

Sveučilište J. J. Strossmayera u Osijeku  
Odjel za matematiku  
Sveučilišni diplomski studij financijske  
i poslovne matematike

**Slobodan Jelić**

APROKSIMACIJSKI ALGORITMI ZA OPTIMALNO OSVJETLJENJE  
SCENE

Diplomski rad

Osijek, 2009.

Sveučilište J. J. Strossmayera u Osijeku  
Odjel za matematiku  
Sveučilišni diplomski studij financijske  
i poslovne matematike

**Slobodan Jelić**

# Aproksimacijski algoritmi za optimalno osvjetljenje scene

Diplomski rad

Mentor: **doc. dr. sc. Domagoj Matijević**

Osijek, 2009.

## Zahvale

*Najveću zahvalnost dugujem mentoru doc. dr. sc. Domagoju Matijeviću koji je neumorno i svesrdno pratio moj rad, usmjeravao ideje i pomagao u njihovom ostvarenju. Zahvalnost dugujem i mojim roditeljima koji su mi svojim podrškama omogućili miran i učinkovit rad.*

# Sadržaj

|  |           |
|--|-----------|
| <b>1. Uvod</b>   | <b>5</b>  |
| <b>2. Osnove linearnog programiranja</b>                               | <b>6</b>  |
| 2.1. Osnovni pojmovi . . . . .   | 6         |
| 2.2. Geometrijska reprezentacija dopustivog skupa i rješenja . . . . . | 8         |
| 2.3. