

FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY  
UNIVERZITA KOMENSKÉHO V BRATISLAVE

# Optimálne navrhovanie blokových experimentov

Dizertačná práca

# Optimálne navrhovanie blokových experimentov

DIZERTAČNÁ PRÁCA

Mgr. Alena Bachratá

Univerzita Komenského v Bratislave  
Fakulta matematiky, fyziky a informatiky  
Katedra aplikovanej matematiky a štatistiky



Štúdijný odbor: 9.1.9 Aplikovaná matematika

Školiteľ:

doc. Mgr. Radoslav Harman, PhD.

Bratislava 2015



Univerzita Komenského v Bratislave  
Fakulta matematiky, fyziky a informatiky

---

## ZADANIE ZÁVEREČNEJ PRÁCE

**Meno a priezvisko študenta:** Mgr. Alena Bachratá  
**Študijný program:** aplikovaná matematika (Jednoodborové štúdium, doktorandské III. st., denná forma)  
**Študijný odbor:** 9.1.9. aplikovaná matematika  
**Typ záverečnej práce:** dizertačná  
**Jazyk záverečnej práce:** slovenský  
**Sekundárny jazyk:** anglický

**Názov:** Optimálne navrhovanie blokových experimentov  
*Optimal design of block experiments*

**Cieľ:** Cieľom práce je vypracovať teoretické a algoritmické metódy optimálneho navrhovania blokových experimentov, so zameraním na efektívnu augmentáciu vopred realizovaných pokusov.

**Školiteľ:** doc. Mgr. Radoslav Harman, PhD.  
**Katedra:** FMFI.KAMŠ - Katedra aplikovanej matematiky a štatistiky  
**Vedúci katedry:** prof. RNDr. Daniel Ševčovič, CSc.

**Spôsob prístupnosti elektronickej verzie práce:**  
bez obmedzenia

**Dátum zadania:** 01.09.2011

**Dátum schválenia:** 01.03.2011

prof. RNDr. Marek Fila, DrSc.  
garant študijného programu

.....  
študent

.....  
školiteľ

### Čestné prehlásenie

Vyhlasujem, že som dizertačnú prácu vypracovala samostatne s použitím uvedenej odbornej literatúry, pod vedením školiteľa.

Bratislava 29. 4. 2015

.....

*podpis*

Názov práce: Optimálne navrhovanie blokových experimentov

Pracovisko: Katedra aplikovanej matematiky a štatistiky, FMFI UK v Bratislave

Autor: Mgr. Alena Bachratá

Vedúci DP: doc. Mgr. Radoslav Harman, PhD.

Dátum: 29. 4. 2015

Kľúčové slová: navrhovanie experimentov, blokové experimenty, optimalizácia, Laplaceove spektrum grafu

Abstrakt: Témou dizertačnej práce je optimálne navrhovanie blokových experimentov. Hľadáme návrh experimentu, ktorý maximalizuje množstvo informácie o neznámych významných parametroch lineárneho regresného modelu. Návrh musí byť z množiny prípustných návrhov, ktoré sú definované pomocou ohraničení na použité zdroje. Zameriame sa na blokové experimenty s veľkosťou bloku dva, ktoré vieme reprezentovať pomocou párovacích grafov.

V práci sú dva typy výsledkov. Prvým sú teoreticky odvodené triedy Schurovsky optimálnych augmentácii návrhov, pri ktorých hľadani sme využili vlastnosti z teórie grafov. Druhým sú stochastické algoritmy, určené na hľadanie optimálnych exaktných návrhov. V závere práce sú výsledky a hypotézy založené na numerických výsledkoch z algoritmov.

Thesis title: Optimal design of block experiments

Department: Department of Applied Mathematics and Statistics, FMPI Comenius University in Bratislava

Author: Mgr. Alena Bachratá

Supervisor: doc. Mgr. Radoslav Harman, PhD.

Date: 29. 4. 2015

Keywords: design of experiment, block design, optimization, Laplacian spectrum of graph

Abstract: The topic of the dissertation thesis is the optimal design of block experiments. We are looking for such design, that maximize amount of information for unknown parameters of interest of linear regression model. The optimal design must be from the set of all feasible designs, which are determined by constraints on resources. We focus on block experiments with the block size two, which can be represented by concurrence graphs.

There are two kinds of results. Firstly, we show some theoretically derived classes of Schur-optimal augmentation of designs, which are found by applying results from graph theory. Secondly, we present three novel stochastic algorithms for finding optimal exact designs. In the end of the thesis, we show numerical results of these algorithms and propose hypotheses based on the results from algorithms.

# Predhovor

Výber témy dizertačnej práce bol jednoduchý. Už v diplomovej práci sme sa venovali navrhovaniu blokových experimentov s veľkosťou bloku dva, pre konkrétnu aplikáciu z praxe. Zistili sme, že táto téma je dostatočne široká a zaujímavá, aby sa dala rozšíriť na dizertačnú prácu. V nej samotnej sa venujeme všeobecnému navrhovaniu blokových experimentov s veľkosťou bloku dva.

Navrhovanie experimentov slúži na optimalizáciu množstva informácie získanej z experimentu, pri daných vstupných podmienkach a ohraničeniach. Potreba optimalizácie množstva získanej informácie vznikla kvôli finančnej a časovej náročnosti niektorých experimentov. Priekopníkom v oblasti navrhovania experimentov je R. A. Fisher (viď. [18]). Túto oblasť štatistiky skúmal počas práce v jednom z najstarších poľnohospodárskych výskumných centier, takže pôvodná myšlienka navrhovania experimentov vznikla na základe riešenia problémov z praxe.

My sa zameriame na blokové návrhy a ich prepojenie s teóriou grafov. Hlavným odrazovým mostíkom sú práce R. A. Baileyovej a P. J. Camerona [5], [7]. Okrem nich sa prepojením navrhovania experimentov a teórie grafov zaoberal aj napríklad T. Tjur v práci [50]. V spomínaných prácach nájdeme viacero prístupov k reprezentovaniu návrhov experimentov pomocou grafov. V teórii grafov sa úlohám, ktoré sa dajú previesť a využiť pre problémy navrhovania experimentov, venovali napr. Kelmans [33], [34], Constantine [9] a iní. Z ich prác sme vychádzali pri dokazovaní nových teoretických výsledkov, ktoré sa nachádzajú v našej práci.

Okrem teoretických výsledkov nás zaujímali aj algoritmy, pomocou ktorých by sme vedeli hľadať optimálne návrhy. Takýchto algoritmov je pomerne veľa a niektoré sú podrobnejšie popísané v práci. Žiaden zo známych algoritmov však nevedel pracovať s väčším množstvom rôznych ohraničení na experiment. Túto medzeru sa nám podarilo zaplniť novým algoritmom, ktorý aj spolu s numerickými výsledkami prezentujeme na konci práce.

Chcela by som sa poďakovať môjmu školiteľovi Radoslavovi Harmanovi za trpezlivosť, ochotu, cenné rady a nápady. Vďaka patrí aj mojej rodine za podporu a pomoc počas celého štúdia.

# Obsah

Úvod	1
<b>1 Základný model</b>	<b>4</b>
1.1 Navrhovanie experimentov . . . . .	4
1.1.1 Lineárny regresný model . . . . .	4
1.1.2 Informačná matica pre subsystém parametrov $\mathbf{P}^T\theta$ . . . . .	5
1.1.3 Kritériá optimality . . . . .	7
1.1.4 Optimálny návrh experimentu . . . . .	9
1.1.5 Ohraničenia na zdroje . . . . .	10
1.1.6 Množina prípustných návrhov . . . . .	12
1.1.7 Ilustračný príklad - nátery . . . . .	13
1.2 Blokové návrhy s veľkosťou bloku dva . . . . .	16
1.2.1 Blokové návrhy s veľkosťou bloku dva a ich reprezentácia grafmi	16
1.2.2 Model a informačná matica pre subsystém významných parametrov	17
1.2.3 Laplacián grafu . . . . .	19
1.2.4 Kritériá optimality pre blokové návrhy . . . . .	20
1.2.5 Kritériá optimality pomocou vlastností grafov . . . . .	25
1.2.6 Aplikácia - mikročipové experimenty . . . . .	27
<b>2 Optimálna augmentácia blokových návrhov s veľkosťou bloku dva</b>	<b>29</b>
2.1 Motivácia . . . . .	29
2.2 Definície a značenie . . . . .	31
2.3 Dve metódy na porovnávanie vlastných čísel grafov . . . . .	33
2.3.1 Laplaceove polynómy a Kelmansova veta . . . . .	34
2.3.2 Majorizácia podľa Constantineho . . . . .	37
2.4 Nové triedy optimálnych augmentovaných návrhov . . . . .	40
2.4.1 Augmentácia kompletného multipartitného grafu I . . . . .	40
2.4.2 Augmentácia hviezdicového grafu . . . . .	44
2.4.3 Augmentácia kompletného grafu . . . . .	47
2.4.4 Augmentácia kompletného multipartitného grafu II . . . . .	50
2.5 Záver . . . . .	54
<b>3 Algoritmy a numerické výsledky</b>	<b>55</b>
3.1 Algoritmy . . . . .	56



---

3.1.1	Prvý algoritmus - simulované žíhanie . . . . .	56
3.1.2	Druhý algoritmus . . . . .	59
3.1.3	Tretí algoritmus . . . . .	62
3.2	Numerické výsledky . . . . .	71
3.2.1	Silno regulárne grafy . . . . .	72
3.2.2	Mooreove grafy . . . . .	73
3.2.3	Jednoduché grafy . . . . .	74
3.2.4	Schurovsky optimálne grafy podľa Constantineho . . . . .	75
3.2.5	Skoro-regulárne kompletne multipartitné grafy . . . . .	76
3.2.6	Porovnanie tretieho algoritmu so simulovaným žíhaním [54] . . .	77
3.2.7	Druhý algoritmus a ohraňenie na počty ošetrení okrem kontrol- ného ošetrenia . . . . .	77
3.3	Záver . . . . .	78
	<b>Záver</b>	<b>80</b>
	<b>Literatúra</b>	<b>81</b>
	<b>Prílohy</b>	<b>86</b>

## Úvod

Hlavným cieľom dizertačnej práce je vypracovať teoretické a algoritmické metódy optimálneho navrhovania blokových experimentov, so zameraním na efektívnu augmentáciu vopred realizovaných pokusov. Vďaka práci s blokovými experimentami s veľkosťou bloku dva existuje prepojenie medzi teóriou navrhovania experimentov a teóriou grafov. Toto prepojenie je vyzerá byť veľmi prínosné a zaujímavé. Na jednu stranu prináša teoretické výsledky z prebádanejšej oblasti teórie grafov, na druhú stranu z teórie navrhovania experimentov prichádzajú do teórie grafov úlohy z praxe.

V prvej časti prvej kapitoly zavedieme klasický lineárny regresný model, na základe ktorého vieme dobre popísať rôzne druhy experimentov a hľadať pre ne optimálne návrhy. Ďalej zavedieme čo znamená exaktný optimálny návrh experimentu. Pre návrhy experimentov zavedieme informačné matice a kritériá optimality, vďaka ktorým vieme jednotlivé návrhy porovnávať. Okrem toho definujeme množinu prípustných návrhov na základe špeciálnych ohraničení na zdroje. Všetky definované pojmy ukážeme na jednoduchom príklade.

V druhej časti prvej kapitoly predstavíme konkrétnu triedu blokových návrhov experimentov s veľkosťou bloku dva, s ktorými budeme pracovať vo zvyšku práce. Tieto návrhy vieme reprezentovať pomocou párovacích grafov a jednotlivé pojmy z prvej časti previesť do pojmov a úloh z teórie grafov. Na záver ukážeme aplikáciu z praxe pri mikročipových experimentoch.

Druhá kapitola obsahuje nami nájdené a dokázané teoretické tvrdenia o optimálnej augmentácii blokových návrhov s veľkosťou bloku dva. Augmentácia experimentu znamená dodatočné dorobenie ďalších pokusov do už vykonaného experimentu, pričom sa opäť snažíme dostať optimálny návrh. Najskôr ukážeme dva známe postupy na prácu s vlastnými číslami Laplaciánov grafov. Potom tieto postupy rozšírime a aplikujeme na naše problémy. Kapitola obsahuje nové výsledky pre augmentáciu návrhov, ktorých párovacie grafy sú regulárne a skoro-regulárne kompletne multipartitné grafy, hviezdicové grafy a kompletne grafy. Celá kapitola je písaná v pojmoch z teórie grafov.

V prvej časti tretej kapitoly predstavíme nami vytvorené tri stochastické algoritmy určené na hľadanie optimálnych grafov. Prvý pracuje na princípe simulovaného žihania. Ďalšie dva prehľadávajú rôznymi spôsobmi množinu všetkých prípustných návrhov.

V druhej časti tretej kapitoly prezentujeme výsledky nájdené pomocou tretieho a v jednom prípade druhého algoritmu. Vďaka týmto výsledkom sa nám podarilo vyvrátiť jednu lemu a niekoľko liem potvrdiť na konkrétnych množinách. Tak isto sme vyslovili niekoľko nových liem, ktoré na základe numerických výsledkov vyzerajú byť zaujímavé a vhodné na ďalšie skúmanie.

## Pomocné vety, definície a označenia

### Pozitívne definitné matice a Loewnerovo usporiadanie:

V práci budeme pracovať s pozitívne semidefinitnými a pozitívne definitnými maticami. Ukážeme niekoľko základných vlastností takýchto matíc a vzťahy medzi nimi.

**Definícia 1.** *Nech  $\mathbf{A}$  je symetrická matica rozmeru  $k \times k$ . Nazveme ju pozitívne semidefinitnou, ak platí*

$$x^T \mathbf{A} x \geq 0, \quad \forall x \in \mathbb{R}^k,$$

*a pozitívne definitnou, ak platí*

$$x^T \mathbf{A} x > 0, \quad \forall x \in \mathbb{R}^k; x \neq 0.$$

*Množinu všetkých pozitívne semidefinitných matíc (pozitívne definitných matíc) rozmeru  $k \times k$  budeme označovať  $\mathcal{S}_+^k$  ( $\mathcal{S}_{++}^k$ ).*

**Lema 0.1.** *Nech  $\mathbf{A}$  je symetrická matica rozmeru  $k \times k$ . Potom všetky vlastné čísla  $\mathbf{A}$  sú reálne. Nech  $\lambda_{\min}(\mathbf{A})$  je najmenšie vlastné číslo matice  $\mathbf{A}$ . Potom platí*

$$\mathbf{A} \in \mathcal{S}_+^k \Leftrightarrow \lambda_{\min}(\mathbf{A}) \geq 0$$

$$\mathbf{A} \in \mathcal{S}_{++}^k \Leftrightarrow \lambda_{\min}(\mathbf{A}) > 0.$$

**Definícia 2.** *Nech  $\mathbf{A}_{k \times k}$ ,  $\mathbf{B}_{k \times k}$  sú symetrické matice. Potom usporiadanie matíc  $\succeq$  definované ako*

$$\mathbf{A} \succeq \mathbf{B} \quad \Leftrightarrow \quad \mathbf{A} - \mathbf{B} \succeq 0 \quad \Leftrightarrow \quad \mathbf{A} - \mathbf{B} \in \mathcal{S}_+^k$$

*budeme nazývať Loewnerovo usporiadanie.*

**Lema 0.2.** *Vlastnosti Loewnerovho usporiadania*

$$\begin{array}{ll}
 \delta \mathbf{A} \succeq 0 & \forall \delta > 0, \forall \mathbf{A} \succeq 0 \\
 \mathbf{A} + \mathbf{B} \succeq 0 & \forall \mathbf{A}, \mathbf{B} \succeq 0 \\
 \mathbf{A} \succeq \mathbf{B} \text{ a } \mathbf{B} \succeq \mathbf{A} \Rightarrow \mathbf{A} = \mathbf{B} & \forall \mathbf{A}, \mathbf{B} \text{ symetrické} \\
 \mathbf{A} \succeq \mathbf{A} & \forall \mathbf{A} \text{ symetrické} \\
 \mathbf{A} \succeq \mathbf{B} \text{ a } \mathbf{B} \succeq \mathbf{C} \Rightarrow \mathbf{A} \succeq \mathbf{C} & \forall \mathbf{A}, \mathbf{B} \text{ symetrické}
 \end{array}$$

Posledné tri vlastnosti sú postupne *antisymetrickosť*, *reflexivita* a *tranzitívnosť*, takže Loewnerovo usporiadanie je *čiasťočné usporiadanie*. Dôkazy uvedených lemm sú v práci [45].

### Pojmy z teórie grafov:

Graf  $G = G(n, e)$  definujeme ako množinu vrcholov  $V(G)$  (kde  $|V(G)| = n$ ), množinu hrán  $E(G)$  (kde  $|E(G)| = e$ ) a reláciu incidencie medzi. Okrem označenia  $G = G(n, e)$  budeme používať značenie  $G_n^e$ , resp. len  $G_n$  alebo  $G^k$ .

V práci budeme používať konečné neorientované grafy. Predpokladáme, že grafy sú bez slučiek (hrán spájajúcich vrchol sám so sebou) a väčšinou budeme pracovať s takzvanými *jednoduchými grafmi*, ktoré okrem slučiek neobsahujú ani násobné hrany (viac hrán medzi rovnakou dvojicou vrcholov).

Potrebné základné pojmy z teórie grafov (strom, kostra, komplementárny graf, izomorfný graf, susedné hrany, stupeň vrchola, komponenta súvislosti a ďalšie) sa dajú nájsť v práci [40]. Definujeme niektoré menej bežné pojmy, ktoré sú v práci dôležité.

**Definícia 3** (Kompletný graf). *Kompletný graf  $K_n$  na  $n$  vrchoch je jednoduchý graf (bez slučiek a násobných hrán), v ktorom sú každé dva rôzne vrcholy spojené hranou.*

**Definícia 4** (Bipartitný a multipartitný graf). *Bipartitný (multipartitný) graf  $K_{m,n}$  ( $K_{n_1, n_2, \dots, n_k}$ ) je graf ktorého vrcholy vieme rozdeliť do dvoch ( $k$ ) partícií, pričom hrany sa nachádzajú iba medzi vrcholmi z rôznych partícií.*

**Definícia 5** (Regulárny graf). *Graf nazývame regulárny ak sú stupne všetkých jeho vrcholov rovnaké.*

**Definícia 6** (Skoro-regulárny graf). *Graf nazývame skoro-regulárny ak sa stupne ľubovoľných dvoch jeho vrcholov líšia maximálne o jedna.*

# 1 Základný model

Prvá kapitola pozostáva z dvoch častí. V prvej zavedieme potrebné pojmy z teórie navrhovania experimentov. V druhej ukážeme prepojenie blokových experimentov a teórie grafov.

## 1.1 Navrhovanie experimentov

Cieľom práce je hľadať exaktné optimálne návrhy experimentov, pomocou ktorých vieme najlepšie odhadnúť neznáme parametre vystupujúce v modeli. Aby sme mohli hľadať optimálny návrh pre konkrétny experiment, potrebujeme zaviesť niekoľko pojmov. Ako prvé potrebujeme vedieť s akým modelom experimentu budeme pracovať. Ďalšia otázka je, či chceme odhadovať všetky neznáme parametre modelu, alebo len nejaký lineárny subsystém významných parametrov. V oboch prípadoch potrebujeme zaviesť informačné matice pre návrhy experimentov.

Ďalej definujeme kritériá optimality, pomocou ktorých budeme vedieť porovnávať rôzne informačné matice. Nakoniec uvedieme čo znamená optimálny návrh experimentu a množinu prípustných návrhov. Ukážeme ako môžu množinu prípustných návrhov ovplyvňovať rôzne materiálne a technické ohraničenia na experiment. Nakoniec všetky zavedené pojmy demonštrujeme na jednoduchom príklade.

### 1.1.1 Lineárny regresný model

V celej práci budeme pracovať s klasickým lineárnym regresným modelom. V niektorých častiach (najmä o algoritmoch) je možné použiť aj komplikovanejšie modely, ale tými sa nebudeme zaoberať.

Majme experiment pozostávajúci z  $N$  pokusov (meraní, pozorovaní), definovaný pomocou lineárneho regresného modelu

$$Y_i = \mathbf{f}^T(\tilde{x}_i)\theta + \varepsilon_i, \quad i = 1, \dots, N. \quad (1.1)$$

Jednotlivé body návrhu experimentu  $\tilde{x}_1, \dots, \tilde{x}_N$  sú prvky z množiny  $\mathfrak{X}$ , ktorá reprezentuje všetky možné body návrhu experimentu. Označme počet prvkov tejto množiny  $|\mathfrak{X}| = n$ . V každom bode návrhu môžeme vykonať viacero pokusov, teda niektoré body návrhu sa môžu opakovať. Všetky možné body návrhu experimentu usporiadajme a očísľujme  $x_1, \dots, x_n$ . Potom množina všetkých bodov návrhu je  $\mathfrak{X} = \{x_1, \dots, x_n\}$ . Teda  $\tilde{x}_i \in \mathfrak{X}$  je v poradí  $i$ -ty pokus experimentu,  $i = 1, \dots, N$

Ďalej hodnoty  $Y_1, \dots, Y_N$  sú pozorovania v jednotlivých bodoch  $\tilde{x}_1, \dots, \tilde{x}_N$ . Funkcia  $\mathbf{f} : \mathfrak{X} \rightarrow \mathbb{R}^m$  je funkcia priradujúca regresor  $\mathbf{f}(\tilde{x}_i)$  bodu  $\tilde{x}_i \in \mathfrak{X}$ ,  $\theta \in \mathbb{R}^m$  je vektor neznámych parametrov modelu a  $\varepsilon_1, \dots, \varepsilon_N$  sú nezávislé, rovnako rozdelené chyby s nulovou strednou hodnotou a konečnou nenulovou varianciou  $\sigma^2$ .

Maticový zápis základného modelu (1.1) je nasledovný

$$\mathbf{Y} = \mathbf{F}\theta + \varepsilon, \quad (1.2)$$

kde  $\mathbf{Y} = (Y_1, \dots, Y_N)^T$  je vektor pozorovaní,  $\mathbf{F} = (\mathbf{f}(\tilde{x}_1), \dots, \mathbf{f}(\tilde{x}_N))^T$  je matica plánu typu  $N \times m$ ,  $\theta \in \mathbb{R}^m$  je stĺpcový vektor neznámych parametrov modelu a  $\varepsilon = (\varepsilon_1, \dots, \varepsilon_N)^T$  je vektor chýb so  $E(\varepsilon) = \mathbf{0}_N$  a  $D(\varepsilon) = \sigma^2 \mathbf{I}_N$ . Pre model platí  $E(\mathbf{Y}) = \mathbf{F}\theta$  a  $D(\mathbf{Y}) = \sigma^2 \mathbf{I}_N$ .

### 1.1.2 Informačná matica pre subsystém parametrov $\mathbf{P}^T\theta$

V nasledujúcej časti čerpáme z knihy [45]. V experimentoch používame dva druhy parametrov. Prvé, ktorých efekty nás pri odhadovaní zaujímajú, nazývame *významné parametre* (z ang. *parameters of interest*, označíme ich  $\tau$ ). Druhé, ktoré nechceme odhadovať, nazývame *pomocné parametre* (z ang. *nuisance parameters*, označíme ich  $\beta$ ). *Pomocné parametre* sú dôležité na definovanie štatistického modelu, aby čo najvernejšie popisoval skutočnosť. Pri hľadaní optimálneho návrhu experimentu pre daný model potrebujeme maximalizovať informáciu o *významných parametroch*, prípadne o nejakej ich lineárnej kombinácii.

Lineárny podsystém *významných parametrov* definujeme pomocou  $\mathbf{P}^T\theta$ , kde  $\mathbf{P}$  je matica koeficientov typu  $m \times s$ ,  $m \geq s$ , ktorá má plnú hodnotu, a  $s$  je počet lineárnych kombinácií parametrov ktoré odhadujeme. Keď chceme odhadovať všetky neznáme parametre modelu, tak volíme  $\mathbf{P} = \mathbf{I}_m$  ( $s = m$ ). Keď chceme odhadovať len jeden konkrétny parameter  $\theta_i$ , tak volíme  $\mathbf{P} = \mathbf{e}_i$  ( $s = 1$ ). Cieľom práce je nájsť návrh experimentu, ktorý vedie k najlepšiemu lineárnemu nevychýlenému odhadu lineárneho subsystému parametrov  $\mathbf{P}^T\theta$ .

Použijeme klasický model v tvare maticového zápisu (1.2). Hľadáme najlepší nevychýlený lineárny odhad pre  $\mathbf{P}^T\theta$ . Subsystém  $\mathbf{P}^T\theta$  nazývame *odhadnuteľný*, práve vtedy keď existuje aspoň jedna matica  $\mathbf{U}$  typu  $N \times m$ , pre ktorú platí

$$E(\mathbf{U}^T\mathbf{Y}) = \mathbf{P}^T\theta, \quad \text{pre } \forall\theta.$$

Úpravou dostaneme

$$\mathbf{U}^T \mathbf{F} \theta = \mathbf{P}^T \theta, \quad \text{pre } \forall \theta,$$

$$\mathbf{F}^T \mathbf{U} = \mathbf{P}.$$

Takže, aby bol subsystém  $\mathbf{P}^T \theta$  odhadnuteľný, potrebujeme aby  $\mathcal{M}(\mathbf{P}) \subset \mathcal{M}(\mathbf{F}^T)$ . Okrem toho vieme, že  $\mathcal{M}(\mathbf{F}^T) = \mathcal{M}(\mathbf{F}^T \mathbf{F})$ , kde  $\mathbf{F}^T \mathbf{F} = \mathbf{M}$  je *momentová matica* experimentu. Takže podmienku odhadnuteľnosti budeme používať v tvare

$$\mathcal{M}(\mathbf{P}) \subseteq \mathcal{M}(\mathbf{M}). \quad (1.3)$$

Ak je splnená podmienka odhadnuteľnosti (1.3), tak najlepší lineárny nevychýlený odhad subsystému  $\mathbf{P}^T \theta$  je

$$\hat{\theta}_{\mathbf{P}} = \mathbf{P}^T \mathbf{M}^{-1} \mathbf{F}^T \mathbf{Y}, \quad (1.4)$$

ktorý navyše nezávisí od voľby pseudoinverzie matice  $\mathbf{M}$  a kovariančná matica  $D(\hat{\theta}_{\mathbf{P}}) = \sigma^2 \mathbf{P}^T \mathbf{M}^{-1} \mathbf{P}$  je regulárna [45].

**Definícia 7.** *Nech matica koeficientov  $\mathbf{P}$  je typu  $m \times s$  ( $m \geq s$ ), má plnú hodnotu  $s$  a nech  $\mathbf{M}$  je momentová matica experimentu (teda je pozitívne semidefinítaná) typu  $m \times m$ . Potom **informačná matica** pre subsystém  $\mathbf{P}^T \theta$  je*

$$\mathbf{C}_{\mathbf{P}}(\mathbf{M}) = \min_{\mathbf{L} \in \mathbb{R}^{s \times m}: \mathbf{L}\mathbf{P} = \mathbf{I}_s} \mathbf{L}\mathbf{M}\mathbf{L}^T. \quad (1.5)$$

Minimalizácia je vzhľadom na Loewnerovo usporiadanie matíc. Toto usporiadanie je iba čiastočné usporiadanie, ale napriek tomu existuje minimum uvedené v definícii. Jeho existenciu zaručuje Gauss-Markovova veta ([45], sekcia 1.21), ktorá navyše ponúka reprezentáciu *informačnej matice* v tvare

$$\mathbf{C}_{\mathbf{P}}(\mathbf{M}) = \mathbf{L}\mathbf{M}\mathbf{L}^T - \mathbf{L}\mathbf{M}\mathbf{R}^T(\mathbf{R}\mathbf{M}\mathbf{R}^T)^{-1}\mathbf{R}\mathbf{M}\mathbf{L}^T, \quad (1.6)$$

kde  $\mathbf{L}$  je ľubovoľná ľavá inverzia matice  $\mathbf{P}$  a  $\mathbf{R}$  je projektor  $\mathbf{R} = \mathbf{I}_m - \mathbf{P}\mathbf{L}$ . Tento zápis využijeme neskôr. Okrem toho vďaka Gauss-Markovovej vete platí, že keď je splnená podmienka odhadnuteľnosti (1.3), tak je informačná matica regulárna a má tvar

$$\mathbf{C}_{\mathbf{P}}(\mathbf{M}) = (\mathbf{P}^T \mathbf{M}^{-1} \mathbf{P})^{-1}. \quad (1.7)$$

Informačnou maticou  $\mathbf{C}_{\mathbf{P}}(\mathbf{M})$  môžeme reprezentovať množstvo informácie, ktorú poskytuje experiment o lineárnom subsystéme parametrov  $\mathbf{P}^T \theta$ . Matica  $\mathbf{C}_{\mathbf{P}}(\mathbf{M})$  je typu  $s \times s$  a  $\mathbf{C}_{\mathbf{P}}(\mathbf{M}) \in \mathcal{S}_+^s$ .

### Subsystém významných parametrov:

Ukázali sme tvar informačnej matice pre ľubovoľný vhodný subsystém parametrov  $\mathbf{P}^T\theta$ . Pozrime sa teraz na tvar matice v prípade, že chceme pracovať len s *významnými parametrami*. Rozdeľme vektor parametrov  $\theta = (\theta_1, \dots, \theta_m)'$  na  $v$  významných parametrov  $\tau_1, \dots, \tau_v$  a  $b$  pomocných parametrov  $\beta_1, \dots, \beta_b$ , teda  $m = v + b$ . Chceme nájsť informačnú maticu pre významné parametre. Zvolíme maticu koeficientov  $\mathbf{P}$  subsystému parametrov  $\mathbf{P}^T\theta$  nasledovne  $\mathbf{P} = (\mathbf{I}_v, \mathbf{0})^T$ . Vyjadríme informačnú maticu  $\mathbf{C}_P(\mathbf{M})$  pomocou tvaru (1.6). Na to potrebujeme zľava inverznú maticu  $\mathbf{L}$  k matici  $\mathbf{P}$  a na základe nej vyjadriť projektor  $\mathbf{R} = \mathbf{I}_m - \mathbf{P}\mathbf{L}$ . Zvoľme

$$\mathbf{L} = (\mathbf{I}_v, \mathbf{0}) \quad \text{a} \quad \mathbf{R} = \mathbf{I}_m - \begin{pmatrix} \mathbf{I}_v & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_b \end{pmatrix}.$$

Označme podmatice matice  $\mathbf{M}$  nasledovne

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{pmatrix},$$

pričom  $\mathbf{M}_{11}$  je typu  $v \times v$ ,  $\mathbf{M}_{12}$  a  $\mathbf{M}_{21}^T$  sú typu  $v \times b$  a  $\mathbf{M}_{22}$  je typu  $b \times b$ .

Potom *informačná matica* je

$$\mathbf{C}_P(\mathbf{M}) = \mathbf{M}_{11} - (\mathbf{0}, \mathbf{M}_{12}) \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{22} \end{pmatrix}^{-} \begin{pmatrix} \mathbf{0} \\ \mathbf{M}_{21} \end{pmatrix} = \mathbf{M}_{11} - \mathbf{M}_{12}\mathbf{M}_{22}^{-}\mathbf{M}_{21}. \quad (1.8)$$

Matica  $\mathbf{M}_{11} - \mathbf{M}_{12}\mathbf{M}_{22}^{-}\mathbf{M}_{21}$  sa volá *Schurov doplnok* podmatice  $\mathbf{M}_{22}$  matice  $\mathbf{M}$  a je pozitívne semidefinitná práve vtedy ak je  $\mathbf{M}$  pozitívne semidefinitná,  $\mathbf{M}_{22}$  je pozitívne semidefinitná a  $\mathcal{M}(\mathbf{M}_{21}) \subseteq \mathcal{M}(\mathbf{M}_{22})$ . Dôkaz je v knihe [45], kapitola 3. Takto vyjadrenú informačnú maticu budeme používať vo zvyšku práce. Pripomenieme, že vďaka všeobecným tvrdeniam pre informačnú maticu subsystému parametrov vieme, že  $\mathbf{C}_P(\mathbf{M}) \in \mathcal{S}_+^v$ .

#### 1.1.3 Kritériá optimality

Keď sú definované informačné matice, tak potrebujeme pridať mieru na ich porovnanie a určovanie množstva informácie z nich získanej. Na to slúžia kritériá optimality definované ako  $\phi : \mathcal{S}_+^v \rightarrow [0, \infty)$ , kde  $v$  je rozmer informačnej matice a  $\mathcal{S}_+^v$  je množina pozitívne semidefinitných matíc. Často používaná trieda kritérií sú ortogonálne invari-



antné kritériá optimality (viď. [22, 45]).

**Definícia 8.** *Nech  $\mathcal{S}_+^v$  je množina všetkých pozitívne semidefinitných matíc rozmeru  $v \times v$ . Potom kritérium  $\phi : \mathcal{S}_+^v \rightarrow [0, \infty)$  spĺňajúce nasledovné vlastnosti nazývame ortogonálne invariantné informačné kritérium.*

1. *Loewnerovská izotónnosť - pre každé  $\mathbf{C}_1, \mathbf{C}_2 \in \mathcal{S}_+^v$  platí*

$$\mathbf{C}_1 \preceq \mathbf{C}_2 \Rightarrow \Phi(\mathbf{C}_1) \leq \Phi(\mathbf{C}_2),$$

*kde symbol  $\preceq$  reprezentuje Loewnerovské usporiadanie matíc.*

2. *Konkávnosť - pre každé  $\mathbf{C}_1, \mathbf{C}_2 \in \mathcal{S}_+^v$  a konštantu  $\alpha \in (0, 1)$  platí*

$$\Phi(\alpha \mathbf{C}_1 + (1 - \alpha) \mathbf{C}_2) \geq \alpha \Phi(\mathbf{C}_1) + (1 - \alpha) \Phi(\mathbf{C}_2).$$

3. *Pozitívna homogenita - pre každú maticu  $\mathbf{C} \in \mathcal{S}_+^v$  a konštantu  $\alpha \geq 0$  platí*

$$\Phi(\alpha \mathbf{C}) = \alpha \Phi(\mathbf{C}).$$

4. *Polospojitosť zhora - množiny  $\{\mathbf{C} \in \mathcal{S}_+^v; \Phi(\mathbf{C}) \geq b\}$  sú uzavreté pre každé  $b \in \mathbb{R}$ .*

5. *Ortogonálna invariancia - existuje funkcia  $\Phi : \mathbb{R}^v \rightarrow \mathbb{R}$  taká, že pre každú maticu  $\mathbf{C} \in \mathcal{S}_+^v$  platí  $\Phi(\mathbf{C}) = \tilde{\Phi}(\boldsymbol{\lambda}(\mathbf{C}))$ , kde  $\boldsymbol{\lambda}(\mathbf{C}) = (\lambda_1(\mathbf{C}), \dots, \lambda_v(\mathbf{C}))'$  je usporiadaný nerastúci vektor vlastných hodnôt matice  $\mathbf{C}$ .*

Príkladom ortogonálne invariantných kritérií optimality sú kritériá  $D$ ,  $A$  a  $E$ . Nás zaujímajú odhady subsystému parametrov  $\mathbf{P}^T \boldsymbol{\theta}$ , takže nás zaujímajú ich parciálne verzie  $D_{\mathbf{P}}$ ,  $A_{\mathbf{P}}$  a  $E_{\mathbf{P}}$ . V definíciách predpokladáme, že  $\mathbf{M}$  je momentová matica rozmeru  $m \times m$  a  $\mathbf{C}_{\mathbf{P}}(\mathbf{M})$  je informačná matica pre subsystém významných parametrov  $\mathbf{P}^T \boldsymbol{\theta}$  rozmeru  $v \times v$ .

**Definícia 9. Kritérium  $D_{\mathbf{P}}$ -optimality  $\Phi_{D_{\mathbf{P}}}(\cdot)$ .** *Kritérium  $\Phi_{D_{\mathbf{P}}} : \mathcal{S}_+^m \rightarrow \langle 0, \infty \rangle$  je zobrazenie definované*

$$\Phi_{D_{\mathbf{P}}}(\mathbf{M}) = (\det(\mathbf{C}_{\mathbf{P}}(\mathbf{M})))^{1/v} = (\det(\mathbf{P}^T \mathbf{M}^{-1} \mathbf{P}))^{-1/v},$$

*pre momentové matice  $\mathbf{M} \in \mathcal{S}_+^m$  spĺňajúce podm. odhadnutelnosti  $\mathcal{M}(\mathbf{P}) \subseteq \mathcal{M}(\mathbf{M})$ .*

*Pre ostatné momentové matice bude  $\Phi_{D_{\mathbf{P}}}(\mathbf{M}) = 0$ .*

**Definícia 10. Kritérium  $A_{\mathbf{P}}$ -optimality  $\Phi_{A_{\mathbf{P}}}(\cdot)$ .** *Kritérium  $\Phi_{A_{\mathbf{P}}} : \mathcal{S}_+^m \rightarrow \langle 0, \infty \rangle$  je zobrazenie definované*

$$\Phi_{A_{\mathbf{P}}}(\mathbf{M}) = \left( \frac{1}{v} \text{trace } \mathbf{C}_{\mathbf{P}}(\mathbf{M})^{-1} \right)^{-1} = \left( \frac{1}{v} \text{trace } \mathbf{P}^T \mathbf{M}^{-1} \mathbf{P} \right)^{-1} = \left( \frac{1}{v \sigma^2} \sum_{i=1}^v \text{Var}(\widehat{\mathbf{P}^T \boldsymbol{\theta}}_i) \right)^{-1},$$

pre momentové matice  $\mathbf{M} \in \mathcal{S}_+^m$  splňajúce podm. odhadnuteľnosti  $\mathcal{M}(\mathbf{P}) \subseteq \mathcal{M}(\mathbf{M})$ . Pre ostatné momentové matice bude  $\Phi_{A_{\mathbf{P}}}(\mathbf{M}) = 0$ .

**Definícia 11. Kritérium  $E_{\mathbf{P}}$ -optimality  $\Phi_{E_{\mathbf{P}}}(\cdot)$ .** Označme  $\lambda_{\min}(\mathbf{C}_{\mathbf{P}}(\mathbf{M}))$  najmenšie vlastné číslo informačnej matice. Kritérium  $\Phi_{E_{\mathbf{P}}} : \mathcal{S}_+^m \rightarrow \langle 0, \infty \rangle$  je zobrazenie definované

$$\Phi_{E_{\mathbf{P}}}(\mathbf{M}) = \lambda_{\min}(\mathbf{C}_{\mathbf{P}}(\mathbf{M})) = \lambda_{\min}((\mathbf{P}^T \mathbf{M}^{-1} \mathbf{P})^{-1}),$$

pre momentové matice  $\mathbf{M} \in \mathcal{S}_+^m$  splňajúce podm. odhadnuteľnosti  $\mathcal{M}(\mathbf{P}) \subseteq \mathcal{M}(\mathbf{M})$ . Pre ostatné momentové matice bude  $\Phi_{E_{\mathbf{P}}}(\mathbf{M}) = 0$ .

#### 1.1.4 Optimálny návrh experimentu

Za *exaktný návrh experimentu* budeme považovať vektor  $\xi = (\xi_1, \dots, \xi_n)'$ , pričom jednotlivé komponenty  $\xi_i \in \{0, 1, 2, \dots\}$  predstavujú počty pokusov vykonaných v bodoch návrhu experimentu  $x_1, \dots, x_n$ . Občas, kvôli zrozumiteľnosti, budeme používať zápis  $\xi_i = \xi(x_i)$ .

Rovnako môžeme definovať aproximatívny návrh  $\xi \in [0, \infty)^n$ , s tým rozdielom, že nepracujeme s celými číslami, ale  $\xi_i$  môžu byť ľubovoľné nezáporné reálne čísla. V práci sa budeme zaoberať exaktnými návrhmi.

*Momentová matica* nezávisí na poradí jednotlivých pokusov vykonávaných v bodoch návrhu experimentu. Preto ju pre konkrétny návrh experimentu môžeme definovať pomocou usporiadanej množiny bodov návrhu  $\mathfrak{X} = \{x_1, \dots, x_n\}$  a ich násobností, namiesto všetkých použitých bodov návrhu  $\tilde{x}_1, \dots, \tilde{x}_N$ .

**Definícia 12. Momentová matica** prislúchajúca návrhu experimentu  $\xi$  je matica typu  $m \times m$

$$\mathbf{M}(\xi) = \sum_{x=1}^n \xi_i \mathbf{f}(x_i) \mathbf{f}^T(x_i). \quad (1.9)$$

Dá sa ľahko nahliadnuť, že *momentová matica* je *informačná matica* pre celý systém parametrov  $\theta$ . Stačí za *maticu koeficientov* zvoliť  $\mathbf{P} = \mathbf{I}_m$ .

Ak  $\mathbf{M}(\xi)$  spĺňa podmienku odhadnuteľnosti (1.3), tak je subsystém  $\mathbf{P}^T \theta$  *odhadnuteľný* pre návrh experimentu  $\xi$ . V práci budeme pracovať len s návrhmi, pre ktoré je subsystém  $\mathbf{P}^T \theta$  *odhadnuteľný*.

**Definícia 13 (Optimálny návrh).** *Hovoríme, že návrh experimentu  $\xi^*$  je optimálny vzhľadom na kritérium  $\Phi$ , ak je návrh  $\xi^*$  prípustný a kritérium  $\Phi$  nadobúda pre tento*

návrh maximum spomedzi všetkých prípustných návrhov  $\xi \in \Xi$

$$\xi^* = \operatorname{argmax}_{\xi \in \Xi} \Phi(\mathbf{M}(\xi)). \quad (1.10)$$

Čo rozumieme pod množinou prípustných návrhov experimentu špecifikujeme v nasledovnej časti 1.1.5.

**Definícia 14** (Schurovsky optimálny návrh). *Návrh, ktorý je optimálny vzhľadom na množinu všetkých ortogonálne invariantných kritérií optimality, nazývame Schurovsky optimálny návrh.*

Keď máme definovaný návrh experimentu, tak vyslovíme ešte jednu podmienku pre kritériá optimality  $\Phi$  - podmienku monotónnosti. Mohli by sme ju vysloviť skôr, ale teraz máme k dispozícii lepší slovník.

(M) Rozšírenie experimentu o ďalšie pokusy nemôže znížiť množstvo informácie o parametroch. Teda ak pre dva návrhy  $\xi$  a  $\zeta$  platí  $\xi \leq \zeta$  po zložkách, potom  $\phi(\mathbf{M}(\xi)) \leq \phi(\mathbf{M}(\zeta))$ .

Táto podmienka je intuitívna (ak pribudne pokus, tak sa nemôže znížiť získaná informácia) a platí skoro pre akékoľvek rozumné kritériá. Ortogonálne invariantné kritériá ju spĺňajú vďaka vlastnosti *Loewnerovskej izotónnosti*.

### 1.1.5 Ohraničenia na zdroje

Pri hľadaní optimálneho návrhu experimentu potrebujeme poznať množinu všetkých prípustných návrhov  $\Xi$  s ktorými pracujeme. Aby sme ju mohli definovať musíme poznať vstupné obmedzenia experimentu. Ak by neboli žiadne obmedzenia, tak nemusíme optimalizovať. Urobíme neobmedzene veľa pokusov vo všetkých bodoch návrhu a nájdeme dostatočne dobrý odhad. V praxi však pracujeme s časovo, finančne alebo inak náročnými experimentmi, ktorých môžeme urobiť len obmedzený počet (napríklad keď je zadaný celkový rozpočet, ktorý nemôžeme presiahnuť).

Najbežnejšie ohraničenie na množinu prípustných návrhov  $\Xi$  je obmedzenie na celkový počet rovnocenných pokusov

$$\xi_1 + \dots + \xi_n \leq N. \quad (1.11)$$

Bežnejšie sa toto ohraničenie používa v tvare rovnosti. Ak však predpokladáme *monotónnosť* kritérií optimality, tak existuje optimálny návrh, v ktorom sa dosahuje rovnosť. Na

riešenie úlohy so štandardným ohraničením 1.11 existuje viacero algoritmov a heuristík (budú ukázané neskôr). V skutočnosti sa však často vyskytujú komplikovanejšie ohraničenia na zdroje (materiálne, ľudské, finančné, časové), alebo viacero ohraničení súčasne, s ktorými klasické algoritmy nevedia pracovať.

Skoro všetky bežné podmienky vyskytujúce sa v praxi vieme zapísať v tvare *k* ohraničení na zdroje zapísateľných ako

$$\sum_{i=1}^n a_{ri}\xi_i \leq b_r \text{ pre všetky } r \in \{1, \dots, k\}; \quad (1.12)$$

kde  $a_{ri}$  predstavuje množstvo čerpania z  $r$ -tého zdroja pri vykonaní jedného pokusu v  $i$ -tom bode návrhu a  $b_r$  predstavuje limit na  $r$ -tý zdroj. Budeme používať aj skrátaný maticový zápis  $\mathbf{A}\xi \leq \mathbf{b}$ . K takto definovaným ohraničeniam (1.12) pridáme ešte tri prirodzené podmienky

- (P1) limity na zdroje sú konečné kladné čísla, tj.  $b_1, \dots, b_k \in (0, \infty)$ ,
- (P2) rozšírenie návrhu nemôže znížiť množstvo čerpania pre žiaden zdroj, tj.  $a_{ri} \geq 0$  pre všetky  $r \in \{1, \dots, k\}$  a  $i \in \{1, \dots, n\}$ ,
- (P3) žiaden pokus nie je úplne zadarmo a jeho opakovanie musí viesť k porušeniu niektorého ohraničenia, tj. pre všetky  $i \in \{1, \dots, n\}$  existuje zdroj  $r \in \{1, \dots, k\}$  taký, že  $a_{ri} > 0$ .

Ukážeme aké podmienky z praxe vieme zapísať v tvare *ohraničení na zdroje* (1.12) a ako pre ne volíme jednotlivé parametre.

- Štandardné ohraničenie (viď. 1.11) na celkový počet pokusov  $N$  pri predpoklade jednotkovej ceny za pokus v hociktorom bode návrhu  $x_1, \dots, x_n$

$$a_{1i} = 1, \forall i \in \{1, \dots, n\}; \quad b_1 = N.$$

- Ohraničenia na počty pokusov  $N_1, \dots, N_n$  vykonaných v jednotlivých bodoch návrhu  $x_1, \dots, x_n$

$$a_{ri} = 1 \text{ pre } i = r, \quad a_{rj} = 0 \text{ pre } j \neq r; \quad b_r = N_r; \quad r = 1, \dots, n.$$

Tieto ohraničenia sa nazývajú priame (z ang. *direct constraints*) a používajú sa

napríklad v článkoch [16], [53]. Často bývajú rovné maximálne jednému pokusu v každom bode návrhu  $N_1 = \dots = N_n = 1$ .

- Ohraničenie na celkový *rozpočet* experimentu, pričom ceny pokusov v jednotlivých bodoch návrhu  $c_1, \dots, c_n$  môžu byť rôzne

$$a_{11} = c_1, \dots, a_{1n} = c_n; \quad b_1 = \text{rozpočet.}$$

- Marginálne ohraničenia na pokusy v určitých disjunktných skupinách bodov návrhov  $\mathfrak{X}_1, \dots, \mathfrak{X}_k$ . Pričom návrhy z jednotlivých skupín mŕňajú rôzne zdroje, ktorých množstvá sú  $z_1, \dots, z_k$

$$\sum_{x \in \mathfrak{X}_r} \xi(x) \leq z_r; \quad r = 1, \dots, k.$$

Použitie marginálnych ohraničení sa dá nájsť v článkoch [13], [39]) alebo všeobecnejšie v ([24]).

- Mnohé ďalšie typy. Napríklad ak nemôžeme vykonať dva pokusy vo veľmi podobných bodoch návrhu. Potom po vykonaní jedného pokusu, musíme zakázať vykonanie ďalších pokusov v podobných bodoch návrhu.

*Ohraničenia na zdroje* definované nerovnosťami tvaru (1.12) a podmienkami (P1), (P2) a (P3), nepokrývajú všetky možné používané ohraničenia pri navrhovaní experimentov (napríklad [49]), ale zahŕňajú väčšinu v praxi používaných ohraničení. Navyše vytvárajú peknú množinu prípustných riešení s ktorou vieme dobre pracovať v algoritmoch. Vďaka tvaru ohraničení vieme prejsť z ľubovoľného prípustného návrhu do ľubovoľného iného prípustného návrhu, len po prípustných návrhoch vznikajúcich z predošlých len pridaním alebo odobraním jedného pokusu.

### 1.1.6 Množina prípustných návrhov

Okrem *ohraničení na zdroje* (1.12) majme exaktný návrh  $\xi^{(0)}$  predstavujúci prípadné vopred vykonané pokusy (ku ktorým chceme dorobiť ďalšie pozorovania). Ak robíme normálny experiment, bez žiadnych vopred vykonaných pokusov, tak  $\xi^{(0)} = \mathbf{0}_n$  je nulový počiatočný návrh. Samozrejme predpokladáme, že  $\xi^{(0)}$  spĺňa všetky *ohraničenia na zdroje*.

Potom na základe návrhu  $\xi^{(0)}$  a *ohraničení na zdroje* (1.12) definujeme množinu všetkých prípustných exaktných návrhov  $\Xi^{\text{ex}}$  ako

$$\Xi^{\text{ex}} = \{\xi \in \{0, 1, 2, \dots\}^n : \xi^{(0)} \leq \xi, \mathbf{A}\xi \leq \mathbf{b}\}, \quad (1.13)$$

a množinu všetkých prípustných aproximatívnych návrhov  $\Xi^{\text{ap}}$  ako

$$\Xi^{\text{ap}} = \{\xi \in [0, \infty)^n : \xi^{(0)} \leq \xi, \mathbf{A}\xi \leq \mathbf{b}\}. \quad (1.14)$$

V týchto množinách hľadáme optimálne návrhy experimentov. Potom môžeme optimálny návrh z definície (1.10) prepísať na tvar

$$\xi^* = \operatorname{argmax}_{\xi \in \Xi^{\text{ex}}} \Phi(\mathbf{M}(\xi)). \quad (1.15)$$

Toto je základná teoretická úloha, ktorú budeme riešiť v celej práci.

### 1.1.7 Ilustračný príklad - nátery

V tejto časti predvedieme zavedené pojmy na konkrétnom príklade. Príklad je inšpirovaný úlohou z knihy [4], str.41, príklad 2.2.

**Príklad 1.** *Chceme testovať vplyv počtu náterov laku na koróziu súčiastok. K dispozícii máme 24 súčiastok a 24 dávok náteru, pričom na jednu súčiastku môžeme aplikovať jednu alebo dve dávky náteru. Po aplikácii náterov súčiastky necháme mesiac v morskej vode a potom zmeriame stav korózie. Na koľko súčiastok máme aplikovať jednu a na koľko dve dávky náteru, keď chceme sledovať rozdiel medzi nimi?*

Použijeme model

$$\mathbf{Y}_i = \mathbf{f}^T(\tilde{x}_i)\theta + \varepsilon_i, \quad i = 1, \dots, N, \quad N \leq 24.$$

Možné body návrhu experimentu sú dva  $\mathfrak{X} = \{x_1, x_2\}$ , kde  $x_1$  predstavuje aplikáciu jednej dávky náteru a  $x_2$  aplikáciu dvoch dávok náteru na jednu súčiastku. K ním prislúchajúce regresory sú  $\mathbf{f}(x_1) = (1 \ 0)^T$  a  $\mathbf{f}(x_2) = (0 \ 1)^T$ . Sledujeme účinky oboch typov náterov, takže potrebujeme dva parametre  $\theta = (\theta_1, \theta_2)'$ , kde  $\theta_1$  sleduje efekt jednej dávky náteru a  $\theta_2$  sleduje efekt dvoch dávok náteru. Zaujímá nás odhad  $\widehat{\theta_1 - \theta_2}$ . Pri jednotlivých pokusoch predpokladáme nezávislé, rovnako rozdelené chyby s nulovou strednou hodnotou a konečnou disperziou  $0 < \sigma^2$ .

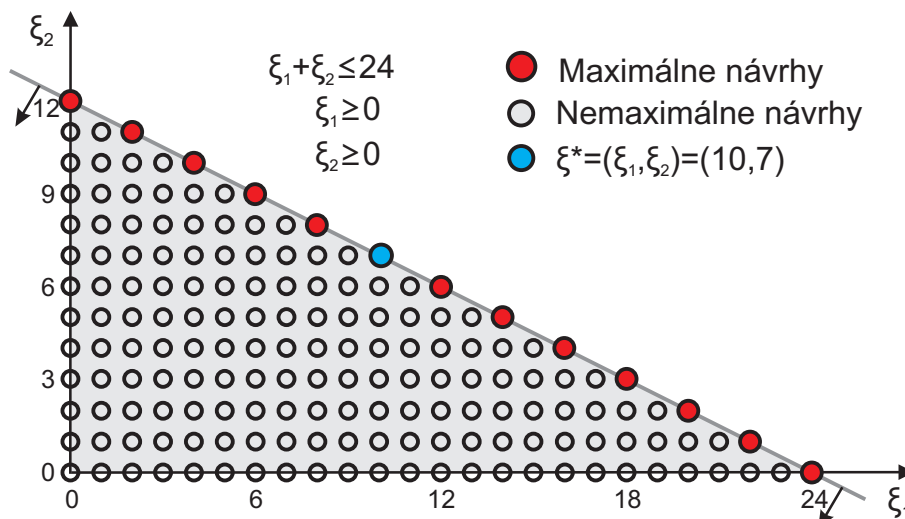
Nech  $\xi_1$  je počet súčiastok, na ktoré bola použitá jedna dávka náteru a  $\xi_2$  je počet súčiastok, na ktoré boli použité dve dávky náteru. Potom *ohraničenia na zdroje* na celkový počet dávok a na celkový počet súčiastok vieme zapísať ako

$$\xi_1 + 2\xi_2 \leq 24 \quad \text{a} \quad \xi_1 + \xi_2 \leq 24.$$

Druhé ohraničenie vyplýva z prvého ( $\xi_1 \geq 0, \xi_2 \geq 0$ ), takže stačí pracovať s ohraničením

$$\xi_1 + 2\xi_2 \leq 24. \quad (1.16)$$

Na obrázku 1–1 zobrazíme množinu  $\Xi^{ex}$  všetkých exaktných návrhov spĺňajúcich ohraničenie (1.16) a kandidátov na optimálny návrh.



**Obrázok 1–1:** Množina  $\Xi^{ex}$  exaktných prípustných návrhov (vyznačená bodmi). Množina  $\Xi^{ap}$  aproximatívnych prípustných návrhov (vyznačená sivou plochou). Maximálne návrhy (červené body), nemaximálne návrhy (bez farby) a optimálny návrh  $\xi^* = (10, 7)$  (modrý bod).

Pre nás dôležité sú návrhy označené červenou farbou, ktoré nazveme *maximálne*.

**Definícia 15** (Maximálny návrh). *Maximálny návrh je taký návrh, do ktorého nemôžeme pridať žiaden ďalší pokus v žiadnom bode experimentu, bez porušenia niektorého z ohraničení na zdroje.*

*Nemaximálne* návrhy sú návrhy spĺňajúce *ohraničenia na zdroje*, ktoré nie sú *maximálne*. Niektorý z *maximálnych* návrhov je určite optimálny, vďaka monotónnosti kritérií optimality. Pretože z *nemaximálneho* návrhu vieme vyrobiť informatívnejší (alebo rovnako informatívny) návrh pridaním pokusu do niektorého z bodov návrhu

experimentu, ktorým neporušíme ohraničenia (1.12) (takýto bod návrhu určite existuje vďaka *nemaximálnosti* návrhu). Keďže kritérium optimality nie je ostro monotónne, tak môžu byť optimálne aj *nemaximálne návrhy*.

Za kritérium optimality zvolíme **c**-optimalitu, lebo sledujeme rozdiel medzi efektmi počtu náterov, teda rozdiel medzi parametrami  $\theta_1$  a  $\theta_2$ . Takže nás zaujíma odhad subsystému parametrov  $\widehat{\theta_1 - \theta_2}$ , ktorý vieme zapísať pomocou *matice koeficientov* ako  $\mathbf{P} = (1 \ -1)^T$ , teda *matica koeficientov* je len vektor **c** a hociktoré z kritérií  $D_{\mathbf{P}}$ ,  $A_{\mathbf{P}}$ , a  $E_{\mathbf{P}}$  sa zúži na kritérium **c**-optimality. Toto kritérium vyzerá nasledovne  $\Phi_c(\mathbf{M}) = (\mathbf{c}^T \mathbf{M}^{-1} \mathbf{c})^{-1}$ , ak je splnená podmienka odhadnuteľnosti  $\mathbf{c} \in \mathcal{M}(\mathbf{M})$ . Táto podmienka je splnená ak sú  $n_1$  a  $n_2$  nenulové (ak by bolo jedno z nich nulové, tak nemá zmysel odhadovať rozdiel medzi efektmi počtu náterov), pretože vieme určiť momentovú maticu **M** a vyjadriť vektor **c** ako lineárnu kombináciu stĺpcov momentovej matice.

$$\mathbf{M} = \begin{pmatrix} \xi_1 & 0 \\ 0 & \xi_2 \end{pmatrix} \quad \mathbf{c} = (1 \ -1)^T = \frac{1}{\xi_1}(\xi_1 \ 0)^T + \frac{1}{\xi_2}(0 \ \xi_2)^T \in \mathcal{M}(\mathbf{M}).$$

Potom kritérium optimality nadobúda hodnotu  $\Phi_c(\mathbf{M}) = \xi_1 \xi_2 / (\xi_1 + \xi_2)$ . Keď vyriešime túto optimalizačnú úlohu, alebo overíme všetky *maximálne* návrhy, nájdeme optimálny návrh  $\xi^* = (\xi_1, \xi_2) = (10, 7)$ . Teda desať súčiastok natrieme jednou dávkou náteru a sedem súčiastok dvomi dávkami. Odhad subsystému parametrov  $\theta_1 - \theta_2$  bude

$$\widehat{\theta_1 - \theta_2} = \frac{\sum_{j=1}^{\xi_1} y_j}{\xi_1} - \frac{\sum_{j=\xi_1+1}^{\xi_1+\xi_2} y_j}{\xi_2}.$$

Tento príklad bol pomerne ľahký a optimálny návrh sa dal nájsť aj bez použitia špeciálnych algoritmov. Stačilo vyriešiť ľahký optimalizačný problém, alebo zrátať hodnotu kritéria pre jedenásť *maximálnych* návrhov. Problém nastane, ak máme viacero bodov návrhu a viacero *ohraničení na zdroje*. Potom môže byť množina *maximálnych* návrhov veľmi zložitá a tým pádom je ťažšie nájsť optimálny návrh. Viac o optimalizačných metódach určených na hľadanie optimálnych návrhov experimentov ukážeme v kapitole 3.



## 1.2 Blokové návrhy s veľkosťou bloku dva

V celej práci sa zameriame na blokové experimenty s veľkosťou bloku dva. Ukážeme ako súvisia so všeobecne definovaným modelom v predošlej časti. Ďalej ako tento typ experimentov vieme reprezentovať pomocou grafov. Potom použijeme známe výsledky z teórie grafov na problémy z navrhovania experimentov a naopak. Toto prepojenie využíva, že teória grafov je staršia oblasť a tým pádom je v nej veľa teoretických výsledkov, ktoré nám pomôžu. Na druhú stranu navrhovanie experimentov ponúka mnoho nových problémov a úloh, ktoré môžu byť motiváciou pre nové úvahy v teórii grafov. Možné využitie blokových experimentov s veľkosťou bloku dva ukážeme na *mikročíповých experimentoch* v genetike.

### 1.2.1 Blokové návrhy s veľkosťou bloku dva a ich reprezentácia grafmi

V celej časti vychádzame z článku [7].

**Definícia 16.** *Blokový návrh*  $(v, b, B)$  definujeme ako dve množiny, množinu ošetrov (veľkosti  $v$ ) a množinu blokov (veľkosti  $b$ ), a binárnu reláciu incidencie medzi týmito množinami. Binárna relácia incidencie určuje, ktoré ošetrenia sa nachádzajú v spoločných blokoch, pričom v každom bloku sa nachádza práve  $B$  ošetrení.

V celej práci sa zaoberáme blokovými návrhmi s veľkosťou bloku dva  $(v, b, 2)$ . Veľkosť bloku dva hovorí, že v každom bloku sa nachádzajú dve ošetrenia. Navyše budeme pracovať s *binárnymi návrhmi*.

**Definícia 17** (Binárny návrh). *Blokový návrh voláme binárny, ak sa žiadne ošetrenie nenachádza v žiadnom bloku viac než raz.*

Predpoklad *binárnosti* návrhov je prirodzený, pretože nás zaujímajú rozdiely medzi efektmi jednotlivých ošetrov a nie samotné efekty ošetrov. Teda ak by sme použili v jednom bloku dvakrát to isté ošetrenie, tak nezískame žiadnu informáciu o rozdieloch medzi efektmi týchto ošetrov.

*Binárne blokové experimenty s veľkosťou bloku dva* vieme reprezentovať pomocou grafov. Jedna používaná reprezentácia je pomocou *Leviho grafu*. Používali ju napr. v práci [50] a jej výhodou je, že vie pracovať s ľubovoľnými veľkosťami bloku. My sa špecializujeme na veľkosť bloku dva, preto bude pre nás vhodnejšia druhá reprezentácia. Tá je pomocou *párovacieho grafu* (z ang. *concurrency graph*). Ošetrenia v nej

predstavujú  $v$  vrcholov grafu a bloky predstavujú  $b$  hrán grafu. Ak sa dve ošetrenia nachádzajú v spoločnom bloku, tak sú vrcholy reprezentujúce príslušné ošetrenie spojené hranou reprezentujúcou príslušný blok (viď. obr. 1–2). Vďaka používaniu *binárnych návrhov* vzniknutý graf neobsahuje slučky. Násobné hrany predstavujú viacnásobné použitie tých istých dvoch ošetrení v rôznych blokoch. Graf prislúchajúci k návrhu  $\xi$  budeme označovať  $G(\xi)$ .



**Obrázok 1 – 2:** Graf a model pre tri ošetrenia  $\tau_1, \tau_2, \tau_3$  a dva bloky  $\beta_1, \beta_2$

### 1.2.2 Model a informačná matica pre subsystém významných parametrov

Pre blokové experimenty, môžeme definovať nasledovný model analýzy rozptylu

$$Y_{ij} = \tau_i + \beta_j + \varepsilon_{ij}. \quad (1.17)$$

Člen  $Y_{ij}$  predstavuje pozorovanie pre ošetrenie  $i$ , kde  $i \in \{1, \dots, v\}$ , a v bloku  $j$ , kde  $j \in \{1, \dots, b\}$ . Predpokladáme, že  $\varepsilon_{ij}$  sú nezávislé, rovnako rozdelené chyby s nulovou strednou hodnotou a konečnou nenulovou varianciou  $\sigma^2$ . Neznáme parametre  $\tau_1, \dots, \tau_v$  označujú efekt jednotlivých ošetrení (pre nás budú predstavovať *významné parametre*) a  $\beta_1, \dots, \beta_b$  označujú efekt jednotlivých blokov v ktorých sa ošetrenia nachádzajú (pre nás budú predstavovať *pomocné parametre*). Ukážka tvorby modelu je na obrázku 1–2. Tento model (1.17) vieme prepísať do tvaru univerzálneho modelu (1.1) z predošlej sekcie nasledovne

$$Y_k = \mathbf{f}^T(\tilde{x}_k)\theta + \varepsilon_k, \quad k = 1, \dots, N, \quad \text{kde } N = 2k.$$

Množina bodov návrhu experimentu  $\mathfrak{X}$  je množina usporiadaných dvojíc  $(i, j)$ , kde  $i \in \{1, \dots, v\}$  predstavuje použité ošetrenie a  $j \in \{1, \dots, b\}$  predstavuje blok v ktorom sa toto ošetrenie nachádza. Index  $k$  bodu návrhu  $\tilde{x}_k \in \mathfrak{X}$  je poradové číslo vybraného bodu návrhu v ktorom sme pozorovali  $i$ -te ošetrenie na  $j$ -tom bloku. Potom  $k$  je rovné  $2j - 1$ , pre menšie z použitých ošetrení v  $j$ -tom bloku a  $k = 2j$  pre väčšie z použitých ošetrení v  $j$ -tom bloku. Hodnota  $Y_k$  je pozorovanie v bode návrhu  $\tilde{x}_k$ .

Regresor  $\mathbf{f} : \mathcal{X} \rightarrow \mathbb{R}^m$  (kde  $m = v + b$ ) sleduje efekt ošetrenia a bloku v ktorom sa toto ošetrenie nachádza.

$$\mathbf{f}((i, j)) = (0, \dots, 0, \underbrace{1}_{i\text{-te miesto}}, 0, \dots, 0, \underbrace{1}_{v+j\text{-te miesto}}, 0, \dots, 0)^T \in \mathbb{R}^m.$$

Vektor neznámych parametrov  $\theta$  je tvaru  $(\tau_1, \dots, \tau_v, \beta_1, \dots, \beta_b)^T \in \mathbb{R}^m$  a  $\varepsilon_k$  sú nezávislé, rovnako rozdelené chyby s nulovou strednou hodnotou a konečnou nenulovou varianciou  $\sigma^2$ .

Na základe modelu (1.17) a použitého návrhu  $\xi$ , vieme vytvoriť maticu plánu experimentu  $\tilde{\mathbf{F}}(\xi)$  a z nej *momentovú maticu*  $\tilde{\mathbf{M}}(\xi)$  pre celý systém parametrov  $\theta$ . Vyzerá nasledovne

$$\tilde{\mathbf{M}}(\xi) = \begin{bmatrix} \mathbf{R}_{v \times v} & \mathbf{N}_{v \times b} \\ \mathbf{N}_{b \times v}^T & 2 \cdot \mathbf{I}_{b \times b} \end{bmatrix}. \quad (1.18)$$

Kde  $\mathbf{I}_{b \times b}$  je identická matica. Matica  $\mathbf{R}_{v \times v}$  je matica počtu použití jednotlivých ošetrení, kde  $r_{ij} = 0; i \neq j$  a  $r_{ii}$  je rovné počtu blokov obsahujúcich ošetrenie  $i$ . Matica  $\mathbf{N}_{v \times b}$  je matica *incidencie*.

**Definícia 18.** *Matica incidencie návrhu*  $\mathbf{N}$  (z ang. *incidence matrix*), pre  $v$  ošetrení a  $b$  blokov je matica rozmeru  $v \times b$ , kde člen  $n_{ij}$  predstavuje počet výskytov  $i$ -teho ošetrenia v  $j$ -tom bloku.

Keďže pracujeme s *binárnymi* návrhmi, tak  $n_{ij}$  môže byť len jedna (ak sa  $i$ -te ošetrenie nachádza v  $j$ -tom bloku) alebo nula (inak).

Ďalej podľa postupu na konci odseku 1.1.2 a tvrdenia 1.8, vytvoríme *informačnú maticu* pre subsystém *významných parametrov*  $\mathbf{P}^T \theta$ , kde  $\mathbf{P} = (\mathbf{I}_v, \mathbf{0})^T$ . Maticu  $\tilde{\mathbf{M}}(\xi)$  rozdelíme na bloky

$$\tilde{\mathbf{M}}_{11}(\xi) = \mathbf{R}_{v \times v}, \quad \tilde{\mathbf{M}}_{12}(\xi) = \mathbf{N}_{v \times b}, \quad \tilde{\mathbf{M}}_{21}(\xi) = \mathbf{N}_{b \times v}^T, \quad \tilde{\mathbf{M}}_{22}(\xi) = 2 \cdot \mathbf{I}_{b \times b}.$$

Potom *informačná matica* pre subsystém *významných parametrov* vyzerá nasledovne

$$\tilde{\mathbf{C}}_{\mathbf{P}}(\xi) = \mathbf{R}_{v \times v} - \mathbf{N}_{v \times b} (2 \cdot \mathbf{I}_{b \times b})^{-1} \mathbf{N}_{b \times v}^T = \mathbf{R}_{v \times v} - \frac{1}{2} \mathbf{N}_{v \times b} \mathbf{N}_{b \times v}^T. \quad (1.19)$$

*Informačná matica* je vyjadrená pomocou niekoľkých špeciálnych matíc, ktoré predstavíme.

**Definícia 19.** *Párovacia matica návrhu*  $\Lambda(\xi)$  (z ang. *concurrency matrix*) pre  $v$  ošetrení a  $b$  blokov je matica rozmeru  $v \times v$ , a definujeme ju ako  $\Lambda(\xi) = \mathbf{N}\mathbf{N}^T$ , kde  $\mathbf{N}$  je matica incidencie.

Pracujeme s binárnymi návrhmi, takže člen  $\lambda_{ij}$  predstavuje koľkokrát sa nachádzajú ošetrenia  $i$  a  $j$  v spoločnom bloku. Člen  $\lambda_{ii}$  hovorí v koľkých blokoch sa nachádzalo  $i$ -te ošetrenie. Matica  $\mathbf{R}_{v \times v}$  je maticu počtu použití jednotlivých ošetrení. Má rozmer  $v \times v$  a  $r_{ij} = 0; i \neq j$ .

Kvôli jednoduchosti budeme v ďalšom namiesto  $\tilde{\mathbf{C}}_{\mathbf{P}}(\xi)$  (pre  $\mathbf{P} = (\mathbf{I}_v, \mathbf{0})^T$ ) používať označenie  $\mathbf{C}(\xi)$ , prípadne len  $\mathbf{C}$ . Potom môžeme *informačnú maticu* pre subsystém významných parametrov (1.19) zapísať zjednodušene ako

$$\mathbf{C}_{v \times v} = \mathbf{R}_{v \times v} - \frac{1}{2}\Lambda_{v \times v}.$$

Táto matica  $\mathbf{C}$  je symetrická pozitívne semidefinitná matica. Súčty členov v riadkoch (respektíve v stĺpcoch) sú nulové, teda  $\mathbf{C}_{v \times v}\mathbf{1}_v = \mathbf{0}_v$ , a záporne vzatý člen na diagonále je rovný súčtu zvyšných členov riadku (resp. stĺpcu). Tieto tvrdenia sú zjavné na základe definície matice  $\mathbf{C}$  pomocou matic  $\mathbf{R}$  a  $\Lambda$ .

### 1.2.3 Laplacián grafu

Nech  $\mathbf{R}$  a  $\Lambda$  sú matice definované v predošlom texte. Dá sa ukázať, že matica  $\mathbf{R}$  je pre graf návrhu matica, ktorá má na diagonále stupne vrcholov a inde samé nuly. Podobne matica  $\Lambda$  je pre graf návrhu matica, ktorá má na diagonále stupne vrcholov a mimo diagonály počet hrán medzi dvojicami vrcholov. Potom definujeme Laplacián grafu nasledovne.

**Definícia 20** (Laplacián grafu). *Laplacián grafu* je  $\mathbf{L}(G) = 2\mathbf{R} - \Lambda$ .

Zjavne platí  $\mathbf{L} = 2\mathbf{C}$ . Takže ak chceme nájsť *informačnú maticu* pre systém významných parametrov pre návrh experimentu  $\xi$ , tak stačí vytvoriť pre návrh  $\xi$  *párovací graf* a zostrojiť k nemu Laplacián. Keďže Laplacián je dvojnásobok informačnej matice a pracujeme s pozitívne homogénnymi kritériami, tak je jedno či budeme hľadať optimálny návrh na základe Laplaciánu, alebo informačnej matice.

Ďalej ukážeme, čo všetko vieme odhadnúť na základe *informačnej matice* pre významné parametre, resp. príslušného Laplaciánu. Z vlastnosti inf. matice  $\mathbf{C}_{v \times v}\mathbf{1}_v = \mathbf{0}_v$

vyplýva, že matica  $\mathbf{C}$  (resp.  $\mathbf{L}$ ) je singulárna. Takže nemôžeme odhadnúť všetky *významné parametre*, pretože *matica koeficientov*  $\mathbf{P} = \mathbf{I}_{v \times v}$  je regulárna a teda nespĺňa podmienku odhadnuteľnosti  $\mathcal{M}(\mathbf{P}) \subseteq \mathcal{M}(\mathbf{C})$ . Sme schopní odhadnúť len niektoré *kontrasty* medzi parametrami (lineárne kombinácie parametrov  $\mathbf{a}^T \boldsymbol{\tau}$ , kde  $\mathbf{a}^T \mathbf{1}_v = 0$ ). Najčastejšie sa zvyknú odhadovať *kontrasty* porovnávajúce dve ošetrenia, t.j.  $\tau_i - \tau_j$  a teda vektor kontrastov  $\mathbf{a}$  tvoria samé nuly, 1 na  $i$ -tom mieste a  $-1$  na  $j$ -tom mieste.

Dá sa ukázať že všetky kontrasty sú *odhadnuteľné* práve vtedy, keď je graf návrhu súvislý. V takom prípade má *Laplacián*  $\mathbf{L}$  grafu (tým pádom aj *informačná matica*  $\mathbf{C}$ ) práve jedno nulové vlastné číslo a ostatné vlastné čísla sú kladné. Ak graf návrhu nie je súvislý, potom počet nulových vlastných čísel *Laplaciánu* (teda aj *informačnej matice*) je rovný počtu komponent súvislosti grafu návrhu (viď. [7], tvrdenie 2.9). Toto využijeme pri definícii *D*-, *A*- a *E*- optimality pomocou vlastných čísel vhodnej matice.

#### 1.2.4 Kritériá optimality pre blokové návrhy

##### Majorizácia a ortogonálne invariantné kritériá:

V predošlej časti sme ukázali vlastnosti *ortogonálne invariantných kritérií* optimality. Teraz pre ne ukážeme inú definíciu, založenú na majorizácii vektorov vlastných čísel *informačných matíc* (resp. *Laplaciánov*). Podklady k majorizácii sme čerpali najmä z [9, 23].

**Definícia 21** (Majorizácia). *Nech  $\mathbf{x}, \mathbf{y}$  sú dva vektory v  $\mathbb{R}^v$ . Hovoríme, že vektor  $\mathbf{x}$  majorizuje vektor  $\mathbf{y}$  (značíme  $\mathbf{x} \prec \mathbf{y}$ ), ak pre ich usporiadané prvky  $x_1 \geq \dots \geq x_v$ ,  $y_1 \geq \dots \geq y_v$  platí*

$$i. \quad \sum_{i=1}^k x_{v-i+1} \leq \sum_{i=1}^k y_{v-i+1}, \quad \text{pre } k = 1, \dots, v-1$$

$$a \quad ii. \quad \sum_{i=1}^v x_i = \sum_{i=1}^v y_i.$$

My budeme ako vektory  $\mathbf{x}$  a  $\mathbf{y}$  používať usporiadané vektory vlastných čísel *informačných matíc* návrhov  $\mathbf{C}(\xi)$  (resp. príslušných *Laplaciánov*). Potom druhá podmienka rovnosti *ii.* bude splnená, ak budú mať oba návrhy rovnako veľa, rovnako veľkých blokov. Pretože  $\sum_{i=1}^v x_i = \text{stopa}(\mathbf{C}(\xi)) = \text{polovica počtu prvkov v blokoch návrhu}$ . Pre nás bude mať význam porovnávať návrhy s rovnakým počtom ošetrení a blokov. V opačnom prípade by sme mohli pracovať so *slabou majorizáciou*, pre ktorú aj druhá vlastnosť *ii.* je v tvare rovnakej nerovnosti ako predošlé.

**Definícia 22** (Schurovsky konkávna funkcia). *Hovoríme, že funkcia  $f : \mathbb{R}^v \rightarrow \mathbb{R}$  je Schurovsky konkávna, ak  $\mathbf{x} \prec \mathbf{y}$  implikuje  $f(\mathbf{x}) \geq f(\mathbf{y})$ .*

Platí, že každá symetrická, konkávna funkcia je aj *Schurovsky konkávna*. Tým pádom sú *ortogonálne invariantné* kritériá optimality, vzhľadom na vektory vlastných čísel informačných matíc, Schurovsky konkávne - sú symetrické (nezáleží na poradí pokusov) a konkávne (vlastnosť 2. z definície). Tým pádom môžeme definíciu 14 previesť na nasledovný tvar.

**Definícia 23** (Schurovsky optimálny návrh). *Návrh  $\xi$ , ktorého vektor vlastných čísel  $\lambda(\xi)$  informačnej matice pre subsystém významných parametrov je majorizovaný vektorom vlastných čísel  $\lambda(\zeta)$  ľubovoľného iného prípustného návrhu  $\zeta$ , nazývame Schurovsky optimálny návrh (resp. najlepší návrh vzhľadom na majorizáciu).*

Ukážeme dve tvrdenia, ktoré budeme potrebovať neskôr.

**Lema 1.1.** *Pre ľubovoľné rovnako veľké vektory  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^v$  platí, že  $\mathbf{x} \prec \mathbf{y}$  práve vtedy keď  $c\mathbf{1}_v + \mathbf{x} \prec c\mathbf{1}_v + \mathbf{y}$ , kde  $\mathbf{1}_v$  je  $v$ -rozmerný vektor samých jednotiek a  $c$  je reálna konštanta.*

*Dôkaz.* V dôkaze budeme vychádzať z definície majorizácie. Podľa nej rozpíšeme  $c\mathbf{1}_v + \mathbf{x} \prec c\mathbf{1}_v + \mathbf{y}$  nasledovne

$$\begin{aligned} \sum_{i=1}^k (c + x_i) &\leq \sum_{i=1}^k (c + y_i) && \text{pre } k = 1, \dots, v, \\ kc + \sum_{i=1}^k x_i &\leq kc + \sum_{i=1}^k y_i && \text{pre } k = 1, \dots, v, \\ \sum_{i=1}^k x_i &\leq \sum_{i=1}^k y_i && \text{pre } k = 1, \dots, v. \end{aligned}$$

Čo je presne  $\mathbf{x} \prec \mathbf{y}$ , pričom sme použili len ekvivalentné úpravy. □

**Lema 1.2.** *Nech  $\mathbf{x}, \mathbf{y} \in \mathbb{R}_+^v$  sú ľubovoľné rovnako veľké vektory s nezápornými prvkami. Nech  $\mathbf{x}_0 = (x, \underbrace{0, \dots, 0}'_k)$ ,  $\mathbf{y}_0 = (y, \underbrace{0, \dots, 0}'_k)$ . Potom platí*

$$\mathbf{x} \prec \mathbf{y} \Leftrightarrow \mathbf{x}_0 \prec \mathbf{y}_0.$$

*Dôkaz.* Rovnako ako v predošlom prípade, priamo z rozpísania pomocou definície. □

### Konkrétne kritériá optimality:

Nech  $\mathbf{C}_{v \times v}$  je *informačná matica* návrhu experimentu pre efekty *významných parametrov*  $\tau_1, \dots, \tau_v$  a nech  $\lambda_1(\mathbf{C}) \geq \dots \geq \lambda_{v-1}(\mathbf{C}) \geq \lambda_v(\mathbf{C}) = 0$  sú jej vlastné čísla. Potom vlastné čísla Laplaciánu  $\mathbf{L} = 2\mathbf{C}$  sú  $2\lambda_1(\mathbf{C}) \geq \dots \geq 2\lambda_{v-1}(\mathbf{C}) \geq 2\lambda_v(\mathbf{C}) = 0$ . Budeme používať tri konkrétne kritériá optimality  $D$ ,  $A$  a  $E$ , v teórii blokových experimentov (pozri prácu [7]).

**Definícia 24. Kritérium  $D$ -optimality**  $\Phi_D(\mathbf{C}(\xi))$  položíme rovné  $v$ -tej odmocnine zo súčiny všetkých vlastných čísel informačnej matice, okrem najmenšieho vlastného čísla spomedzi nich, teda  $v$ -tej odmocnine zo súčiny vlastných čísel  $\lambda_1(\mathbf{C}), \dots, \lambda_{v-1}(\mathbf{C})$ . Ak je niektoré vlastné číslo spomedzi  $\lambda_1(\mathbf{C}), \dots, \lambda_{v-1}(\mathbf{C})$  rovné nule, tak aj kritérium  $\Phi_D$  je rovné nule.

**Definícia 25. Kritérium  $A$ -optimality**  $\Phi_A(\mathbf{C})$  položíme rovné harmonickému priemeru všetkých vlastných čísel informačnej matice, okrem najmenšieho vlastného čísla spomedzi nich. Teda harmonickému priemeru vlastných čísel  $\lambda_1(\mathbf{C}), \dots, \lambda_{v-1}(\mathbf{C})$ . Ak je niektoré vlastné číslo spomedzi  $\lambda_1(\mathbf{C}), \dots, \lambda_{v-1}(\mathbf{C})$  rovné nule, tak aj kritérium  $\Phi_A$  je rovné nule.

**Definícia 26. Kritérium  $E$ -optimality**  $\Phi_E(\mathbf{C})$  položíme rovné druhému najmenšiemu vlastnému číslu  $\mathbf{C}$ , teda  $\lambda_{v-1}(\mathbf{C})$ . Ak je  $\lambda_{v-1}(\mathbf{C})$  rovné nule, tak aj kritérium  $\Phi_E$  je rovné nule.

V nasledujúcej vete dokážeme, že tieto kritériá vieme interpretovať v klasickej teórii navrhovania experimentov ako parciálne kritériá  $D_{\mathbf{P}}$ -,  $A_{\mathbf{P}}$ - a  $E_{\mathbf{P}}$ -optimality pre odhad akéhokoľvek ortonormálneho systému  $(v - 1)$  kontrastov efektov *v významných parametrov*.

Majme návrh experimentu, ktorého graf je súvislý a má informačnú maticu  $\mathbf{C}_{v \times v}$ . Potom  $\mathbf{C}$  má práve jednu nulovú vlastnú hodnotu. Vďaka vete 1.1 existuje ortonormálna transformácia matice  $\mathbf{C}$  na maticu, ktorá bude mať s  $\mathbf{C}$  rovnaké vlastné čísla okrem najmenšieho z nich (takže bude mať všetky vlastné čísla nenulové).

**Veta 1.1.** Nech  $\mathbf{Z}_{v \times (v-1)} = (\mathbf{z}_1, \dots, \mathbf{z}_{v-1})$ ,  $\mathbf{Z}^T \mathbf{Z} = \mathbf{I}_{v-1}$ ,  $\mathbf{Z}^T \mathbf{1}_v = \mathbf{0}_{v-1}$  je matica, ktorej stĺpce sú vektory definujúce kontrasty medzi významnými parametrami  $\tau_1, \dots, \tau_v$ . Nech  $\mathbf{P}^T = (\mathbf{Z}^T, \mathbf{0}_{(v-1) \times b})$ . Nech  $\lambda_i(\cdot)$  sú vlastné hodnoty pozitívne semidefinítnej matice usporiadané od najväčšej po najmenšiu. Nech návrh  $\xi$  spĺňa podmienku odhadnutelnosti

$\mathcal{M}(\mathbf{P}) \subseteq \mathcal{M}(\mathbf{M}(\xi))$ . Potom platí

$$i. \quad \mathbf{C}_{\mathbf{P}}(\xi) = \mathbf{Z}^T \mathbf{C}_{\begin{pmatrix} \mathbf{I}_v \\ \mathbf{0}_{b \times v} \end{pmatrix}}(\xi) \mathbf{Z} \quad (1.20)$$

$$ii. \quad \lambda_i(\mathbf{C}_{\begin{pmatrix} \mathbf{I}_v \\ \mathbf{0}_{b \times v} \end{pmatrix}}(\xi)) = \lambda_i(\mathbf{C}_{\mathbf{P}}(\xi)), \quad i = 1, \dots, v-1 \quad (1.21)$$

Ešte pred dôkazom vety sa pozrime čo presne znamená. Tvrdenie *i.* hovorí, že informačnú maticu pre akýkoľvek systém  $v-1$  ortonormálnych kontrastov efektov vzoriek  $\mathbf{C}_{\mathbf{P}}(\xi)$  vieme vyjadriť pomocou informačnej matice pre celý vektor efektov *významných parametrov*. Tvrdenie *ii.* hovorí, že vlastné čísla informačnej matice  $\mathbf{C}_{\mathbf{P}}(\xi)$  sú rovnaké ako vlastné čísla informačnej matice pre celý vektor efektov *významných parametrov*, okrem najmenšieho (nulového) vlastného čísla. Čo je vo zvolenom usporiadaní presne  $v$ -te vlastné číslo pôvodnej, väčšej informačnej matice pre celý subsystém parametrov  $\tau$ .

*Dôkaz. i.* Najskôr využijeme vetu 3.19 z knihy [45]. Tá tvrdí

$$\mathbf{C}_{\mathbf{KH}}(\xi) = \mathbf{C}_{\mathbf{H}}(\mathbf{C}_{\mathbf{K}}(\xi)),$$

kde matica  $\mathbf{K}_{m \times v}$  má plnú hodnosť  $v$  a vyberá subsystém parametrov  $\mathbf{K}^T \theta$ , matica  $\mathbf{H}_{v \times (v-1)}$  má plnú hodnosť  $v-1$  a vyberá subsystém parametrov  $\mathbf{H}^T \mathbf{K}^T \theta$ . My zvolíme matice  $\mathbf{K} = (\mathbf{I}_v, \mathbf{0}_{v \times b})^T$  a  $\mathbf{H} = \mathbf{Z}$ . Potom

$$\mathbf{C}_{\mathbf{P}}(\xi) = \mathbf{C}_{\begin{pmatrix} \mathbf{z}_{v \times (v-1)} \\ \mathbf{0}_{b \times (v-1)} \end{pmatrix}}(\xi) = \mathbf{C}_{\begin{pmatrix} \mathbf{I}_v \\ \mathbf{0}_{b \times v} \end{pmatrix} \mathbf{Z}}(\xi) = \mathbf{C}_{\mathbf{Z}}(\mathbf{C}_{\begin{pmatrix} \mathbf{I}_v \\ \mathbf{0}_{b \times v} \end{pmatrix}}(\xi)) = (\mathbf{Z}^T (\mathbf{C}_{\begin{pmatrix} \mathbf{I}_v \\ \mathbf{0}_{b \times v} \end{pmatrix}}(\xi))^{-1} \mathbf{Z})^{-1}.$$

Teda tvrdenie (1.20) vieme prepísať do tvaru

$$(\mathbf{Z}^T (\mathbf{C}_{\begin{pmatrix} \mathbf{I}_v \\ \mathbf{0}_{b \times v} \end{pmatrix}}(\xi))^{-1} \mathbf{Z})^{-1} = \mathbf{Z}^T \mathbf{C}_{\begin{pmatrix} \mathbf{I}_v \\ \mathbf{0}_{b \times v} \end{pmatrix}}(\xi) \mathbf{Z}.$$

Takže chceme ukázať

$$\left( \mathbf{Z}^T \mathbf{C}_{\begin{pmatrix} \mathbf{I}_v \\ \mathbf{0}_{b \times v} \end{pmatrix}}(\xi) \mathbf{Z} \right) \left( \mathbf{Z}^T (\mathbf{C}_{\begin{pmatrix} \mathbf{I}_v \\ \mathbf{0}_{b \times v} \end{pmatrix}}(\xi))^{-1} \mathbf{Z} \right) = \mathbf{I}_{v-1} \quad (1.22)$$

Nech  $\lambda_1, \dots, \lambda_v$  sú vlastné čísla matice  $\mathbf{C}_{\begin{pmatrix} \mathbf{I}_v \\ \mathbf{0}_{b \times v} \end{pmatrix}}(\xi)$  a  $\mathbf{u}_1, \dots, \mathbf{u}_v$  k nim prislúchajúce vlastné vektory, pričom  $\mathbf{u}_1, \dots, \mathbf{u}_v$  vyberieme tak, aby tvorili ortonormálny systém. Keďže matica  $\mathbf{C}_{\begin{pmatrix} \mathbf{I}_v \\ \mathbf{0}_{b \times v} \end{pmatrix}}(\xi)$  je singulárna, tak musí mať  $\lambda_v = 0$  a vďaka predpokladu odhadnuteľnosti je  $\lambda_{v-1} > 0$ . Vlastný vektor prislúchajúci k nulovému vlastnému číslu zvolíme  $\mathbf{u}_v = \mathbf{1}_v / \sqrt{v}$ , čo môžeme vďaka vlastnosti  $\mathbf{C}_{\begin{pmatrix} \mathbf{I}_v \\ \mathbf{0}_{b \times v} \end{pmatrix}}(\xi) (\mathbf{1}_v / \sqrt{v}) = \mathbf{0}_v$ . Pomocou



spektrálneho rozkladu matíc môžeme zapísať

$$\mathbf{C}_{(\mathbf{o}_{b \times v}^{\mathbf{I}_v})}(\xi) = \sum_{i=1}^{v-1} \lambda_i \mathbf{u}_i \mathbf{u}_i^T, \quad \mathbf{C}_{(\mathbf{o}_{b \times v}^{\mathbf{I}_v})}^{-}(\xi) = \sum_{j=1}^{v-1} \frac{1}{\lambda_j} \mathbf{u}_j \mathbf{u}_j^T.$$

Vektory  $\mathbf{u}_1, \dots, \mathbf{u}_{v-1}, (\mathbf{1}_v/\sqrt{v})$  sme vybrali ako ortonormálne. Preto  $\mathbf{u}_i \perp (\mathbf{1}_v/\sqrt{v})$ , resp.  $\mathbf{u}_i \in (\mathbf{1}_v/\sqrt{v})^\perp$  pre ľubovoľné  $i = 1, \dots, v-1$ . Zároveň vieme, že  $\mathbf{Z}^T(\mathbf{1}_v/\sqrt{v}) = \mathbf{0}_{v-1}$ , teda  $\mathbf{z}_i \perp (\mathbf{1}_v/\sqrt{v})$  a navyše  $\mathbf{z}_i \perp \mathbf{z}_j; i \neq j; i, j \in \{1, \dots, v-1\}$ . Takže  $\mathbf{z}_1, \dots, \mathbf{z}_{v-1}$  tvoria bázu  $(\mathbf{1}_v/\sqrt{v})^\perp$ . Potom  $\mathbf{u}_i \in (\mathbf{1}_v/\sqrt{v})^\perp$  vieme vyjadriť ako lineárnu kombináciu bázy  $\mathbf{z}_i$  nasledovne  $\mathbf{u}_i = \mathbf{Z}\mathbf{a}^{(i)}$ , pre nejaké vektory  $\mathbf{a}^{(i)}$ . Po dosadení za  $\mathbf{u}_i$  dostávame

$$\mathbf{C}_{(\mathbf{o}_{b \times v}^{\mathbf{I}_v})}(\xi) = \mathbf{Z} \left[ \sum_{i=1}^{v-1} \lambda_i \mathbf{a}^{(i)} (\mathbf{a}^{(i)})^T \right] \mathbf{Z}^T, \quad \mathbf{C}_{(\mathbf{o}_{b \times v}^{\mathbf{I}_v})}^{-}(\xi) = \mathbf{Z} \left[ \sum_{j=1}^{v-1} \frac{1}{\lambda_j} \mathbf{a}^{(j)} (\mathbf{a}^{(j)})^T \right] \mathbf{Z}^T.$$

**Lema 1.3.** *Nech  $\mathbf{u}_i = \mathbf{Z}\mathbf{a}^{(i)}$  pre ľubovoľné  $i = 1, \dots, v-1$ , kde  $\mathbf{Z}, \mathbf{a}^{(i)}$  a  $\mathbf{u}_i$  sú definované v predošlom texte. Potom*

$$i. \quad (\mathbf{a}^{(i)})^T \mathbf{a}^{(i)} = 1, \quad i = 1, \dots, v-1$$

$$ii. \quad (\mathbf{a}^{(i)})^T \mathbf{a}^{(j)} = 0 \quad i \neq j.$$

*Dôkaz.*

$$\mathbf{1}_v = \mathbf{u}_i^T \mathbf{u}_i = (\mathbf{a}^{(i)})^T \mathbf{Z}^T \mathbf{Z} \mathbf{a}^{(i)} = (\mathbf{a}^{(i)})^T \mathbf{a}^{(i)}$$

$$\mathbf{0}_v = \mathbf{u}_i^T \mathbf{u}_j = (\mathbf{a}^{(i)})^T \mathbf{Z}^T \mathbf{Z} \mathbf{a}^{(j)} = (\mathbf{a}^{(i)})^T \mathbf{a}^{(j)}, \text{ pre } i \neq j \quad \square$$

Lema dokazuje, že  $\mathbf{a}^{(i)}$  tvoria ortonormálny systém  $v-1$  vektorov rozmeru  $v-1$ . Pomocou toho a s využitím vlastností matice  $\mathbf{Z}$  dokážeme rovnosť (1.22)

$$\begin{aligned} & \left( \mathbf{Z}^T \mathbf{C}_{(\mathbf{o}_{b \times v}^{\mathbf{I}_v})}(\xi) \mathbf{Z} \right) \left( \mathbf{Z}^T (\mathbf{C}_{(\mathbf{o}_{b \times v}^{\mathbf{I}_v})}^{-}(\xi))^{-} \mathbf{Z} \right) = \\ & = \left( \mathbf{Z}^T \mathbf{Z} \left[ \sum_{i=1}^{v-1} \lambda_i \mathbf{a}^{(i)} (\mathbf{a}^{(i)})^T \right] \mathbf{Z}^T \mathbf{Z} \right) \left( \mathbf{Z}^T \mathbf{Z} \left[ \sum_{j=1}^{v-1} \frac{1}{\lambda_j} \mathbf{a}^{(j)} (\mathbf{a}^{(j)})^T \right] \mathbf{Z}^T \mathbf{Z} \right) = \\ & = \left[ \sum_{i=1}^{v-1} \lambda_i \mathbf{a}^{(i)} (\mathbf{a}^{(i)})^T \right] \left[ \sum_{j=1}^{v-1} \frac{1}{\lambda_j} \mathbf{a}^{(j)} (\mathbf{a}^{(j)})^T \right] = \sum_{i=1}^{v-1} \sum_{j=1}^{v-1} \left[ \lambda_i \mathbf{a}^{(i)} (\mathbf{a}^{(i)})^T \frac{1}{\lambda_j} \mathbf{a}^{(j)} (\mathbf{a}^{(j)})^T \right] = \\ & = \sum_{i=1}^{v-1} \mathbf{a}^{(i)} (\mathbf{a}^{(i)})^T = \mathbf{I}_{v-1}. \end{aligned}$$

Teda

$$\mathbf{C}_P(\xi) = (\mathbf{Z}^T (\mathbf{C}_{(\mathbf{I}_v)}(\xi))^{-1} \mathbf{Z})^{-1} = \mathbf{Z}^T \mathbf{C}_{(\mathbf{0}_{b \times v})}(\xi) \mathbf{Z}.$$

Čím sme dokázali (1.20).

*ii.* Druhú časť dokážeme ľahko pomocou informácií z dôkazu prvej časti. Ukázali sme

$$\mathbf{C}_P(\xi) = \mathbf{Z}^T \mathbf{C}_{(\mathbf{0}_{b \times v})}(\xi) \mathbf{Z} = \sum_{i=1}^{v-1} \lambda_i \mathbf{a}^{(i)} (\mathbf{a}^{(i)})^T,$$

kde  $\mathbf{a}^{(i)}$  sú ortonormálne a  $\lambda_i, i = 1, \dots, v-1$  sú vlastné čísla  $\mathbf{C}_{(\mathbf{0}_{b \times v})}$  okrem najmenšieho z nich. Zároveň sme dostali rozklad matice  $\mathbf{C}_P(\xi) = \sum_{i=1}^{v-1} \lambda_i \mathbf{a}^{(i)} (\mathbf{a}^{(i)})^T$  pomocou lineárnej kombinácie ortonormálnych vektorov, teda spektrálny rozklad. Takže  $\lambda_i, i = 1, \dots, v-1$  sú vlastné čísla matice  $\mathbf{C}_P(\xi)$ . Teda platí

$$\lambda_i(\mathbf{C}_{(\mathbf{0}_{b \times v})}(\xi)) = \lambda_i(\mathbf{C}_P(\xi)), \quad i = 1, \dots, v-1.$$

□

### 1.2.5 Kritériá optimality pomocou vlastností grafov

Ukázali sme, že kritériá optimality vieme vyjadriť pomocou vlastných čísel informačnej matice a tým pádom aj pomocou vlastných čísel Laplaciánov grafov, prislúchajúcich k návrhu experimentu. O vlastnostiach vlastných čísel Laplaciánov sa vie pomerne veľa. Vďaka tomu môžeme kritériá optimality zaviesť aj alternatívne, pomocou vlastností z teórie grafov. Kritérium  $D$ -optimality vyjadríme pomocou počtu kostier grafu, kritérium  $A$ -optimality pomocou priemerného odporu v elektrických sieťach a kritérium  $E$ -optimality pomocou algebraickej konektivity grafu.

#### **$D$ -optimality:**

Pri kritériu  $D$ -optimality vychádzame z Kirchhoffovej vety o počte kostier grafu ([36]).

**Veta 1.2.** *Nech  $G$  je graf s  $v$  vrcholmi. Potom nasledujúce hodnoty sú rovnaké:*

- i.* počet kostier grafu;
- ii.* súčin vlastných čísel Laplaciánu  $\lambda_1(\mathbf{L}), \dots, \lambda_{v-1}(\mathbf{L})$  vydelený počtom vrcholov  $v$ .

Parciálne kritérium  $D_P$ -optimality pre návrhy so súvislými grafmi je  $v$ -ta odmocnina zo súčinu vlastných čísel  $\lambda_1(\mathbf{C}), \dots, \lambda_{v-1}(\mathbf{C})$  informačnej matice  $\mathbf{C}$ . Laplacián

$\mathbf{L} = 2\mathbf{C}$ , takže súčinu vlastných čísel  $\lambda_1(\mathbf{C}), \dots, \lambda_{v-1}(\mathbf{C})$  je rovný  $v$ -násobku počtu kostier grafu deleno  $2^{v-1}$  (vďaka súvislosti grafu sú vlastné čísla okrem najmenšieho nenulové a je ich  $v - 1$ ). Pri hľadaní  $D$ -optimálneho návrhu teda stačí hľadať návrh reprezentovaný grafom, ktorý má maximálny počet kostier spomedzi grafov s  $v$  vrcholmi a  $b$  hranami. Pretože  $v/2^{v-1}$  je konštanta a  $v$ -ta odmocnina, pre  $v$  kladné je rýdzomonotónna. Hľadanie grafu s maximálnym počtom kostier, pri danom počte vrcholov a hrán, je v teórii grafov značená ako úloha  $t$ -optimality. V tomto znení ju spomíname v druhej kapitole.

Počet kostier grafu  $G$  je rovný počtu možností ako odobrať z grafu  $b - v + 1$  hrán tak, aby ostal súvislý (kde  $b$  je počet hrán a  $v$  je počet vrcholov). Počet kostier grafu sa dá zistiť pomerne jednoducho (existujú teoreticky odvodené optimálne návrhy), ak  $b$  nie je o veľa väčšie ako  $v$  (viď. [5]). Napr. ak je  $b = v - 1$  potom je graf *strom* a má jedinú kosť. Ďalšie známe výsledky sú pre kompletne grafy (Cayleyho veta) a kompletne bipartitné grafy (viď. [11]). Tiež existujú výsledky o optimálnych návrhoch pre regulárne a skoro-regulárne multipartitné grafy (viď. [44]).

### **A-optimality:**

Na základe parciálneho kritéria  $A_{\mathbf{P}}$ -optimality  $\Phi_{A_{\mathbf{P}}}(\mathbf{M}) = \left( \frac{1}{s\sigma^2} \sum_{i=1}^s \text{Var}(\widehat{\mathbf{P}^T\theta_i}) \right)^{-1}$  chceme minimalizovať priemernú varianciu. Medzi ľubovoľnými dvomi ošetreniami  $i$  a  $j$  nachádzajúcimi sa v spoločnom bloku je variancia  $v_{ij} = 2\sigma^2$  (na základe modelu analýzy rozptylu s varianciou chýb  $\text{Var}(\varepsilon_{ij}) = \sigma^2$ ).

Na nájdenie návrhu s minimálnou priemernou varianciou použijeme Kirchhoffov zákonu o odpore v elektrických sieťach (viď. [36]). Odpor hrany prislúchajúcej bloku zodpovedá variancii medzi vrcholmi prislúchajúcimi ošetreniam  $i$  a  $j$  spojenými touto hranou. Varianciu medzi dvomi ošetreniami, ktoré sa nenachádzali v spoločnom bloku, vypočítame na základe zákonov o sériovom a paralelnom zapojení odporov a bude rovná celkovému odporu medzi týmito dvomi ošetreniami.

**Definícia 27** (Sériové zapojenie odporov). *Pri zapojení rezistorov do série (za sebou) sa výsledný odpor rovná súčtu odporov jednotlivých rezistorov.*

**Definícia 28** (Paralelne zapojenie odporov). *Pri zapojení rezistorov paralelne sa výsledný odpor rovná prevrátenej hodnote súčtu prevrátенých hodnôt jednotlivých odporov.*

Takže ak existuje medzi dvomi vrcholmi reprezentujúcimi ošetrenia  $i$  a  $j$  jediná cesta dĺžky  $d$ , tak variancia  $v_{ij} = 2d\sigma^2$  (sériové zapojenie odporov). Ak existuje viacero

ciest, tak potrebujeme použiť oba typy zapojení rezistorov a výpočet súčtu variancií medzi ošetreniami bude zložitejší.

Vďaka uvedenej interpretácii  $A_{\mathbf{P}}$ -optimality bude optimálny návrh experimentu návrh, ktorého párovací graf má minimálny priemerný medziuzlový odpor. Opäť sú teoreticky odvodené optimálne návrhy, pre prípady kde počet blokov nie je o veľa väčší ako počet ošetrení (článok [5]). V diplomovej práci sme na základe algoritmu *simulovaného žitia* našli niekoľko ďalších  $D$ - a  $A$ -optimálnych návrhov.

### **$E$ -optimalita:**

V prípade  $E$ -optimality je možné ukázať, že  $E$ -optimálny návrh zodpovedá párovaciemu grafu, ktorý má maximálnu algebraickú konektivitu grafu.

**Definícia 29** (Algebraická konektivita). *Druhá najmenšia vlastná hodnota Laplaciánu grafu  $\lambda_{v-1}$  sa nazýva algebraická konektivita grafu.*

Pojem *algebraickej konektivity* nemá takú priamu interpretáciu v teórii grafov ako predošlé dve kritériá. Súvisí s hranovými rezmi grafu a podrobnejší prehľad známych výsledkov je napr. v článku [43], alebo [17].

### **1.2.6 Aplikácia - mikročipové experimenty**

Využitie blokových experimentov v praxi ukážeme na mikročipových experimentoch. Čerpáme z prác pani R. A. Baileyovej [4], [5] a viac podrobností je v diplomovej práci [2].

Dvojfarebné mikročipové experimenty sú určené na porovnávanie informácie medzi rôznymi vzorkami DNA (Deoxyribonukleová kyselina). Môžeme porovnávať napr. vzorku DNA zdravého jedinca so vzorkami DNA od rôznych geneticky chorých jedincov, alebo vzorky DNA vo viacerých vývojových štádiách bunky a podobne. Väčšinou sledujeme v ktorých génoch sa jednotlivé vzorky DNA líšia.

Problém je, že mikročipové experimenty sú drahé. Takže je dôležité vhodne vybrať ktoré vzorky porovnáme, aby sme o nich získali čo najväčšiu informáciu vzhľadom na pevne daný rozpočet. Ukážeme ako prebieha mikročipový experiment a ako ho vieme zapísať ako blokový návrh s veľkosťou bloku dva.

### Priebeh experimentu:

Vychádzame z článku [51]. Experiment pozostáva z troch častí. V prvej časti sa pripravia biologické jednotky (napr. myši, baktérie) a rozdelia sa do skupín podľa hodnôt vlastnosti, ktorú chceme sledovať (napr. sledujeme stav choroby v rôznych vývinových štádiách, alebo rôzne množstvá použitého lieku). Jednotlivé skupiny spolu s ich vlastnosťou budeme nazývať *vzorkami*. Ak sa bude dať (napr. pri rôznom množstve liekov), tak rozdeľujeme biologické jednotky do vzoriek náhodne. *Vzorky predstavujú ošetrenia v blokovom návrhu*.

V druhej časti vyberieme vždy dve vzorky, ktoré chceme porovnať a odoberieme z nich časť mRNA. Odoberatú mRNA z prvej vzorky ofarbíme fluorescenčným farbivom na červeno a mRNA z druhej vzorky ofarbíme na zeleno. Potom tieto vzorky hybridizujeme na spoločnom mikročipe. *Mikročipy predstavujú bloky v blokovom návrhu*.

Nakoniec sledujeme koľko z ktorej farby sa vyžiarilo v jednotlivých génoch mRNA na mikročipe. Políčko na mikročipe prislúchajúce génu v ktorom je aktívnejšia prvá vzorka, by malo vyžiarit viac červenej farby a naopak. Na výstupné dáta má vplyv ktoré vzorky sa použili, na ktorom mikročipe sa hybridizovali a na ktorú vzorku bola použitá ktorá farba. Vplyv farby do modelu nezahrnieme (zaoberali sme sa tým v práci [2]). Robíme to najmä kvôli zjednodušeniu modelu, ale dá sa odstrániť aj pri reálnom experimente, ak sa každý dielčí pokus urobí dvakrát s opačným nafarbeným mRNA vzoriek. Návrhy experimentov nezohľadňujúce efekt farby sa nazývajú *návrhy bez efektu farby* (z ang. Dye-swap designs).

Návrh experimentu určuje koľko a ktorých pokusov sa vykoná, teda ktoré vzorky (ošetrenia) sa budú spoločne hybridizovať na mikročipoch (blokoch). Pre takto zvolené ošetrenia a bloky potom využijeme vybudovanú teóriu. Vyberieme systém parametrov, ktoré chceme odhadovať (efekty vzoriek), vytvoríme informačnú maticu (alebo Laplacián), vyberieme kritérium optimality a nájdeme optimálny návrh experimentu.

## 2 Optimálna augmentácia blokových návrhov s veľkosťou bloku dva

Predstavme si, že po vykonaní experimentu dostaneme k dispozícii ďalšie zdroje. Teda môžeme urobiť doplňujúce merania, ale už nemôžeme meniť merania vykonané v pôvodnom experimente. Takúto situáciu nazývame *augmentácia* návrhu experimentu a budeme sa jej podrobne venovať v tejto aj v nasledujúcej kapitole.

Najskôr predstavíme samotnú augmentáciu – kedy a prečo sa používa. Ďalej spomenieme triedy optimálnych návrhov, ktoré budeme augmentovať. Hlavným cieľom celej kapitoly je odvodiť nové triedy optimálnych augmentovaných návrhov experimentov. Tieto triedy odvodíme pomocou rozšírenia dvoch známych postupov, ktoré podrobne vysvetlíme. Prvý postup využíva *charakteristické polynómy* Laplaciánov grafov prislúchajúcich k návrhu. Tejto téme sa veľmi obsérne venoval Kelmans [31], [32], [33] a ďalší [19]. Druhý postup využíva *majorizáciu* vektorov vlastných čísel Laplaciánov grafov prislúchajúcich k návrhom a prvý ho predstavil Constantine [9].

### 2.1 Motivácia

Dôvody prečo použiť dodatočnú augmentáciu experimentu môžu byť rôznorodé. Vieme ich rozdeliť na dva hlavné typy. Prvý typ dôvodov môžeme všeobecne nazvať zlyhanie experimentu. To môže nastať na základe rôznych príčin:

- **Použitý model nie je vhodný** - buď model nezahŕňa dostatok vysvetľujúcich premenných, alebo sme ich použili zbytočne veľa. V prvom prípade model nedostatočne vysvetľuje namerané dáta. V druhom prípade model výborne popisuje namerané dáta, ale kvôli tomu že je veľmi presný na tréningových dátach, tak nebude fungovať pre nezávislú množinu nových dát.
- **Viaceré pokusy v experimente zlyhali** - ak prideme o dáta kvôli náhodným technickým problémom (rozbitie prístroja, chyba pri prenose dát), tak vieme chýbajúce pokusy zopakovať. Ak chýbajú dáta dôsledkom niečoho systémového (prístroj nefunguje pri nízkych teplotách, vysoká dávka lieku je nebezpečná), tak chýbajúce dáta musíme nahradiť inak.
- **Hľadané optimum leží mimo množiny prípustných návrhov** - ak teoreticky nájdené výsledky vôbec nesúhlasia s reálne nameranými dátami.

V literatúre sa dajú nájsť rôzne spôsoby ako riešiť uvedené problémy. Niektoré sa dajú odstrániť ešte pred vykonaním experimentu, napríklad nevhodnosť modelu [1]. Ďalšie môžu byť vyriešené vhodnou augmentáciou experimentu. Na augmentáciu existuje viacero algoritmov. Niektoré z algoritmov predstavíme v kapitole 3. Ďalšie algoritmy, ako aj viac o tejto téme, sa dá nájsť napríklad v knihe [1].

Druhý, priaznivejší a pre nás dôležitejší typ dôvodov na augmentáciu experimentu je, keď vykonáme vhodný experiment a následne dostaneme možnosť na nameranie ďalších pozorovaní. Najtypickejšie príčiny sú:

- **Testovanie modelu** - ak sme v prvej časti experimentu úspešne testovali vhodnosť modelu a v druhej časti chceme na základe ďalších pozorovaní odhadovať samotné efekty parametrov.
- **Nové zdroje** - ak po úspešnom ukončení experimentu, dostaneme k dispozícii dodatočné zdroje na vykonanie ďalších pozorovaní.

V oboch prípadoch môžeme vykonať ďalšie pokusy, ktoré spolu s pôvodne vykonanými pokusmi dajú presnejšie výsledky ako z pôvodného experimentu. Preto je dôležité by boli nové pokusy vyberané čo najvhodnejšie aj vzhľadom na množinu pôvodných pokusov.

Druhý typ dôvodov definoval oblasť, ktorou sa budeme zaoberať v celej nasledujúcej kapitole. Vždy začneme s nejakým optimálnym návrhom experimentu a potom budeme hľadať jeho najlepšiu možnú augmentáciu pri daných podmienkach. Ako počiatočné návrhy budeme používať známe  $A$ - a  $D$ -optimálne návrhy. Konkrétne budeme pracovať s návrhmi reprezentovanými nasledovnými grafmi.

- **Hviezdicové grafy** - sú  $A$ - aj  $D$ -optimálne, viď. článok [5].
- **Kompletné grafy** - sú Schurovsky optimálne. Vlastné čísla ich Laplaciánov (okrem triviálneho nulového) sú všetky rovnaké, tým pádom sú majorizované vlastnými číslami Laplaciánu ľubovoľného iného grafu s rovnakým počtom vrcholov a hrán.
- **Skoro-regulárne a regulárne kompletné multipartitné grafy** - sú  $t$ -optimálne a teda aj  $D$ -optimálne, viď. článok [44].

## 2.2 Definície a značenie

V celej kapitole budeme pracovať prevažne s terminológiou z teórie grafov (namiesto terminológie z navrhovania experimentov), kvôli jednoduchšiemu a rýchlejšiemu zápisu. Teda namiesto augmentácie blokových návrhov, budeme hľadať augmentáciu grafov prislúchajúcich k blokovým návrhom, skrátene augmentáciu grafov.

Najdôležitejší pojem, s ktorým budeme pracovať, je nerastúci vektor vlastných čísel  $\lambda(\cdot)$  Laplaciánu  $\mathbf{L}(\cdot)$  pre graf  $G$  s  $v$  vrcholmi, prislúchajúci k návrhu  $\xi$ .

$$\lambda(\mathbf{L}(G(\xi))) = [\lambda_1(\mathbf{L}(G(\xi))), \lambda_2(\mathbf{L}(G(\xi))), \dots, \lambda_v(\mathbf{L}(G(\xi)))].$$

Namiesto značenia  $\lambda(\mathbf{L}(G(\xi)))$  budeme používať skrátenejší zápis  $\lambda(G)$ , prípadne  $\lambda(G^\xi)$ . Podobne pri slovnom vyjadrení budeme pre jednoduchosť používať pojem vlastné čísla grafu  $G$ , namiesto pojmu vlastné čísla Laplaciánu grafu  $G$  prislúchajúcemu k návrhu  $\xi$ .

Na meranie množstva informácie získanej z konkrétneho experimentu budeme používať triedu ortogonálne invariantných kritérií optimality. Pre túto triedu kritérií dostávame Schurovsky optimálne návrhy. Rovnako ako Constantine [9] budeme párovací graf Schurovsky optimálneho návrhu nazývať *najlepším*.

**Definícia 30** (Najlepší graf). *Graf  $G^*$  nazývame najlepším, ak je návrh z ktorého vznikol Schurovsky optimálny, spomedzi všetkých návrhov s rovnakým počtom ošetrení a blokov (vrcholov a hrán).*

Čo znamená, že vektor vlastných čísel Laplaciánu *najlepšieho* grafu majorizuje vektor vlastných čísel Laplaciánu ľubovoľného iného grafu s rovnakým počtom vrcholov a hrán.

**Definícia 31** (Najlepší návrh). *Návrh  $\xi$  nazývame najlepším, ak jeho párovací graf  $G^*$  je najlepším.*

**Definícia 32.** *Majme grafy  $G$  a  $H$  s rovnakým počtom vrcholov a hrán. Potom nazývame graf  $G$  lepším ako graf  $H$  (vzhľadom na ortogonálne invariantné kritériá, resp. majorizáciu), ak platí  $\lambda(G) \prec \lambda(H)$ .*

Najbežnejší spôsob, ako vyjadriť počet kostier grafu (ktorý maximalizujeme pri  $t$ -optimalite, resp.  $D$ -optimalite), je použitie Kirchhoffovej vety o počte kostier grafu [36].



**Veta 2.1** (Kirchhoffova veta). *Nech  $\lambda_1(G) \geq \lambda_2(G) \geq \dots \geq \lambda_{v-1}(G) \geq \lambda_v(G) = 0$  sú vlastné čísla Laplaciánu grafu  $G$  na  $v$  vrchoch. Potom počet kostier  $\tau(\cdot)$  grafu  $G$  je*

$$\tau(G) = \frac{1}{v} \prod_{k=1}^{v-1} \lambda_k.$$

Budeme využívať úzke prepojenie medzi vlastnosťami vlastných čísel Laplaciánu grafu a jeho komplementu. Uvedenú vlastnosť prvý spozoroval A. K. Kelmans, [32].

**Veta 2.2** (Kelmans, 1966). *Nech  $G$  je graf na  $v$  vrchoch s vlastnými číslami  $\lambda_1(G), \dots, \lambda_v(G)$ , potom vlastné čísla  $k$  nemu komplementárneho grafu  $\bar{G}$  sú*

$$\lambda_i(\bar{G}) = v - \lambda_{v-i+1}(G), \quad i = 1, \dots, v.$$

**Lema 2.1.** *Nech  $G$  a  $H$  sú jednoduché grafy. Potom  $\lambda(G) \prec \lambda(H)$ , práve vtedy keď  $\lambda(\bar{G}) \prec \lambda(\bar{H})$ .*

Dôkaz priamo vyplýva z predchádzajúcej vety a definície majorizácie (21). Táto lema bude užitočná, pretože budeme pracovať s grafmi s veľkým počtom hrán. Prácu s nimi, vďaka uvedenej leme, môžeme nahradiť za prácu s  $k$  nim komplementárnymi grafmi s malým počtom hrán.

Ďalej ukážeme dva spôsoby ako spájať grafy.

**Definícia 33** (Zjednotenie grafov). *Zjednotenie grafov  $G_1$  a  $G_2$  je graf s vrcholmi  $V(G_1) \cup V(G_2)$  a hranami  $E(G_1) \cup E(G_2)$ . Označujeme ho  $G_1 + G_2$ .*

**Definícia 34** (Spojenie grafov). *Spojenie grafov  $G_1$  a  $G_2$  je graf získaný zjednotením grafov  $G_1$  a  $G_2$  a pridaním všetkých možných hrán medzi vrcholy jedného a druhého pôvodného grafu, t.j.  $e = (s, t); s \in V(G_1), t \in V(G_2)$ . Označujeme ho  $G_1 * G_2$ .*



Existujú aj ďalšie spôsoby kombinovania grafov, pre ktoré je ukázané ako sa chovajú ich vlastné čísla. Napríklad kartézsky súčin dvoch grafov. Nám budú stačiť uvedené dva spôsoby.

**Definícia 35** (Laplaceov polynóm). *Laplaceovým polynómom  $\mu(G, x)$  pre graf  $G_v$  v bode  $x$ , nazývame charakteristický polynóm Laplaciánu grafu  $G$ . Korene Laplaceovho polynómu sú vlastné čísla Laplaciánu  $\lambda_1(G) \geq \dots \geq \lambda_{v-1}(G) \geq \lambda_v(G) = 0$ .*

$$\mu(G, x) = (x - \lambda_1(G))(x - \lambda_2(G)) \cdots (x - \lambda_v(G)).$$

Uvedieme dve vety, na základe ktorých bude pracovať nasledovná metóda 2.3.1.

**Veta 2.3** (Kelmans, 1965). *Nech  $G_{v_1}$  a  $G_{v_2}$  sú grafy s  $v_1$  a  $v_2$  vrcholmi. Potom Laplaceov polynóm pre spojenie grafov  $G_{v_1} * G_{v_2}$  v bode  $x$  je*

$$\mu(G_{v_1} * G_{v_2}, x) = \frac{x(x - v_1 - v_2)}{(x - v_1)(x - v_2)} \mu(G_{v_1}, x - v_2) \mu(G_{v_2}, x - v_1).$$

Dôkaz sa nachádza v článku [31]. Túto vetu vieme jednoducho rozšíriť na nasledovnú.

**Veta 2.4** (Rozšírený Kelmans). *Nech  $G_{v_1}, \dots, G_{v_k}$  sú grafy s  $v_1, \dots, v_k$  vrcholmi,  $\sum_{i=1}^k v_i = v$ . Potom Laplaceov polynóm pre spojenie grafov  $G_{v_1} * \dots * G_{v_k}$  v bode  $x$  je*

$$\mu(G_{v_1} * \dots * G_{v_k}, x) = \frac{x(x - v)^{k-1}}{\prod_{i=1}^k (x - v + v_i)} \prod_{i=1}^k \mu(G_{v_i}, x - v + v_i).$$

*Dôkaz.* Použijeme predošlú vetu 2.3 postupne na dvojice grafov  $G_{v_1}$  a  $G_{v_2}$ , potom  $(G_{v_1} * G_{v_2})$  a  $G_{v_3}$ , a tak ďalej až po  $(G_{v_1} * \dots * G_{v_{k-1}})$  a  $G_{v_k}$ .  $\square$

## 2.3 Dve metódy na porovnávanie vlastných čísel grafov

Predstavíme dve metódy a výsledky získané pomocou nich. Neskôr tieto metódy rozšírime a použijeme na nájdenie nových tried Schurovsky optimálnych návrhov experimentov.

Prvá metóda pracuje na základe výsledkov Kelmansa [33] a jeho spolupráce s Chelnokovom [34]. V prvej uvedenej práci Kelmans pracuje iba s  $t$ -optimalitou, ale jeho výsledky sa dajú rozšíriť na celú triedu ortogonálne invariantných kritérií optimality. V druhej uvedenej práci už porovnávajú Laplaceove polynómy grafov a predstavujú niekoľko tried najlepších grafov. Pracujú ale prevažne s grafmi s nízkym počtom hrán a tým pádom často nesúvislými grafmi, pre ktoré nemá zmysel skúmať optimalitu vzhľadom na naše kritériá.

Druhá metóda je založená na majorizácii vektorov vlastných hodnôt Laplaciánov grafov, teda sa dá ľahko previesť na hľadanie Schurovsky optimálnych návrhov. Za-

oberal sa ňou Constantine. V článku [9] dokázal pomocou tejto metódy rovnaké výsledky ako Kelmans.

### 2.3.1 Laplaceove polynómy a Kelmansova veta

Ukážeme využitie *rozšírenej Kelmansovej vety* 2.4 pri dôkaze konkrétneho tvrdenia o optimálnej augmentácii návrhu experimentu. Tento spôsob bude užitočný, keď budeme augmentovať *kompletné multipartitné grafy*.

**Veta 2.5** (Multipartitný graf s jednou pridanou hranou). *Majme daný kompletný multipartitný skoro-regulárny graf  $K_{v_1 \times k_1, v_2 \times k_2}$  s  $k_1$  partíciami veľkosti  $v_1$  a  $k_2$  partíciami veľkosti  $v_2$ , pričom platí  $v_1 = v_2 + 1$ , a celkový počet vrcholov je  $v = v_1 k_1 + v_2 k_2$ . Potom počet kostier  $\tau(G)$  jednoduchého grafu  $G$  vytvoreného z grafu  $K_{v_1 \times k_1, v_2 \times k_2}$  pridaním jednej hrany je*

- a)  $\tau(G) = \tau(K_{v_1 \times k_1, v_2 \times k_2}) \left(1 + \frac{2}{v - v_1}\right)$ ,  
ak pridáme hranu medzi ľub. dva vrcholy vo väčšej partícii s počtom hrán  $v_1$ .
- b)  $\tau(G) = \tau(K_{v_1 \times k_1, v_2 \times k_2}) \left(1 + \frac{2}{v - v_2}\right)$ ,  
ak pridáme hranu medzi ľub. dva vrcholy v menšej partícii s počtom hrán  $v_2$ .

**Dôsledok 2.1.** *Graf  $G$  je  $t$ -optimálny, spomedzi všetkých jednoduchých grafov vytvorených pridaním jednej hrany do grafu  $K_{v_1 \times k_1, v_2 \times k_2}$ , ak bol vytvorený pridaním hrany medzi dva ľubovoľné vrcholy väčšej partície.*

*Dôkaz.* (Dôsledok 2.1). Vieme že  $v_1 > v_2$  a  $v_1, v_2$  sú prirodzené čísla. Potom

$$\begin{aligned} \tau(K_{v_1 \times k_1, v_2 \times k_2}) \left(1 + \frac{2}{v - v_1}\right) &> \tau(K_{v_1 \times k_1, v_2 \times k_2}) \left(1 + \frac{2}{v - v_2}\right) \\ \Rightarrow \tau(K_{v_1 \times k_1, v_2 \times k_2} + e_{1,1}) &> \tau(K_{v_1 \times k_1, v_2 \times k_2} + e_{2,2}), \end{aligned}$$

kde  $e_{1,1}$  je hrana pridaná medzi dva ľubovoľné vrcholy v partícii veľkosti  $v_1$  (bez ujmy na všeobecnosti si môžeme vybrať ľubovoľnú z partície veľkosti  $v_1$  a v nej ľubovoľné dva vrcholy) a  $e_{2,2}$  je hrana pridaná medzi dva ľubovoľné vrcholy v partícii veľkosti  $v_2$ .  $\square$

Pri predošlej vete 2.5 a jej dôsledku 2.1 sa naskytá otázka, prečo neskúsime pridať hranu medzi partície. Hlavný dôvod je, že chceme pracovať iba s jednoduchými grafmi, teda s grafmi bez slučiek a bez násobných hrán. No a v prípade pridanía hrany medzi partície by sme vytvorili násobnú hranu. Takto vzniknutý graf by už nebol jednoduchý

a tým pádom si myslíme, že nemôže byť Schurovsky optimálny (viď. 3.3 v numerickej časti). Tým pádom sa snažíme nevytvárať iné ako jednoduché grafy.

*Dôkaz. (Veta 2.5).* Dokážeme časť *a*). Pri dôkaze časti *b*) sa postupuje analogicky. Vzniknutý graf  $K_{v_1 \times k_1, v_2 \times k_2} + e_{1,1}$  môžeme zapísať ako *spojenie*

$$K_{v_1 \times k_1, v_2 \times k_2} + e_{1,1} = \underbrace{G_{v_1}^0 * \dots * G_{v_1}^0}_{k_1-1} * \underbrace{G_{v_2}^0 * \dots * G_{v_2}^0}_{k_2} * G_{v_1}^I,$$

kde  $G_{v_1}^0$  je graf na  $v_1$  vrcholoch bez žiadnej hrany,  $G_{v_2}^0$  je graf na  $v_2$  vrcholoch bez žiadnej hrany a  $G_{v_1}^I$  je graf na  $v_1$  vrcholoch s jednou hranou  $e_{1,1}$  medzi dvomi vrcholmi. Laplaceove polynómy pre grafy  $G_{v_1}^0$ ,  $G_{v_2}^0$  a  $G_{v_1}^I$  v bode  $x$  sú  $\mu(G_{v_1}^0, x) = x^{v_1}$ ,  $\mu(G_{v_2}^0, x) = x^{v_2}$  a  $\mu(G_{v_1}^I, x) = x^{v_1-1}(x-2)$ . Tvorba týchto Laplaceových polynómov je ukázaná v prílohe A 3.3. Potom môžeme využiť rozšírenú Kelmansovu vetu 2.4 a vytvoriť hľadaný Laplaceov polynóm pre *spojenie*

$$\begin{aligned} \mu(K_{v_1 \times k_1, v_2 \times k_2} + e_{1,1}) &= \mu(\underbrace{G_{v_1}^0 * \dots * G_{v_1}^0}_{k_1-1} * \underbrace{G_{v_2}^0 * \dots * G_{v_2}^0}_{k_2} * G_{v_1}^I, x) = \\ &= \frac{x(x-v)^{k_1+k_2-1}}{(x-v+v_1)^{k_1}(x-v+v_2)^{k_2}} \cdot \mu(G_{v_1}^0, x-v+v_1)^{k_1-1} \mu(G_{v_2}^0, x-v+v_2)^{k_2} \mu(G_{v_1}^I, x-v+v_1) = \\ &= \frac{x(x-v)^{k_1+k_2-1}}{(x-v+v_1)^{k_1}(x-v+v_2)^{k_2}} \cdot (x-v+v_1)^{v_1(k_1-1)} (x-v+v_2)^{v_2 k_2} (x-v+v_1)^{v_1-1} (x-v+v_1-2) = \\ &= x(x-v)^{k_1+k_2-1} (x-v+v_1)^{v_1 k_1 - k_1 - 1} (x-v+v_2)^{v_2 k_2 - k_2} (x-v+v_1-2). \end{aligned}$$

Korene tohto polynómu sú vlastné čísla (usporiadame ich nerastúco)  $\lambda_1 = v$ ,  $\lambda_2 = v - v_1 + 2$ ,  $\lambda_3 = v - v_2$ ,  $\lambda_4 = v - v_1$  a  $\lambda_5 = 0$  s násobnosťami  $\alpha_1 = k_1 + k_2 - 1$ ,  $\alpha_2 = 1$ ,  $\alpha_3 = v_2 k_2 - k_2$ ,  $\alpha_4 = v_1 k_1 - k_1 - 1$  a  $\alpha_5 = 1$ . Potom počet kostier grafu  $K_{v_1 \times k_1, v_2 \times k_2} + e_{1,1}$  je (na základe vety 2.1)

$$\begin{aligned} \tau(K_{v_1 \times k_1, v_2 \times k_2} + e_{1,1}) &= \frac{v^{k_1+k_2-1} (v-v_1+2) (v-v_2)^{v_2 k_2 - k_2} (v-v_1)^{v_1 k_1 - k_1 - 1}}{v} = \\ &= \tau(K_{v_1 \times k_1, v_2 \times k_2}) \left( \frac{v-v_1+2}{v-v_1} \right) = \tau(K_{v_1 \times k_1, v_2 \times k_2}) \left( 1 + \frac{2}{v-v_1} \right). \end{aligned}$$

Kde počet kostier *kompletného multipartitného grafu*  $K_{v_1 \times k_1, v_2 \times k_2}$  vieme odvodiť rov-

nako:

$$\tau(K_{v_1 \times k_1, v_2 \times k_2}) = v^{k_1+k_2-1}(v-v_2)^{v_2 k_2 - k_2}(v-v_1)^{v_1 k_1 - k_1}. \quad (2.1)$$

Rovnakým spôsobom dostaneme výsledok z časti *b*)

$$\tau(K_{v_1 \times k_1, v_2 \times k_2} + e_{2,2}) = \tau(K_{v_1 \times k_1, v_2 \times k_2}) \left(1 + \frac{2}{v-v_2}\right).$$

□

Kelmans a mnoho ďalších sa zaoberalo hľadáním  $t$ -optimálnych grafov najmä kvôli informatickej aplikácii pri zabezpečovaní spoľahlivosti počítačových sietí (z ang. network reliability). Keďže sieťovanie počítačov je stále atraktívne, tak pre  $t$ -optimalitu pribúdajú naďalej nové výsledky. Na druhú stranu je väčšina výsledkov (nájdenných *Laplaceových* polynómov pre určité triedy grafov) veľmi špecifická a zaoberá sa špeciálne vymedzenými triedami grafov. Kelmans s Chelenkovom [34] odvodili *Laplaceove* polynómy pre viacero špecifických príkladov s malým počtom hrán, vzhľadom na počet vrcholov. Kelmans dokázal, ako sa správajú Laplaciány grafov pre viaceré druhy spájania dvoch grafov [33]. Ďalšie triedy grafov pre ktoré vieme porovnávať počet kostier predstavili ich nasledovníci. Napríklad Gilbert a Myrvold [19] vyjadrili formulu pre počet kostier triedy grafov, ktorých komplementy sú zjednotenia cyklov a ciest.

Najväčšou nevýhodou tohto postupu je, že pracuje len s jedným kritériom  $t$ -optimality, zatiaľ čo nás zaujímajú aj iné kritériá (v navrhovaní blokových experimentov sa častejšie používa  $A$ -optimality). My budeme pracovať s ortogonálne invariantnými kritériami optimality, ktoré v sebe zahŕňajú aj uvedenú  $t$ -optimalitu. Vetu 2.5 vieme rozšíriť z  $t$ -optimality na celú triedu ortogonálne invariantných kritérií nasledovne.

**Veta 2.6.** *Majme daný kompletný multipartitný skoro-regulárny graf  $K_{v_1 \times k_1, v_2 \times k_2}$  s  $k_1$  partíciami veľkosti  $v_1$  a  $k_2$  partíciami veľkosti  $v_2$ , pričom platí  $v_1 = v_2 + 1$  a celkový počet vrcholov je  $v = v_1 k_1 + v_2 k_2$ . Potom jednoduchý graf  $G$ , vytvorený z grafu  $K_{v_1 \times k_1, v_2 \times k_2}$  pridaním jednej hrany medzi dva ľubovoľné vrcholy v ľubovoľnej väčšej partícii, je Schurovsky optimálny na množine všetkých jednoduchých grafov vytvorených z  $K_{v_1 \times k_1, v_2 \times k_2}$  pridaním jednej hrany.*

*Dôkaz.* V tomto prípade sú iba dvaja kandidáti (keďže neberieme do úvahy izomorfné grafy a vzniknutý graf  $G$  musí byť jednoduchý) na najlepší graf spĺňajúci podmienky zo

zadania. Prvý možný graf je keď pridáme hranu medzi dva ľubovoľné vrcholy vo väčšej partícii a druhý keď pridáme hranu medzi dva ľubovoľné vrcholy v menšej partícii. Pre oba prípady sme už našli vektor vlastných čísel v predošlom dôkaze

$$\begin{aligned} \lambda(K_{v_1 \times k_1, v_2 \times k_2} + e_{1,1}) &= \\ &= \underbrace{(v, \dots, v)}_{k_1+k_2-1}, (v - v_1 + 2), \underbrace{(v - v_2), \dots, (v - v_2)}_{v_2 k_2 - k_2}, \underbrace{(v - v_1), \dots, (v - v_1)}_{v_1 k_1 - k_1 - 1}, 0)' \end{aligned}$$

$$\begin{aligned} \lambda(K_{v_1 \times k_1, v_2 \times k_2} + e_{2,2}) &= \\ &= \underbrace{(v, \dots, v)}_{k_1+k_2-1}, (v - v_2 + 2), \underbrace{(v - v_2), \dots, (v - v_2)}_{v_2 k_2 - k_2 - 1}, \underbrace{(v - v_1), \dots, (v - v_1)}_{v_1 k_1 - k_1}, 0)' \end{aligned}$$

Jednoduchým porovnaním týchto dvoch vektorov (s dosadením  $v_1 = v_2 + 1$ ) vidíme, že prvý vektor je majorizovaný druhým

$$\lambda(K_{v_1 \times k_1, v_2 \times k_2} + e_{1,1}) \prec \lambda(K_{v_1 \times k_1, v_2 \times k_2} + e_{2,2}).$$

Takže graf  $K_{v_1 \times k_1, v_2 \times k_2} + e_{1,1}$  je Schurovsky optimálny.  $\square$

*Rozšírenú Kelmansovú vetu* môžeme využívať napr. pre grafy tvorené množstvom menších partícií, kompletne pospajaných so všetkými ostatnými partíciami. Pričom do jednotlivých partícií môžeme pridávať ďalšie hrany.

### 2.3.2 Majorizácia podľa Constantineho

Nasledujúca časť je založená na výsledku G. M. Constantineho v článku [9] z roku 1983. Tento výsledok sa netýka augmentácie, ale práve naopak sa zaoberá odoberaním hrán. Použitý postup však vieme zovšeobecniť a využiť pri augmentácii.

Ukážeme dve Constantineho vety a hlavné časti ich dôkazov, ktoré budeme používať. V nich ukázané triedy najlepších jednoduchých grafov boli objavené skôr na základe Laplaceových polynómov v prácach [33] a [34]. Constantine ich odvodil iným spôsobom a rozšíril triedu kritérií, vzhľadom na ktoré sú nájdené grafy optimálne.

V pôvodnom znení viet pracovali s inak definovaným vektorom vlastných čísel, ktorý neobsahoval triviálne vlastné číslo rovné nule. My naďalej používame naše označenie, ale význam viet, ani dôkazy, toto iné značenie nemení.

**Veta 2.7** (Constantine 1). *Majme kompletný graf  $K_v$  na  $v$  vrcholoch. Graf vytvorený odobratím  $k$ -nesusedných hrán z grafu  $K_v$  je Schurovsky optimálny (viď obr. 2-1, a)).*

*Dôkaz.* Označme graf zo znenia vety, o ktorom tvrdíme že je *najlepší* ako  $G^*$ . Graf  $G^*$  má  $v$  vrcholov a  $\binom{v}{2} - k$  hrán. Nech  $H$  je ľubovoľný jednoduchý graf s rovnakým počtom vrcholov a hrán ako  $G^*$ . Cieľom je dokázať, že platí  $\lambda(G^*) \prec \lambda(H)$ . Grafy  $G^*$  a  $H$  majú veľké množstvo hrán, tým pádom sa s nimi ťažšie pracuje. Preto využijeme lemu 2.1, vďaka ktorej stačí porovnať iba  $\bar{G}^*$  a  $\bar{H}$ , ktoré majú iba  $k$  hrán. Teda potrebujeme dokázať  $\lambda(\bar{G}^*) \prec \lambda(\bar{H})$ .

Pre graf  $\bar{G}^*$  pozostávajúci z  $k$  nesusedných hrán vieme jednoducho nájsť vlastné čísla (3.3).

$$\lambda(\bar{G}^*) = (\underbrace{2, \dots, 2}_k, \underbrace{0, \dots, 0}_{v-k})'$$

Chceme ukázať, že graf  $\bar{H}$  má horší vektor vlastných čísel (vzhľadom na majorizáciu) ako je vektor vlastných čísel pre  $\bar{G}^*$ .

Ako prvé sa ukáže, že vektor  $\lambda(\bar{H})$  má aspoň toľko nulových členov ako vektor  $\lambda(\bar{G}^*)$ . Dokáže sa to na základe hodností Laplaciánov a počtov komponentov súvislosti. Ďalej sa nahradia všetky nenulové vlastné čísla grafu  $\bar{H}$  za ich aritmetický priemer  $\bar{h} = (\lambda_1(\bar{H}) + \lambda_2(\bar{H}) + \dots + \lambda_k(\bar{H}))/k$ . Z nich sa vytvorí vektor

$$\lambda(H_{\text{priemer}}) = (\underbrace{\bar{h}, \dots, \bar{h}}_k, \underbrace{0, \dots, 0}_{v-k})'$$

o ktorom sa dá ľahko ukázať že je majorizovaný vektorom  $\lambda(\bar{H})$ .

Nakoniec ukážeme, že  $\bar{h}$  musí byť rovný dvom a teda vektor  $\lambda(H_{\text{priemer}})$  je rovný vektoru vlastných čísel  $\lambda(\bar{G}^*)$ . Z čoho dostaneme

$$\lambda(\bar{G}^*) = \lambda(H_{\text{priemer}}) \prec \lambda(\bar{H}), \quad (2.2)$$

a teda

$$\lambda(G^*) \prec \lambda(H).$$

Čo sme chceli dokázať. □

Celý dôkaz je podrobne vysvetlený v práci [9]. Druhá Constantineho veta rozšíri tvrdenie prvej vety pre nepárny počet vrcholov.

**Veta 2.8** (Constantine 2). *Nech  $v$  je nepárne. Nech  $N$  je graf vytvorený z kompletného grafu  $K_v$  na  $v$  vrcholoch, odobratím ľubovoľných  $\frac{1}{2}(v-1)$  nesusedných hrán. Potom graf*



**Obrázok 2 – 1:** Ilustrácia k vetám 2.7 a 2.8. a) kompletný graf na šiestich vrchoch s odobratím troch nesusedných (červených) hrán; b) kompletný graf na siedmych vrchoch s odobratím troch nesusedných (červených) hrán a jednej ďalšej (modrej) hrany incidentnej s vrcholom stupňa šesť.

vytvorený z grafu  $N$  odobratím ľubovoľnej ďalšej hrany, incidentnej s vrcholom stupňa  $v - 1$ , je Schurovsky optimálny (viď obr. 2 – 1, b)).

*Dôkaz.* Dôkaz má podobnú myšlienku ako predošlý, ale je komplikovanejší a treba ošetriť niekoľko špeciálnych prípadov. Opäť označíme  $G^*$  hľadaný optimálny graf zo znenia vety a definujeme  $H$  ako ľubovoľný jednoduchý graf s rovnakým počtom vrcholov a hrán ako  $G^*$ . Opäť budeme pracovať s komplementárnymi grafmi  $\bar{G}^*$  a  $\bar{H}$ .

Nájdeme Laplacián a vektor vlastných čísel pre  $\bar{G}^*$  (viď. príloha B, 3.3)

$$\boldsymbol{\lambda}(\bar{G}^*) = (3, \underbrace{2, \dots, 2}_{(v-3)/2}, 1, \underbrace{0, \dots, 0}_{(v-1)/2})'.$$

Zistíme minimálny počet nulových vlastných čísel vo vektore  $\boldsymbol{\lambda}(\bar{H})$ . Ďalej ohraničíme dve najväčšie vlastné čísla vektoru  $\boldsymbol{\lambda}(\bar{H})$  pomocou dvoch najväčších vlastných čísel vektoru  $\boldsymbol{\lambda}(\bar{G}^*)$ . Potom využijeme aritmetický priemer zvyšných vlastných čísel, rovnako ako v predošlom dôkaze a vytvoríme vektor

$$\boldsymbol{\lambda}(H_{\text{priemer}}) = (\lambda_1(\bar{H}), \lambda_2(\bar{H}), \underbrace{\bar{h}, \dots, \bar{h}}_{(v-3)/2}, \underbrace{0, \dots, 0}_{(v-1)/2})'.$$

Potom ukážeme

$$\boldsymbol{\lambda}(\bar{G}^*) = \boldsymbol{\lambda}(H_{\text{priemer}}) \prec \boldsymbol{\lambda}(\bar{H}), \quad (2.3)$$

a teda

$$\boldsymbol{\lambda}(G^*) \prec \boldsymbol{\lambda}(H).$$

Čo sme chceli dokázať. □

Celé znenie dôkazu sa dá nájsť v článku [9]. Constantine navyše tvrdí, že ak odoberieme viac ako  $(v + 1)/2$  hrán, tak neexistuje Schurovsky najlepší návrh a pre



hľadanie optimálnych návrhov musíme pracovať s konkrétnymi kritériami optimality. Toto tvrdenie je vyslovené na konci článku [9], pričom nie je dokázané a neodkazuje sa na žiadnu ďalšiu prácu. K tomuto tvrdeniu sa vrátíme v numerickej časti 3.1.

Pre nás najdôležitejšia vlastnosť uvedených viet je, že sa pracovalo s najlepšimi grafmi vzhľadom na ortogonálne invariantné kritériá optimality. Tým pádom tieto vety budeme môcť využiť pre našu augmentáciu, najmä tvrdenia 2.2 a 2.3.

## 2.4 Nové triedy optimálnych augmentovaných návrhov

Vhodným použitím dvoch metód predstavených vyššie, sme objavili tri nové triedy optimálnych augmentácií návrhov experimentov.

### 2.4.1 Augmentácia kompletného multipartitného grafu I

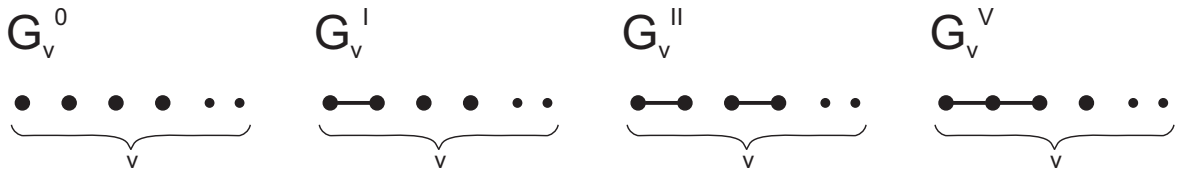
Využitím Laplaceových polynómov a Kelmansových viet vieme nájsť optimálne augmentácie jednoduchých kompletných skoro-regulárnych multipartitných grafov pri pridaní dvoch a troch hrán. Zaujímajú nás len skoro-regulárne grafy, vďaka práci [44]. Tým pádom počty vrcholov v ľubovoľných dvoch partiáciách sa líšia maximálne o jedna.

Rovnakým spôsobom by sa dali nájsť optimálne augmentácie aj pre väčší počet pridaných hrán, ale potom by už bolo treba prechádzať veľa možností ako pridať hrany - tento problém vyriešime všeobecnejšie neskôr. Pri pridávaní dvoch a troch hrán budeme postupovať úplne rovnako ako vo vete 2.5 a 2.6. Prejdeme všetky možné typy pridania hrán a pozrieme sa na vektory vlastných čísel vytvorených grafov. Potom tieto vektory porovnáme a nájdeme *najlepší* vzhľadom na majorizáciu.

#### Dve pridané hrany:

Je sedem neizomorfných možností ako vytvoriť jednoduchý graf pridaním dvoch hrán do  $K_{v_1 \times k_1, v_2 \times k_2}$ ,  $v_1 = v_2 + 1$ . Jednotlivé hrany môžeme pridávať iba medzi vrcholy v rovnakej partiícii, inak by sme dostali graf s násobnými hranami. Všetkých sedem typov grafov vieme dostať ako *spojenie* fragmentov z obrázka 2 – 2.

Prvý typ grafov sú grafy izomorfné s  $H_1 = K_{v_1 \times k_1, v_2 \times k_2} + e_{1,1} + e_{1,1} = K_{v_1 \times (k_1 - 1), v_2 \times k_2} * G_{v_1}^{II}$ . Prvý spôsob zápisu hovorí, že graf  $H_1$  vznikol z grafu  $K_{v_1 \times k_1, v_2 \times k_2}$  pridaním dvoch nesusedných hrán do partiície s  $v_1$  vrcholmi. Tento zápis je vhodnejší na sledovanie zmien v pôvodnom grafe počas augmentácie. Druhý spôsob zápisu ukazuje ako vieme daný graf vytvoriť ako *spojenie*  $K_{v_1 \times (k_1 - 1), v_2 \times k_2}$  a fragmentu  $G_{v_1}^{II}$ . Tento zápis umožňuje rovno použiť *Kelmansovu vetu* a nájsť vektor vlastných čísel jednotlivých typov grafov.



**Obrázok 2–2:** Možné fragmenty multipartitného grafu s pridanými dvomi hranami, kde  $v$  je počet vrcholov v partícii.  $G_v^0$  je jedna partícia grafu s nulou pridanými hranami;  $G_v^I$  je jedna partícia grafu s jednou pridanou hranou;  $G_v^{II}$  je jedna partícia grafu s dvomi pridanými nesusednými hranami;  $G_v^V$  je jedna partícia grafu s dvomi pridanými susednými hranami.

Pomocou druhého zápisu ukážeme zvyšných šesť kandidátov na najlepší graf.

$$\begin{aligned}
 H_2 &= K_{v_1 \times k_1, v_2 \times (k_2 - 1)} * G_{v_2}^{II} & H_5 &= K_{v_1 \times (k_1 - 2), v_2 \times k_2} * G_{v_1}^I * G_{v_1}^I \\
 H_3 &= K_{v_1 \times (k_1 - 1), v_2 \times k_2} * G_{v_1}^V & H_6 &= K_{v_1 \times k_1, v_2 \times (k_2 - 2)} * G_{v_2}^I * G_{v_2}^I \\
 H_4 &= K_{v_1 \times k_1, v_2 \times (k_2 - 1)} * G_{v_2}^V & H_7 &= K_{v_1 \times (k_1 - 1), v_2 \times (k_2 - 1)} * G_{v_1}^I * G_{v_2}^I
 \end{aligned}$$

Vieme ľahko vyjadriť Laplaceove polynómy pre jednotlivé fragmenty (príloha A, 3.3) a poznáme ich aj pre kompletne multipartitné grafy (viď. [38]). Tým pádom (rovnako ako v dôkaze vety 2.5) vieme nájsť vektory vlastných čísel pre všetkých sedem typov augmentovaných grafov. Tieto vektory sú zobrazené v tabuľke 2–1.

V prvom stĺpci sú mená porovnávaných grafov. V ďalších stĺpcoch sú vlastné čísla jednotlivých grafov zoradené zostupne. Okrem grafov a ich vlastných čísel je v tabuľke aj porovnanie vlastných čísel grafu s vlastnými číslami grafu  $H_1$ . Pričom porovnáваме vždy súčet prvých niekoľkých vlastných čísel až po miesto kde je zapísaný výsledok porovnania. Toto porovnávanie je presne porovnanie z definície majorizácie 21. Symbol  $\Sigma \parallel$  znamená, že súčet predošlých vlastných čísel je rovnaký ako súčet odpovedajúcich vlastných čísel pre graf  $H_1$ . Symbol  $\Sigma \wedge$  hovorí to isté, len namiesto rovná sa je symbol menší.

