnocení. Náš problém záleží v určení CI s minimálním ohodnocením. Ve větě 14 jsme již ukázali metodu pro nalezení nejmenšího ohodnocení; v další větě je popsán způsob nalezení odpovídajícího CI.

Věta 15: Nechť $j_0 \in \{1, 2, \ldots, n\}$ je přirozené číslo, pro které platí $f_{j_0} = f^*$.

a) Jestliže $f^* \geqq 0$, potom je $\{j_0\}$ CI s minimálním ohodnocením.

b) Jestliže $f^* < 0$, zvolme za i_0 nejmenší přirozené číslo z množiny $\{1, 2, \ldots, j_0\}$, pro které platí $f_{i_0} < 0$, $f_{i_0+1} < < 0, \ldots, f_{j_0-1} < 0$, $f_{j_0} < 0$. Potom je $\{i_0, i_0 + 1, \ldots, j_0\}$ CI s nejmenším ohodnocením.

Důkaz: a) V případě $j_0 = 1$ platí $f^* = f_1 = a_1$. V případě $j_0 > 1$ dostáváme $f^* = f_{j_0} = a_{j_0} + \min(0, f_{j_0-1})$, kde $f_{j_0-1} \geqq f_{j_0} \geqq 0$. Odtud vyplývá

$$\min(0, f_{j_0-1}) = 0 \text{ a tedy } f^* = f_{j_0} = a_{j_0}$$

V obou případech je tedy $\{j_0\}$ CI s minimálním ohodnocením.

b) Podle předpokladu platí $f^* = f_{j_0} < 0$, $f_{j_0-1} < 0, \ldots, f_{i_0+1} < 0$, $f_{i_0} < 0$, ale $f_{i_0-1} \geqq 0$, nebo $i_0 = 1$. Odtud dostáváme na základě věty 14:

$f_{j_0} = a_{j_0} + f_{j_0-1}, f_{j_0-1} = a_{j_0-1} + f_{j_0-2}, \ldots, f_{i_0+1} = a_{i_0+1} +$
$$+ f_{i_0}, f_{i_0} = a_{i_0}.$$
Z posledního řetězce vztahů vyplývá
$$f^* = f_{j_0} = a_{i_0} + a_{i_0+1} + \ldots + a_{j_0};$$
je tedy $\{i_0, i_0 + 1, \ldots, j_0\}$ CI s minimálním ohodnocením a důkaz věty 15 je dokončen.

Poznámka: Čtenář snadno sám ukáže, že v případě a) platí $a_j \geqq 0$ pro $j = 1, 2, \ldots, n$, zatímco v případě b) je alespoň jedno z čísel $a_1, a_2, \ldots, a_n$ záporné.

Použití vyložené metody dynamického programování ilustrujeme na následujícím numerickém příkladu.

Příklad 5: Máme nalézt minimum funkce (6) pro $n = 10$ a pro čísla a_j uvedená ve druhém řádku následující tabulky.

j	1	2	3	4	5	6	7	8	9	10
a_j	—1	0	1	—2	3	—4	0	1	—2	1
f_j	—1	—1	0	—2	1	—4	—4	—3	—5	—4

Tabulka 4

Čísla f_j vypočtená podle věty 14 zapisujeme do třetího řádku tabulky 4. Dále dostáváme
$$f^* = \min \{f_j \mid j = 1, 2, \ldots, 10\} = f_9 = -5$$
Dále z tabulky vidíme, že platí $f_9 < 0$, $f_8 < 0$, $f_7 < 0$, $f_6 < 0$, ale $f_5 \geqq 0$, takže $\{6, 7, 8, 9\}$ je CI s minimálním ohodnocením. Pro zkoušku správnosti výpočtu se můžeme přesvědčit, že skutečně platí $a_6 + a_7 + a_8 + a_9 = -5$.

Cvičení

Cvičení 1: Popište a odůvodněte netriviální metodu pro nalezení maxima funkce (6), analogickou metodě pro nalezení minima.

Cvičení 2: V podmínkách příkladu 5 řešte úlohu na maximum.

Cvičení 3: Extremální problém z příkladu 5 řešte pomocí triviálního algoritmu.

Cvičení 4: Určete počet operací sčítání a srovnání pro určení minima funkce (5), vyžadovaný

a) triviálním algoritmem

b) algoritmem dynamického programování

Do získaných vzorců dosaďte $n = 10, 100, 1000$.

Cvičení 5: Nechť I' a I'' jsou dva CI množiny $\{1, 2, \ldots, n\}$. I' a I'' nazveme *(navzájem) styčné*, jestliže neobsahují společný prvek a $I' \cup I''$ je rovněž CI. Nechť dále jsou dána čísla $a_1, a_2, \ldots, a_n$. Definujme funkci $\boldsymbol{a}$ na množině všech CI množiny $\{1, 2, \ldots, n\}$ tímto předpisem:

$\boldsymbol{a}(I) = a_i + a_{i+1} + \ldots + a_j$, jestliže $I = \{i, i + 1, \ldots, j\}$.

Dokažte, že pro libovolné dva styčné CI I' a I'' množiny $\{1, 2, \ldots, n\}$ platí

$$\boldsymbol{a}(I' \cup I'') = \boldsymbol{a}(I') + \boldsymbol{a}(I'')$$

Cvičení 6: Dokažte následující tvrzení: Nechť $\boldsymbol{a}^*$ je funkce definovaná na množině všech CI množiny $\{1, 2, \ldots, n\}$ a vyhovující podmínce: Pro libovolné dva styčné CI (viz předcházející cvičení) I' a I'' platí

$$\boldsymbol{a}^*(I' \cup I'') = \boldsymbol{a}^*(I') + \boldsymbol{a}^*(I'')$$

Potom existuje jediná n-tice reálných čísel $a_1^*, a_2^*, \ldots, a_n^*$ tak, že platí

$$\boldsymbol{a}^*(I) = a_i^* + a_{i+1}^* + \ldots + a_j^* \text{ pro } I = \{i, i + 1, \ldots, j\}$$

6. kapitola

ÚLOHA O NEJCENNĚJŠÍM NÁKLADU LODI

Obecný extremální problém, kterým se v tomto odstavci chceme zabývat, může být motivován touto praktickou úlohou:

Obchodní loď s nosností b tun se vydává na cestu z jednoho přístavu do druhého a vzniká otázka o vhodném složení nákladu. Přitom je k dispozici n různých zboží*), přičemž váha j-tého zboží ($j = 1, 2, \ldots, n$) je a_j tun a jeho cena je c_j jednotek měny. Nyní chceme sestavit takové složení nákladu, aby nebyla překročena nosnost lodi a přitom aby celková cena nákladu byla maximální. Hledané složení nákladu popíšeme uspořádanou n-ticí $(x_1, x_2, \ldots, x_n)$, kde položíme $x_j = 1$, jestliže j-té zboží je naloženo a $x_j = 0$ v opačném případě. Při tomto označení je celková cena nákladu $c_1x_1 + c_2x_2 + \ldots + c_nx_n$ jednotek měny, jeho váha je $a_1x_1 + a_2x_2 + \ldots + a_nx_n$ tun, a přicházíme tedy k následujícímu extremálnímu problému: máme nalézt maximum funkce

$$y = c_1x_1 + c_2x_2 + \ldots + c_nx_n \tag{7}$$

na množině

$$\left\{(x_1, x_2, \ldots, x_n) \,\middle|\, \begin{array}{l} x_j \in \{0, 1\}\ (j = 1, \ldots, n) \\ a_1x_1 + a_2x_2 + \ldots + a_nx_n \leqq b \end{array}\right\} \tag{8}$$

*) Předpokládáme, že každé zboží je nedělitelné, tj. v případě, že se nakládá, musí se naložit celé, nikoliv pouze jeho část.

Vyložíme algoritmus dynamického programování pro řešení zformulovaného extremálního problému (7), (8) za předpokladu, že c_j jsou libovolná reálná čísla, a_j jsou přirozená čísla a b je celé nezáporné.

Poznámka: Číselné údaje vyskytující se v praxi, mají obvykle tvar racionálních čísel (dokonce čísel s konečným dekadickým rozvojem). Je tedy rozumné předpokládat, že v případě úlohy o nejcennějším nákladu jsou čísla a_j, b a c_j racionální, kladná; přechodem k menším jednotkám lze pak dosáhnout toho, že jsou to čísla přirozená.

Algoritmus pro řešení maximalizačního problému (7), (8) se zakládá na rekurentním výpočtu jistých hodnot $g_j(x)$ $(j = 1, 2, \ldots, n;\ x = 0, 1, \ldots, b)$, zavedených takto:

1. $g_j(x)$ není definováno, jestliže

$$\left\{(x_1, \ldots, x_j) \;\middle|\; \begin{array}{l} x_i \in \{0, 1\}\ (i = 1, \ldots, j) \\ a_1x_1 + \ldots + a_jx_j = x \end{array}\right\} = \emptyset$$

2. $g_j(x)$ je definováno, jestliže

$$\left\{(x_1, \ldots, x_j) \;\middle|\; \begin{array}{l} x_j \in \{0, 1\}\ (i = 1, \ldots, j) \\ a_1x_1 + \ldots + a_jx_j = x \end{array}\right\} \neq \emptyset\,;$$

v posledním případě položíme

$$g_j(x) = \max\left\{c_1x_1 + \ldots + c_jx_j \;\middle|\; \begin{array}{l} x_i \in \{0, 1\}\ (i = 1, \ldots, j) \\ a_1x_1 + \ldots + a_jx_j = x \end{array}\right\}$$

Hodnoty $g_j(x)$ jsou tedy zavedeny analogicky jako hodnoty $f_j(x)$ v kapitole 3, avšak hlavní rozdíl, určený specifikou tohoto extremálního problému je v tom, že $g_j(x)$ nejsou obecně definovány pro všechny dvojice (j, x), kde $j = 1, \ldots, n$ a $x = 0, 1, \ldots, b$. Při pevném j odpovídá definovaným hodnotám $g_j(x)$ jistá funkce

$y = g_j(x)$, definovaná na nějaké podmnožině množiny $\{0, 1, \ldots, b\}$ (ve speciálním případě však může být definičním oborem celá množina $\{0, 1, \ldots, b\}$). Z důvodu stručnosti zápisu budeme dále místo „$g_j(x)$ není definováno“ psát pouze „$g_j(x)$ nedef“ a místo „$g_j(x)$ je definováno“ pouze „$g_j(x)$ def“.

Dále položíme

$$g^* = \max \left\{ c_1 x_1 + \ldots + c_n . x_n \,\middle|\, \begin{array}{l} x_i \in \{0, 1\}\ (i = 1, \ldots, n) \\ a_1 x_1 + \ldots + a_n x_n \leqq b \end{array} \right\}$$

Maximum na pravé straně posledního vztahu skutečně existuje, neboť množina

$$\left\{ (x_1, \ldots, x_n) \,\middle|\, \begin{array}{l} x_i \in \{0, 1\}\ (i = 1, \ldots, n) \\ a_1 x_1 + \ldots + a_n x_n \leqq b \end{array} \right\}$$

je za předpokladů o číslech a_j a b, uvedených na začátku kapitoly, neprázdná. Algoritmus pro řešení problému (7), (8) je založen na rekurentním výpočtu hodnot $g_j(x)$ a g^*, jak ukazuje následující věta.

Věta 16:

a) Platí $g_1(0) = 0$, $g_1(a_1) = c_1$ a $g_1(x)$ nedef pro všechna ostatní x.

b) Položme

$$\bar{g}_j(x) \begin{cases} \text{nedef, jestliže } g_j(x - a_{j+1}) \text{ nedef} \\ = g_j(x - a_{j+1}) + c_{j+1}, \text{ jestliže } g_j(x - a_{j+1}) \text{ def} \end{cases}$$

pro $j = 1, 2, \ldots, n$; $x = 0, 1, \ldots, b$.

Potom platí

$$g_{j+1}(x) \begin{cases} \text{nedef, jestliže } g_j(x) \text{ nedef a } \bar{g}_j(x) \text{ nedef} \\ = g_j(x), \text{ jestliže } g_j(x) \text{ def a } \bar{g}_j(x) \text{ nedef} \\ = \bar{g}_j(x), \text{ jestliže } g_j(x) \text{ nedef a } \bar{g}_j(x) \text{ def} \\ = \max\, (g_j(x), \bar{g}_j(x)), \text{ jestliže } g_j(x) \text{ def a } \bar{g}_j(x) \text{ def} \end{cases}$$

$(j = 1, 2, \ldots, n-1;\ x = 0, 1, \ldots, b)$

c) Platí $g^* = \max \{\mathbf{g}_n(x) | x = 0, \ldots, b;\ \mathbf{g}_n(x) \text{ def}\}$.

Důkaz: a) $\mathbf{g}_1(0) = \max \left\{c_1 x_1 \middle| x_1 \in \{0, 1\};\ a_1 x_1 = 0\right\} = \max \{c_1\, 0\} = 0$;

$\mathbf{g}_1(a_1) = \max \{c_1 x_1 | x_1 \in \{0, 1\};\ a_1 x_1 = a_1\} = \max \{c_1\, 1\} = c_1$;

$0 < x \neq a_1 \Rightarrow \left\{(x_1) \middle| x_1 \in \{0, 1\};\ a_1 x_1 = x\right\} = \emptyset \Rightarrow \mathbf{g}_1(x)$ nedef

b) 1. $\mathbf{g}_j(x)$ nedef a $\bar{\mathbf{g}}_j(x)$ nedef, tj. $\mathbf{g}_j(x - a_{j+1})$ nedef.

Odtud vyplývá, že rovnice

$a_1 x_1 + \ldots + a_j x_j = x$ a $a_1 x_1 + \ldots + a_j x_j = x - a_{j+1}$ nemají žádné $\{0, 1\}$ — řešení $(x_1, \ldots, x_j)$*), a tedy ani rovnice

$$a_1 x_1 + \ldots + a_j x_j + a_{j+1} x_{j+1} = x$$

nemá žádné $\{0, 1\}$ — řešení $(x_1, \ldots, x_j, x_{j+1})$. Tudíž platí $\mathbf{g}_{j+1}(x)$ nedef, což jsme měli ukázat.

2. $\mathbf{g}_j(x)$ def a $\bar{\mathbf{g}}_j(x)$ nedef, tj. $\mathbf{g}_j(x - a_{j+1})$ nedef. Potom rovnice $a_1 x_1 + \ldots + a_j x_j = x$ má $\{0, 1\}$ — řešení $(x_1, \ldots, x_j)$, ale rovnice $a_1 x_1 + \ldots + a_j x_j = x - a_{j+1}$ žádné $\{0, 1\}$ — řešení $(x_1, \ldots, x_j)$ nemá. Odtud vyplývá, že množina

$$\left\{(x_1, \ldots, x_j, x_{j+1}) \middle| \begin{array}{l} x_i \in \{0, 1\}\ (i = 1, \ldots, j, j+1) \\ a_1 x_1 + \ldots + a_j x_j + a_{j+1} x_{j+1} = x \end{array}\right\}$$

je neprázdná, avšak pro každou $(j + 1)$-tici $(x_1, \ldots, x_j, x_{j+1})$ z této množiny platí $x_{j+1} = 0$. Dostáváme tedy

$\mathbf{g}_{j+1}(x) =$

*) Pro stručnost vyjadřujeme obratem „$\{0, 1\}$ — řešení $(x_1, \ldots, x_j)$“ okolnost, že $x_1 \in \{0, 1\}, \ldots, x_j \in \{0, 1\}$.

$$= \max \left\{ c_1 x_1 + \ldots + c_j x_j + c_{j+1} x_{j+1} \;\middle|\; \begin{array}{l} x_i \in \{0, 1\} \, (i = \\ = 1, \ldots, j+1) \\ a_1 x_1 + \ldots + a_j x_j + \\ + a_{j+1} x_{j+1} = x \end{array} \right\} =$$

$$= \max \left\{ c_1 x_1 + \ldots + c_j x_j + c_{j+1} \, 0 \;\middle|\; \begin{array}{l} x_i \in \{0, 1\} \, (i = \\ = 1, \ldots, j) \\ a_1 x_1 + \ldots + \\ + a_j x_j + a_{j+1} \, 0 = x \end{array} \right\} =$$

$= \boldsymbol{g}_j(x)$, což jsem měli ukázat.

3. $\boldsymbol{g}_j(x)$ nedef a $\bar{\boldsymbol{g}}_j(x)$ def, tj. $\boldsymbol{g}_j(x - a_{j+1})$ def. Zcela analogicky jako v případě 2. zde dostaneme, že $\boldsymbol{g}_{j+1}(x)$ def a $\boldsymbol{g}_{j+1}(x) =$

$$= \max \left\{ c_1 x_1 + \ldots + c_j x_j + c_{j+1} x_{j+1} \;\middle|\; \begin{array}{l} x_i \in \{0, 1\} \, (i = \\ = 1, \ldots, j+1) \\ a_1 x_1 + \ldots + \\ + a_j x_j + a_{j+1} x_{j+1} = x \end{array} \right\} =$$

$$= \max \left\{ c_1 x_1 + \ldots + c_j x_j + c_{j+1} \, 1 \;\middle|\; \begin{array}{l} x_i \in \{0, 1\} \, (i = \\ = 1, \ldots, j) \\ a_1 x_1 + \ldots + \\ a_j x_j = x - a_{j+1} \end{array} \right\}$$

Odtud vyplývá na základě věty 10 z druhé kapitoly a definice hodnot $\bar{\boldsymbol{g}}_j(x)$: $\boldsymbol{g}_{j+1}(x) = c_{j+1} + \boldsymbol{g}_j(x - a_{j+1}) = \bar{\boldsymbol{g}}_j(x)$, což jsme měli ukázat.

4. $\boldsymbol{g}_j(x)$ def a $\bar{\boldsymbol{g}}_j(x)$ def, tj. $\boldsymbol{g}_j(x - a_{j+1})$ def. Potom platí

$$\left\{ (x_1, \ldots, x_j, x_{j+1}) \;\middle|\; \begin{array}{l} x_i \in \{0, 1\} \, (i = 1, \ldots, j, j+1) \\ a_1 x_1 + \ldots + a_j x_j + a_{j+1} x_{j+1} = x \end{array} \right\} \neq 0$$

odkud vyplývá $\boldsymbol{g}_{j+1}(x)$ def. Vyjádřeme dále poslední množinu ve tvaru sjednocení dvou množin

$$\left\{(x_1, \ldots, x_j, 0) \middle| \begin{array}{l} x_i \in \{0, 1\}\ (i = 1, \ldots, j) \\ a_1x_1 + \ldots + a_jx_j = x \end{array}\right\} \cup$$

$$\cup \left\{(x_1, \ldots, x_j, 1) \middle| \begin{array}{l} x_i \in \{0, 1\}\ (i = 1, \ldots, j) \\ a_1x_1 + \ldots + a_jx_j = x - a_{j+1} \end{array}\right\}$$

a všimněme si, že obě dvě množiny jsou neprázdné. Odtud dostáváme na základě vět 8 a 10 (kapitola 2) a definice symbolu $\bar{g}_j(x)$:

$$g_{j+1}(x) =$$

$$= \max\left\{c_1x_1 + \ldots + c_jx_j + c_{j+1}x_{j+1} \middle| \begin{array}{l} x_i \in \{0, 1\} \\ (i = 1, \ldots, j + 1) \\ a_1x_1 + \ldots + \\ + a_{j+1}x_{j+1} = x \end{array}\right\} =$$

$$= \max\left(\max\left\{c_1x_1 + \ldots + c_jx_j \middle| \begin{array}{l} x_i \in \{0, 1\}\ (i = 1, \ldots, j) \\ a_1x_1 + \ldots + a_jx_j = x \end{array}\right\}\right.$$

$$\left. c_{j+1} + \max\left\{c_1x_1 + \ldots + c_jx_j \middle| \begin{array}{l} x_i \in \{0, 1\}\ (i = 1, \ldots, j) \\ a_1x_1 + \ldots + a_jx_j = \\ = x - a_{j+1} \end{array}\right\}\right) =$$

$$= \max\,(g_j(x), \bar{g}_j(x)).$$

c) Na základě věty 8 dostáváme $g^* =$

$$= \max\left\{c_1x_1 + \ldots + c_nx_n \middle| \begin{array}{l} x_i \in \{0, 1\}\ (i = 1, \ldots, n) \\ a_1x_1 + \ldots + a_nx_n \leqq b \end{array}\right\} =$$

$$= \max\left\{\max\left\{\begin{array}{l} c_1x_1 + \\ + \ldots + c_nx_n \end{array} \middle| \begin{array}{l} x_i \in \{0, 1\}\ (i = 1, \ldots, n) \\ a_1x_1 + \ldots + a_nx_n = x \end{array}\right\} \middle| \begin{array}{l} x = 0, \ldots, b; \\ a_1x_1 + \ldots + a_nx_n = x \\ \text{má } \{0,\ 1\} \text{— řešení} \end{array}\right\} =$$

$= \max \{g_n(x) | x = 0, 1, \ldots, b;\ g_n(x) \text{ def}\}$

Důkaz věty 16 je dokončen.

Obsah části b) věty 16 přehledně vyjádříme pomocí tabulky 5.

$g_j(x)$	$\bar{g}_j(x)$	$g_{j+1}(x)$
nedef	nedef	nedef
def	nedef	$g_j(x)$
nedef	def	$\bar{g}_j(x)$
def	def	$\max\,[g_j(x), \bar{g}_j(x)]$

Tabulka 5

K úplnému výkladu metody zbývá popsat způsob nalezení n-tice $(x_1, x_2, \ldots, x_n)$, která je řešením našeho extremálního problému. K dosažení tohoto cíle určujeme současně s rekurentním výpočtem hodnot $g_j(x)$ jisté hodnoty $Q_j(x)$ ($x \in \{0, \ldots, b\}$; $j \in \{1, 2, \ldots, n\}$), zavedené následujícím způsobem:

(i) $Q_1(a_1) = 1$, $Q_1(0) = 0$ a $Q_1(x)$ nedef, jestliže $0 < x \neq a_1$.

(ii) Pro $j = 1, 2, \ldots, n - 1$ položíme

$$Q_{j+1}(x) \begin{cases} \text{nedef}, \ldots & \text{jestliže } g_{j+1}(x) \text{ nedef} \\ = 0, \ldots & \text{jestliže } g_j(x) \text{ def a } \bar{g}_j(x) \text{ nedef; nebo } \bar{g}_j(x) < g_j(x) \\ = 1, \ldots & \text{jestliže } \bar{g}_j(x) \text{ def a } g_j(x) \text{ nedef; nebo } g_j(x) < \bar{g}_j(x) \\ = 0 \text{ nebo } 1 & \text{(libovolně) ve zbývajícím případě} \end{cases}$$

Poznámka: Nejednoznačnost při zavedení $Q_{j+1}(x)$

v posledním případě opět úzce souvisí s faktem, že náš extremální problém nemá v obecném případě jediné řešení.

Důkaz následujícího pomocného tvrzení přenecháváme čtenáři.

Lemma 2: a) Jestliže $g_1(x)$ def, potom $Q_1(x)$ def, $g_1(x) = c_1 Q_1(x)$ a $x = a_1 Q_1(x)$.
b) Jestliže $g_{j+1}(x)$ def, potom $Q_{j+1}(x)$ def a

$$g_{j+1}(x) = g_j[x - a_{j+1} Q_{j+1}(x)] + c_{j+1} Q_{j+1}(x) .$$

Nalezení n-tice, která je řešením maximalizačního problému (7), (8), se provádí pomocí následující věty.

Věta 17: a) Položme $x_j^{(0)} = Q_j(x)$, $x_{j-1}^{(0)} = Q_{j-1}(x - a_j x_j^{(0)})$, $x_{j-2}^{(0)} = Q_{j-2}(x - a_j x_j^{(0)} - a_{j-1} x_{j-1}^{(0)})$, ...

$$\ldots, x_1^{(0)} = Q_1(x - a_j x_j^{(0)} - a_{j-1} x_{j-1}^{(0)} - \ldots - a_2 . x_2^{(0)})$$

Potom platí:

1. $x_i^{(0)} \in \{0, 1\}$ $(i = 1, 2, \ldots, j)$
2. $a_1 x_1^{(0)} + \ldots + a_j x_j^{(0)} = x$
3. $c_1 x_1^{(0)} + \ldots + c_j x_j^{(0)} = g_j(x)$

b) Zvolme číslo $x^* \in \{0, 1, \ldots, b\}$ tak, aby platilo $g_n(x^*) = g^*$ a určeme n-tici $(x_1^*, x_2^*, \ldots, x_n^*)$ ve smyslu části a). Tato n-tice je řešením maximalizačního problému (7), (8).

Důkaz: Část 1. je zřejmá z definice $Q_j(x)$. Na základě lemmatu 2 platí $x - a_j x_j^{(0)} - \ldots - a_2 x_2^{(0)} = a_1 x_1^{(0)}$, odkud vyplývá platnost 2. Zbývá dokázat část 3. Na základě lemmatu 2 dostáváme řetězec vztahů

$$g_j(x) \qquad = g_{j-1}(x - a_j x_j^{(0)}) + c_j x_j^{(0)}$$

$$g_{j-1}(x - a_j x_j^{(0)}) = g_{j-2}(x - a_j x_j^{(0)} - a_{j-1} x_{j-1}^{(0)}) + c_{j-1} x_{j-1}^{(0)}$$

$$g_2(x - a_j x_j^{(0)} - \ldots - a_3 x_3^{(0)}) = g_1(x - a_j x_j^{(0)} - \ldots -$$
$$- a_2 x_2^{(0)}) + c_2 x_2^{(0)}$$
$$g_1(x - a_j x_j^{(0)} - \ldots - a_2 x_2^{(0)}) = c_1 x_1^{(0)}$$

odkud vyplývá postupným dosazováním

$$g_j(x) = c_1 x_1^{(0)} + c_2 x_2^{(0)} + \ldots + c_j x_j^{(0)}$$

b) Na základě tvrzení části a) je $(x_1^*, x_2^*, \ldots, x_n^*)$ $\{0, 1\}$ — řešení rovnice $a_1 x_1 + a_2 x_2 + \ldots + a_n x_n = x^*$, a tedy patří do množiny (8). Kromě toho platí na základě a) $c_1 x_1^* + c_2 x_2^* + \ldots + c_n x_n^* = g^*$, takže $(x_1^*, x_2^*, \ldots, x_n^*)$ je řešením maximalizačního problému (7), (8). Důkaz věty 17 je dokončen.

Ukážeme použití vyloženého algoritmu na následujícím číselném příkladu.

Příklad 6: Řešme maximalizační problém (7), (8) pro $n = 7$, $b = 7$ a pro čísla a_j a c_j uvedená v horní části tabulky 6. Průběh výpočtu zapisujeme v dolní části této tabulky. V případě nedefinovaných hodnot ponecháváme příslušná pole tabulky prázdná. Hledaná maximální hodnota je $g_7(4) = 16$. Konstrukci příslušného řešení podle věty 17 lze provádět názorným způsobem s využitím tabulky (viz silná čára). Dostáváme řešení (1, 0, 0, 0, 1, 0, 0). Všimněme si pro kontrolu správnosti výpočtu, že skutečně platí

$$8.1 - 7.0 + 5.0 + 6.0 + 8.1 - 5.0 - 7.0 = 16$$

a

$$3.1 + 1.0 + 4.0 + 4.0 + 1.1 + 2.0 + 2.0 = 4$$

j	1			2			3			4			5			6			7	
a_j	3			1			4			4			1			2			2	
c_j	8			−7			5			6			8			−5			−7	
x	g_1	Q_1	$\bar{g}_1$	g_2	Q_2	$\bar{g}_2$	g_3	Q_3	$\bar{g}_3$	g_4	Q_4	$\bar{g}_4$	g_5	Q_5	$\bar{g}_5$	g_6	Q_6	$\bar{g}_6$	g_7	Q_7
0	0	0		0	0		0	0		0	0		0	0		0	0		0	0
1			−7	−7	1		−7	0		−7	0	8	8	1		8	0		8	0
2												1	1	1	−5	1	0	−7	1	0
3	8	1		8	0		8	0		8	0		8	0	3	8	0	1	8	0
4			1	1	1	5	5	1	6	6	1	16	16	1	−4	16	0	−6	16	0
5						−2	−2	1	−1	−1	1	14	14	1	3	14	0	1	14	0
6												7	7	1	11	11	1	9	11	0
7						13	13	1	14	14	1		14	0	9	14	0	7	14	0

Tabulka 6

Na závěr kapitoly se zmíníme o efektivnosti popsaného algoritmu. Pro získání představy srovnejme jej s následujícím triviálním algoritmem: Postupně se uvažují všechny n-tice $(x_1, \ldots, x_n)$ z nul a jedniček a pro ty z nich, které splňují nerovnost $a_1x_1 + \ldots + a_nx_n \leqq b$ se určí hodnota $c_1x_1 + \ldots + c_nx_n$; ze získaných součtů $c_1x_1 + \ldots + c_nx_n$ se nalezne největší. Je zřejmé, že tento algoritmus vyžaduje prozkoumat celkem 2^n n-tic $(x_1, x_2, \ldots, x_n)$, zatímco podstatná část algoritmu dynamického programování záleží v „zaplnění" $n(b + 1)$ políček jisté obdélníkové tabulky pro $g_j(x)$. Efektivnost netriviálního algoritmu dynamického programování záleží tedy v našem případě na velikosti čísla b. Pokud bude b „podstatně menší" než 2^n (např. „řádově srovnatelné" s n), bude algoritmus dynamického programování jistě mnohem efektivnější než algoritmus triviální.

Cvičení

Cvičení 1: Dokažte lemma 2.

Cvičení 2: Nalezněte a odůvodněte metodu pro určení minima funkce (7) na množině (8), analogickou vyložené metodě pro maximalizační problém.

Cvičení 3: V podmínkách příkladu 6 řešte minimalizační problém.

Cvičení 4: Co lze říci o řešení extremálního problému (7), (8) za obecnějšího předpokladu, že a_j jsou celá nezáporná? (Co lze říci o x_j v řešení $(x_1, x_2, \ldots, x_n)$ za předpokladu $a_j = 0$?)

Cvičení 5: Všimněte si, že též tato konkrétní úloha vede na extremální problém (7), (8): Pašerák se vydává na cestu přes hranici a rozhoduje se, které z n věcí vzít s sebou do rance. Přitom je známo, že j-tá věc váží a_j kg, na druhé straně hranice za ni dostane zaplaceno c_j jednotek měny a na zádech unese nejvýše b kg. (Poznámka: Extremální problém (7), (8) se v literatuře nazývá též „úloha o ranci".)

Cvičení 6: Jak lze nalézt všechna řešení extremálního problému (7), (8)?

Cvičení 7: [Neúspěšný pokus o sestrojení jednoduššího algoritmu pro maximalizační problém (7), (8).] V podmínkách maximalizačního problému (7), (8) sestrojme n-tici $(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$ takto: Uspořádejme čísla $c_j a_j^{-1}$ $(j = 1, \ldots, n)$ podle velikosti

$$c_{i_1} a_{i_1}^{-1} \geqq c_{i_2} a_{i_2}^{-1} \geqq \ldots \geqq c_{i_n} a_{i_n}^{-1}$$

a určeme maximální k tak, aby platilo $a_{i_1} + a_{i_2} + \ldots + a_{i_k} \leqq b$. Nyní položme

$$\bar{x}_i = \begin{cases} 1, \text{ jestliže } i \in \{i_1, i_2, \ldots, i_k\} \\ 0, \text{ jestliže } i \notin \{i_1, i_2, \ldots, i_k\} \end{cases}$$

Ukažte pomocí vhodně zvoleného číselného příkladu, že $(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_k)$ není obecně řešením maximalizačního problému (7), (8).

7. kapitola

O JEDNOM PŘIŘAZOVACÍM PROBLÉMU

Nechť jsou dány dvě n-tice reálných čísel

$$(a_1, a_2, \ldots, a_n) \quad \text{a} \quad (b_1, b_2, \ldots, b_n)$$

kde n je dané přirozené číslo. Náš problém záleží v tom, přiřadit číslům a_1, a_2, ..., a_n vzájemně jednoznačně v nějakém obecně novém pořadí čísla $b_1, b_2, \ldots, b_n$ tak, aby součet součinů přiřazených dvojic byl maximální. Jinými slovy, hledáme takové pořadí $b_{k_1}, b_{k_2}, \ldots, b_{k_n}$ čísel $b_1, b_2, \ldots, b_k$, pro které platí

$$a_1 b_{k_1} + \ldots + a_n b_{k_n} \geqq a_1 b_{l_1} + \ldots + a_n b_{l_n}$$

pro všechna možná pořadí $b_{l_1}, b_{l_2}, \ldots, b_{l_n}$ čísel b_1, b_2, ..., b_n.

Řešení zformulovaného problému je v principu možné provádět následující triviální metodou: Vyzkouší se všechna možná pořadí čísel $b_1, b_2, \ldots, b_n$ a určí se jim odpovídající součty součinů a z nich se vyhledá maximální. Protože všech zmíněných pořadí je $n! = 1 . 2 . \ldots . n$, je takováto metoda pro „větší" hodnoty n velmi neefektivní. Velmi jednoduché řešení problému nám umožní další věta, jejíž důkaz je založen na myšlence dynamického programování. Při formulaci věty budeme předpokládat, že platí

$$a_1 \leqq a_2 \leqq \ldots \leqq a_n \tag{9}$$

(Splnění tohoto předpokladu lze snadno dosáhnout vhodným přečíslováním.)

Věta 18: Nechť n-tice čísel $(a_1, a_2, \ldots, a_n)$ splňuje podmínku (9) a nechť $b_{k_1}, b_{k_2}, \ldots, b_{k_n}$ je takové pořadí čísel $b_1, b_2, \ldots, b_n$, pro které platí $b_{k_1} \leqq b_{k_2} \leqq \ldots \leqq b_{k_n}$. Potom je $a_1 b_{k_1} + \ldots + a_n b_{k_n}$ maximální.

K důkazu věty budeme potřebovat následující pomocnou větu, jejíž snadný důkaz přenecháváme čtenáři.

Lemma 3: Nechť pořadí $b_{r_1}, b_{r_2}, \ldots, b_{r_n}$ splňuje nerovnost $b_{r_i} \geqq b_{r_j}$ pro některou uspořádanou dvojici indexů (i, j), kde $i < j$. Sestrojme nové pořadí b_{s_1}, $b_{s_2}, \ldots, b_{s_n}$, kde $s_j = r_i$, $s_i = r_j$ a $s_k = r_k$ pro $k = 1, 2, \ldots, n$; $i \neq k \neq j$. Potom platí $a_1 b_{r_1} + \ldots + a_n b_{r_n} \leqq \leqq a_1 b_{s_1} + \ldots + a_n b_{s_n}$.

Budeme říkat, že pořadí $b_{s_1}, \ldots, b_{s_n}$ z lemmatu 3 vzniklo z pořadí $b_{r_1}, \ldots, b_{r_n}$ *transpozicí* dvojice (b_{r_i}, b_{r_j}).

Důkaz věty 18: Větu dokážeme matematickou indukcí. Pro $n = 1$ je tvrzení zřejmé. Předpokládejme, že tvrzení věty platí pro přirozené číslo n a dokažme je pro $n + 1$. Nechť $b_{k_1}, b_{k_2}, \ldots, b_{k_{n+1}}$ je pořadí splňující předpoklad věty a nechť $b_{l_1}, b_{l_2}, \ldots, b_{l_{n+1}}$ je libovolné pořadí čísel $b_1, \ldots, b_{n+1}$. Naším cílem je ukázat, že platí

$$a_1 b_{l_1} + \ldots + a_{n+1} b_{l_{n+1}} \leqq a_1 b_{k_1} + \ldots + a_{n+1} b_{k_{n+1}}, \quad (10)$$

Rozlišíme dva případy: a) $l_{n+1} = k_{n+1}$, b) $l_{n+1} \neq k_{n+1}$. V případě a) platí $a_{n+1} b_{l_{n+1}} = a_{n+1} b_{k_{n+1}}$, takže dokazovaná nerovnost vyplývá ze vztahu $a_1 b_{l_1} + \ldots + a_n b_{l_n} \leqq$

$\leq a_1b_{k_1} + \ldots + a_nb_{k_n}$, který však platí na základě indukčního předpokladu. V případě b) postupujeme takto: V pořadí $b_{l_1}, b_{l_2}, \ldots, b_{l_n}$ nalezneme člen b_{l_i}, pro který $l_i = k_{n+1}$ a transpozicí dvojice $(b_{l_i}, b_{l_{n+1}})$ získáme z pořadí $b_{l_1}, \ldots, b_{l_n}, b_{l_{n+1}}$ nové pořadí $b_{p_1}, \ldots, b_{p_n}, b_{p_{n+1}}$, pro které $p_{n+1} = k_{n+1}$. Platí tedy jednak $a_1b_{p_1} + \ldots + a_{n+1}b_{p_{n+1}} \leqq a_1b_{k_1} + \ldots + a_{n+1}b_{k_{n+1}}$ (případ a)), a dále, protože $b_{l_i} = b_{k_{n+1}} \geqq b_{l_{n+1}}$, dostáváme na základě lemmatu 3 $a_1b_{l_1} + \ldots + a_{n+1}b_{l_{n+1}} \leqq a_1b_{p_1} + \ldots + a_{n+1}b_{p_{n+1}}$.

Spojením posledních dvou nerovností získáváme nerovnost (10), čímž je důkaz dokončen.

Věta 18 nám poskytuje jednoduchou metodu pro řešení zformulovaného extremálního problému, záležející v tom, že se čísla $a_1, \ldots, a_n$ i $b_1, \ldots, b_n$ uspořádají „podle velikosti“ a přiřadí se navzájem členy nacházející se na stejném místě v obou pořadích.