Na základe porovnania vektorov vlastných čísel vidno, že vektor vlastných čísel grafu  $H_1$  majorizuje všetky ostatné vektory vlastných čísel, teda návrh prislúchajúci ku grafu  $H_1$  je Schurovsky optimálny. Zároveň je najlepší aj vektor  $H_5$ , ktorý má rovnaké vlastné čísla ako  $H_1$ . Keďže sme prešli všetky neizomorfné možnosti ako pridať hrany, tak platí nasledovná veta.

**Veta 2.9.** *Schurovsky optimálnu augmentáciu kompletneho skoro-regulárneho multipartitného grafu na jednoduchý graf pridaním dvoch hrán dostaneme, ak tieto hrany pridáme ako nesusedné hrany medzi ľubovoľné vrcholy z väčších partícií.*

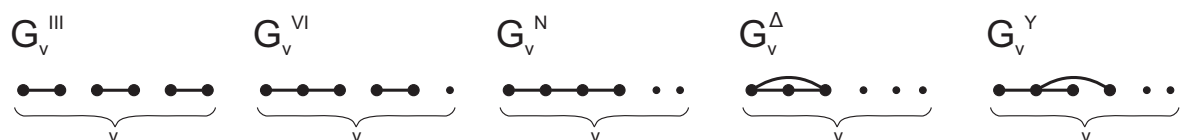


Je dôležité si uvedomiť, že je jedno či nesusedné hrany pridávame len do jednej partície s väčším počtom vrcholov, alebo každú hranu do inej partície s väčším počtom vrcholov. Vo všetkých prípadoch dostaneme Schurovsky optimálny graf s rovnakými vlastnými číslami. Jediné čo treba dodržať je, že vzniknutý graf musí byť *jednoduchý*, teda jednotlivé pridané hrany musia mať vždy oba vrcholy v rovnakej partícii.

**Tri pridané hrany:**

Možností, ako vytvoriť rôzne typy jednoduchých grafov pridaním troch hrán do  $K_{v_1 \times k_1, v_2 \times k_2}$ , je viacero. Aby sme nemuseli prechádzať všetky, treba si uvedomiť, že stačí pracovať s pridávaním hrán do väčšej partície (veľkosti  $v_1$ ).

Ak by sme pridali všetky tri hrany nejakým spôsobom do väčšej partície (vytvoríme graf  $G_1$ ) a potom by sme pridali rovnakým spôsobom všetky tri hrany do menšej partície (vytvoríme graf  $G_2$ ), tak je ľahko ukázateľné, že vektor vlastných čísel grafu  $G_1$  majorizuje vektor vlastných čísel grafu  $G_2$ . Podobne ak pri pridávaní hrán do rôznych partícií máme na výber či pridať hranu do menšej partície veľkosti  $v_2$  alebo do väčšej partície veľkosti  $v_1$ , tak pridanie hrany do väčšej partície povedie k lepšiemu (vzhľadom na majorizáciu) vektoru vlastných čísel grafu. Takže stačí porovnať možné pridania hrán do väčšej partície a pridania hrán do rôznych väčších partícií. Potom jednotlivé typy vzniknutých grafov vieme dostať ako *spojenie* predošlých fragmentov 2–2 a novej sady fragmentov pre tri pridané hrany 2–3.



**Obrázok 2–3:** Možné fragmenty multipartitného grafu s pridanými dvomi hranami, kde  $v$  je počet vrcholov v partícii.  $G_v^{III}$  je jedna partícia grafu s tromi pridanými nesusednými hranami;  $G_v^{VI}$  je jedna partícia grafu s dvomi pridanými susednými hranami a jednou s nimi nesusednou hranou;  $G_v^N$  je jedna partícia grafu s pridanými tromi hranami, pričom dve dvojice hrán sú susedné a jedna dvojica hrán je nesusedná;  $G_v^\Delta$  je jedna partícia grafu s pridanými tromi po dvoch susednými hranami susediacimi v troch rôznych bodoch;  $G_v^Y$  je jedna partícia grafu s pridanými tromi hranami incidentnými s rovnakým bodom.

Prvý typ grafov sú grafy izomorfné s  $J_1 = K_{v_1 \times k_1, v_2 \times k_2} + e_{11} + e_{11} + e_{11} = K_{v_1 \times (k_1 - 1), v_2 \times k_2}^*$   $G_{v_1}^{III}$ . Používame rovnaký spôsob zápisu ako v prípade pridania dvoch hrán. Opäť je pre nás vhodnejší druhý spôsob zápisu. Z prvého je síce lepšie vidno akým spôsobom sme nové grafy vytvorili, ale od druhého spôsobu zápisu vieme rovno prejsť k použitiu *Kelmansovej vety*. Máme ďalších šesť možných typov grafov.

$$J_2 = K_{v_1 \times (k_1-1), v_2 \times k_2} * G_{v_1}^{VI}$$

$$J_5 = K_{v_1 \times (k_1-1), v_2 \times k_2} * G_{v_1}^Y$$

$$J_3 = K_{v_1 \times (k_1-1), v_2 \times k_2} * G_{v_1}^N$$

$$J_6 = K_{v_1 \times (k_1-2), v_2 \times k_2} * G_{v_1}^{II} * G_{v_1}^I$$

$$J_4 = K_{v_1 \times (k_1-1), v_2 \times k_2} * G_{v_1}^{\Delta}$$

$$J_7 = K_{v_1 \times (k_1-3), v_2 \times k_2} * G_{v_1}^I * G_{v_1}^I * G_{v_1}^I$$

Vektory vlastných čísel jednotlivých grafov vypočítame rovnako ako v predošlom prípade (výsledky pre dielčie fragmenty sú v prílohe A, 3.3) a nájdeme ich v tabuľke 2–2. Tabuľka je vyplnená rovnakým spôsobom ako predošlá tabuľka 2–1.

Na základe porovnania vektorov vlastných čísel v nej vidno, že vektory vlastných čísel grafov  $J_1$ ,  $J_6$  a  $J_7$  sú rovnaké a majorizujú vektory vlastných čísel všetkých ostatných grafov. Teda k nim prislúchajúce návrhy sú Schurovsky optimálne. Keďže sme prešli všetky možnosti ako pridať hrany, tak platí nasledovná veta.

**Veta 2.10.** *Schurovsky optimálnu augmentáciu kompletného skoro-regulárneho multipartitného grafu na jednoduchý graf pridaním troch hrán dostaneme, ak tieto hrany pridáme ako nesusedné hrany medzi ľubovoľné vrcholy z väčších partícií.*

Opäť je jedno do ktorých z väčších partícií pridávame nesusedné hrany, pretože všetky možnosti majú rovnaké vlastné čísla.

Uvedené tvrdenia sa dajú v určitých prípadoch zovšeobecniť aj pre väčší počet hrán, ako bude ukázané v časti 2.4.4.

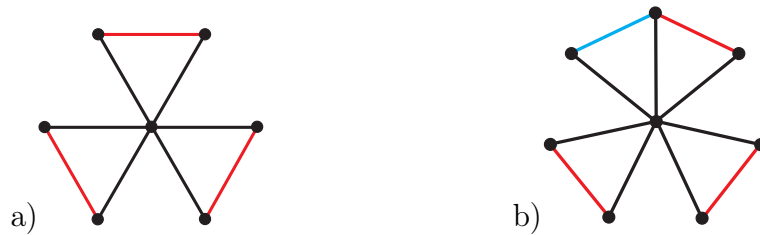
## 2.4.2 Augmentácia hviezdicového grafu

Pri optimálnej augmentácii hviezdicového návrhu a k nemu prislúchajúcemu hviezdicovému grafu využijeme obe predstavené metódy. Z dôkazov Constantineho viet 2.7 a 2.8 využijeme, že množina  $k$  nesusedných hrán je pre daný počet vrcholov a hrán *najlepšia* (tvrdenia 2.2 a 2.3). Pomocou *zovšeobecnenej Kelmansovej vety* 2.4 zase nájdeme vlastné čísla pre celý výsledný graf.

**Veta 2.11.** *Majme hviezdicový graf  $S_v$  na  $v$  vrchoch s  $v - 1$  hranami. Jednoduchý graf, vytvorený pridaním  $k$ -nesusedných hrán do grafu  $S_v$ , je Schurovsky najlepšia augmentácia grafu  $S_v$  (obr. 2–4, a)).*

**Veta 2.12.** *Nech  $v$  je nepárne. Jednoduchý graf  $N$ , vytvorený pridaním  $(v - 1)/2$  nesusedných hrán a ešte jednej hrany incidentnej s vrcholom stupňa  $v - 1$  do hviezdicového grafu  $S_v$  na  $v$  vrchoch, je Schurovsky najlepšia augmentácia grafu  $S_v$  (obr. 2–4, b)).*





**Obrázok 2 – 4:** Ilustrácia k vetám 2.11 a 2.12. a) hviezdicový graf na šiestich vrcholoch s pridaním troch nesusedných (červených) hrán; b) hviezdicový graf na siedmich vrcholoch s pridaním troch nesusedných (červených) hrán a jednej ďalšej (modrej) hrany incidentnej s vrcholom stupňa šesť.

Tvrdenia viet 2.11 a 2.12 budeme dokazovať súčasne. Jediný rozdiel v dôkazoch je, že v určitej časti použijeme v jednej vete tvrdenie 2.2 a v druhej vete tvrdenie 2.3. Inak sú dôkazy identické. Skôr ako začneme samotný dôkaz uvedieme lemu.

*Dôkaz. (veta 2.11 a veta 2.12).* Označme grafy zo znenia viet o ktorých tvrdíme, že sú *najlepšie* ako  $H^*$ . Z dôkazov viet 2.7 a 2.8 vieme, že graf tvorený množinou nesusedných hrán, alebo množinou nesusedných hrán a ešte jednej vhodne pridanej hrany (označovali sa  $\bar{G}^*$ ), má vektor vlastných čísel majorizovaný vektorom vlastných čísel ľubovoľného iného jednoduchého grafu  $G$  na rovnakom počte vrcholov a hrán, teda  $\lambda(\bar{G}^*) \prec \lambda(G)$ . Toto sú presne hrany, ktoré pridávame do *hviezdicového grafu*  $S_v$ .

Majme ľubovoľný jednoduchý graf  $H$  na  $v$  vrcholoch vytvorený z *hviezdicového grafu*  $S_v$  pridaním  $k$  ľubovoľných hrán. Vieme ho rozdeliť na dva grafy. Na graf  $G_h$  obsahujúci hlavný vrchol *hviezdicového grafu* stupňa  $v - 1$  (stred hviezdice) a na graf  $G_z$  obsahujúci zvyšných  $v - 1$  vrcholov a hrany medzi nimi (čo je presne  $k$  pridaných hrán do  $S_v$  zo znenia vety). Pôvodný graf  $H$  vieme vyjadriť ako *spojenie*

$$H = G_h * G_z.$$

*Laplaceov polynóm* grafu  $H$  vieme vyjadriť pomocou *Kelmansovej vety* 2.3 a *Laplaceových polynómov* pre  $G_h$  a  $G_z$ . Kde  $\mu(G_h, x) = x$  a  $G_h$  má jeden vrchol,  $\mu(G_z, x) = (x - \lambda_1(G_z))(x - \lambda_2(G_z)) \cdots (x - \lambda_{v-1}(G_z))$  a  $G_z$  má  $v - 1$  vrcholov. Vieme, že triviálna vlastná hodnota je  $\lambda_{v-1}(G_z) = 0$  a najväčšia vlastná hodnota je  $\lambda_1(G_z) \leq v - 1$  (vďaka vete 2.2).

Potom

$$\begin{aligned}\mu(H, x) &= \mu(G_h * G_z, x) = \frac{x(x-1-(v-1))}{(x-1)(x-(v-1))} \mu(G_h, x-(v-1)) \mu(G_z, x-1) = \\ &= \frac{x(x-v)}{(x-1)(x-v+1)} (x-v+1)(x-1-\lambda_1(G_z))(x-1-\lambda_2(G_z)) \cdots (x-1-\lambda_{v-1}(G_z)).\end{aligned}$$

Keďže  $\lambda_{v-1}(G_z)$  je rovná nule (triviálna), tak posledná zátvorka bude rovná  $(x-1)$  a vykrátí sa s časťou menovateľa zlomku. Tým pádom dostávame

$$\mu(H, x) = x(x-v)(x-1-\lambda_1(G_z))(x-1-\lambda_2(G_z)) \cdots (x-1-\lambda_{v-2}(G_z)).$$

Potom vlastné čísla grafu  $H$  sú  $v, 1 + \lambda_1(G_z), 1 + \lambda_2(G_z), \dots, 1 + \lambda_{v-2}(G_z)$  a triviálne vlastné číslo nula. Vektor vlastných čísel bude

$$\boldsymbol{\lambda}(H) = (v, 1 + \lambda_1(G_z), 1 + \lambda_2(G_z), \dots, 1 + \lambda_{v-2}(G_z), 0)'$$

Takýto vektor dostaneme pre ľubovoľné pridané hrany do *hviezdicového grafu* (tak aby vznikol jednoduchý graf), pričom závisí len na vlastných číslach podgrafu  $G_z$  obsahujúceho pridané hrany. Najväčším vlastným číslom pre ľubovoľný jednoduchý graf vytvorený z *hviezdicového* bude vždy  $v$ , pretože  $\lambda_1(G_z) \leq v-1$ . Tým pádom najväčšie vlastné číslo nemusíme, v rámci majorizácie, porovnávať medzi rôznymi grafmi a stačí porovnať ostatné vlastné čísla. Tie sú  $1 + \lambda_1(G_z), 1 + \lambda_2(G_z), \dots, 1 + \lambda_{v-2}(G_z)$ . Potom vďaka leme 1.1 a výsledkov z dôkazov Constantineho ( $\boldsymbol{\lambda}(\bar{G}^*) \prec \boldsymbol{\lambda}(G_z)$  pre  $v(\bar{G}^*) = v(G_z) = v-1$  a  $e(\bar{G}^*) = e(G_z) = k$ , a posledné vlastné číslo nulové) dostávame

$$\begin{aligned}\boldsymbol{\lambda}(H^*) &= (v, 1 + \lambda_1(\bar{G}^*), 1 + \lambda_2(\bar{G}^*), \dots, 1 + \lambda_{v-2}(\bar{G}^*), 0)' \prec \\ &\prec (v, 1 + \lambda_1(G_z), 1 + \lambda_2(G_z), \dots, 1 + \lambda_{v-2}(G_z), 0)' = \boldsymbol{\lambda}(H),\end{aligned}$$

pre ľubovoľný jednoduchý graf  $H$  na rovnakom počte vrcholov a hrán ako  $H^*$ . Teda  $H^*$  je *najlepšia*.  $\square$

### 2.4.3 Augmentácia kompletného grafu

Tvrdenia ktoré vyslovíme sú dokázateľné podobnými technikami ako Constantineho vety 2.7 a 2.8, ale zatiaľ neboli dostatočne zaujímavé motivované, na to aby sa doka-

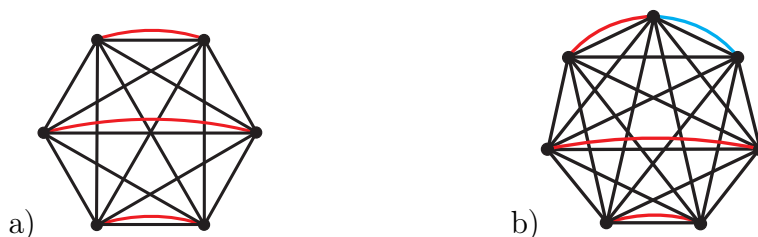


zovali. Naša motivácia augmentáciou už vykonaných návrhov je úplne prirodzená a vyžaduje tieto nové tvrdenia. Jeden z hlavných z dôvodov prečo doteraz neboli potrebné je, že v teórii grafov sa častejšie pracuje s jednoduchými grafmi.

V našom prípade je to naopak - odoberanie hrán v pôvodných Constantineho vetách vedie k grafom Schurovsky optimálnych návrhov, ktoré sa pri navrhovaní experimentov nevyužívajú veľmi často. Naopak pridávanie hrán do optimálneho kompletného grafu je pekne motivované jednoduchou augmentáciou. Síce pri tom vznikajú násobné hrany, ale tie zodpovedajú opakovaniam experimentu za rovnakých podmienok, čo je pre štatistické pozorovania úplne prirodzené.

**Veta 2.13.** *Majme kompletný graf  $K_v$  na  $v$  vrchoch. Graf vytvorený pridaním  $k$  nesusedných hrán do grafu  $K_v$  je Schurovsky najlepšia augmentácia grafu  $K_v$  (viď obr. 2–5, a)).*

**Veta 2.14.** *Nech  $v$  je nepárne. Nech  $N$  je graf vytvorený z kompletného grafu  $K_v$  na  $v$  vrchoch pridaním ľubovoľných  $\frac{1}{2}(v-1)$  nesusedných hrán. Potom graf vytvorený z grafu  $N$  pridaním ľubovoľnej ďalšej hrany incidentnej s vrcholom stupňa  $v-1$  je Schurovsky najlepšia augmentácia grafu  $K_v$ . (viď obr. 2–5, b)).*



**Obrázok 2–5:** Ilustrácia k vetám 2.13 a 2.14. a) kompletný graf na šiestich vrchoch s pridaním troch nesusedných (červených) hrán; b) kompletný graf na siedmich vrchoch s pridaním troch nesusedných (červených) hrán a jednej ďalšej (modrej) hrany incidentnej s vrcholom stupňa šesť.

Na dôkaz vety 2.14 použijeme nasledovnú lemu a Constantineho vety. V leme budeme používať namiesto značenia  $\lambda_i(G)$  značenie  $\lambda_i^G$  aby sa nemýlili vlastné čísla s násobením Laplaciánom grafu. Tak isto vlastný vektor Laplaciánu grafu  $u_i(\mathbf{L}(G))$  budeme skrátene označovať  $u_i^G$ .

**Lema 2.2.** *Keď zjednotíme kompletný graf  $K_v$  na  $v$  vrchoch (s vlastnými číslami  $\lambda_1^{K_v}, \dots, \lambda_v^{K_v} = 0$ ) s ľubovoľným jednoduchým grafom  $G$  na  $v$  vrchoch (s vlastnými číslami  $\lambda_1^G, \dots, \lambda_v^G = 0$ ), tak vektor vlastných čísel vzniknutého grafu je*

$$\boldsymbol{\lambda}^{K_v+G} = (\lambda_1^{K_v} + \lambda_1^G, \dots, \lambda_v^{K_v} + \lambda_v^G)'.$$

*Dôkaz. (lema 2.2).* Vlastné čísla kompletneho grafu  $K_v$  sú všetky rovnaké až na triviálne nulové vlastné číslo (viď. 3.3)

$$\boldsymbol{\lambda}^{K_v} = (\underbrace{v, \dots, v}_{v-1}, 0)'.$$

K vlastným číslam  $\lambda_1^{K_v}, \dots, \lambda_{v-1}^{K_v}$  prislúcha  $v-1$  lineárne nezávislých vlastných vektorov  $u_1^{K_v}, \dots, u_{v-1}^{K_v}$ , pričom pracujeme s normalizovanými vlastnými vektormi s veľkosťou jedna. Tieto vlastné vektory generujú  $(v-1)$ -rozmerný priestor (hodnosť Laplaciánu  $K_v$  je  $v-1$ ), ktorý tvorí ortogonálny doplnok k vektoru  $u_v^{K_v}$  prislúchajúcemu k triviálnemu vlastnému číslu rovnému nule. Tento vektor je normalizovaný vektor samých jednotiek, teda  $u_v^{K_v} = (1/\sqrt{v}, \dots, 1/\sqrt{v})'$ .

Označme ortonormálne vlastné vektory prislúchajúce k vlastným číslam Laplaciánu grafu  $G$  postupne  $u_1^G, \dots, u_{v-1}^G, u_v^G$ . Posledný vlastný vektor  $u_v^G$  prislúchajúci k nulovému vlastnému číslu, je opäť  $(1/\sqrt{v}, \dots, 1/\sqrt{v})'$ . Tým pádom ostatné vlastné vektory generujú tiež  $(v-1)$ -rozmerný priestor (hodnosť Laplaciánu  $G$  je  $v-1$ ) tvoriaci ortogonálny doplnok k rovnakému vektoru ako pri grafe  $K_v$ . Vďaka tomu vieme ľubovoľný vlastný vektor  $u_i^G, i = 1, \dots, v-1$  zapísať ako lineárnu kombináciu netriviálnych normalizovaných vlastných vektorov Laplaciánu grafu  $K_v$

$$u_i^G = \sum_{j=1}^{v-1} \alpha_{ij} u_j^{K_v}, \quad i = 1, \dots, v-1; \quad \sum_{j=1}^{v-1} \alpha_{ij}^2 = 1.$$

Potom vieme vybrať vlastné vektory Laplaciánu grafu vzniknutého ako *zjednotenie* grafov  $K_v$  a  $G$  rovnaké, ako vlastné vektoru grafu  $G$ . Potom pre všetky  $i = 1, \dots, v-1$  platí

$$\begin{aligned} u_i^G \mathbf{L}(K_v + G) &= u_i^G (\mathbf{L}(K_v) + \mathbf{L}(G)) = u_i^G \mathbf{L}(K_v) + u_i^G \mathbf{L}(G) = \\ &= \sum_{j=1}^{v-1} \alpha_{ij} u_j^{K_v} \mathbf{L}(K_v) + u_i^G \mathbf{L}(G) = \sum_{j=1}^{v-1} \alpha_{ij} \lambda_j^{K_v} u_j^{K_v} + \lambda_i^G u_i^G = \\ &= \sum_{j=1}^{v-1} \alpha_{ij} v u_j^{K_v} + \lambda_i^G u_i^G = v \sum_{j=1}^{v-1} \alpha_{ij} u_j^{K_v} + \lambda_i^G u_i^G = \\ &= v u_i^G + \lambda_i^G u_i^G = (v + \lambda_i^G) u_i^G. \end{aligned}$$

Pričom sme využili, že vlastné vektory prislúchajúce Laplaciánu grafu  $G$  vieme vy-

jadriť ako lineárnu kombináciu vlastných vektorov prislúchajúcich Laplaciánu grafu  $K_v$ . Navyše sú všetky netriviálne vlastné čísla grafu  $K_v$  rovnaké. Tým pádom dostávame, že pre  $i = 1, \dots, v-1$ , sú vlastné čísla grafu  $K_v + G$  rovné  $v + \lambda_i^G = \lambda_i^{K_v} + \lambda_i^G$ . Posledné vlastné číslo Laplaciánu grafu  $K_v + G$  musí byť triviálna nula (pretože je to najmenšie vlastné číslo Laplaciánu), ktorú dostaneme ako  $\lambda_v^{K_v+G} = \lambda_v^{K_v} + \lambda_v^G$ . Čo sme chceli dokázať.  $\square$

*Dôkaz. (veta 2.13 a veta 2.14).* V dôkazoch viet 2.7 a 2.8 bolo ukázané, že grafy tvorené množinou nesusedných hrán, alebo množinou nesusedných hrán a ešte jednej vhodne pridanej hrany (označme ich  $\bar{G}^*$ ) majú vektor vlastných čísel majorizovaný vektorom vlastných čísel ľubovoľného iného jednoduchého grafu  $G$  na rovnakom počte vrcholov a hrán, teda  $\lambda(\bar{G}^*) \prec \lambda(G)$ . Vďaka leme 2.2 vieme nasledovné

$$\lambda(K_v + G) = ((v-1) + \lambda_1(G), \dots, (v-1) + \lambda_{v-1}(G), (v-1))' = (v-1)\mathbf{1}_v + \lambda(G),$$

pre ľubovoľný jednoduchý graf  $G$ . Potom využitím lemy 1.1 dostávame, že ak  $\lambda(\bar{G}^*) \prec \lambda(G)$ , tak

$$\lambda(K_v + \bar{G}^*) = (v-1)\mathbf{1}_v + \lambda(\bar{G}^*) \prec (v-1)\mathbf{1}_v + \lambda(G) = \lambda(K_v + G).$$

Čo sme chceli dokázať.  $\square$

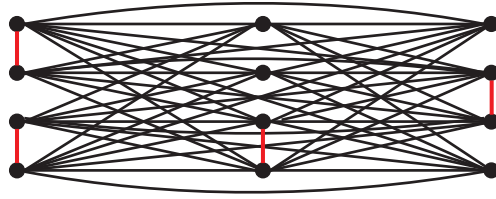
#### 2.4.4 Augmentácia kompletného multipartitného grafu II

Budeme postupovať podobne ako pri *hviezdicovom grafe* a opäť využijeme obe predstavené metódy. Budeme pracovať s *kompletným multipartitným grafom*, ktorý má všetky partície s rovnakým počtom vrcholov.

**Veta 2.15.** *Majme daný kompletný multipartitný graf  $K_{v_1 \times k_1}$  s  $k_1$  rovnako veľkými partíciami, veľkosti  $v_1$  a celkovým počtom vrcholov  $v = v_1 k_1$ . Jednoduchý graf  $H^*$ , vytvorený pridaním  $k$ -nesusedných hrán do ľubovoľných partícií  $K_{v_1 \times k_1}$ , je Schurovsky najlepšia augmentácia grafu  $K_{v_1 \times k_1}$ . (viď obr. 2–6.)*

Treba si uvedomiť, že počet pridaných hrán  $k$  môže byť  $k = 1, 2, \dots, k_1 \cdot \lfloor v_1/2 \rfloor$ .

*Dôkaz.* Chceme dokázať, že vektor vlastných čísel grafu  $H^*$  je majorizovaný vektorom vlastných čísel ľubovoľného iného jednoduchého grafu  $H$ , ktorý vznikol z  $K_{v_1 \times k_1}$  pridaním  $k$  hrán (keďže graf  $H$  má byť jednoduchý, tak hrany doň môžeme pridávať iba vrámci partícií a nie medzi partície).



**Obrázok 2 – 6:** Ilustrácia k vete 2.15. Kompletný multipartitný graf  $K_{v_1 \times k_1}$  s pridaním štyroch nesusedných (červených) hrán.

Označme grafy tvorené vrcholmi a hranami jednotlivých  $k_1$  partícií grafu  $H$  ako  $H_1, \dots, H_{k_1}$ . Teda  $H_i$  je graf obsahujúci vrcholy  $i$ -tej partície a všetky hrany medzi týmito vrcholmi (keďže nové hrany pridávame iba medzi partície, tak každá pridaná hrana bude obsiahnutá v niektorom z grafov  $H_i$ ). Platí, že  $H_1, \dots, H_{k_1}$  sú disjunktné a ich zjednotenie tvorí graf na všetkých vrcholoch grafu  $H$ , s  $k$  pridanými hranami. Nazvime tento graf  $H_{\text{pridané}} = H_1 + \dots + H_{k_1}$  a rovnakým spôsobom definujeme aj  $H_{\text{pridané}}^* = H_1^* + \dots + H_{k_1}^*$ . Navyše pôvodný graf  $H$  (vytvorený z grafu  $K_{v_1 \times k_1}$  pridaním  $k$  hrán) vieme vyjadriť ako *spojenie*  $H = H_1 * H_2 * \dots * H_{k_1}$ . Analogicky vieme vytvoriť grafy  $H_1^*, \dots, H_{k_1}^*$  pre graf  $H^*$ . Pričom platí  $H^* = H_1^* * H_2^* * \dots * H_{k_1}^*$ .

Nech  $\lambda_1(H_i), \dots, \lambda_{v_1}(H_i)$  sú vlastné čísla  $H_i$ , pre ľubovoľné  $i = 1, \dots, k_1$ . Potom Laplaceov polynóm pre graf  $H$  v bode  $x$  vyzerá na základe *rozšírenej Kelmansovej vety* 2.4 nasledovne

$$\begin{aligned} \mu(H, x) &= \frac{x(x-v)^{k_1-1}}{(x-v+v_1)^{k_1}} \prod_{i=1}^{k_1} \mu(G_i, x-v+v_1) = \\ &= \frac{x(x-v)^{k_1-1}}{(x-v+v_1)^{k_1}} \prod_{i=1}^{k_1} (x-v+v_1 - \lambda_1(H_i)) \cdot \dots \cdot (x-v+v_1 - \lambda_{v_1}(H_i)). \end{aligned}$$

Vieme, že posledné vlastné číslo každého grafu  $H_i$  musí byť triviálna nula, teda posledné činitele dielčích súčinnov sú  $(x-v+v_1-0)$  a spoločne ich je presne  $k_1$ . Tým pádom ich môžeme vykrátiť s menovateľom zlomku a dostávame

$$\mu(H, x) = x(x-v)^{k_1-1} \prod_{i=1}^{k_1} (x-v+v_1 - \lambda_1(H_i)) \cdot \dots \cdot (x-v+v_1 - \lambda_{v_1-1}(H_i)).$$

Z toho vieme vyjadriť neusporiadaný vektor vlastných čísel (budeme ho značiť  $\tilde{\lambda}(\cdot)$ ) grafu  $H$  ako  $\tilde{\lambda}(H) =$

$$\underbrace{(v, \dots, v)}_{k_1-1}, v-v_1 + \lambda_1(H_1), \dots, v-v_1 + \lambda_{v_1-1}(H_1), v-v_1 + \lambda_1(H_2), \dots, v-v_1 + \lambda_{v_1-1}(H_{k_1}), 0)'$$

Rovnako dostaneme neusporiadaný vektor vlastných čísel pre graf  $H^*$  ako  $\tilde{\boldsymbol{\lambda}}(H^*) = (\underbrace{v, \dots, v}_{k_1-1}, v - v_1 + \lambda_1(H_1^*), \dots, v - v_1 + \lambda_{v_1-1}(H_1^*), v - v_1 + \lambda_1(H_2^*), \dots, v - v_1 + \lambda_{v_1-1}(H_{k_1}^*), 0)'$ .

Vďaka vlastnostiam Laplaciánu vieme, že vlastné čísla rovné  $v$  sú najväčšie a že nulové vlastné číslo je v oboch vektoroch najmenšie. Tieto dva vektory chceme porovnať (po usporiadaní ich prvkov zostupne) a zistiť či  $\boldsymbol{\lambda}(H)$  majorizuje  $\boldsymbol{\lambda}(H^*)$ . Keďže najväčších  $k_1 - 1$  vlastných čísel a najmenšie nulové vlastné číslo je v oboch vektoroch vždy rovnaké, tak stačí porovnať zvyšné vlastné čísla 1.2. Navyše ak využijeme lemu 1.1, tak môžeme od všetkých vlastných čísel odčítať rovnakú hodnotu  $(v - v_1)$  a nakoniec dostávame na porovnanie  $(v_1 k_1 - k_1)$ –rozmerné upravené neusporiadané vektory vlastných čísel

$$\tilde{\boldsymbol{\gamma}}(H) = (\lambda_1(H_1), \dots, \lambda_{v_1-1}(H_1), \lambda_1(H_2), \dots, \lambda_{v_1-1}(H_{k_1}))',$$

$$\tilde{\boldsymbol{\gamma}}(H^*) = (\lambda_1(H_1^*), \dots, \lambda_{v_1-1}(H_1^*), \lambda_1(H_2^*), \dots, \lambda_{v_1-1}(H_{k_1}^*))'.$$

Ak bude (po usporiadaní týchto vektorov zostupne) platiť  $\boldsymbol{\gamma}(H^*) \prec \boldsymbol{\gamma}(H)$ , tak bude platiť aj  $\boldsymbol{\lambda}(H^*) \prec \boldsymbol{\lambda}(H)$ , a graf  $H^*$  bude Schurovsky najlepší.

Vektory  $\tilde{\boldsymbol{\gamma}}(H^*)$  a  $\tilde{\boldsymbol{\gamma}}(H)$  pozostávajú z vektorov vlastných čísel dielčích grafov jednotlivých partícií, okrem najmenšieho vlastného čísla každého dielčieho grafu  $\lambda_{v_1}(\cdot)$ . Vieme, že najmenšie vlastné číslo Laplaciánu ľubovoľného grafu je nulové. Ak by sme teda do oboch vektorov  $\tilde{\boldsymbol{\gamma}}(H^*)$  a  $\tilde{\boldsymbol{\gamma}}(H)$  pridali  $k_1$  núl, tak pridáme “chýbajúce” nulové najmenšie vlastné čísla  $\lambda_{v_1}(H_1), \dots, \lambda_{v_1}(H_{k_1})$  a  $\lambda_{v_1}(H_1^*), \dots, \lambda_{v_1}(H_{k_1}^*)$ . Potom dostaneme vektory

$$\tilde{\boldsymbol{\gamma}}_0(H) = (\lambda_1(H_1), \dots, \lambda_{v_1-1}(H_1), \lambda_1(H_2), \dots, \lambda_{v_1-1}(H_{k_1}), \lambda_{v_1}(H_1), \dots, \lambda_{v_1}(H_{k_1}))',$$

$$\tilde{\boldsymbol{\gamma}}_0(H^*) = (\lambda_1(H_1^*), \dots, \lambda_{v_1-1}(H_1^*), \lambda_1(H_2^*), \dots, \lambda_{v_1-1}(H_{k_1}^*), \lambda_{v_1}(H_1^*), \dots, \lambda_{v_1}(H_{k_1}^*))'.$$

Čo vieme zjednodušene zapísať ako

$$\tilde{\boldsymbol{\gamma}}_0(H) = (\boldsymbol{\lambda}(H_1), \dots, \boldsymbol{\lambda}(H_{k_1}))', \quad \tilde{\boldsymbol{\gamma}}_0(H^*) = (\boldsymbol{\lambda}(H_1^*), \dots, \boldsymbol{\lambda}(H_{k_1}^*))'.$$

Ak pre ich usporiadané verzie platí  $\boldsymbol{\gamma}_0(H^*) \prec \boldsymbol{\gamma}_0(H)$ , tak platí aj  $\boldsymbol{\gamma}(H^*) \prec \boldsymbol{\gamma}(H)$ , pretože sme do pôvodných vektorov pridali rovnako veľa núl, o ktorých vieme že sú menšie, nanajvýš rovné, ostatným komponentom týchto vektorov. Takáto operácia zachováva

majorizáciu 1.2.

Na porovnanie vektorov  $\gamma_0(H^*)$  a  $\gamma_0(H)$  budeme potrebovať nasledovnú lemu.

**Lema 2.3.** *Majme graf  $G$ , ktorý vieme vyjadriť ako zjednotenie disjunktných grafov  $G_1, \dots, G_{k_1}$ . Potom platí, že vektor vlastných čísel  $\boldsymbol{\lambda}(G)$  grafu  $G$  dostaneme, keď usporiadame zostupne vlastné čísla všetkých dielčích grafov  $G_1, \dots, G_{k_1}$ . Respektíve  $\tilde{\boldsymbol{\lambda}}(G) = (\boldsymbol{\lambda}(G_1), \dots, \boldsymbol{\lambda}(G_{k_1}))'$ .*

*Dôkaz.* Keďže  $G_1, \dots, G_{k_1}$  sú disjunktné, tak Laplacián grafu  $G = G_1 + \dots + G_{k_1}$  je bloková matica s  $k_1$ -blokmi. Potom pri hľadaní vlastných čísel Laplaciánu pracujeme s rovnicou  $\det(\mathbf{L}(G) - \boldsymbol{\lambda}\mathbf{I}) = 0$ , teda opäť s determinantom blokovej matice. Pre determinant blokovej matice platí, že je rovný súčinu determinantov jednotlivých blokov. Tým pádom dostávame vlastné čísla Laplaciánu  $G$ , ako zjednotenie vlastných čísel jednotlivých blokov. Čo sú presne vlastné čísla Laplaciánov dielčích grafov  $G_1, \dots, G_{k_1}$ .  $\square$

Vďaka tomu, že  $H_1, \dots, H_{k_1}$  sú disjunktné môžeme využiť lemu 2.3. Potom dostávame, že graf  $H_{\text{pridané}} = H_1 + \dots + H_{k_1}$  má neusporiadaný vektor vlastných čísel rovný

$$\tilde{\boldsymbol{\lambda}}(H_{\text{pridané}}) = (\boldsymbol{\lambda}(H_1), \dots, \boldsymbol{\lambda}(H_{k_1}))'$$

Rovnako môžeme vyjadriť aj

$$\tilde{\boldsymbol{\lambda}}(H_{\text{pridané}}^*) = (\boldsymbol{\lambda}(H_1^*), \dots, \boldsymbol{\lambda}(H_{k_1}^*))'$$

Z tvaru uvedených vektorov vyplýva, že  $\tilde{\boldsymbol{\lambda}}(H_{\text{pridané}}) = \tilde{\gamma}_0(H)$  a  $\tilde{\boldsymbol{\lambda}}(H_{\text{pridané}}^*) = \tilde{\gamma}_0(H^*)$ .