Příklad 6: Použijeme větu 18 pro $n = 7$,

$$a_1 = -2,\ a_2 = 3,\ a_3 = 4,\ a_4 = 0,$$
$$a_5 = 1,\ a_6 = 3,\ a_7 = -1$$
$$b_1 = 10,\ b_2 = 5,\ b_3 = -4,\ b_4 = 6,$$
$$b_5 = 0,\ b_6 = -3,\ b_7 = 2$$

Sestrojíme nová pořadí

$$a_1 = -2,\ a_7 = -1,\ a_4 = 0,\ a_5 = 1,$$
$$a_2 = 3,\ a_6 = 3,\ a_3 = 4$$
$$b_3 = -4,\ b_6 = -3,\ b_5 = 0,\ b_7 = 2,$$
$$b_2 = 5,\ b_4 = 6,\ b_1 = 10$$

odkud dostáváme hledané přiřazení. Maximální hodnota, odpovídající potom našemu maximalizačnímu problému je $a_1b_3 + a_7b_6 + \ldots + a_3b_1 = 86$.

Analogický problém o minimu se řeší na základě analogické věty, jejíž důkaz přenecháváme čtenáři.

Věta 19: Nechť n-tice čísel $(a_1, a_2, \ldots, a_n)$ splňuje podmínku (9) a nechť $b_{m_1}, \ldots, b_{m_n}$ je takové pořadí čísel $b_1, \ldots, b_n$, že platí $b_{m_1} \geqq b_{m_2} \geqq \ldots \geqq b_{m_n}$. Potom má součet $a_1 b_{m_1} + \ldots + a_n b_{m_n}$ nejmenší možnou hodnotu ze všech takových součtů.

Cvičení

Cvičení 1: Dokažte lemma 3.

Cvičení 2: Dokažte větu 19.

Cvičení 3: Řešte minimalizační problém v podmínkách příkladu 6.

Cvičení 4: Dokažte následující tvrzení: Nechť čísla $a_1, \ldots, a_n$ i čísla $b_1, \ldots, b_n$ jsou navzájem různá, a nechť platí $a_1 < a_2 < \ldots < a_n$. Potom je $a_1 b_{k_1} + \ldots + a_n b_{k_n}$ maximální tehdy a jen tehdy, jestliže $b_{k_1} < b_{k_2} < \ldots < b_{k_n}$.

Cvičení 5: Řešte následující extremální problémy:

a) Nalezněte pořadí $b_{k_1}, b_{k_2}, \ldots, b_{k_n}$ čísel $b_1, b_2, \ldots, b_n$, pro které $\max \{a_j b_{k_j} \mid j = 1, 2, \ldots, n\}$ nabývá své minimální hodnoty.

b) Nalezněte pořadí $b_{r_1}, b_{r_2}, \ldots, b_{r_n}$ čísel $b_1, b_2, \ldots, b_n$, pro které $\min \{a_j b_{r_j} \mid j = 1, \ldots, n\}$ nabývá své maximální hodnoty.

Cvičení 6: Cestující chce postupně navštívit 5 měst **A, B, C, D** a **E** v uvedeném pořadí. Všechny úseky cesty **AB, BC, CD,** a **DE** mají navzájem různou délku a k dispozici jsou celkem čtyři typy letadel s navzájem různými cestovními rychlostmi, přičemž všechny typy létají na každém ze čtyř úseků. Dokažte: Jestliže cestující chce vyzkoušet každý typ letadla a přitom vykonat cestu za nejkratší možnou dobu, musí používat tím rychlejší letadlo, čím je delší úsek.*)

*) Srovnej s úlohou č. 84 z knížky *Dynkina, Molčanova a Rozentala*, „*Matěmatičeskije sorevnovanija* (Arifmetika i algebra)“, Nauka, Moskva, 1970.

8. kapitola

PŘÍPAD NEKONEČNÝCH MNOŽIN

Zatím jsme metodu dynamického programování aplikovali pouze v případě funkcí s konečným definičním oborem. V tomto případě se nevyskytovaly žádné potíže s existencí extrémů, neboť každá funkce definovaná na konečné neprázdné množině má maximum i minimum. V případě funkce definované na nekonečné množině však již existenci extrémů obecně zaručit nelze (vezměme např. funkci f definovanou na množině $\{x \in \mathbf{R} | 0 < x < 1\}$ předpisem $f(x) = x$; tato funkce nemá ani minimum ani maximum). Otázky existence extrémů funkcí definovaných na nekonečných množinách se vyšetřují metodami matematické analýzy. Za příklad nám může sloužit fakt, že každý polynom, tj. funkce tvaru $y = a_0x^n + a_1x^{n-1} + \ldots + a_n$ ($a_0, a_1, \ldots, a_n$ jsou reálné konstanty) nabývá na každém intervalu tvaru $\{x \in \mathbf{R} | a \leqq x \leqq b\}$, kde $a \leqq b$ jsou daná reálná čísla, své minimální i maximální hodnoty. S posledním tvrzením i s mnoha jinými tohoto typu se čtenář seznámí již na počátku vysokoškolského studia matematiky. My se obecnými existenčními otázkami pro extrémy funkcí nebudeme zabývat, tím spíše, že zpravidla z nich nevyplývají žádné metody pro skutečné nalezení extrémů.

V řadě konkrétních i důležitých případů lze však extrémy funkcí definovaných na nekonečných množinách nalézt přímo s použitím různých elementárnějších metod. V této kapitole uvádíme dva příklady použití metody dynamického programování.

Příklad 7: Je dáno reálné číslo a a přirozené číslo n. Máme zjistit, zdali funkce n proměnných

$$y = x_1^2 + x_2^2 + \ldots + x_n^2 \tag{11}$$

nabývá na množině

$$\left\{(x_1, \ldots, x_n) \middle| \begin{array}{l} x_1, \ldots, x_n \in \mathbf{R} \\ x_1 + \ldots + x_n = a \end{array}\right\} \tag{12}$$

své minimální hodnoty.

Ukážeme nejen, že funkce (11) nabývá na množině (12) minima, ale nalezneme toto minimum a nalezneme příslušnou n-tici, pro kterou se toto minimum nabývá. Dokážeme následující tvrzení:

Funkce (11) nabývá minimální hodnoty a^2n^{-1}, a sice v jediné n-tici $(x_1, x_2, \ldots, x_n) = (an^{-1}, an^{-1}, \ldots, an^{-1})$.

Důkaz: Větu dokážeme metodou matematické indukce podle n. (Hlavní myšlenkou důkazu je opět použití dynamického programování.)

(i) V případě $n = 1$ je tvrzení zřejmé.

(ii) Předpokládejme, že tvrzení platí pro přirozené číslo n a dokažme je pro $n + 1$: Nechť $(x_1, x_2, \ldots, x_{n+1})$ je $(n + 1)$-tice reálných čísel, splňující vztah

$x_1 + \ldots + x_n + x_{n+1} = a$, odkud vyplývá $x_1 + \ldots + x_n = a - x_{n+1}$.

Na základě indukčního předpokladu nyní dostáváme, že platí

$$x_1^2 + x_2^2 + \ldots + x_n^2 \geq (a - x_{n+1})^2\, n^{-1}, \text{ a tedy}$$

$$\left.\begin{aligned} x_1^2 + \ldots + x_n^2 + x_{n+1}^2 &\geqq \frac{(a - x_{n+1})^2}{n} + x_{n+1}^2 = \\ = \frac{n+1}{n} \cdot &\left[\left(x_{n+1} - \frac{a}{n+1}\right)^2 + \frac{a^2 n}{(n+1)^2}\right] \geqq \\ \geqq \frac{n+1}{n} \; &\frac{a^2 n}{(n+1)^2} = \frac{a^2}{n+1} \end{aligned}\right\} \qquad (13)$$

Platí tedy nerovnost $x_1^2 + \ldots + x_{n+1}^2 \geqq a^2 (n+1)^{-1}$ pro všechny $(n+1)$-tice $(x_1, \ldots, x_{n+1})$ reálných čísel, splňující vztah $x_1 + \ldots + x_{n+1} = a$.

Dosazením $(x_1, \ldots, x_{n+1}) = [a(n+1)^{-1}, \ldots, a(n+1)^{-1}]$ se snadno přesvědčíme, že $a^2(n+1)^{-1}$ je skutečně minimum funkce (11) na množině (12). K zakončení důkazu zbývá nalézt všechny $(n+1)$-tice, pro které funkce (11) nabývá svého minima na množině (12). Nechť tedy pro $(x_1, \ldots, x_{n+1})$ platí $x_1^2 + \ldots + x_n^2 + x_{n+1}^2 = a^2(n+1)^{-1}$ a $x_1 + x_2 + \ldots + x_n + x_{n+1} = a$. Odtud a z řetězce vztahů (13) dostáváme jednak

$$x_1^2 + \ldots + x_n^2 = (a - x_{n+1})^2 \, n^{-1} \qquad (14)$$

jednak

$$\left(x_{n+1} - \frac{a}{n+1}\right)^2 = 0 \qquad (15)$$

Ze (14) a ze vztahu $x_1 + \ldots + x_n = a - x_{n+1}$ vyplývá na základě indukčního předpokladu $x_1 = \ldots = x_n = (a - x_{n+1}) \, n^{-1}$ a z (15) dostáváme $x_{n+1} = a(n+1)^{-1}$. Z posledních dvou vztahů dostáváme nakonec $x_1 = x_2 = \ldots = x_n = x_{n+1} = a(n+1)^{-1}$, což dokončuje důkaz.

Použití principu dynamického programování záleželo v tomto případě podobně jako v předchozích kapitolách v převedení extremálního problému pro funkci více proměnných na několik extremálních problémů pro funkce jedné proměnné.

Příklad 8: Uvažujme funkci

$$y = x_1 + x_2 + \ldots + x_n \tag{16}$$

definovanou na množině

$$\left\{(x_1, \ldots, x_n) \,\middle|\, \begin{matrix} x_1, \ldots, x_n \in \mathbf{R} \\ x_1^2 + \ldots + x_n^2 = r^2 \end{matrix}\right\} \tag{17}$$

kde r je dané nezáporné číslo. Dokažte, že funkce (16) nabývá na množině (17) maximální hodnoty $r\sqrt{n}$, a sice v jediné n-tici

$$(x_1, \ldots, x_n) = \left(\frac{r}{\sqrt{n}}, \ldots, \frac{r}{\sqrt{n}}\right)$$

Důkaz: Pro $n = 1$ je tvrzení zřejmé. Předpokládejme, že je tvrzení dokázáno pro přirozené číslo n a uvažujme libovolnou $(n + 1)$-tici $(x_1, \ldots, x_n, x_{n+1})$ reálných čísel, splňující vztah

$$x_1^2 + \ldots + x_n^2 + x_{n+1}^2 = r^2$$

Potom platí $r^2 - x_{n+1}^2 \geqq 0$, $r\sqrt{n+1} - x_{n+1} \geqq 0$ a $x_1^2 + \ldots + x_n^2 = (\sqrt{r^2 - x_{n+1}^2})^2$. Odtud dostáváme $x_1 + \ldots + x_n + x_{n+1} \leqq \sqrt{r^2 - x_{n+1}^2}\,\sqrt{n} + x_{n+1} = \sqrt{-(x_{n+1}\sqrt{n+1} - r)^2 + (r\sqrt{n+1} - x_{n+1})^2} + x_{n+1} \leqq \sqrt{(r\sqrt{n+1} - x_{n+1})^2} + x_{n+1} = |r\sqrt{n+1} - x_{n+1}| + x_{n+1} = r\sqrt{n+1}$, což dokazuje první část věty.

Z posledního řetězce nerovností je dále zřejmé, že vztah $x_1 + \ldots + x_{n+1} = r\sqrt{n+1}$ platí právě tehdy, jestliže $x_1 + \ldots + x_n = \sqrt{r^2 - x_{n+1}^2}\sqrt{n}$ (tj., na základě indukčního předpokladu, jestliže $x_1 = \ldots = x_n = \sqrt{r^2 - x_{n+1}^2} : \sqrt{n}$) a

$$(x_{n+1}\sqrt{n+1} - r)^2 = 0, \text{ tj. } x_{n+1} = \frac{r}{\sqrt{n+1}}$$

Indukční důkaz je dokončen.

9. kapitola

CAUCHY-LAGRANGEOVA NEROVNOST*)

V této kapitole zobecníme výsledek z příkladu 8 předcházející kapitoly a z tohoto zobecnění získáme důkaz dvou klasických nerovností, důležitých v geometrii a matematické analýze.

Platí tato věta:

Věta 10: Nechť jsou dána reálná čísla $a_1, a_2, \ldots, a_n$ (n je dané přirozené číslo) a nezáporné číslo x. Pro libovolnou n-tici $(x_1, x_2, \ldots, x_n)$ reálných čísel, pro kterou $x_1^2 + x_2^2 + \ldots + x_n^2 = x^2$, platí

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n \leqq x\sqrt{a_1^2 + a_2^2 + \ldots + a_n^2}$$

Důkaz: Tvrzení dokážeme indukcí. Pro $n = 1$ je zřejmé; předpokládejme, že platí pro přirozené číslo n a nechť $(x_1, \ldots, x_n, x_{n+1})$ je libovolná $(n+1)$-tice reálných čísel, splňující vztah $x_1^2 + \ldots + x_n^2 + x_{n+1}^2 = x^2$. Odtud vyplývá $x_1^2 + \ldots + x_n^2 = (\sqrt{x^2 - x_{n+1}^2})^2$ a tedy na základě indukčního předpokladu $a_1x_1 + \ldots + a_nx_n \leqq \sqrt{x^2 - x_{n+1}^2}\sqrt{a_1^2 + \ldots + a_n^2}$.

Odtud dostáváme $a_1x_1 + \ldots + a_nx_n + a_{n+1}x_{n+1} \leqq \sqrt{x^2 - x_{n+1}^2}\sqrt{a_1^2 + \ldots + a_n^2} + a_{n+1}x_{n+1} = \sqrt{(x\sqrt{a_1^2 + \ldots + a_{n+1}^2} - a_{n+1}x_{n+1})^2 - (x_{n+1}\sqrt{a_1^2 + \ldots + a_{n+1}^2} - a_{n+1}x)^2} + a_{n+1}x_{n+1} \leqq$

*) Cauchy, čti koši; Lagrange, čti lagranže.

$\leqq |x\sqrt{a_1^2 + \ldots + a_{n+1}^2} - a_{n+1} \cdot x_{n+1}| + a_{n+1}x_{n+1} = x \sqrt{a_1^2 + \ldots + a_{n+1}^2}$, čímž je důkaz dokončen.

Pomocí věty 20 lze dokázat následující nerovnost, která má sama, ale hlavně její různá zobecnění, důležitou úlohu v geometrii a analýze.

Věta 21: (Nerovnost Cauchy-Lagrangeova) Nechť $(a_1, \ldots, a_n)$ a $(b_1, \ldots, b_n)$ jsou libovolné dvě uspořádané n-tice reálných čísel. Potom platí $|a_1b_1 + \ldots + a_nb_n| \leqq \leqq \sqrt{a_1^2 + \ldots + a_n^2}\sqrt{b_1^2 + \ldots + b_n^2}$.

Důkaz: Nejprve dokážeme, že z předpokladů věty vyplývá nerovnost

$$a_1b_1 + \ldots + a_nb_n \leqq \sqrt{a_1^2 + \ldots + a_n^2}\sqrt{b_1^2 + \ldots + b_n^2} \tag{18}$$

Skutečně, položíme-li ve větě 20 $x_j = b_j$ $(j = 1, 2, \ldots, n)$ a $x = \sqrt{b_1^2 + \ldots + b_n^2}$, dostáváme $a_1b_1 + \ldots + a_nb_n = a_1x_1 + \ldots + a_nx_n \leqq x\sqrt{a_1^2 + \ldots + a_n^2} = \sqrt{a_1^2 + \ldots + a_n^2}\sqrt{b_1^2 + \ldots + b_n^2}$, což jsme měli ukázat. Ze vztahu (18) však vyplývá s použitím elementárních vlastností absolutní hodnoty $|a_1b_1 + \ldots + a_nb_n| \leqq |a_1|\,|b_1| + \ldots + |a_n|\,|b_n| \leqq \sqrt{|a_1|^2 + \ldots + |a_n|^2}\sqrt{|b_1|^2 + \ldots + |b_n|^2} = \sqrt{a_1^2 + \ldots + a_n^2}\sqrt{b_1^2 + \ldots + b_n^2}$, čímž je důkaz dokončen.

Pomocí Cauchy-Langrangeovy nerovnosti dokážeme dále tzv. trojúhelníkovou nerovnost pro vzdálenost v Eukleidově prostoru. Ze školy je známo, že na přímce lze zvolit souřadnicový systém, tj. poloha každého bodu na přímce je vzájemně jednoznačně určena jistým reálným číslem. Jestliže poloha bodu **A** přímky je určena číslem a, píšeme $\mathbf{A} = (a)$. V takto zvoleném souřadnicovém systému je vzdálenost mezi dvěma body $\mathbf{A} = (a)$ a $\mathbf{B} = (b)$ dána číslem $|a - b|$, které lze zapsat též ve tvaru $\sqrt{(a - b)^2}$. Podobně, je-li v rovině zavedena obvyklým způsobem soustava pravoúhlých souřadnic, takže poloha libovolného bodu $\mathbf{A}$ roviny je vzájemně jednoznačně určena uspořádanou dvojicí reálných čísel a_1, a_2 (což zapisujeme pomocí $\mathbf{A} = (a_1, a_2)$), je vzdálenost mezi body $\mathbf{A} = (a_1, a_2)$ a $\mathbf{B} = (b_1, b_2)$ rovna $\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}$, jak se snadno dokáže pomocí Pythagorovy věty. Též v prostorových souřadnicích se vzdálenost mezi body $\mathbf{A} = (a_1, a_2, a_3)$ a $\mathbf{B} = (b_1, b_2, b_3)$ prostoru počítá podle analogického vzorce $\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2}$.

Abstrakcí se v matematice z pojmů přímky, roviny a (obyčejného) prostoru vyvinul pojem *Eukleidova n-rozměrného prostoru**) (n je přirozené číslo), kterým rozumíme množinu všech uspořádaných n-tic $\mathbf{A} = (a_1, a_2, \ldots, a_n)$ reálných čísel, spolu s funkcí ϱ definovanou pro všechny uspořádané dvojice n-tic ($\mathbf{A} = (a_1, \ldots, a_n)$, $\mathbf{B} = (b_1, \ldots, b_n)$) předpisem

$$\varrho(\mathbf{A}, \mathbf{B}) = \sqrt{(a_1 - b_1)^2 + \ldots + (a_n - b_n)^2}$$

Přitom se n-tice $\mathbf{A} = (a_1, \ldots, a_n)$ nazývají *body* prostoru, čísla $a_1, \ldots, a_n$ *souřadnicemi* bodu $\mathbf{A} = (a_1, \ldots, a_n)$ a $\varrho(\mathbf{A}, \mathbf{B})$ *Eukleidovou vzdáleností* bodů $\mathbf{A}$, $\mathbf{B}$.

Z elementární geometrie je známo, že v případě přímky, roviny a „obyčejného" prostoru platí pro vzdálenost bodů tzv. *trojúhelníková nerovnost*, kterou lze for-

*) Pro solidnější seznámení s pojmem Eukleidova n-rozměrného prostoru odkazujeme čtenáře na knížku K. Havlíčka „Prostory o čtyřech a více rozměrech"; vydala v edici „Škola mladých matematiků" Mladá fronta (Praha 1965).

mulovat takto: Pro libovolné tři body **A**, **B**, **C** je vzdálenost mezi **A**, **C** menší nebo rovna součtu vzdáleností mezi **A**, **B** a mezi **B**, **C**. Vzniká přirozená otázka, zdali obdobné tvrzení platí i v obecném případě Eukleidova n-rozměrného prostoru. Následující věta obsahuje kladnou odpověď na tuto otázku.