O grafe  $H_{\text{pridané}}^*$  vieme, že sa skladá s  $k$  nesusedných hrán (alebo  $\lfloor v/2 \rfloor$  nesusedných a jednej vhodnej navyše) a prípadných izolovaných vrcholov. Graf  $H_{\text{pridané}}$  je ľubovoľný jednoduchý graf s rovnakým počtom hrán ako  $H_{\text{pridané}}^*$ , ktorý navyše neobsahuje hrany spájajúce partície v  $K_{n \times k}$ . Tým pádom sú rovnaké ako grafy z tvrdenia 2.2 z dôkazu vety 2.7 (resp. 2.3 z vety 2.8), pričom náš graf  $H_{\text{pridané}}^*$  je rovnaký ako graf  $\bar{G}^*$  z tvrdenia a náš graf  $H_{\text{pridané}}$  korešponduje s grafom  $\bar{H}$  z tvrdenia.

Z dôkazu Constantineho viet vieme, že pre takéto grafy platí  $\boldsymbol{\lambda}(\bar{G}^*) \prec \boldsymbol{\lambda}(\bar{H})$ . Čo vieme prepísať na  $\boldsymbol{\lambda}(H_{\text{pridané}}^*) \prec \boldsymbol{\lambda}(H_{\text{pridané}})$ , ďalej na  $\gamma_0(H^*) \prec \gamma_0(H)$ , a nakoniec na  $\gamma(H^*) \prec \gamma(H)$ . Z čoho vyplýva, že graf  $H^*$  je Schurovsky *najlepšia* augmentácia kompletného multipartitného grafu.  $\square$

## 2.5 Záver

Predstavili sme si tri nové triedy optimálne augmentovaných blokových návrhov s veľkosťou bloku dva. Každý návrh je reprezentovaný párovacím grafom. Tieto návrhy sú optimálne augmentované vzhľadom na širokú triedu ortogonálne invariantných kritérií optimality. Okrem toho sme predstavili dve známe metódy 2.3.1 a 2.3.2 pomocou ktorých vieme dobre pracovať s vlastnými číslami Laplaciánov grafov. Použitie týchto metód sme demonštrovali pri dokazovaní optimality jednotlivých tried. Vďaka týmto metódam a poznatkom z lineárnej algebry o vlastných číslach, môžeme hľadať a analyzovať ďalšie triedy optimálne augmentovaných grafov. Ako vhodný kandidát sa napríklad javí augmentácia skoro-regulárneho kompletného multipartitného grafu, pre ktorú máme zatiaľ len numerické výsledky, ktoré ukážeme v druhej časti poslednej kapitoly.

### 3 Algoritmy a numerické výsledky

Praktická časť dizertačnej práce pozostáva z dvoch podčastí. V prvej predstavíme tri nové algoritmy určené na hľadanie optimálnych návrhov a optimálnej augmentácie už vykonaných návrhov experimentov. V druhej ukážeme numerické výsledky získané pomocou nich.

Numerické výsledky a algoritmy sú v práci adaptované pre optimálne blokové experimenty s veľkosťou bloku dva. Všetky tri algoritmy fungujú pre omnoho širšiu triedu experimentov, ale my sme sa zamerali len na experimenty z témy práce.

Algoritmy nám pôvodne slúžili najmä na odhaľovanie zákonitostí v optimálnych návrhoch, ktoré sme potom dokazovali teoreticky. Takto sme získali myšlienky tvrdení dokázaných v kapitole 2. Na druhú stranu tieto algoritmy v porovnaní s inými používanými metódami dosahujú natoľko dobré výsledky, že sú vhodné aj na samotné numerické hľadanie optimálnych návrhov experimentov (bez dokázanej optimality). Pracovali sme so stochastickými algoritmami, ktoré bývajú pomalšie ako deterministické, ale majú väčšiu šancu neuviaznuť v lokálnych minimách.

Najdôležitejšou novou črtou týchto algoritmov je, že sú schopné pracovať s viacerými *ohraničeniami na zdroje* (1.12). Ostatné existujúce algoritmy buď vedeli pracovať len s jedným ohraničením na celkový počet pokusov (ktoré sa nedalo triviálne rozšíriť), alebo boli veľmi úzko špecializované na konkrétny typ úloh (ako napr. algoritmus [55], ktorý ale nevie zaručiť nájdenie optima). Toto je hlavný dôvod, prečo sme vyvíjali nové algoritmy. Požiadavka na viacero ohraničení na zdroje je v mnohých experimentoch prirodzená a nenašli sme žiadnu metódu ktorá by s takouto požiadavkou vedela pracovať. Okrem toho sú všetky tri algoritmy použiteľné na augmentáciu nejakého už vykonaného návrhu  $\xi^{(0)}$ . Čo samozrejme zahŕňa aj prípady, keď robíme experiment prvýkrát a ešte neboli vykonané žiadne návrhy.

Numerické výsledky uvedené v druhej časti sú získané prevažne tretím algoritmom. Na základe týchto numerických výsledkov vyvrátíme niektoré staršie hypotézy a vyslovíme nové. Ukážeme zaujímavé optimálne návrhy a k nim prislúchajúce grafy, na základe ktorých by sa dali vyextrahovať ďalšie vlastnosti optimálnych návrhov.



## 3.1 Algoritmy

Postupne ukážeme tri nové algoritmy. Prvým z nich je vhodne upravený algoritmus *simulovaného žihania*. S týmto algoritmom sme pracovali v diplomovej práci [2]. Používali sme v ňom len niektoré konkrétne parciálne kritériá optimality a keďže ďalšie dva algoritmy sa javili byť lepšie, tak sme ho nerozširovali pre ďalšie kritériá. Podobne v ňom ešte nie sú zapracované ohraničenia na zdroje (1.12). Dali by sa pridať, ale nevracali sme sa k tomu skrz dva nové algoritmy.

Druhý algoritmus je už úplne prispôsobený používaniu ohraničení na zdroje (1.12) a prezentovali sme ho v zborníku prác z konferencie EYSM [3]. V zborníku sme na ukážku použili len dve kritériá, ale rovnako by pracoval aj s ostatnými ortogonálne invariantnými kritériami optimality.

Tretí z algoritmov je najkomplexnejší a dáva dobré výsledky pre širokú triedu problémov navrhovania experimentov. Pracuje s ohraničeniami na zdroje (1.12) a so všetkými ortogonálne invariantnými kritériami optimality (vie použiť aj ľubovoľné iné monotónne kritériá optimality). Na tomto algoritme sme pracovali spoločne so školiteľom Radoslavom Harmanom a kolegyňou Lenkou Filovou a je podrobne spracovaný v článku [21].

### 3.1.1 Prvý algoritmus - simulované žihanie

*Simulované žihanie* je inšpirované procesom žihania ocele. Žihanie je proces tepelného spracovania zliatiny, vďaka ktorému získa výsledný produkt lepšie mechanické vlastnosti zmenou mikroskopickej štruktúry. Napríklad takto vieme zvýšiť pevnosť, či ťažnosť ocele. Princíp žihania spočíva v tom, že najskôr zliatinu zahrejeme na vysokú teplotu (roztopíme) a potom ju nechávame pomaly chladnúť. Od rýchlosti chladnutia potom závisia vlastnosti zliatiny.

Na podobnom princípe je založené aj *simulované žihanie*. Na začiatku hľadania optimálneho návrhu pracujeme s vysokou pravdepodobnosťou prechodu z lepšieho návrhu (vzhľadom na hodnotu kritériá optimality) do horšieho návrhu, ktorú postupne znižujeme. Teda na začiatku behu algoritmu je veľká šanca opustiť lokálne optimum, zatiaľ čo na konci behu algoritmu sa už len dohľadáva optimum v konkrétnej oblasti riešení. Viac podrobností o *simulovanom žihaní* sa dá nájsť v prácach [48] a [8], kde bolo prvýkrát zavedené.

Schéma 1 predstavuje schematický zápis nášho algoritmu. Je v ňom popísaný celý

beh algoritmu *simulovaného žihania*, pričom ešte podrobne vysvetlíme jednotlivé jeho časti. Základný princíp algoritmu spočíva v postupnom prehľadávaní nových kandidátov  $\xi^{(k)}$  na optimálny návrh v okolí aktuálneho návrhu experimentu  $\xi^{(a)}$ . Ak je skúmaný kandidát lepší (vzhľadom na hodnotu kritéria  $\Phi$ ) ako aktuálny návrh, tak do neho prejde vždy a ďalej s ním pracuje ako s novým aktuálnym návrhom. Ak je kandidát horší ako aktuálny návrh, tak do neho môže tiež prejsť, ale len s určitou pravdepodobnosťou závislou na hodnotách kritérií jednotlivých návrhov a aktuálnej teplote. Ako presne to funguje ukážeme za chvíľu.

<p><b>Vstup</b> : model, kritérium optimality <math>\Phi</math>, počiatočný návrh <math>\xi^{(1)}</math>, kritérium ukončenia, kritérium zastavenia, počiatočná teplota <math>T</math>, spôsob znižovania teploty <math>g(T, time)</math> vzhľadom na čas <math>time</math></p> <p><b>Výstup</b>: <math>\xi^+</math> najlepší nájdený návrh experimentu</p> <p><math>\xi^{(a)} \leftarrow \xi^{(1)}</math>; <math>\xi^+ \leftarrow \xi^{(a)}</math>; <math>time \leftarrow 0</math>;</p> <p><b>repeat</b></p> <table border="1"> <tr> <td> <p><b>repeat</b></p> <p>vyberte kandidáta na nový návrh <math>\xi^{(k)}</math> z okolia návrhu <math>\xi^{(a)}</math></p> <p><b>if</b> <math>\Phi(\mathbf{M}(\xi^{(a)})) &lt; \Phi(\mathbf{M}(\xi^{(k)}))</math> <b>then</b></p> <table border="1"> <tr> <td> <p><math>\xi^{(a)} \leftarrow \xi^{(k)}</math>;</p> <p><b>if</b> <math>\Phi(\mathbf{M}(\xi^+)) &lt; \Phi(\mathbf{M}(\xi^{(k)}))</math> <b>then</b></p> <table border="1"> <tr> <td> <p><math>\xi^+ \leftarrow \xi^{(k)}</math></p> </td> </tr> </table> <p><b>end</b></p> </td> </tr> <tr> <td> <p><b>else</b></p> <p><b>if</b> <math>\text{random}(0, 1) &lt; e^{\frac{\Phi(\mathbf{M}(\xi^{(k)})) - \Phi(\mathbf{M}(\xi^{(a)}))}{T}}</math> <b>then</b></p> <table border="1"> <tr> <td> <p><math>\xi^{(a)} \leftarrow \xi^{(k)}</math></p> </td> </tr> </table> <p><b>end</b></p> </td> </tr> </table> <p><b>end</b></p> </td> </tr> </table> <p><b>until</b> <i>kritérium ukončenia</i>;</p> <p><math>T \leftarrow g(T, time)</math>;</p> <p><math>time \leftarrow time + 1</math>;</p> <p><b>until</b> <i>kritérium zastavenia</i>;</p>	<p><b>repeat</b></p> <p>vyberte kandidáta na nový návrh <math>\xi^{(k)}</math> z okolia návrhu <math>\xi^{(a)}</math></p> <p><b>if</b> <math>\Phi(\mathbf{M}(\xi^{(a)})) &lt; \Phi(\mathbf{M}(\xi^{(k)}))</math> <b>then</b></p> <table border="1"> <tr> <td> <p><math>\xi^{(a)} \leftarrow \xi^{(k)}</math>;</p> <p><b>if</b> <math>\Phi(\mathbf{M}(\xi^+)) &lt; \Phi(\mathbf{M}(\xi^{(k)}))</math> <b>then</b></p> <table border="1"> <tr> <td> <p><math>\xi^+ \leftarrow \xi^{(k)}</math></p> </td> </tr> </table> <p><b>end</b></p> </td> </tr> <tr> <td> <p><b>else</b></p> <p><b>if</b> <math>\text{random}(0, 1) &lt; e^{\frac{\Phi(\mathbf{M}(\xi^{(k)})) - \Phi(\mathbf{M}(\xi^{(a)}))}{T}}</math> <b>then</b></p> <table border="1"> <tr> <td> <p><math>\xi^{(a)} \leftarrow \xi^{(k)}</math></p> </td> </tr> </table> <p><b>end</b></p> </td> </tr> </table> <p><b>end</b></p>	<p><math>\xi^{(a)} \leftarrow \xi^{(k)}</math>;</p> <p><b>if</b> <math>\Phi(\mathbf{M}(\xi^+)) &lt; \Phi(\mathbf{M}(\xi^{(k)}))</math> <b>then</b></p> <table border="1"> <tr> <td> <p><math>\xi^+ \leftarrow \xi^{(k)}</math></p> </td> </tr> </table> <p><b>end</b></p>	<p><math>\xi^+ \leftarrow \xi^{(k)}</math></p>	<p><b>else</b></p> <p><b>if</b> <math>\text{random}(0, 1) &lt; e^{\frac{\Phi(\mathbf{M}(\xi^{(k)})) - \Phi(\mathbf{M}(\xi^{(a)}))}{T}}</math> <b>then</b></p> <table border="1"> <tr> <td> <p><math>\xi^{(a)} \leftarrow \xi^{(k)}</math></p> </td> </tr> </table> <p><b>end</b></p>	<p><math>\xi^{(a)} \leftarrow \xi^{(k)}</math></p>
<p><b>repeat</b></p> <p>vyberte kandidáta na nový návrh <math>\xi^{(k)}</math> z okolia návrhu <math>\xi^{(a)}</math></p> <p><b>if</b> <math>\Phi(\mathbf{M}(\xi^{(a)})) &lt; \Phi(\mathbf{M}(\xi^{(k)}))</math> <b>then</b></p> <table border="1"> <tr> <td> <p><math>\xi^{(a)} \leftarrow \xi^{(k)}</math>;</p> <p><b>if</b> <math>\Phi(\mathbf{M}(\xi^+)) &lt; \Phi(\mathbf{M}(\xi^{(k)}))</math> <b>then</b></p> <table border="1"> <tr> <td> <p><math>\xi^+ \leftarrow \xi^{(k)}</math></p> </td> </tr> </table> <p><b>end</b></p> </td> </tr> <tr> <td> <p><b>else</b></p> <p><b>if</b> <math>\text{random}(0, 1) &lt; e^{\frac{\Phi(\mathbf{M}(\xi^{(k)})) - \Phi(\mathbf{M}(\xi^{(a)}))}{T}}</math> <b>then</b></p> <table border="1"> <tr> <td> <p><math>\xi^{(a)} \leftarrow \xi^{(k)}</math></p> </td> </tr> </table> <p><b>end</b></p> </td> </tr> </table> <p><b>end</b></p>	<p><math>\xi^{(a)} \leftarrow \xi^{(k)}</math>;</p> <p><b>if</b> <math>\Phi(\mathbf{M}(\xi^+)) &lt; \Phi(\mathbf{M}(\xi^{(k)}))</math> <b>then</b></p> <table border="1"> <tr> <td> <p><math>\xi^+ \leftarrow \xi^{(k)}</math></p> </td> </tr> </table> <p><b>end</b></p>	<p><math>\xi^+ \leftarrow \xi^{(k)}</math></p>	<p><b>else</b></p> <p><b>if</b> <math>\text{random}(0, 1) &lt; e^{\frac{\Phi(\mathbf{M}(\xi^{(k)})) - \Phi(\mathbf{M}(\xi^{(a)}))}{T}}</math> <b>then</b></p> <table border="1"> <tr> <td> <p><math>\xi^{(a)} \leftarrow \xi^{(k)}</math></p> </td> </tr> </table> <p><b>end</b></p>	<p><math>\xi^{(a)} \leftarrow \xi^{(k)}</math></p>	
<p><math>\xi^{(a)} \leftarrow \xi^{(k)}</math>;</p> <p><b>if</b> <math>\Phi(\mathbf{M}(\xi^+)) &lt; \Phi(\mathbf{M}(\xi^{(k)}))</math> <b>then</b></p> <table border="1"> <tr> <td> <p><math>\xi^+ \leftarrow \xi^{(k)}</math></p> </td> </tr> </table> <p><b>end</b></p>	<p><math>\xi^+ \leftarrow \xi^{(k)}</math></p>				
<p><math>\xi^+ \leftarrow \xi^{(k)}</math></p>					
<p><b>else</b></p> <p><b>if</b> <math>\text{random}(0, 1) &lt; e^{\frac{\Phi(\mathbf{M}(\xi^{(k)})) - \Phi(\mathbf{M}(\xi^{(a)}))}{T}}</math> <b>then</b></p> <table border="1"> <tr> <td> <p><math>\xi^{(a)} \leftarrow \xi^{(k)}</math></p> </td> </tr> </table> <p><b>end</b></p>	<p><math>\xi^{(a)} \leftarrow \xi^{(k)}</math></p>				
<p><math>\xi^{(a)} \leftarrow \xi^{(k)}</math></p>					

Schéma 1: Prvý algoritmus

### Parametre a nastavenia algoritmu:

Pri *simulovanom žihaní* potrebujem na začiatku zvoliť niekoľko vstupných parametrov. Vždy potrebujeme poznať *model* s ktorým pracujeme, aby sme vedeli vytvoriť

informačné matice  $\mathbf{M}(\cdot)$  pre porovnávané návrhy. Budeme pracovať s klasickým modelom (1.1) zavedeným v úvode práce. Zároveň s modelom zvolíme aj *kritériá optimality*  $\Phi(\cdot)$  na základe ktorých budeme porovnávať informačné matice jednotlivých návrhov. Používali sme parciálne kritériá optimality  $D_{\mathbf{P}}, A_{\mathbf{P}}, E_{\mathbf{P}}$  (1.1.3). Úloha by sa dala rozšíriť aj na ďalšie kritériá.

Ďalej zadávame *počiatočný návrh* experimentu  $\xi^{(1)}$  z ktorého budeme vychádzať. Musí spĺňať *ohraničenia na zdroje* a byť po zložkách rovný alebo väčší ako návrh  $\xi^{(0)}$ , ktorý chcem augmentovať. Inak môžeme  $\xi^{(1)}$  vybrať úplne náhodne, alebo použiť výsledný návrh z inej optimalizácie (napr. zaokrúhlený optimálny aproximatívny návrh). Tak isto môžeme algoritmus spustiť viackrát s rôznymi počiatočnými návrhmi, aj keď to nemá tak veľký význam ako pri deterministických algoritmoch. Pri stochastických algoritmoch totiž môžeme napriek rovnakým počiatočným nastaveniam dostať rôzne výsledky.

Ako ďalšie potrebujeme vybrať *kritérium zastavenia* a *kritérium ukončenia*. *Kritérium zastavenia* určuje ako dlho bude bežať celý algoritmus. Môžeme ho zvoliť napríklad ako celkový počet iterácií, celkový čas behu algoritmu, alebo na akú hodnotu sa chceme dostať pri znižovaní teploty. *Kritérium ukončenia* hovorí kedy ukončiť jednu iteráciu algoritmu. Po ukončení každej iterácie znižujeme teplotu  $T$ . Ukončenie môže nastať napríklad po dostatočnom zlepšení hodnoty kritéria, po odskúšaní určitého množstva kandidátov na nový návrh, alebo po každej zmene aktuálneho návrhu.

Nakoniec ešte musíme zvoliť *počiatočnú teplotu*  $T$  a spôsob jej znižovania  $g(T, \text{time})$ . Teplotu  $T$  sme volili v závislosti na konkrétnu úlohu. Ak sa spôsob znižovania teploty zvolí úmerne  $g(T, \text{time}) = T / \log(\text{time} + 1)$ , tak je dokázaná konvergencia k optimálnemu návrhu [54]. Tento pokles sa ale v praxi nepoužíva kvôli veľkej časovej náročnosti. Namiesto toho sa používa napr. lineárny alebo exponenciálny pokles [48], pre ktoré ale nie je dokázaná konvergencia.

Najdôležitejšou časťou algoritmu je *výber kandidátov na prechod*. Túto voľbu môžeme robiť rôznymi spôsobmi. Kandidátov môžeme voliť úplne náhodne, náhodne z určitého okolia aktuálneho návrhu  $\xi^{(a)}$ , môžeme vyberať len najlepší návrh z okolia návrhu  $\xi^{(a)}$ , môžeme systematicky prehľadať celé okolie  $\xi^{(a)}$ . Je dobré sa obmedziť len na malé okolie  $\xi^{(a)}$  najmä kvôli rýchlosti algoritmu (ak by sme počas jednej iterácie prehľadávali vždy celý priestor možných návrhov, tak by sme síce mali zaručené nájdenie optimálneho návrhu, ale za neúnosnú časovú náročnosť). My sme pracovali s okolím tvoreným návrhmi líšiacimi sa od aktuálneho návrhu len v jednom bode návrhu. Samo-

zrejme by sa okolie dalo voliť aj inými spôsobmi. V prípade keď boli aktuálne návrhy prezentované grafmi sme okolia tvorili dvomi spôsobmi. Buď sme v aktuálnom návrhu vybrali náhodnú hranu a zmenili jej náhodne jeden koncový vrchol, alebo sme odobrili celú hranu a pridali ľubovoľnú novú hranu. Podrobnejšie sa o spôsobe výberu kandidátov píše v článku [54].

Posledná nevysvetlená časť algoritmu je *pravdepodobnosť prechodu do kandidátskeho návrhu*. Táto je rovná jednej, ak je hodnota kritéria kandidáta lepšia (vyššia) ako hodnota aktuálneho kritéria. Vtedy prejdeme do kandidátskeho návrhu vždy. Ak je hodnota kritéria kandidátskeho návrhu horšia, tak je stále možnosť na prechod. Pravdepodobnosť prechodu je rovná

$$e^{\frac{\Phi(\mathbf{M}(\xi^{(k)})) - \Phi(\mathbf{M}(\xi^{(a)}))}{T}}.$$

Táto pravdepodobnosť závisí na rozdiel medzi hodnotami kritéria  $\Phi$  v aktuálnom návrhu kritéria  $\xi^{(a)}$  a v kandidátovi na nový návrh  $\xi^{(k)}$ , a na teplote  $T$ . Pričom platí, že čím je menší rozdiel kritérií, alebo čím je vyššia teplota, tým je pravdepodobnosť prechodu väčšia. Keďže teplotu postupne znižujeme, tak na začiatku behu algoritmu je väčšia šanca prechodu do horších kandidátov (s nižšou hodnotou kritéria). Na konci behu algoritmu je teplota nízka a tým pádom prechádzame väčšinou už len do lepších, alebo veľmi podobných návrhov.

### Vlastnosti a využitie:

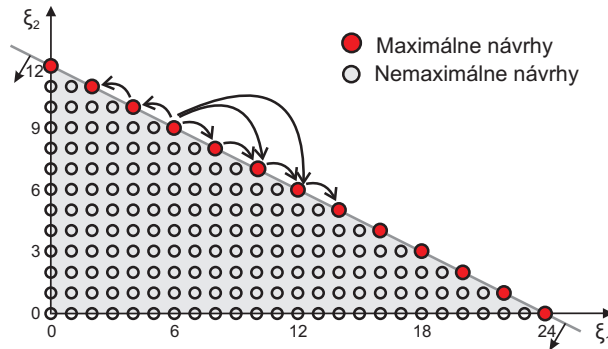
Nevýhodou prvého algoritmu je, že pre každý typ experimentu musíme nastaviť veľké množstvo vstupných parametrov. Tým pádom musíme danému experimentu rozumieť do hĺbky a až potom vieme dostať rozumné riešenia. Preto sa tomuto algoritmu nebudeme venovať dlhšie a prejdeme k ďalším dvom algoritmom. V nich obmedzíme počet vstupných voliteľných parametrov a výsledné algoritmy budú univerzálnejšie.

Viac o algoritme a voľbe parametrov sa dá nájsť v knihe [41], ale voľba parametrov sa do značnej miery odvíja od konkrétneho typu optimalizačnej úlohy. Ako využiť *simulované žihanie* pri hľadaní optimálnych mikročipových experimentov (pre parciálne kritéria  $D_{\mathbf{P}}$  a  $A_{\mathbf{P}}$  optimality) som riešila v diplomovej práci [2].

### 3.1.2 Druhý algoritmus

Druhý algoritmus sme vytvorili, keď sme mali zavedený model (1.1), parciálne kritéria optimality a najmä *ohraničenia na zdroje* (1.12). Na základe *ohraničení na zdroje* sme vytvorili množinu prípustných návrhov  $\Xi$ , spĺňajúcich tieto ohraničenia. Množinu prí-

pustných návrhov  $\Xi$  sme rozdelili na *maximálne* a *nemaximálne* návrhy (viď. 15). Úlohou druhého algoritmu bolo rozumne prehľadávať množinu *maximálnych* návrhov, lebo sa musí medzi nimi nachádzať aspoň jeden optimálny návrh.



Obrázok 3–1: Pohyb po *maximálnych* (červených) návrhoch.

Náš algoritmus funguje podobne ako algoritmus uvedený v článku [55], ale porovnaní s ním dokáže pracovať s ľubovoľným množstvom *ohraničení na zdroje* a vždy má šancu dokonvergovať k optimálnemu riešeniu. Zatiaľ čo algoritmus z článku [55] pracuje len s jedným ohraničením a môže sa stať, že skončí v lokálnom optime. Na rozdiel od nášho algoritmu totiž nevie zameniť viacero "lacných a menej informatívnych" pokusov za jeden "drahší a informatívnejší", teda pravdepodobnosti prechodu medzi niektorými návrhmi sú nulové.

Druhý algoritmus budeme používať pre optimalizáciu na základe parciálnych kritérií optimality  $D_{\mathbf{A}}$ ,  $A_{\mathbf{A}}$ ,  $E_{\mathbf{A}}$  a kritéria  $\mathbf{c}$ -optimality. V schematickom zápise (schéma 2) ukážeme v čom spočíva princíp algoritmu a potom podrobnejšie vysvetlíme jednotlivé jeho časti. Následne ukážeme ako pracuje na modelovom príklade. *Maximálne* návrhy budeme označovať písmenom  $\xi$  s rôznymi indexmi a *nemaximálne* návrhy písmenom  $\zeta$ .

### Parametre a nastavenia algoritmu:

Podobne ako pri *simulovanom žíhaní* aj tu potrebujeme definovať niekoľko vstupných parametrov. Okrem modelu 1.1 a kritérií optimality 1.1.3, potrebujeme *počiatočný návrh* experimentu  $\zeta^{(1)}$ . Ten môžeme zvoliť ako návrh v ktorom nevykonáme žiaden nový pokus, vzhľadom na návrh  $\xi^{(0)}$ , ktorý chceme augmentovať. Teda  $\xi^{(1)} = \xi^{(0)}$ . Samozrejme predpokladáme, že návrh  $\xi^{(0)}$  neporušuje *ohraničenia na zdroje*. Počiatočný návrh môžeme zvoliť aj inak.

Ďalej potrebujeme *kritérium zastavenia* algoritmu, za ktoré zvolíme celkový počet

**Vstup** : model, kritérium optimality  $\Phi$ , počiatočný návrh  $\zeta^{(1)}$ , počet iterácií  $n$ , množina prípustných návrhov  $\Xi$

**Výstup**:  $\xi^+$  najlepší nájdený návrh experimentu

$\xi^+ \leftarrow \zeta^{(1)}$  (priebežne najlepší návrh);

$\Phi^+ \leftarrow 0$  (hodnota kritéria priebežne najlepšieho návrhu);

$it \leftarrow 0$  (poradové číslo iterácie);

**repeat**

*saturácia*  $\zeta^{(it)}$  do  $\xi^{(it)}$ , kde  $\xi^{(it)}$  je *maximálny*;

**if**  $\Phi^+ = \Phi(\mathbf{M}(\xi^+)) \leq \Phi(\mathbf{M}(\xi^{(it)}))$  **then**

$\xi^+ \leftarrow \xi^{(it)}$ ;  $\Phi^+ \leftarrow \Phi(\mathbf{M}(\xi^+))$ ;

*subsaturácia*  $\xi^+$  do  $\zeta^{(it+1)}$ ;

$it \leftarrow it + 1$ ;

**end**

**until** *kritérium zastavenia*:  $it = n$ ;

### Schéma 2: Druhý algoritmus

iterácií  $n$ . Tento počet odvíjame od toho, ako často sa mení priebežne najlepší návrh (ak zmena nenastane dostatočne dlho, tak zastavíme algoritmus), na základe skúseností s konkrétnym experimentom.

Najdôležitejšia časť algoritmu je spôsob výberu a prechodu k novému priebežne najlepšiemu návrhu. Z aktuálneho *maximálneho* návrhu  $\xi^{(i)}$  sa dostaneme do nového *maximálneho* návrhu  $\xi^{(i+1)}$  procesom *subsaturácie* do *nemaximálneho* návrhu  $\zeta^{(i+1)}$  a následnej *saturácie*. Dôležité je ako tieto dve fázy prebiehajú. Vo fáze *saturácie* vyberáme náhodne bod experimentu  $x_i \in \mathfrak{X}$  na základe lokálnych vlastností použitého kritéria. S väčšou pravdepodobnosťou vyberáme body o ktorých predpokladáme, že nesú viac informácie. Potom v tomto bode návrhu pridáme jeden pokus. Tento postup opakujeme pokým nedosiahneme *maximálny* návrh. Táto fáza je v našom algoritme aj v algoritme [55] podobná. Rozdiel je vo fáze *subsaturácie*.

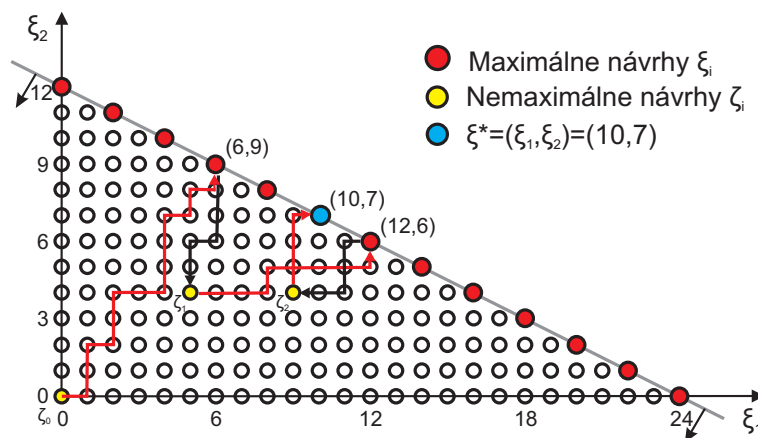
V algoritme z článku [55] je *subsaturácia* nasledovná. Rovnomerne náhodne vyberieme jeden bod návrhu a odoberieme (ak sa dá) z neho jeden pokus. Potom nastupuje fáza *saturácie* a pridávania pokusov. Pri tomto postupe sa nemusí podariť zvýšiť počet pokusov v niektorých návrhoch. Keď sa pozrieme na jednoduchý príklad o náteroch 1, tak na to aby sme z jedného *maximálneho* návrhu dostali druhý z vyšším počtom pokusov v bode návrhu  $x_2$ , potrebovali by sme pri *subsaturácii* odobrať aspoň dva pokusy z bodu návrhu  $x_1$ . Ale máme dovolené odobrať len jeden pokus.

U nás tento problém riešime nasledovnou *subsaturáciou*. Odoberieme náhodný počet

pokusov (na základe Poissonovho rozdelenia so zvoleným parametrom  $\lambda$ ) z rovnomerne náhodne vybraných bodov návrhu. Potom nastupuje fáza saturácie a pridávania návrhov. Tým pádom máme teoreticky zaručené (pri dostatočne veľkom  $\lambda$ ), že vieme odobrať dostatok pokusov na prechod do hociktorého iného *maximálneho* návrhu.

### Príklad o náteroch:

Keď sme druhý algoritmus použili na nájdenie optimálneho návrhu pre príklad (1), tak našiel optimálny návrh pri 50 iteráciách za menej ako 0,05 sekundy.



**Obrázok 3–2:** Jeden priebeh optimalizácie pomocou druhého algoritmu. Červené body návrhu sú *maximálne*. Modrý bod návrhu je *optimálny*. Červené cesty sledujú *saturáciu* do *maximálneho* návrhu, čierne *subsaturáciu*.

### Vlastnosti a využitie:

Hlavné výhody druhého algoritmu sú, že vie pracovať s parciálnymi kritériami optimality a ľubovoľným množstvom *ohraničení na zdroje*. Nevýhodou je určovanie *kritéria zastavenia* na základe vlastností experimentu. Tento algoritmus sme prezentovali na konferencii EYSM v Osijeku [3]. Výsledky získané pomocou neho sú v nasledujúcej numerickej časti pri hľadaní optimálnych návrhov experimentov pre blokové návrhy s veľkosťou bloku dva.

#### 3.1.3 Tretí algoritmus

Tretí algoritmus predstavíme najpodrobnejšie, pretože má najširšie využitie a dosahoval najlepšie výsledky z uvedených troch algoritmov, ale aj porovnaní s inými používanými

algoritmami. Má niekoľko vstupných parametrov, ale pre väčšinu z nich uvádzame všeobecné nastavenia.

Algoritmus sme porovnali s niekoľkými inými známymi a používanými algoritmami a dosiahli sme podobné, alebo lepšie výsledky. Náš algoritmus vie navyše riešiť problémy s viac *ohraničeniami na zdroje* (1.12) ako doteraz známe metódy a je použiteľný na veľmi širokú triedu modelov a kritérií.

Okrem úloh navrhovania optimálnych experimentov vieme pre vstup do algoritmu upraviť viacero ťažkých problémov z diskkrétnej optimalizácie a kombinatoriky. Napríklad takzvaný knapsack problém ([26], [30]), teóriu spoľahlivosť počítačových sietí (z ang. optimal redundancy allocation in reliability theory ([12], [37])), a hľadanie  $t$ -optimálnych grafov v teórii grafov (ktoré bolo ukázané v kapitole 2).

Budeme pracovať s modelom (1.1) a používať ľubovoľné kritérium optimality  $\Phi$ , spĺňajúce kritérium monotónnosti (1.1.4). V teoretických výsledkoch sme sa zamerali najmä na *ortogonálne invariantné kritériá optimality*, v numerickej časti budeme pracovať s niektorými konkrétnymi z nich. Budeme pracovať s *ohraničeniami na zdroje* (1.12), s návrhom určeným na augmentáciu  $\xi^{(0)}$ , a s množinou  $\Xi^{ex}$  prípustných exaktných návrhov (1.13). Budeme riešiť základnú úlohu (1.15).

Typickým príkladom zadania je lineárny regresný model s nekorelovanými, homoskedastickými chybami na ktorý použijeme kritérium  $D$ -optimality. Tento model budeme využívať aj v numerickej časti, ale s rôznymi kritériami. Náš algoritmus vieme použiť aj pre iné ako lineárne modely (článok [21], časť 3.2 o kvadratickej regresii).

### **Prehľad známych algoritmov na riešenie problému 1.15:**

Na riešenie problémov typu (1.15) s malou množinou *prípustných návrhov* je možné použiť metódy kompletnej enumerácie, napríklad *rozdeľuj a ohraničuj* (z ang. branch-and-bound). Tieto metódy zaručujú nájdenie globálneho optima (viď [52], [47], alebo [25]). Na druhú stranu tieto metódy pri možnostiach dnešnej výpočtovej techniky nevieme rozšíriť pre väčšie problémy. Pre tie existujú len heuristiky väčšinou vedúce k efektívnemu prípustnému návrhu.

Prirodzený spôsob riešenia úlohy (1.15) je vytvoriť heuristiku založenú na “exkurziách” po množine prípustných návrhov, podobne ako v niektorých starších algoritmoch určených na nájdenie  $D$ -optimálnych návrhov za platnosti štandardnej podmienky (1.11), ako napr. [15], [56] a [42]. Pre nás najzaujímavejší z týchto algoritmov je *Det-max* predstavený Mitchellom [42]. Tento algoritmus je založený na tabu prehľadávaní



([20] a [29]), s ktorým budeme pracovať aj my.

V dnešnej dobe najviac používané metódy na riešenie štandardného problému navrhovania experimentov (s podmienkou eqstandard) sú výmenné algoritmy (viď napr. kapitola 12 v [1], a [28]). Bohužiaľ tieto výmenné algoritmy nemôžu byť priamo použité na riešenie všeobecného problému (1.15), pretože vopred nepoznáme počet pokusov v optimálnom návrhu. Tak isto počet pokusov môže variovať aj pri lokálnych optimálnych návrhoch. Navyše sa ukazuje, že problémy s viacerými *ohraničeniami na zdroje* zvyknú mať viacero lokálnych optím, pričom výmenné algoritmy nevieme jednoducho modifikovať tak, aby sme sa zbavili ich inklinácie k lokálnym optimám.

Čo sa týka úloh so všeobecnými *ohraničeniami na zdroje*, tak existuje metóda podobná *Detmaxu*, ktorá ich rieši v teórii spoľahlivosti (pozri [35]). V oblasti optimálneho navrhovania experimentov študoval Welch [53] možné modifikácie *Detmaxu* pre všeobecné ohraničenia. V práci [55], bola predstavená metóda založená na postupnom odoberaní jednotlivých pokusov a následnej pažravej augmentácii vzniknutých návrhov. Táto metóda je obmedzená len na riešenie úloh s konkrétnym druhom priamych obmedzení kombinovaných s jedným obmedzením na celkovú cenu.