Věta 22: Pro libovolné tři body **A**, **B**, **C** Eukleidova n-rozměrného prostoru platí

$$\varrho(\mathbf{A}, \mathbf{C}) \leqq \varrho(\mathbf{A}, \mathbf{B}) + \varrho(\mathbf{B}, \mathbf{C}) .$$

Důkaz: Na základě Cauchy-Lagrangeovy nerovnosti platí

$$(a_1 - b_1)(b_1 - c_1) + \ldots + (a_n - b_n)(b_n - c_n) \leqq$$
$$\leqq \sqrt{(a_1 - b_1)^2 + \ldots + (a_n - b_n)^2} \,.$$
$$. \sqrt{(b_1 - c_1)^2 + \ldots + (b_n - c_n)^2}$$

Odtud dále vyplývá

$$[(a_1 - b_1)^2 + \ldots + (a_n - b_n)^2] +$$
$$+ [(b_1 - c_1)^2 + \ldots + (b_n - c_n)^2] +$$
$$+ 2[(a_1 - b_1)(b_1 - c_1) + \ldots + (a_n - b_n)(b_n - c_n)] \leqq$$
$$\leqq [(a_1 - b_1)^2 + \ldots + (a_n - b_n)^2] +$$
$$+ [(b_1 - c_1)^2 + \ldots + (b_n - c_n)^2] +$$
$$+ 2\sqrt{(a_1 - b_1)^2 + \ldots + (a_n - b_n)^2} \,.$$
$$\cdot \sqrt{(b_1 - c_1)^2 + \ldots + (b_n - c_n)^2}$$

a po jednoduché úpravě obou stran získané nerovnosti dostáváme nakonec vztah

$$(a_1 - c_1)^2 + \ldots + (a_n - c_n)^2 \leqq$$
$$\leqq [\sqrt{(a_1 - b_1)^2 + \ldots + (a_n - b_n)^2} +$$
$$+ \sqrt{(b_1 - c_1)^2 + \ldots + (b_n - c_n)^2}]^2$$

Protože na obou stranách poslední nerovnosti jsou nezáporná čísla, vyplývá z ní

$$\sqrt{(a_1 - c_1)^2 + \ldots + (a_n - c_n)^2} \leqq$$
$$\leqq \sqrt{(a_1 - b_1)^2 + \ldots + (a_n - b_n)^2} +$$
$$+ \sqrt{(b_1 - c_1)^2 + \ldots + (b_n - c_n)^2}$$

což jsme měli dokázat.

Cvičení

Cvičení 1: Nalezněte všechny n-tice $(x_1, x_2, \ldots, x_n)$ z věty 20, pro které platí $x_1^2 + x_2^2 + \ldots + x_n^2 = x^2$ a $a_1x_1 + \ldots + a_nx_n = x\sqrt{a_1^2 + \ldots + a_n^2}$.

Cvičení 2: Dokažte, že za předpokladů věty 20 platí $|a_1x_1 + \ldots + a_nx_n| \leqq x\sqrt{a_1^2 + \ldots + a_n^2}$.

Cvičení 3: (Jiný důkaz Cauchy-Lagrangeovy nerovnosti)
a) Nechť $\alpha_1, \alpha_2, \ldots, \alpha_n$ a $\beta_1, \beta_2, \ldots, \beta_n$ jsou libovolná reálná čísla, splňující vztahy $\alpha_1^2 + \ldots + \alpha_n^2 = \beta_1^2 + \ldots + \beta_n^2 = 1$. Potom platí $\alpha_1\beta_1 + \ldots + \alpha_n\beta_n \leqq 1$. Dokažte zformulované tvrzení! (Návod: Vyšetřujte výraz $(\alpha_1 - \beta_1)^2 + \ldots + (\alpha_n - \beta_n)^2$.)
b) Pomocí výsledku a) dokažte Cauchy-Lagrangeovu nerovnost. (Návod: Položte

$$\alpha_j = \frac{a_j}{\sqrt{a_1^2 + \ldots + a_n^2}}, \beta_j = \frac{b_j}{\sqrt{b_1^2 + \ldots + b_n^2}}$$

pro $j = 1, 2, \ldots, n$.)

Cvičení 4: (Ještě jeden důkaz Cauchy-Lagrangeovy nerovnosti.)

a) Dokažte rovnost $(a_1^2 + \ldots + a_n^2) \cdot (b_1^2 + \ldots + b_n^2) -$

$$-(a_1b_1 + \dots + a_nb_n)^2 = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (a_ib_j - a_ib_j)^2\text{*)}$$ (Lagrangeova identita).

b) Pomocí výsledku a) dokažte Cauchy-Lagrangeovu nerovnost.

Cvičení 5: Ukažte, že Eukleidova vzdálenost má též tyto dvě vlastnosti (**A, B** označují libovolné dva body Eukleidova n-rozměrného prostoru): 1. ϱ (**A, B**) = ϱ (**B, A**) 2. ϱ (**A, B**) = **0,** právě když **A** = **B**.

Cvičení 6: Dokažte, že pro Eukleidovu vzdálenost je nerovnost ϱ **(A, B)** $\geqq$ |ϱ **(A, C)** — ϱ **(B, C)**| splněna pro všechny trojice bodů **A, B, C** Eukleidova n-rozměrného prostoru. (Návod: Použijte vhodně trojúhelníkové nerovnosti.)

*) Zápis znamená v souhlasu s obvyklou matematickou symbolikou součet všech čísel $(a_ib_j - b_ia_j)^2$, kde i, j jsou celá, $l \leq i < j \leq n$.

10. kapitola

O PRINCIPU OPTIMÁLNOSTI

V této závěrečné kapitole si řekneme několik slov o tzv. principu optimálnosti dynamického programování, na kterém se zakládaly všechny konkrétní algoritmy i důkazy probrané v této knížce. Během naší diskuse vysvětlíme rovněž, proč se používá termínu „dynamické programování".

Všimneme si první z našich aplikací dynamického programování, která záležela v určení maxima funkce

$$y = g_1(x_1) + \ldots + g_n(x_n) \tag{19}$$

na množině

$$\left\{(x_1, \ldots, x_n) \;\middle|\; \begin{array}{l} x_1, \ldots, x_n \text{ celá a nezáp.} \\ x_1 + \ldots + x_n = a \end{array}\right\} \tag{20}$$

Každé řešení maximalizačního problému (19), (20) má jistou zajímavou vlastnost, jak ukazuje následující věta.

Věta 23: Nechť $(x_1^{(0)}, \ldots, x_n^{(0)})$ je řešení maximalizačního problému (19), (20) a nechť i, j jsou daná celá čísla, splňující vztahy $1 \leqq i \leqq j \leqq n$. Potom pro $(j - i + 1)$-tici $(x_i^{(0)}, x_{i+1}^{(0)}, \ldots, x_j^{(0)})$ nabývá funkce

$$y = g_i(x_i) + g_{i+1}(x_{i+1}) + \ldots + g_j(x_j) \tag{21}$$

své maximální hodnoty na množině

$$\left\{(x_i, \ldots, x_j) \;\middle|\; \begin{array}{l} x_i, \ldots, x_j \text{ celá, nezáp.} \\ x_i + \ldots + x_j = a(i, j) \end{array}\right\} \tag{22}$$

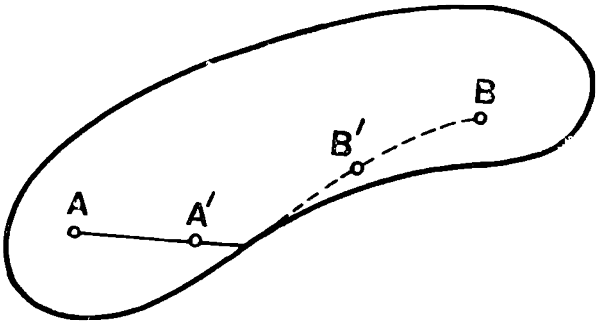
kde $$a(i, j) = x_i^{(0)} + \ldots + x_j^{(0)}$$

Důkaz: Větu dokážeme sporem. Připusťme, že v množině (22) existuje nějaká $(j - i + 1)$-tice $(\hat{x}_i, \ldots, \hat{x}_j)$, pro kterou $g_i(\hat{x}_i) + g_{i+1}(\hat{x}_{i+1}) + \ldots + g_j(\hat{x}_j) > g_i(x_i^{(0)}) + \ldots + g_j(x_j^{(0)})$. Sestrojme nyní n-tici $(\tilde{x}_1, \ldots, \tilde{x}_n)$ tak, že $\tilde{x}_r = x_r^{(0)}$ pro $r = 1, \ldots, i - 1, j + 1, \ldots, n$, a $\tilde{x}_r = \hat{x}_r$ pro $r = i, i + 1, \ldots, j$. Snadno lze ověřit, že $(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n)$ je prvkem množiny (20), a že platí $g_1(\tilde{x}_1) + \ldots + g_n(\tilde{x}_n) > g_1(x_1^{(0)}) + \ldots + g_n(x_n^{(0)})$. Získaná nerovnost je však ve sporu s tím, že $(x_1^{(0)}, \ldots, x_n^{(0)})$ je řešením problému (19), (20), což dokončuje důkaz.