Nakoniec sme v predošlej podkapitole ukázali náš druhý algoritmus ktorý vie riešiť všeobecný problém (1.15). Z tohto algoritmu sme vychádzali aj pri tvorbe tretieho algoritmu, ale je tam významný posun a tretí algoritmus je prepracovanejší.

### Algoritmus - hlavná myšlienka:

Budeme hovoriť, že  $n$ -rozmerný návrh  $\zeta \in \Xi^{\text{ex}}$  je vytvorený z návrhu  $\xi \in \Xi^{\text{ex}}$  *krokom dopredu* (resp. *krokom dozadu*) ak platí  $\zeta = \xi + \mathbf{e}_i$  (resp.  $\zeta = \xi - \mathbf{e}_i$ ), kde  $\mathbf{e}_i \in \mathbb{R}^n$  je  $n$ -rozmerný vektor samých núl s jednotkou na  $i$ -tej pozícii. Návrh je *maximálny*, ak ľubovoľný *krok dopredu* z neho vytvorí neprípustný návrh. Teda do neho nemôžeme pridať žiaden pokus v žiadnom bode návrhu bez toho, aby sme porušili niektoré z *ohraničení na zdroje*.

Ďalej zavedieme pojem *horných susedov* návrhu  $\xi \in \Xi^{\text{ex}}$ , ako množinu všetkých prípustných návrhov, do ktorých sa vieme dostať z  $\xi$  pomocou jedného *kroku dopredu*. Takže množina *horných susedov* je

$$\mathcal{U}(\xi) = \{\xi + \mathbf{e}_1, \dots, \xi + \mathbf{e}_n\} \cap \Xi^{\text{ex}}.$$

Podobne množina *dolných susedov* návrhu  $\xi$  bude množina všetkých prípustných ná-

vrhov, do ktorých sa vieme dostať z  $\xi$  pomocou jedného *kroku dozadu*

$$\mathcal{L}(\xi) = \{\xi - \mathbf{e}_1, \dots, \xi - \mathbf{e}_n\} \cap \Xi^{\text{ex}}.$$

Môžeme si všimnúť, že platí  $\mathcal{L}(\xi^{(0)}) = \emptyset$  a  $\mathcal{U}(\xi) = \emptyset$  práve vtedy keď  $\xi$  je *maximálny*. Navyše budeme predpokladať, že  $\xi^{(0)}$  nie je *maximálny* (inak by sme nemali čo optimalizovať), takže pre ľubovoľný prípustný návrh  $\xi$  platí  $\mathcal{L}(\xi) \cup \mathcal{U}(\xi) \neq \emptyset$ .

Vďaka podmienkam (P1), (P2), (P3) platí, že ľubovoľný prípustný exaktný návrh je dostupný z ľubovoľného iného prípustného exaktného návrhu len pomocou *krokov dopredu* a *krokov dozadu* po množine  $\Xi^{\text{ex}}$ . Toto je hlavná výhoda *ohraničení na zdroje*, pretože vďaka tomu budeme vedieť rozumne prechádzať a prehľadávať množinu prípustných návrhov. V prípade komplikovanejších podmienok by sa mohlo stať, že sa medzi niektorými návrhmi vôbec nedá pohybovať pomocou *krokov*, a tým pádom by sme len používaním *krokov dopredu* a *krokov dozadu* nemali zaručenú možnosť nájsť globálne optimum. Rovnako ako pri druhom algoritme aj tu platí, že optimálny návrh je jeden z *maximálnych* návrhov vďaka *monotónnosti* kritéria.

Náš algoritmus začína v *počiatočnom* návrhu  $\xi^{(1)} \in \Xi^{\text{ex}}$  a vykonáva exkurzie po množine prípustných exaktných návrhov. Počas exkurzií využíva *tabu list*  $V$  *charakteristických atribútov* návrhov  $\text{attr}(\xi)$ , ktoré už navštívil a snaží sa do nich nevracať. Ďalej zavedieme pojem *lokálneho odhadu významu* návrhu  $\text{val}(\xi)$ , ako reálne číslo odhadujúce ako veľmi sľubný je návrh  $\xi$  ako súčasť exkurzie vedúcej k optimálnemu návrhu. Oba tieto parametre presnejšie vysvetlíme v samostatných podkapitolách.

Nech  $\xi$  prezentuje aktuálny návrh v ktorom sa počas exkurzie nachádzame. Algoritmus najskôr skúsi vykonať *krok dopredu* (ak sa *atribút* aktuálneho návrhu nenachádza na *tabu liste*  $\text{attr}(\xi) \notin V$ ) alebo *krok dozadu* (ak sa *atribút* aktuálneho návrhu nachádza na *tabu liste*  $\text{attr}(\xi) \in V$ ). Týmto krokom sa presunieme do susedného prípustného exaktného návrhu  $\zeta$ . Pričom návrh  $\zeta$  vyberáme tak, že má maximálnu hodnotu  $\text{val}(\zeta)$  spomedzi všetkých susedných návrhov ktoré nie sú na *tabu liste*  $\text{attr}(\zeta) \notin V$ .

Ak sa algoritmus pokúsi o *krok dopredu* ale neexistuje žiaden *horný sused*  $\zeta \in \mathcal{U}(\xi)$  ktorého *atribút* by nebol na *tabu liste*  $\text{attr}(\zeta) \notin V$ , alebo naopak ak sa algoritmus pokúsi o *krok dozadu* ale neexistuje žiaden *dolný sused*  $\zeta \in \mathcal{L}(\xi)$  ktorého *atribút* by nebol na *tabu liste*  $\text{attr}(\zeta) \notin V$ , tak sa algoritmus pokúsi urobiť *krok* opačným smerom ako chcel pôvodne. Ak ani potom nenájde žiaden návrh do ktorého môže prejsť (tj. návrh ktorého *atribút* by nebol na *tabu liste*), tak túto patovú situáciu vyrieši náhod-

ným prechodom do ľubovoľného prípustného susedného návrhu  $\zeta \in \mathcal{L}(\xi) \cup \mathcal{U}(\xi)$  bez ohľadu na *tabu list*.

Vždy keď algoritmus navštívi *maximálny* návrh, tak si skontroluje či tento návrh nie je lepší ako doteraz najlepší dosiahnutý návrh  $\xi^+$ . Ak počet *krokov dozadu* prekročí nastavenú hranicu  $back_{\max}$ , tak danú exkurziu prehlásime za neúspešnú a začneme novú exkurziu z doteraz najlepšieho nájdeného návrhu. Treba si ale uvedomiť, že aj neúspešné exkurzie sa zapisujú do *tabu listu*  $V$ , tým pádom ďalšia exkurzia bude iná ako predošlá. Navyše bude smerovať k ešte nenavštíveným návrhom. Algoritmus skončí po vopred zadanom čase  $t_{\max}$  a vráti najlepší dovtedy nájdený návrh. Presný priebeh algoritmu sa dá pozrieť na schematickom zápise 3.

Dôvod prečo je náš algoritmus trochu podobný na algoritmus Detmax je, že vyberá *kroky dopredu* alebo *dozadu* na základe hodnoty *atribútu* aktuálneho návrhu. Na rozdiel od nás ale pracuje maximálne s jedným ohraničením na zdroje. Okrem toho i pre klasické podmienky sa náš algoritmus líši od algoritmu Detmax v niekoľkých dôležitých veciach. Detmax napríklad nikdy nedovoľuje *krok dozadu* do návrhu s *atribútom* na *tabu liste*, čo často vedie k opakovaniam rovnakých exkurzií a zaseknutí sa v určitej oblasti návrhov.

### Voľba počiatočného návrhu $\xi^{(1)}$ :

Voľba *počiatočného* návrhu  $\xi^{(1)}$  je dôležitou súčasťou celého algoritmu. Na základe našich skúseností tvrdíme, že dostatočne eficientný návrh vieme dosiahnuť voľbou  $\xi^{(1)} = \xi^{(0)}$ . Ale pre komplexnejšie úlohy je vhodnejšie spustiť radšej viacero behov algoritmu s rôznymi *počiatočnými* návrhmi. Tie môžeme voliť napríklad ako *maximálne* návrhy vytvorené na základe sady náhodných *krokov dopredu* začínajúc v návrhu  $\xi^{(0)}$ .

Iná možnosť je začať z optimálneho aproximatívneho návrhu  $\tilde{\xi} = \arg \max\{\phi(\xi) : \xi \in \Xi^{\text{ap}}\}$  z ktorého vytvoríme exaktný návrh zaokrúhlením všetkých počtov pokusov na najbližšie celé číslo nadol  $\xi^{(1)} = (\lfloor \tilde{\xi}_1 \rfloor, \dots, \lfloor \tilde{\xi}_n \rfloor)$ . Vlastnosti *ohraničení na zdroje* zaručujú, že takto vytvorený návrh  $\xi^{(1)}$  je prípustný exaktný návrh. Navyše na nájdenie optimálneho aproximatívneho návrhu s viacerými ohraničeniami na zdroje sa dá použiť konvexná optimalizácia (dá sa pozrieť napríklad v článku [60]).

### Voľba charakteristického atribútu návrhu $attr(\xi)$ :

*Charakteristický atribút* (skrátene len *atribút*) návrhov sme volili tak aby dával rôzne hodnoty pre diametrálne rozličné návrhy a rovnaké hodnoty pre určitým spôsobom

podobné, či rovnaké návrhy. Napríklad ak sú dva návrhy algebraicky izomorfné (napr. dva symetrické grafy), tak budú mať rovnakú hodnotu *atribútu*.

Na základe skúseností z viacerých úloh sme sa rozhodli zvoliť hodnotu *atribútu*  $attr(\zeta)$  rovnakú ako hodnotu kritéria v danom návrhu  $\phi(\zeta)$  zaokrúhlenú na rozumné množstvo desatinných miest  $n_{\text{round}}$ . Hodnoty *atribútov* navštívených návrhov potom ukladáme do *tabu listu*  $V$ . Treba si uvedomiť, že ukladanie hodnôt *atribútu* namiesto ukladania celých návrhov má niekoľko výhod. Jedna je, že *atribút* je len jedno reálne číslo, zatiaľ čo  $n$ -rozmerný návrh môže zaberat' mnoho viac priestoru. Tým pádom šetríme pamäť a zvyšujeme rýchlosť algoritmu.

Ďalšia výhoda je, že pridaním hodnoty *atribútu* vylúčime rovno celú skupinu podobných (izomorfných) návrhov s rovnakou hodnotou *atribútu*. Keby sme priadli len konkrétny návrh, tak pri ďalších exkurziách môžeme navštíviť aj s ním izomorfné návrhy. Toto má pre nás veľký význam, pretože v tejto práci sa venujeme najmä blokovým návrhom s veľkosťou bloku dva. Ak navyše sú porovnávané body rovnako významné, tak nám môže vzniknúť veľké množstvo symetricky izomorfných návrhov. Tým pádom takto zvolená tvorba *tabu listu* veľmi urýchlí celé prehľadávanie.

### Voľba lokálneho odhadu významu návrhu $val(\xi)$ :

Pomocou tohto parametru chceme hodnotiť ako veľmi nádejný sa javí aktuálny návrh, ako súčasť exkurzie vedúcej k optimálnemu návrhu. Voľbu jednotlivých parametrov je možné vždy upraviť na mieru riešenej úlohe. Nám sa najvhodnejšie (a najuniverzálnejšie) javila pre *lokálny odhad významu* návrhu  $val(\xi)$  voľba vychádzajúca z nasledovnej úvahy.

Ak by sme pre aktuálny návrh  $\zeta$  vedeli ľahko nájsť najlepší návrh spomedzi *maximálnych* exaktných návrhov  $\eta$ , na ktorý ho vieme augmentovať, potom by sme mohli zvoliť ideálnu  $val^*(\zeta)$  ako hodnotu kritéria optimality v tomto najlepšom z *maximálnych* návrhov  $\eta$

$$val^*(\zeta) = \max\{\phi(\eta) : \eta \in \Xi^{\text{ex}}, \zeta \leq \eta\}.$$

Keby sme vedeli ľahko hľadať najlepšiu augmentáciu ľubovoľného návrhu a hodnoty  $val^*(\zeta)$ , tak už máme vyriešenú celú úlohu optimálneho navrhovania. Stačilo by spustiť algoritmus z návrhu  $\xi^{(1)}$ , ktorý by vždy spravil *krok dopredu* do najlepšieho horného suseda (vzhľadom na  $val^*$ ).

Presné hodnoty  $val^*(\zeta)$  ale nevieme. Takže ich potrebujeme nejako rozumne odhad-

núť. Definujme pre návrh  $\zeta \in \Xi^{\text{ex}}$  vektor zatiaľ nevyčerpaných zdrojov nasledovne

$$\mathbf{r}(\zeta) = (b_1 - \sum_{i=1}^n a_{1i}\zeta_i, \dots, b_k - \sum_{i=1}^n a_{ki}\zeta_i)^T,$$

kde  $b_j$  a  $a_{ji}$  sú parametre z definície ohraničení na zdroje (1.12). Tento vektor vieme pri vykonaní *kroku dopredu* alebo *dozadu* rýchlo upraviť nasledovne

$$\mathbf{r}(\zeta \pm \mathbf{e}_i) = \mathbf{r}(\zeta) \mp (a_{1i}, \dots, a_{ki})^T, \quad i \in \{1, \dots, n\}.$$

Ďalej pre každé  $i \in \{1, \dots, n\}$  definujme koľko maximálne vieme pridať pokusov do  $i$ -teho bodu návrhu tak, aby sme neporušili žiadne ohraničenie na zdroje

$$\begin{aligned} d_i(\zeta) &= \max\{d \geq 0 : \zeta + d\mathbf{e}_i \in \Xi^{\text{ex}}\} \\ &= \lfloor \min\{a_{ri}^{-1}\mathbf{r}_r(\zeta) : r \in \{1, \dots, k\}, a_{ri} > 0\} \rfloor. \end{aligned} \quad (3.1)$$

Druhé vyjadrenie hovorí koľko pokusov zmestíme do nevyčerpaných zdrojov pre  $i$ -ty bodu návrhu aby sme neporušili ohraničenia na zdroje.

Potom vektor  $\mathbf{d}(\zeta) = (d_1(\zeta), \dots, d_n(\zeta))^T$  odhaduje smer k “veľkým” prípustným návrhom. Navyše ak  $\mathbf{d}(\zeta) \neq \mathbf{0}_n$  môžeme definovať vzdialenosť k najlepšiemu aproximátnemu návrhu v smere  $\mathbf{d}$  ako

$$\begin{aligned} \gamma(\zeta) &= \max\{\gamma \geq 0 : \zeta + \gamma\mathbf{d}(\zeta) \in \Xi^{\text{ap}}\} \\ &= \min\{h_r^{-1}(\zeta)\mathbf{r}_r(\zeta) : r \in \{1, \dots, k\}, h_r(\zeta) > 0\}, \end{aligned} \quad (3.2)$$

kde  $h_r(\zeta) = \sum_{i=1}^n a_{ri}d_i(\zeta)$ . Ak je  $\mathbf{d}(\zeta) = \mathbf{0}_n$ , napr. ak  $\zeta$  je *maximálny* návrh, tak položíme  $\gamma(\zeta)$  rovné nule.

Vektor  $\zeta + \gamma(\zeta)\mathbf{d}(\zeta)$  je *maximálnym* prípustným aproximátnym návrhom v smere  $\mathbf{d}(\zeta)$ . Potom hodnota

$$\text{val}(\zeta) = \phi(\zeta + \gamma(\zeta)\mathbf{d}(\zeta)) \quad (3.3)$$

je slušný odhad ideálnej hodnoty *lokálneho odhadu významu návrhu*  $\text{val}^*(\zeta)$ .

Podotkneme že výsledný vektor  $\zeta + \gamma(\zeta)\mathbf{d}(\zeta)$  nemusí byť exaktný návrh (väčšinou ani nebude) s celočíselnými komponentami. Vytvorený odhad totiž využíva maximálne aproximatívne návrhy, s ktorými sa pracuje omnoho jednoduchšie.

Treba si uvedomiť, že pre inak definované ohraničenia, ako *ohraničenia na zdroje*, sa dajú výrazy (3.1) - (3.3) výrazne zjednodušiť a tým pádom ešte urýchliť celý priebeh

algoritmu. Ak napríklad pracujeme iba so štandardným ohraničením na celkovú cenu (1.11), tak

$$val(\zeta) = \phi(\zeta + \gamma(\zeta)\mathbf{1}_n),$$

kde  $\gamma(\zeta) = n^{-1}(N - \mathbf{1}_n^T \zeta)$ , pre ľubovoľné  $\zeta \in \Xi^{\text{ex}}$  a  $\mathbf{1}_n = (1, \dots, 1)^T \in \mathbb{R}^n$ .

### Vlastnosti a využitie:

Na záver zhrnieme niektoré vlastnosti nášho algoritmu. Hodnota  $val(\zeta)$  je volená tak, že závisí iba na výslednej množine prípustných aproximatívnych návrhov  $\Xi^{\text{ap}}$ , a nie na voľbe algebraickej definície tejto množiny  $\Xi^{\text{ap}}$ . Navyše priebeh exkurzií závisí iba na poradí aproximatívnych návrhov vzhľadom na kritérium optimality  $\phi$ , a nie na zvolenej “verzii” tohto kritéria. Takže by mal dávať rovnaké výsledné návrhy aj pre nie úplne rovnaké vyjadrenia kritérií optimality.

Ešte sme nespomenuli tri menšie parametre  $back_{max}$ ,  $n_{round}$  a  $t_{max}$ . Prvý hovorí, koľko najviac môžeme urobiť *krokov dozadu* vrámci jednej exkurzie. Druhý hovorí, na koľko desatinných miest budeme zaokrúhľovať hodnotu kritéria  $\Phi$  pri výpočte parametra  $attr(\cdot)$ . Posledný je čas, na ako dlho spúšťame celý algoritmus. Presné hodnoty týchto parametrov si uvedieme v konkrétnych príkladoch. Väčšinou sme volili (vychádzali najvhodnejšie) nastavenia  $back_{max} = 16$ ,  $n_{round} = 9$  desatinných miest a  $t_{max} = 120$  sekúnd.

Kód algoritmu s programovacích jazykoch Matlab a R, sa spolu s ukázkovými príkladmi nachádza na stránke <http://www.iam.fmph.uniba.sk/design/>.

Určite je veľa ďalších spôsobov ako vytvárať exkurzie po množine prípustných návrhov. Napríklad by sa na výber *krokov dopredu* dali použiť vhodné verzie nejakých *pažravých* heuristik (z ang. greedy heuristic). Možno by urýchlili smerovanie exkurzií do *maximálnych* návrhov, na druhú stranu by mohli poukázať výhodu odstraňovania symetrických návrhov. Každopádne v nami vykonaných experimentoch s rôznymi modifikáciami uvedených postupov sme nenašli nič čo by výrazne zlepšilo kvalitu výsledkov.

V spoločnom článku so školiteľom a kolegyňou [21] uvádzame tri diametrálne odlišné príklady na ukážku flexibility nášho algoritmu. Na týchto príkladoch náš algoritmus porovnávame s rôznymi inými používanými algoritmami určenými na riešenie týchto špecifických úloh. Vo všetkých porovnaniach vyšiel buď porovnateľne, alebo lepšie. My sa budeme v numerickej časti zaoberať iba jednou z týchto úloh, ktorá sa venuje navrhovaniu blokových experimentov s veľkosťou bloku dva.

**Vstup** : model, kritérium optimality  $\Phi$ , návrh experimentu  $\xi^{(0)}$  na augmentáciu, počiatočný návrh  $\xi^{(1)}$ , ohraničenia na zdroje, čas behu algoritmu  $t_{max}$ , maximálny počet *krokov dozadu*  $back_{max}$

**Výstup**:  $\xi^+$  najlepší nájdený návrh experimentu

$\xi^+ \leftarrow \xi \leftarrow \xi^{(1)}$ ;  $V \leftarrow \emptyset$ ;  $back_{no} \leftarrow 0$ ;

**repeat**

- if**  $attr(\xi) \notin V$  **then**
  - $V \leftarrow V \cup \{attr(\xi)\}$ ;
  - if**  $\{attr(\zeta) : \zeta \in \mathcal{U}(\xi)\} \not\subseteq V$  **then**
    - $\xi \leftarrow \operatorname{argmax}\{val(\zeta) : \zeta \in \mathcal{U}(\xi), attr(\zeta) \notin V\}$ ;
  - else**
    - if**  $\mathcal{U}(\xi) = \emptyset$  and  $\phi(\xi^+) < \phi(\xi)$  **then**
      - $\xi^+ \leftarrow \xi$ ;  $back_{no} \leftarrow 0$ ;
    - end**
    - if**  $\{attr(\zeta) : \zeta \in \mathcal{L}(\xi)\} \not\subseteq V$  **then**
      - $\xi \leftarrow \operatorname{argmax}\{val(\zeta) : \zeta \in \mathcal{L}(\xi), attr(\zeta) \notin V\}$ ;
      - $back_{no} \leftarrow back_{no} + 1$ ;
    - else**
      - $\xi \leftarrow$  náhodný návrh  $\zeta$  z  $\mathcal{L}(\xi) \cup \mathcal{U}(\xi)$
    - end**
  - end**
  - else**
    - if**  $\{attr(\zeta) : \zeta \in \mathcal{L}(\xi)\} \not\subseteq V$  **then**
      - $\xi \leftarrow \operatorname{argmax}\{val(\zeta) : \zeta \in \mathcal{L}(\xi), attr(\zeta) \notin V\}$ ;
      - $back_{no} \leftarrow back_{no} + 1$ ;
    - else if**  $\{attr(\zeta) : \zeta \in \mathcal{U}(\xi)\} \not\subseteq V$  **then**
      - $\xi \leftarrow \operatorname{argmax}\{val(\zeta) : \zeta \in \mathcal{U}(\xi), attr(\zeta) \notin V\}$ ;
    - else**
      - $\xi \leftarrow$  náhodný návrh  $\zeta$  z  $\mathcal{L}(\xi) \cup \mathcal{U}(\xi)$
    - end**
  - end**
  - if**  $back_{no} > back_{max}$  **then**  $\xi \leftarrow \xi^+$ ;  $back_{no} \leftarrow 0$

**until**  $time > t_{max}$ ;

Schéma 3: Tretí algoritmus

## 3.2 Numerické výsledky

Numerické výsledky uvedené v nasledovnej časti sú získané, až na jednu výnimku, pomocou tretieho algoritmu. Pracovali sme s nastaveniami  $back_{max} = 16$ ,  $n_{round} = 9$ , čas behu algoritmu sme menili podľa riešenej úlohy. Keď sme používali algoritmus na nájdenie optimálneho návrhu pre úlohu s veľkým počtom ošetrení, alebo keď sme na základe výsledkov chceli nájsť protipríklad k nejakej hypotéze, tak sme ho spúšťali na hodinu a dlhšie. Toto sme nemuseli robiť pri jednoduchších úlohách, kde nám stačilo nájsť aj len dostatočne eficientný návrh (alebo sme dokázali nájsť optimálny návrh dostatočne rýchlo). Pretože okrem nájdenia optimálneho návrhu záleží aj na tom, ako rýchlo sa nám to podarí v porovnaní s inými algoritmami.

V celej časti pracujeme s blokovými návrhmi s  $v$  ošetreniami a  $b$  blokmi veľkosti dva. Ako príklad môžeme uvažovať *mikročipové experimenty*, kde vzorky predstavujú  $v$  ošetrení a mikročipy predstavujú  $b$  blokov. Efekty ošetrení sú *významné parametre* a efekty blokov sú *pomocné parametre*. Pričom chceme hľadať najlepší návrh pre odhad rozdielov efektov významných parametrov.

Budeme predpokladať nasledovný model s nekorelovanými pozorovaniami

$$E(Y_j) = \tau(t_1(j)) - \tau(t_2(j)), \quad j \in \{1, \dots, n\},$$

kde  $t_1(j), t_2(j) \in \{1, \dots, v\}$  sú ošetrenia použité v  $j$ -tom bloku, s efektami  $\tau(t_1(j)), \tau(t_2(j))$ , a  $\text{Var}(Y_j) = \sigma^2 < \infty$ ,  $j \in \{1, \dots, b\}$ . Pri takom nastavení je množina bodov experimentu rovná množine všetkých možných dvojíc ošetrení  $\mathfrak{X} = \{(1, 2), (1, 3), \dots, (v - 1, v)\}$ .

Návrh experimentu je daný výberom ošetrení  $t_1(j)$  a  $t_2(j)$ , ktoré sa majú porovnať v  $j$ -tom bloku, pre všetky  $j \in \{1, \dots, b\}$ . Optimálne návrhy pre takto definovaný klasický model (s jedným pozorovaním pre každý blok), sú rovnaké ako optimálne návrhy pre blokový model s veľkosťou bloku dva. Toto platí preto, že keď porovnáваме efekty dvoch ošetrení v spoločnom bloku, tak sa pri sledovaní rozdielu týchto efektov ošetrení vzájomne vyruší efekt bloku.

Tretí algoritmus sme implementovali do prostredí *Matlab* a *R* (viď. [46]). Väčšinu všeobecných úloh sme riešili pre počet ošetrení (vrcholov)  $v = 16$  a počty blokov (hrán)  $b = 15, \dots, 120$ , aby sme pracovali len s jednoduchými súvislými grafmi (pre  $b < 15$  je graf nesúvislý a pre  $b > 120$  graf nie je jednoduchý). Pozrime sa teraz na konkrétne výsledky a hypotézy.



### 3.2.1 Silno regulárne grafy

Prvá otázka, ktorú sme zodpovedali pomocou výsledkov získaných z tretieho algoritmu, je hypotéza dva z článku [10].

**Hypotéza 3.1** (Constantine 1990). *Majú súvislé, silno regulárne grafy maximálnu komplexitu?*

Komplexita grafu znamená počet kostier grafu. Graf s maximálnou komplexitou je  $t$ -optimálny, resp.  $D$ -optimálny. Na hľadanie takýchto grafov môžeme použiť tretí algoritmus. Navyše silno regulárne grafy sú vysoko symetrické, takže by mohli byť vhodnými kandidátmi na grafy optimálnych návrhov. Pozrime sa na ne bližšie, pričom čerpáme z knihy [14].

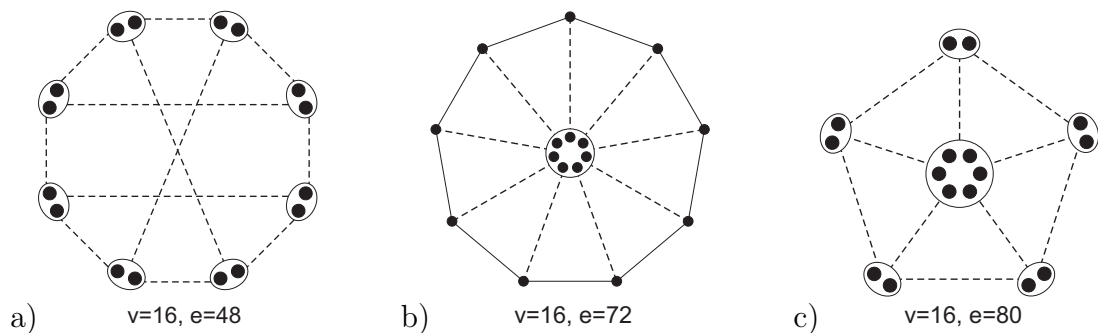
**Definícia 36** (Silno regulárny graf). *Majme graf  $G(v, e)$  na  $v$  vrchoch s  $e$  hranami, pričom každý vrchol je rovnakého stupňa  $k$ . Graf  $G$  nazývame silno regulárny (z ang. *strongly regular graph*), ak existujú také nezáporné celé čísla  $\alpha, \beta$ , že každé dva susedné vrcholy majú presne  $\alpha$  spoločných susedných vrcholov, a každé dva nesusedné vrcholy majú presne  $\beta$  spoločných susedných vrcholov. Budeme ich označovať  $srg(v, k, \alpha, \beta)$ .*

Platí, že ak je nejaký graf silno regulárny, tak potom aj k nemu komplementárny graf je silno regulárny. Silno regulárne grafy delíme na *primitívne* a *neprimitívne*. *Primitívne* grafy  $G$  sú tie, pre ktoré platí, že aj  $G$  aj jeho komplement  $\bar{G}$  sú súvislé. Tým pádom *neprimitívne* sú buď nesúvislé, alebo majú nesúvislý komplement, a teda  $\beta$  je rovné nula alebo  $k$ . Špeciálne prípady *neprimitívnych* silno regulárnych grafov sú kompletne grafy,  $srg(v, v-1, v-2, 0)$ , kompletne multipartitné grafy s rovnako veľkými partíciami,  $srg(v, k, 2k-v, k)$ . Alebo grafy k nim komplementáre, ktoré ale nie sú súvislé a teda ich počet kostier bude vždy nulový. Kompletne a kompletne multipartitné grafy s rovnako veľkými partíciami sú Schurovsky optimálne, teda aj  $t$ -optimálne (viď. druhá kapitola). Takže pre tieto dva špeciálne prípady *neprimitívnych* silno regulárnych grafov hypotéza 3.1 platí.

Na  $v = 16$  vrchoch existujú štyri *primitívne* silno regulárne grafy (viď. [14], kapitola VII.11). Najmenší (vzhľadom na počet hrán) z nich je  $srg(16, 10, 6, 6)$ , takzvaný Clebschov graf so 40 hranami. Tento graf vyšiel ako  $t$ -optimálny na základe tretieho algoritmu spusteného na jednu hodinu. Predpokladáme, že je naozaj  $t$ -optimálny. Pre zvyšné tri grafy (Shrikhande graf -  $srg(16, 6, 2, 2)$  s 48 hranami; komplement k Shrikhande grafu -  $srg(16, 9, 4, 6)$  s 72 hranami; komplement k Clebschovmu grafu -  $srg(16, 5, 0, 2)$  s 80 hranami), sme našli na rovnakom počte hrán a vrcholov grafy

s väčším počtom kostier. Pričom hodnoty kritérií  $D$ -optimality návrhov prislúchajúcich k týmto trom silno regulárnym grafom sú postupne 98.65%, 99.68% a 99.61% z hodnôt kritérií  $D$ -optimality optimálnych návrhov nájdených tretím algoritmom. Párovacie grafy nami nájdených  $D$ -optimálnych a teda aj  $t$ -optimálnych návrhov sú zobrazené na obrázku 3–3. Hlavný rozdiel medzi našimi grafmi a príslušnými silno regulárnymi grafmi je, že naše grafy obsahujú málo trojuholníkov a veľké množstvo kompletných bipartitných podgrafov.

Z istotou nemôžeme tvrdiť, že nami nájdené tri grafy sú  $D$ -optimálne, ale majú väčšiu hodnotu  $D$ -kritéria ako k nim príslušné silno regulárne grafy, takže hypotéza 3.1 vo všeobecnosti neplatí.



**Obrázok 3–3:** Grafy o ktorých na základe tretieho algoritmu predpokladáme, že sú  $t$ -optimálne (resp.  $D$ -optimálne), pre  $v = 16$  vrcholov a postupne  $e = 48, 72$  a  $80$  hrán. Grafy sú reprezentované pomocou zhlukov vrcholov a špeciálnych hrán. Plné čiary predstavujú normálne hrany, čiarkované čiary reprezentujú v prípade  $e = 48$  kompletne bipartitné podgrafy  $K_{2,2}$ , kde zhluky vrcholov sú jednotlivé dvojprvkové partície. V prípade  $e = 72$  sú čiarkované čiary kompletne bipartitné grafy  $K_{1,7}$  a pre  $e = 80$  sú čiarkované čiary buď  $K_{2,2}$  alebo  $K_{2,6}$ .

### 3.2.2 Mooreove grafy

Ďalšia trieda grafov, ktoré nás budú zaujímať sú Mooreove grafy. Hypotéza o tom že sú  $t$ -optimálne je z článku [27].

**Hypotéza 3.2.** *Mooreove grafy majú maximálny počet kostier spomedzi všetkých jednoduchých grafov s rovnakým počtom vrcholov a hrán.*

Túto hypotézu podporíme teoretickými výsledkami, aj výsledkami nášho algoritmu. Najskôr sa pozrime, čo sú to Mooreove grafy (viď. [6]).

**Definícia 37** (Mooreov graf). *Majme súvislý graf  $G$  s priemerom  $d$  (maximum spomedzi najkratších vzdialeností medzi ľubovoľnými dvomi vrcholmi). Potom  $G$  je Mooreov, ak je dĺžka najkratšieho cyklu v grafe rovná  $g = 2d + 1$ .*

Dá sa ukázať, že Mooreov graf musí byť regulárny, teda má rovnaké stupne vrcholov. Označme ich pre ďalší postup  $k$ .

Pre špeciálne voľby parametrov  $(k, g)$  dostaneme známe grafy. Pre veľkosť najmenšieho cyklu  $g = 3$  sú Mooreove grafy kompletne grafy  $K_{k+1}$ . Pre veľkosť najmenšieho cyklu  $g = 4$  sú Mooreove grafy kompletne bipartitné grafy  $K_{k,k}$  (viď. [6]). O týchto grafoch sme v druhej kapitole dokázali, že sú  $t$ -optimálne. Takže Mooreove grafy pre  $g = 3, 4$  sú  $t$ -optimálne.

V knihe [6], kapitola 23, je dokázané, že pre  $g > 4$  Mooreove existujú iba pre  $g = 5$  a  $k = 3$  (Petersenov graf),  $k = 7$  (Hoffmann-Singletonov graf) a možno pre  $k = 57$ , alebo pre  $g = 6, 8$  alebo 12. Grafmi s  $g > 5$  sa nebudeme zaoberať.

Pozrieme sa pomocou tretieho algoritmu na Mooreove grafy  $(5, 3)$  (Petersenov graf s 10 vrcholmi a 15 hranami) a  $(5, 7)$  (Hoffmann-Singletonov graf s 50 vrcholmi a 175 hranami). Pre 10 vrcholov a 15 hrán sme spustili tretí algoritmus desaťkrát na dve minúty. Vo všetkých prípadoch našiel ako výsledný optimálny návrh taký návrh, ktorého párovací graf je Petersenov graf.

Čo sa týka grafu Hoffmann-Singleton, tak sú známe vlastné čísla jeho Laplaciánu -  $\lambda_1 = \dots = \lambda_{21} = 10$ ,  $\lambda_{22}, \dots, \lambda_{49} = 5$  a  $\lambda_{50} = 0$ . Tým pádom hodnota  $D$ -optimality je rovná  $(10^{21} \cdot 5^{28})^{1/49}$ , čo je približne 6,7295.

Spustili sme tretí algoritmus pre 50 vrcholov a 175 hrán na rôzne časy - na päť minút, na jednu hodinu a na šesť hodín. Ako najlepšie návrhy nám v jednotlivých prípadoch vrátil návrhy s hodnotou kritéria  $D$ -optimality postupne rovnou 6,5629; 6,7186 a 6.7201. Čo predstavuje postupne 97,52%; 99,84% a 99.86% z návrhu, ktorého párovací graf je Hoffman-Singletonov graf.

Síce sa nám nepodarilo nájsť potenciálne optimálny návrh, ktorému prislúcha Hoffman-Singletonov graf, ale už za päť minút sme našli návrh s pomerne vysokou hodnotou kritéria  $D$ -optimality. Pri spustení algoritmu na dlhšie časy sme sa ešte viac priblížili potencióálne optimálnemu návrhu.

Na základe výsledkov pokus hypotézu 3.2 nezamietame a myslíme si, že platí.

### 3.2.3 Jednoduché grafy

V tejto podkapitole vyslovíme hypotézu, ktorá je síce veľmi prirodzená, ale ani v teórii navrhovania experimentov, ani v teórii grafov sme ju nenašli dokázanú a nepodarilo sa ju dokázať ani nám. Preto ponúkame len jej empirické overenie. Pripomíname, že jednoduché grafy sú neorientované grafy bez slučiek a násobných hrán.

**Hypotéza 3.3.** *Nech  $G(v, e)$  je graf na  $v$  vrchoch s  $e$  hranami, pričom  $e \leq v(v-1)/2$ . Potom graf prislúchajúci k Schurovsky optimálnemu návrhu je jednoduchý.*

Počas hľadania optimálnych návrhov s rôznymi zadaniami sme nenašli jediný návrh, ktorý by toto pravidlo porušil. Navyše sme vykonali nasledovné overenie.