Tvrzení věty 23 dáme nyní názornější význam. Problém tvaru (21), (22) nazveme *úsekem problému* (19), (20), a $(j - i + 1)$-tici $(x_i^{(0)}, \ldots, x_j^{(0)})$ nazveme *úsekem řešení* $(x_1^{(0)}, \ldots, x_n^{(0)})$. Nyní lze smysl věty 23 vyjádřit názorně takto: „Úsek řešení problému (19), (20) je řešením úseku tohoto problému."

Zformulovaná jednoduchá vlastnost je právě speciálním důsledkem obecného principu optimálnosti dynamického programování, který umožňuje použití rekurentní metody. Abychom se ještě více přiblížili k jeho neformálnímu smyslu, rozebereme tento geometrický příklad. Na povrchu nějakého (geometrického) tělesa **T** jsou zvoleny dva různé body **A**, **B**. Naším úkolem je spojit oba body nejkratší čarou $\mathscr{C}$, ležící na povrchu tělesa. Čtenář se středoškolskou úrovní matematického vzdělání musí ovšem zformulovanou úlohu chápat pouze intuitivně, neboť pojmy jako čára (křivka), délka čáry apod. se definují s dostačující úrovní přesnosti až ve vysokoškolských přednáškách z matematické analýzy a geometrie.

Zformulovaný problém je v obecném případě značně obtížný, neboť povrch tělesa může být složitá plocha s množstvím „výstupků“ a „proláklin“. Za velmi obecných předpokladů o vlastnostech povrchu tělesa **T** lze však dokázat následující, intuitivně velmi názorné tvrzení: **Čára** $\mathscr{C}$ na povrchu tělesa **T**, spojující **A** s **B** je nejkratší ze všech takových čar právě tehdy, jestliže pro každé dva body **A′** a **B′** ležící na $\mathscr{C}$, je úsek $\mathscr{C}$(**A′**, **B′**) čáry $\mathscr{C}$ ležící mezi **A′** a **B′** sám nejkratší čarou na povrchu **T**, spojující **A′** s **B′** (viz obr. 2).



Obr. 2

Tato zdánlivě jednoduchá vlastnost vyjadřuje výstižně podstatu hlubokého principu optimálnosti, podmiňujícího použití metod dynamického programování. Speciálně extremální úlohy řešené v naší knížce by bylo možné interpretovat jako problémy určení jistých nejkratších čar; vyložené rekurentní metody pak vlastně „slepují“ úseky hledané nejkratší čáry, až je nakonec určeno celé řešení.

Úloze o nejkratší čáře lze dát ještě názornou fyzikální interpretaci. Představme si, že po povrchu (reálného fyzikálního) tělesa leze z **A** do **B** konstantní rychlostí brouk, který chce vykonat tuto cestu v nejkratším možném čase, čili — což je v daném případě totéž — po dráze nejkratší délky. Tuto cestu budeme nazývat *optimální*. Slovo „optimální" je latinského původu a jeho význam odpovídá významu českého slova „nejlepší". V daném případě máme konkrétně na mysli, že hledaná cesta má být optimální (nejlepší) ve smyslu minimální doby pohybu (neboli minimální délky dráhy pohybu). Princip optimálnosti lze nyní vyslovit též takto:

Pohyb (v našem konkrétním případě pohyb brouka) probíhá optimálním způsobem právě tehdy, jestliže optimálním způsobem probíhá na každém úseku.

V této obecné formulaci lze principu optimálnosti použít při určování pohybů letadel, raket a družic, optimálních vzhledem k minimální době letu, minimální spotřebě pohonných hmot apod. Lze jej však použít i k určování různých optimálních procesů, např. v technologii, technice a ekonomii, neboť takovéto procesy lze též chápat jako jisté pohyby v některém (obecně vícerozměrném) prostoru. Odtud je též patrný původ názvu „dynamické programování" (slovo „dynamický" odpovídá svým významem českému ekvivalentu „pohybový"). Tím ovšem nechceme říci, že dynamické programování se hodí pouze pro řešení problémů, kde ve zjevné formě vystupuje prostor a čas. (Extremální problémy řešené v naší knížce to ostatně jasně ukazují.) Formulace problémů a metod dynamického programování se obvykle provádí ve zcela abstraktní formě a interpretace pomocí pohybu probíhajícího v prostoru a čase se většinou používá pouze pro názornost a heuristickou cenu.

DOSLOV

Pojem dynamického programování pochází od amerického matematika R. E. Bellmana, který spolu se svými spolupracovníky a žáky tuto metodu aplikoval na velké množství nejrozmanitějších extremálních problémů čisté a aplikované matematiky, od výpočtu letových trajektorií umělých družic až po aplikace na hru šachy. Čtenář, který se chce poučit o některých z těchto zajímavých aplikací, může nahlédnout do inspirující knihy R. E. Bellmana a S. E. Dreyfuse „Applied Dynamic Programming“ (Princeton University Press, Princeton, N. J. 1962), která je u nás snadno dostupná v ruském překladu pod názvem „Prikladnyje zadači dinamičeskogo programmirovanija“ („Nauka“, Moskva, 1965). Je však nutno uvést, že knihu nelze považovat za učebnici, neboť u celé řady témat jsou popsány pouze základní myšlenky a detailní matematický rozbor není proveden.

Pokud jde o českou literaturu, lze čtenáři doporučit přehledný článek *D. Glückaufové a M. Vlacha „Diskrétní dynamické programování“*, který vyšel v časopisu „*Pokroky matematiky, fyziky a astronomie*“, ročník 12, č. 4 a 5.

V obou citovaných pramenech lze pak též nalézt i bohatý seznam další literatury o dynamickém programování.

ODPOVĚDI A NÁVODY KE CVIČENÍM

2. kapitola

Cvičení 7: Návod: Je-li $\mathbf{M}$ neprázdná množina čísel a c reálné číslo, položme $\mathbf{M}' = \{x \in \mathbf{R} \mid x - c \in \mathbf{M}\}$.

3. kapitola

Cvičení 3: Minimum: — 27; řešení: (1, 0, 0, 4, 1, 0, 0).

Cvičení 4: Sestrojí se tabulka hodnot $\boldsymbol{g}_j(x)$ a určí se max $\{\boldsymbol{g}_n(x) \mid x$ celé, $a' \leqq x \leqq a''\}$.

Cvičení 5: a) **max:** 31; (3, 0, 0, 0, 0);
min: —34; (0, 0, 0, 1, 2)
b) **max:** 246; (0, 0, 0, 0, 7);
min: —40; (0, 1, 3, 1, 2)
c) **max:** 133; (0, 0, 0, 0, 6);
min: —42; (0, 0, 3, 1, 2)

Cvičení 6: Návod: Ukažte, že $\boldsymbol{f}_j(x_1^* + x_2^* + \ldots + x_j^*) = \boldsymbol{g}_1(x_1^*) + \ldots + \boldsymbol{g}_j(x_j^*)$ $(j = 1, 2, \ldots, n)$ a použijte definici $\boldsymbol{P}_j$.

4. kapitola

Cvičení 2: $\binom{n + a}{n}$

5. kapitola

Cvičení 2: $\{5\}$ je CI s maximálním ohodnocením.

Cvičení 4: Triviální: $\frac{n(n-1)}{2}$ sčítání a $\frac{n(n+1)}{2} - 1$ srovnání; netriviální: $n - 1$ sčítání a $2(n - 1)$ srovnání.

6. kapitola

Cvičení 4: $c_j > 0 \Rightarrow x_j = 1$; $c_j < 0 \Rightarrow x_j = 0$

Cvičení 6: Viz návod ke cvičení 6 v kapitole 3.

Cvičení 7: Např.: $(a_1, a_2, a_3) = (5, 4, 2)$; $(c_1, c_2, c_3) = (11, 10, 10)$; $b = 9$

7. kapitola

Cvičení 4: Návod: Jedna část tvrzení vyplývá přímo z věty 18. Druhá část se dokáže sporem. Jestliže $b_{k_i} > b_{k_j}$ a $i < j$, dostaneme transpozicí dvojice b_{k_i}, b_{k_j} větší součet než $a_1 b_{k_1} + \ldots + a_n b_{k_n}$.

Cvičení 6: Nechť v_1, v_2, v_3, v_4 jsou cestovní rychlosti. Použijte cvičení 4 pro $a_1 = \overline{\mathbf{AB}}, \ldots, a_4 = \overline{\mathbf{DE}}, b_1 = v_1^{-1}, \ldots, b_4 = v_4^{-1}$.

9. kapitola

Cvičení 1: Srovnej s příkladem 8 v kapitole 8.

Cvičení 6: Návod: $\varrho(\mathbf{A}, \mathbf{B}) \geqq \varrho(\mathbf{A}, \mathbf{C}) - \varrho(\mathbf{B}, \mathbf{C})$

$$\varrho(\mathbf{A}, \mathbf{B}) = \varrho(\mathbf{B}, \mathbf{A}) \geqq \varrho(\mathbf{B}, \mathbf{C}) - \varrho(\mathbf{A}, \mathbf{C})$$

OBSAH

ŠKOLA MLADÝCH MATEMATIKŮ

JAROSLAV MORÁVEK

o dynamickém programování

Pro účastníky matematické olympiády
vydává ÚV Matematické olympiády
v nakladatelství Mladá fronta
Řídí akademik Josef Novák
Obálku navrhl Jaroslav Příbramský
Odpovědný redaktor Ladislav Smoljak
Publikace číslo 3334
Edice Škola mladých matematiků
svazek 33
Vytiskl Mír, novinářské závody, n. p.
závod 1, Praha 1, Václavské nám. 15
3,21 AA, 3,34 VA. 80 stran
Náklad 5000 výtisků. 1. vydání
Praha 1973. 508/21/8.5

23-097-73 03/2 Cena brož. výt. Kčs 6,—

23 - 097 - 73
03/2
Cena brož.
Kčs 6,—