Na overenie hypotézy sme vybrali kritérium  $E$ -optimality. Toto kritérium závisí len na druhej najmenej vlastnej hodnote Laplaciánu grafu a súvisí s algebraickou konektivitou. Teda je to určitý spodný odhad toho, s ako veľmi poprepájaným grafom pracujeme. Na tom by sa malo rýchlo prejavíť, či sa oplatia používať násobné hrany. Pracovali sme s návrhmi, ktorých párovacie grafy majú  $v = 16$  vrcholov a postupne  $e = 15, \dots, 120$  hrán. Pre každý počet hrán sme spustili tretí algoritmus na desať minút. Použili sme dva rôzne druhy ohraničení. V prvej sade sme hľadali optimálny návrh (a k nemu prislúchajúci graf) len s ohraničením na celkový počet blokov (hrán), teda sa mohli vytvárať aj násobné hrany (v jednom bode návrhu sme mohli vykonať viacero pokusov). V druhej sade sme okrem ohraničenia na celkový počet hrán pridali aj ohraničenia, aby sme nemohli vytvárať násobné hrany. Teda v každom bode návrhu mohol byť vykonaný maximálne jeden pokus.

Porovnali sme hodnoty kritéria  $E$ -optimality všetkých optimálnych návrhov z týchto dvoch sád. Skoro všetky porovnania vyšli rovnako, až na pár prípadov, keď vyšla hodnota kritéria  $E$ -optimality vyššia pre návrh s obmedzením, že nemôže vytvárať násobné hrany. Dôvod, prečo to vyšlo takto je, že množina prípustných návrhov ohraničených len celkovým počtom hrán je omnoho väčšia a teda trvá dlhšie, pokiaľ ju postupne prehľadáme.

Takže hypotézu 3.3 sa nám nepodarilo vyvrátiť a na základe prehľadu v tejto téme a vykonaných experimentov si myslíme, že by mala byť pravdivá.

### 3.2.4 Schurovsky optimálne grafy podľa Constantineho

Ako ďalšie podporíme tvrdenie, ktoré vyslovil Constantine na konci článku [9]. Toto tvrdenie naväzuje na časť 2.3.2.

**Tvrdenie 3.1.** *Pri odobratí viac ako  $(v + 1)/2$  hrán z kompletneho grafu  $G$  na  $v$  vrchoch nemôžeme byť návrh, ktorému prislúcha výsledný párovací graf, Schurovsky optimálny (optimálny vzhľadom na všetky ortogonálne kritéria optimality).*

Myslíme si, že je to pomerne netriviálne tvrdenie, ale nepodarilo sa nám ho nájsť v iných zdrojoch a ani ho dokázať. Jediné čo sa nám podarilo urobiť, bolo overiť toto

tvrdenie kompletnou enumeráciou vlastných čísel všetkých možných grafov. Bohužiaľ tento prístup funguje len pre grafy na malých počtoch vrcholov - nám sa to skrz vysoký výpočtový čas podarilo len pre grafy na siedmych a menej vrchoch.

Pre každý z grafov sme našli vektor vlastných čísel Laplaciánu a porovnali ho s ostatnými vektormi vlastných čísel Laplaciánov grafov na rovnakom počte vrcholov. Pre počet hrán menší ako  $v(v-1)/2 - (v+1)/2$  sme nenašli žiaden párovací graf, ktorý by prislúchal Schurovsky optimálnemu návrhu (vektor vlastných čísel jeho Laplaciánu by majorizoval vektory vlastných čísel Laplaciánov všetkých ostatných grafov na rovnakom počte vrcholov.)

Takže pre grafy s počtom vrcholov sedem a menej tvrdenie 3.1 platí. Pre väčšie grafy to zatiaľ nevieme ani dokázať, ani numericky overiť.

### 3.2.5 Skoro-regulárne kompletné multipartitné grafy

V podkapitole 2.4.3 sme ukázali, že augmentácia regulárneho kompletného multipartitného grafu pomocou pridania nesusedných hrán (prípadne ešte jednej hrany navyše) je Schurovsky optimálna. Myslíme si, že rovnaké tvrdenie platí aj pre skoro-regulárne kompletné multipartitné grafy. Multipartitné grafy s inými veľkosťami nás nebudú zaujímať, lebo vďaka článku [44] vieme, že grafy optimálnych návrhov musia byť regulárne, alebo skoro-regulárne.

**Hypotéza 3.4.** *Jednoduchý graf, ktorý vznikol zo skoro-regulárneho kompletného multipartitného grafu pridaním  $k$ -nesusedných hrán (pre  $v$  nepárne ešte aj s pridaním  $(v-1)/2$  nesusedných hrán a jednej hrany, ktorú pridáme k vrcholu stupňa nula) je Schurovsky najlepšia augmentácia skoro-regulárneho kompletného multipartitného grafu.*

Túto hypotézu sme numericky overili pre  $v = 16$  vrcholov a počty hrán rovné počtom hrán v možných skoro-regulárnych kompletných multipartitných grafoch zvýšeným o  $k$ , kde  $k \in \{1, \dots, 8\}$ . Konkrétne pre počty hrán  $e = 85, 102, 106, 109, 113, 114, \dots, 120$  postupne zvýšene o  $1, \dots, 8$ , pričom sme počet hrán nikdy nezvýšili nad 120, aby sme pracovali iba s jednoduchými grafmi.

Myslíme si, že táto hypotéza (3.4) sa bude dať dokázať podobným spôsobom ako veta (2.15). Využije sa, že vlastné čísla Laplaciánu skoro-regulárneho kompletného grafu (okrem nuly) sa líšia iba o jedna. Potom by sa pre ne mohli dať použiť podobné tvrdenia ako v 2.15. Bohužiaľ, zatiaľ sa nám to nepodarilo dokázať.

### 3.2.6 Porovnanie tretieho algoritmu so simulovaným žíhaním [54]

V tejto časti ukážeme, ako tretí algoritmus funguje pri jedinom štandardnom ohraničení 1.11, kde nesmieme prekročiť celkový počet pokusov. Budeme pracovať s blokovými návrhmi zavedenými na začiatku numerickej časti. Síce tým nevyužijeme naplno možnosti tretieho algoritmu, pretože jeho hlavnou prednosťou je, že vie pracovať s viacerými *ohraničeniami na zdroje*, ale pre úlohy s viacerými ohraničeniami neexistujú všeobecné algoritmy, s ktorými by sme ho mohli porovnať. Existujú iba veľmi špecifické algoritmy určené na riešenie konkrétnych praktických úloh.

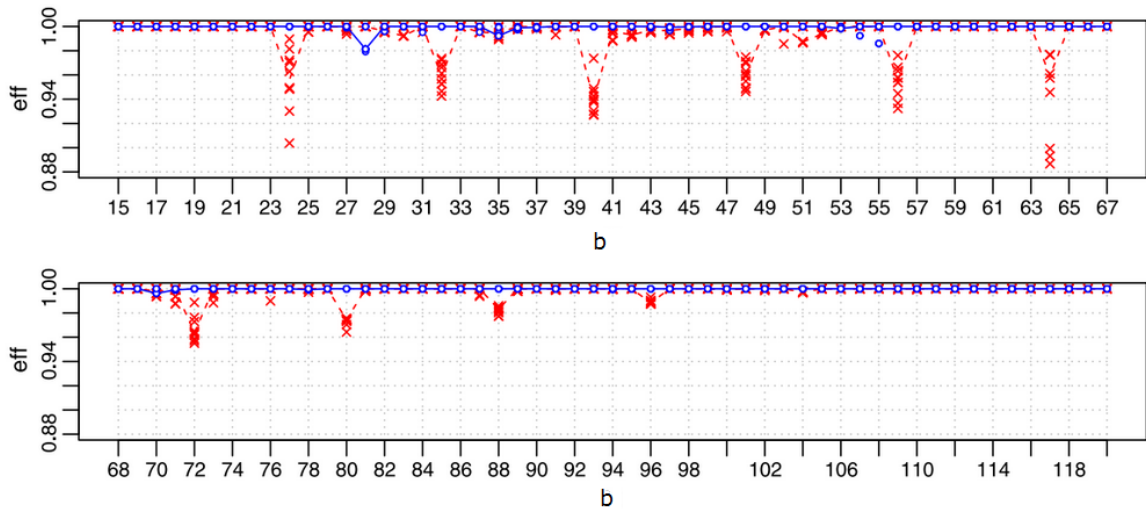
Použijeme implementáciu tretieho algoritmu v prostredí R a vypočítame  $D$ -optimálne návrhy pre  $v = 16$  ošetrení a  $b = 15, \dots, 120$  blokov, pre čas  $t_{\max} = 120$  sekúnd. Potom tieto návrhy porovnáme s výsledkami algoritmu simulovaného žíhania pre procedúru *od* implementovanú v programe R, v balíčku “smida” (viď. [54]), pre parametre *criterion* = “ $D$ ”, *dye* = FALSE, a počet iterácií *n.iter* nastavený tak, aby algoritmus bežal približne rovnako dlho ako náš algoritmus, teda  $t_{\max} = 120$  sekúnd. Oba algoritmy spustíme desaťkrát zakaždým z iného náhodného počiatočného návrhu.

Z obrázku 3–4 vidno, že tretí algoritmus našiel rovnaké, alebo lepšie návrhy ako simulované žíhanie (až na malé výnimky  $b = 28, 36, 53, 54, 55, 78$ ). Z týchto výsledkov sme dostali aj výsledky použité v kapitole o *silno regulárnych grafoch*.

### 3.2.7 Druhý algoritmus a ohraničenie na počty ošetrení okrem kontrolného ošetrenia

Nakoniec ukážeme fungovanie druhého algoritmu. Okrem štandardného ohraničenia 1.11 na celkový počet pokusov pridáme ďalšie *ohraničenia na zdroje*. Pomerne často sa v experimentoch pracuje s jedným kontrolným ošetrením (placebo, bežné ošetrenie, pri mikročipových experimentoch mRNA zo zdravej bunky) s ktorým sa porovnávajú ostatné ošetrenia (rôzne lieky, nové ošetrenie, mRNA z chorých buniek). V takomto prípade je prirodzené obmedziť počet použitých ostatných ošetrení, zatiaľ čo kontrolné ošetrenie môžeme používať neobmedzene veľa krát, pokiaľ to dovoľí celkový počet pokusov. Pozrime sa na konkrétnu úlohu s ohraničeným počtom ošetrení okrem kontrolného ošetrenia.

**Úloha 3.1.** *Je daný experiment v ktorom sa pracuje s  $v$  ošetreniami a  $b$  blokmi. Jedno ošetrenie je kontrolné (nech mu prislúcha parameter  $\tau_1$ ) a je ho neobmedzené množstvo. Ostatné ošetrenia sa môžu použiť len  $r$ -krát. Chceme odhadnúť rozdiely efektov medzi kontrolným ošetrením a ostatnými ošetreniami  $\tau_1 - \tau_2, \dots, \tau_1 - \tau_v$ .*



**Obrázok 3–4:** Numerické výsledky pre úlohu 3.2.6. Na vodorovnej osi sú veľkosti blokov (počty hrán). Na zvislej osi je  $D$ –eficiencia exaktných návrhov získaných tretím algoritmom (krúžky) a  $D$ –eficiencia exaktných návrhov získaných simulovaným žiháním (krížiky). Pričom  $D$ –eficiencia je pomer hodnoty kritéria  $D$ –optimality návrhu, vzhľadom na najlepší nájdený návrh pre daný počet blokov. Plná čiara spája mediány množín výsledkov tretieho algoritmu pre jednotlivé počty blokov. Čiarkovaná čiara spája mediány množín výsledkov algoritmu simulovaného žihania pre jednotlivé počty blokov.

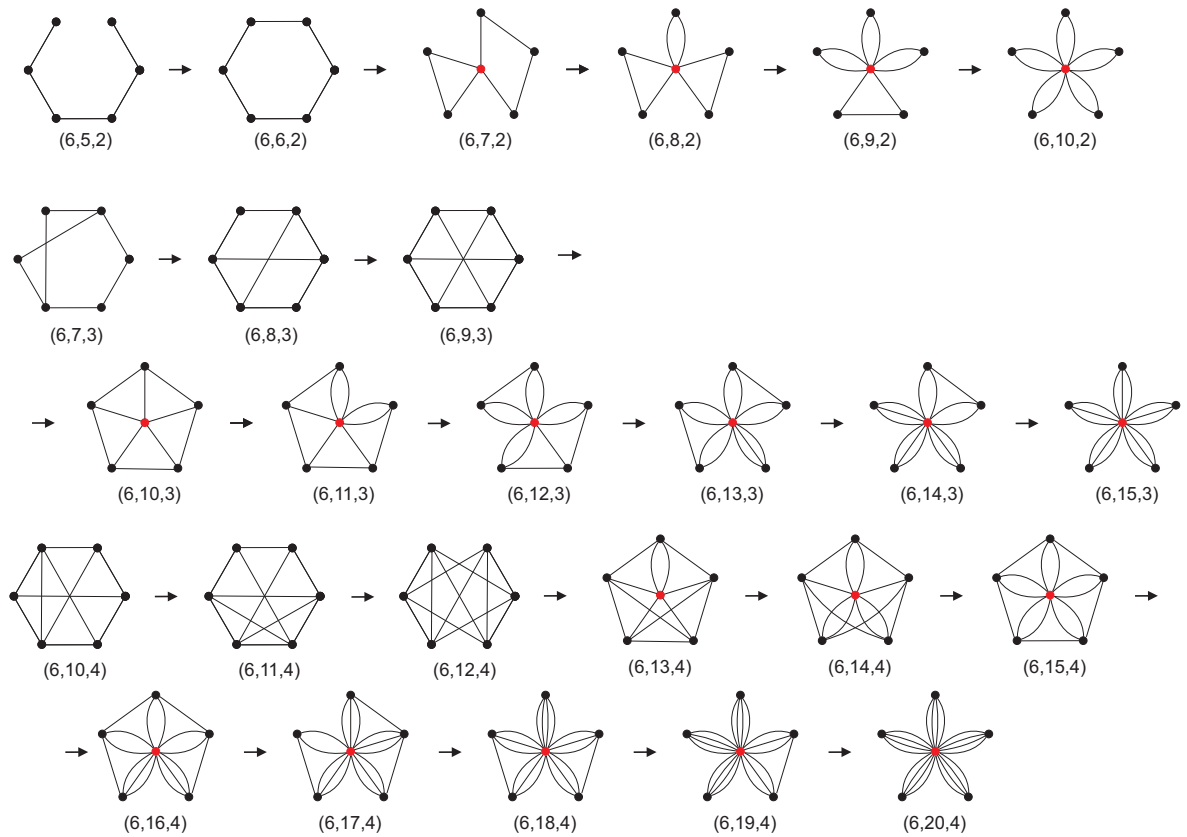
Úlohu riešime pomocou modelu zavedeného na začiatku numerickej časti, pričom využijeme parciálne kritérium  $\mathbf{D}_P$ , kde  $P$  vyberá subsystém pre lineárnu kombináciu rozdielov efektov ošetrení.

Ohraničenia vymenované v úvode úlohy vieme zaznačiť v tvare *ohraničení na zdroje* 1.12. Ohraničenie na celkový počet pokusov pomocou štandardného ohraničenia 1.11 a ohraničenia na ošetrenia, okrem kontrolného, pomocou 1.1.5.

Spustením druhého algoritmu sme našli  $D_P$ –optimálne návrhy experimentov pre  $v = 6$  ošetrení,  $b = 5, \dots, 20$  blokov a ohraničenie  $r = 2, 3, 4$  na počet použítí ošetrení okrem kontrolného ošetrenia. Výsledné návrhy sú zobrazené pomocou *párovacích grafov* (viď. 3–5), pričom údaje o návrhu sú zaznačené v tvare  $(v, b, r)$ , kde  $v$  je počet ošetrení,  $b$  je počet blokov a  $r$  je horné ohraničenie na počet použitých ošetrení okrem kontrolného. Vrchol prislúchajúci kontrolnému ošetreniu je zafarbený na červeno.

### 3.3 Záver

Ukázali sme použitie druhého a tretieho algoritmu na konkrétnych úlohách. Podarilo sa nám vyvrátiť jednu hypotézu a niekoľko ďalších podporiť. Veríme, že by sa ďalšími nu-



**Obrázok 3–5:** Párovacie grafy pre  $D_{\mathcal{P}}$ -optimálne návrhy experimentov, pre  $v = 6$  ošetrení,  $b = 5, \dots, 20$  blokov a ohraňenie  $r = 2, 3, 4$  na počet použítí ošetrení, okrem kontrolného ošetrenia.

merickými experimentmi dali objaviť ďalšie zaujímavé hypotézy, ktoré potom môžeme skúsiť dokázať. Hlavné využitie algoritmov by ale malo byť na hľadanie dostatočne dobrých návrhov pre praktické úlohy.



## Záver

V dizertačnej práci sme sa zaoberali prepojením medzi dvomi matematickými odbormi - teóriou navrhovania experimentov a teóriou grafov. Problém bol zadaný ako hľadanie exaktných optimálnych blokových návrhov s veľkosťou bloku dva. Príslušná teória bola zavedená v prvej kapitole. Exaktné blokové návrhy s veľkosťou bloku dva vieme prezentovať pomocou párovacích grafov. Potom veľa s ich vlastností vieme vyjadriť ako vlastnosti týchto párovacích grafov.

V druhej kapitole sme sa zaoberali optimálnou augmentáciou už vykonaných experimentov. Predstavili dve známe metódy uvedené v prácach 2.3.1 a 2.3.2, pomocou ktorých vieme dobre pracovať s vlastnými číslami Laplaciánov grafov prislúchajúcich k návrhom experimentov. Pomocou rozšírenia týchto metód sme našli a dokázali optimalitu pre tri nové triedy augmentovaných blokových návrhov s veľkosťou bloku dva. Optimálne sú vzhľadom na širokú triedu ortogonálne invariantných kritérií optimality.

V poslednej, tretej, kapitole sme ukázali tri stochastické algoritmy určené na hľadanie optimálnych grafov. Ukázali sme ako a prečo fungujú a ako sa dajú použiť. Potom sme pomocou týchto algoritmov vygenerovali viacero numerických výsledkov. Podarilo sa nám vyvrátiť jednu hypotézu pre silno-regulárne grafy a viacero ďalších hypotéz potvrdiť pre menšie prípady. Na základe numerických výsledkov sme vyslovili niekoľko nových hypotéz, ktorých teoretický dôkaz však zostáva pre budúci výskum.

Celkovo si myslím, že téma prepojenia teórie navrhovania experimentov a teórie grafov ešte nie je vyčerpaná, a je v nej priestor na ďalší výskum. Takto získané výsledky môžu byť užitočné v mnohých praktických aplikáciách.

**\*Literatúra**

- [1] Atkinson, A.C., Donev, A.N., Tobias, R.D.: Optimum Experimental Designs, With SAS. Oxford University Press (2007)
- [2] Bachratá, A.: Návrh mikročipových experimentov, diplomová práca (2011)
- [3] Bachratá, A., Harman, R.: A stochastic optimization method for constructing optimal block designs with linear constraints. Proceedings from the European Young Statisticians Meeting, Osijek, 2013 (2014)
- [4] Bailey, R.A.: Design of Comparative Experiments. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press (2008)
- [5] Bailey, R.A.: Designs for two-colour microarray experiments. J. Roy. Stat. Soc. C Appl. Stat. 56(4), 365-394 (2007)
- [6] Biggs, N.: Algebraic graph theory (2nd ed). Cambridge Univ. Press, (1993)
- [7] Bailey, R.A., Cameron, P.J.: Combinatorics of optimal designs. Surveys in Combinatorics - London Mathematical Society Lecture Note Series 365, 19-73 (2009)
- [8] Černý, V.: A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm. Journal of Optim, Theory and Applications 45, pages 41-51, (1985)
- [9] Constantine, G. M.: Schur convex functions on the spectra of graphs, Discrete Mathematics 45: 181-188, (1983)
- [10] Constantine, G. M.: Graph complexity and the Laplacian matrix in blocked experiments, Linear and Multilinear Algebra 28: 49-56, (1990)
- [11] Cheng, C.S.: Maximizing the total number of spanning trees in a graph: two related problems in graph theory and optimum design theory. J. Combin. Theor. B 31, 240-248 (1981)
- [12] Chern, M.S.: On the computational complexity of reliability redundancy allocation in a series system. Oper. Res. Lett. 11, 309-315 (1992)
- [13] Cook, R.D., Thibodeau, L.A.: Marginally restricted D-optimal designs. J. Am. Stat. Assoc. 75(370), 366-371 (1980)

- 
- [14] Colbourn, C.J., Dinitz, J.H.: CRC Handbook of Combinatorial Designs (2nd ed.). CRC Press (2007)
- [15] Dykstra, O.: The augmentation of experimental data to maximize  $|X'X|$ . *Technometrics* 13, 682-688 (1971)
- [16] Fedorov, V.V.: Optimal design with bounded density: Optimization algorithms of the exchange type. *J. Stat. Plann. Infer.* 22, 1-13 (1982)
- [17] Fiedler, M.: Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(2):298–305, (1975)
- [18] Fisher, R. A.: *The design of Experiments*, Edinburg: Oliver and Boyd (1935)
- [19] Gilbert, B., Myrvold, W.: Maximizing Spanning Trees in Almost Complete Graphs, *Networks* 30: 97–104, (1997)
- [20] Glover, F., Laguna, M.: *Tabu search*. Springer (1999)
- [21] Harman, R., Bachratá, A. and Filová, L.: Construction of efficient experimental designs under multiple resource constraints. *Appl. Stochastic Models Bus. Ind.*, doi: 10.1002/asmb.2117. (2015)
- [22] Harman, R.: Minimal efficiency of designs under the class of orthogonally invariant information criteria, *Metrika*, Volume 60, No. 2, pp. 137-153 (2004)
- [23] Harman, R.: Equivalence theorem for Schur optimality of experimental designs. *J. Stat. Plann. Infer.* 138, 1201 - 1209 (2008)
- [24] Harman, R.: Multiplicative Methods for Computing D-Optimal Stratified Designs of Experiments. *J. Stat. Plann. Infer.* 146, 82 - 94 (2014)
- [25] Harman, R., Filová, L.: Computing efficient exact designs of experiments using integer quadratic programming. *Comput. Stat. Data Anal.* 71, 1159-1167 (2014)
- [26] Hochbaum, D.S.: A nonlinear Knapsack problem. *Oper. Res. Lett.* 17(3), 103 - 110 (1995)
- [27] Jakobson, D., Rivin, I.: On some extremal problems in graph theory. [arXiv:math.CO/9907050](https://arxiv.org/abs/math.CO/9907050).(1999)

- 
- [28] Goos, P., Jones, B.: Optimal Design of Experiments: A Case Study Approach. John Wiley & Sons, New York (2011)
- [29] Jung, J.S., Yum, B.J.: Construction of exact D-optimal designs by tabu search. Computational Statistics & Data Analysis 21, 181-191 (1996)
- [30] Kellerer, H., Pferschy, U., Pisinger, D. : Knapsack Problems. Springer (1993)
- [31] Kelmans, A. K.: The number of trees of a graph I. (Russian) Avtomat. Telemekh. 26: 2194-2204, (1965)
- [32] Kelmans, A. K.: The number of trees of a graph II. (Russian) Automat. Remote Control 27: 56-65, (1966)
- [33] Kelmans, A. K.: Comparison of graphs by their number of spanning trees Discrete Math 16: 241-261, (1976)
- [34] Kelmans, A. K., Chelnokov, V. M.: A certain polynomial of a Graph and Graphs with an Extremal Number of Trees. J. Combin. theory (B) 16: 197-214, (1974)
- [35] Kim, J., Yum, B.: A heuristic method for solving redundancy optimization problems in complex systems. IEEE Trans. Reliab. 42(4), 572-578 (1993)
- [36] Kirchhoff, G.: 'Über die Auflösung der Gleichung, auf welche man bei der Untersuchung der linearen Verteilung galvanischer Ströme geführt wird. Annals of Physical Chemistry, 72, 497-508 (1847)
- [37] Kuo, W., Prasad, V.R., Tillman, F., Hwang, C.: Optimal reliability design: fundamentals and applications. Cambridge university press (2001)
- [38] Lewis, R.P.: The number of spanning trees of a complete multipartite graph. Discrete Math. 197/198(0), 537 - 541 (1999)
- [39] Martín-Martín, R., Torsney, B., López-Fidalgo, J.: Construction of marginally and conditionally restricted designs using multiplicative algorithms. Comput. Stat. Data Anal. 51(12), 5547-5561 (2007)
- [40] Matoušek, J., Nešetřil, J.: Invitation to Discrete Mathematics. Oxford University Press (2008)
- [41] Michalewicz, Z., Fogel, D.B.: How to Solve It: Modern Heuristics. Springer (2008)

- 
- [42] Mitchell, T.J.: An Algorithm for the Construction of “D-Optimal” Experimental Designs. *Technometrics* 16(2), 203-210 (1974)
- [43] Nair Maria Maia de Abreu.: Old and new results on algebraic connectivity of graphs. *Linear Algebra and its Applications*, 423(1):53 – 73, (2007)
- [44] Petingi, L., Rodriguez, J.: A new technique for the characterization of graphs with a maximum number of spanning trees. *Discrete Math.* 244, 351-373 (2002)
- [45] Pukelsheim, F.: *Optimal Design of Experiments*. SIAM (2006)
- [46] R Development Core Team: *R: A Language and Environment for Statistical Computing*, Foundation for Statistical Computing. Vienna, Austria (2011)
- [47] Sagnol, G., Harman, R.: Computing exact D-optimal designs by mixed integer second order cone programming. arXiv preprint, arXiv:1307.4953 (2013)
- [48] Schneider, J., Kirkpatrick, S.: *Stochastic Optimization*. Springer (2006)
- [49] Tack, V., Vandebroek, M.: Budget constrained run orders in optimum design. *J. Stat. Plann. Infer.* 124, 231-249 (2004)
- [50] Tjur, T.: Block designs and electrical networks, *Annals of Statistic* 19, 1010-1027 (1991)
- [51] Yee Hwa Yang, Speed T.: Design issues for cDNA microarray experiment, *Nature reviews, Genetics*, Volume 3, (2002)
- [52] Welch, W.J.: Branch-and-bound search for experimental designs based on D-optimality and other criteria. *Technometrics* 24(1), 41-48 (1982)
- [53] Welch, W.J.: *Computer-Aided Design of Experiments for Response Estimation*. *Technometrics* 26(3), 217-224 (1984)
- [54] Wit, E., Nobile, A., Khanin, R.: Near-optimal designs for dual-channel microarray studies. *Jurnal of the Royal Stat. Society, Series C (Applied Statistics)*, Volume 54, Issue 5, pages 817-830, (2005)
- [55] Wright, S.E., Sigal, B.M., Bailer, A.J.: Workweek Optimization of Experimental Designs: Exact Designs for Variable Sampling Costs. *J. Agr. Biol. Environ. Stat.* 15(4), 491-509 (2010)

- [56] Wynn, H.P.: The sequential generation of D-optimum experimental designs. Ann.Math.Statist. 41, 1655-1664 (1970)

## Prílohy:

### Príloha A - Laplaciány, Laplaceove polynómy a vlastné čísla grafov fragmentov a kompletného grafu

Ukážeme Laplaciány, Laplaceove polynómy a vlastné čísla grafov fragmentov z obrázkov 2-2, 2-3 a kompletného grafu  $K_v$ . Budeme pracovať s grafmi na  $v$  vrcholoch. Matice Laplaciánov obsahujú veľké množstvo núl, nebudeme ich vypisovať všetky. Teda ak nie sú niektoré prvky Laplaciánu vypísané, tak sú nulové.

- Graf  $G_v^0$  neobsahujúci žiadnu hranu.

Laplacián:

$$\mathbf{L}(G_v^0) = (\mathbf{0}_{v \times v});$$

Laplaceov polynóm:

$$\mu(G_v^0, x) = \det[x \cdot \mathbf{I}_v - \mathbf{L}(G_v^0)] = x^v;$$

Vektor usporiadaných vlastných čísel:

$$\lambda(G_v^0) = \underbrace{(0, \dots, 0)'}_v.$$

- Graf  $G_v^I$  s jednou hranou.

Laplacián:

$$\mathbf{L}(G_v^I) = \begin{pmatrix} 1 & -1 & & \dots & \\ -1 & 1 & & \dots & \\ \vdots & \vdots & \mathbf{0}_{(v-2) \times (v-2)} & & \end{pmatrix};$$

Laplaceov polynóm:

$$\mu(G_v^I, x) = \det[x \cdot \mathbf{I}_v - \mathbf{L}(G_v^I)] = x^{v-1}(x-2);$$

Vektor usporiadaných vlastných čísel:

$$\lambda(G_v^I) = (2, \underbrace{0, \dots, 0}')._{v-1}$$

- Graf  $G_v^{II}$  s dvomi nesusednými hranami.

Laplacián:

$$\mathbf{L}(G_v^{II}) = \begin{pmatrix} 1 & -1 & 0 & 0 & \dots & \\ -1 & 1 & 0 & 0 & \dots & \\ 0 & 0 & 1 & -1 & \dots & \\ 0 & 0 & -1 & 1 & \dots & \\ \vdots & \vdots & \vdots & \vdots & \mathbf{0}_{(v-4) \times (v-4)} & \end{pmatrix};$$

Laplaceov polynóm:

$$\mu(G_v^{II}, x) = \det[x \cdot \mathbf{I}_v - \mathbf{L}(G_v^{II})] = x^{v-2}(x-2)^2;$$

Vektor usporiadaných vlastných čísel:

$$\lambda(G_v^{II}) = (2, 2, \underbrace{0, \dots, 0}')._{v-2}$$

- **Graf  $G_v^V$  s dvomi susednými hranami.**

Laplacián:

$$\mathbf{L}(G_v^V) = \begin{pmatrix} 1 & -1 & 0 & \dots \\ -1 & 2 & -1 & \dots \\ 0 & -1 & 1 & \dots \\ \vdots & \vdots & \vdots & \mathbf{0}_{(v-3) \times (v-3)} \end{pmatrix};$$

Laplaceov polynóm:

$$\mu(G_v^V, x) = \det[x \cdot \mathbf{I}_v - \mathbf{L}(G_v^V)] = x^{v-2}(x-3)(x-1);$$

Vektor usporiadaných vlastných čísel:

$$\lambda(G_v^V) = (3, 1, \underbrace{0, \dots, 0}'_{v-2}).$$

- **Graf  $G_v^{III}$  s tromi nesusednými hranami.**

Laplacián:

$$\mathbf{L}(G_v^{III}) = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & \dots \\ -1 & 1 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & -1 & 0 & 0 & \dots \\ 0 & 0 & -1 & 1 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 1 & -1 & \dots \\ 0 & 0 & 0 & 0 & -1 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \mathbf{0}_{(v-6) \times (v-6)} \end{pmatrix};$$

Laplaceov polynóm:

$$\mu(G_v^{III}, x) = \det[x \cdot \mathbf{I}_v - \mathbf{L}(G_v^{III})] = x^{v-3}(x-2)^3;$$

Vektor usporiadaných vlastných čísel:

$$\lambda(G_v^{III}) = (2, 2, 2, \underbrace{0, \dots, 0}'_{v-3}).$$

- **Graf  $G_v^{VI}$  s dvomi susednými hranami a jednou s nimi nesusednou hranou.**

Laplacián:

$$\mathbf{L}(G_v^{VI}) = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & \dots \\ -1 & 2 & -1 & 0 & 0 & \dots \\ 0 & -1 & 1 & 0 & 0 & \dots \\ 0 & 0 & 0 & 1 & -1 & \dots \\ 0 & 0 & 0 & -1 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \mathbf{0}_{(v-5) \times (v-5)} \end{pmatrix};$$

Laplaceov polynóm:

$$\mu(G_v^{VI}, x) = \det[x \cdot \mathbf{I}_v - \mathbf{L}(G_v^{VI})] = x^{v-3}(x-3)(x-2)(x-1);$$

Vektor usporiadaných vlastných čísel:

$$\lambda(G_v^{VI}) = (3, 2, 1, \underbrace{0, \dots, 0}'_{v-3}).$$



- Graf  $G_v^N$  s tromi hranami, pričom jedna hrana je susedná s oboma zvyšnými hranami, a zvyšné dve sú vzájomne nesusedné.

Laplacián:

$$\mathbf{L}(G_v^N) = \begin{pmatrix} 1 & -1 & 0 & 0 & \dots \\ -1 & 2 & -1 & 0 & \dots \\ 0 & -1 & 2 & -1 & \dots \\ 0 & 0 & -1 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \mathbf{0}_{(v-4) \times (v-4)} \end{pmatrix};$$

Laplaceov polynóm:

$$\mu(G_v^N, x) = \det[x \cdot \mathbf{I}_v - \mathbf{L}(G_v^N)] = x^{v-3}(x-2-\sqrt{2})(x-2)(x-2+\sqrt{2});$$

Vektor usporiadaných vlastných čísel:

$$\boldsymbol{\lambda}(G_v^N) = (2 + \sqrt{2}, 2, 2 - \sqrt{2}, \underbrace{0, \dots, 0}_{v-3})'.$$

- Graf  $G_v^\Delta$  s tromi, po dvoch susednými hranami.

Laplacián:

$$\mathbf{L}(G_v^\Delta) = \begin{pmatrix} 2 & -1 & -1 & \dots \\ -1 & 2 & -1 & \dots \\ -1 & -1 & 2 & \dots \\ \vdots & \vdots & \vdots & \mathbf{0}_{(v-3) \times (v-3)} \end{pmatrix};$$

Laplaceov polynóm:

$$\mu(G_v^\Delta, x) = \det[x \cdot \mathbf{I}_v - \mathbf{L}(G_v^\Delta)] = x^{v-3}(x-3)^2(x-1);$$

Vektor usporiadaných vlastných čísel:

$$\boldsymbol{\lambda}(G_v^\Delta) = (3, 3, 1, \underbrace{0, \dots, 0}_{v-3})'.$$

- Graf  $G_v^Y$  s tromi hranami, vedúcimi z rovnakého vrcholu.

Laplacián:

$$\mathbf{L}(G_v^Y) = \begin{pmatrix} 3 & -1 & -1 & -1 & \dots \\ -1 & 1 & 0 & 0 & \dots \\ -1 & 0 & 1 & 0 & \dots \\ -1 & 0 & 0 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \mathbf{0}_{(v-4) \times (v-4)} \end{pmatrix};$$

Laplaceov polynóm:

$$\mu(G_v^Y, x) = \det[x \cdot \mathbf{I}_v - \mathbf{L}(G_v^Y)] = x^{v-3}(x-4)(x-1)^2;$$

Vektor usporiadaných vlastných čísel:

$$\boldsymbol{\lambda}(G_v^Y) = (4, 1, 1, \underbrace{0, \dots, 0}_{v-3})'.$$

## Príloha B - Laplaciány, Laplaceove polynómy a vlastné čísla grafov pozostávajúcich z nesusedných hrán

Ak nie sú prvky Laplaciánu vypísané, tak sú nulové.

- Graf  $G_1$  na  $v$  vrcholov obsahujúci iba  $k$  nesusedných hrán.

Kde  $k \in \{0, 1, \dots, \lfloor v/2 \rfloor\}$ .

Laplacián je bloková matica obsahujúca  $k$  blokov tvaru  $\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$ .

Laplacián:

$$\mathbf{L}(G_1) = \begin{pmatrix} 1 & -1 & 0 & 0 & \dots & 0 & 0 & \dots & \\ -1 & 1 & 0 & 0 & \dots & 0 & 0 & \dots & \\ 0 & 0 & 1 & -1 & \dots & 0 & 0 & \dots & \\ 0 & 0 & -1 & 1 & \dots & 0 & 0 & \dots & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \\ 0 & 0 & 0 & 0 & \dots & 1 & -1 & \dots & \\ 0 & 0 & 0 & 0 & \dots & -1 & 1 & \dots & \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \mathbf{0}_{(v-2k) \times (v-2k)} & \end{pmatrix};$$

Laplaceov polynóm:

$$\mu(G_1, x) = \det[x \cdot \mathbf{I}_v - \mathbf{L}(G_1)] = x^{v-k}(x-2)^k;$$

Vektor usporiadaných vlastných čísel:

$$\lambda(G_1) = (\underbrace{2, \dots, 2}_k, \underbrace{0, \dots, 0}_{v-k})'.$$

- Graf  $G_2$  na  $v$  vrcholov, pre  $v$  nepárne, obsahujúci  $(v-1)/2$  nesusedných hrán a navyše ešte jednu hranu pridanú k vrcholu stupňa nula.

Laplacián:

$$\mathbf{L}(G_2) = \begin{pmatrix} 1 & -1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & \dots & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & -1 & 1 \end{pmatrix};$$

Laplaceov polynóm:

$$\mu(G_2, x) = \det[x \cdot \mathbf{I}_v - \mathbf{L}(G_2)] = x^{(v-1)/2}(x-3)(x-2)^{(v-3)/2}(x-1);$$

Vektor usporiadaných vlastných čísel:

$$\lambda(G_2) = (3, \underbrace{2, \dots, 2}_{(v-3)/2}, 1, \underbrace{0, \dots, 0}_{(v-1)/2})'.$$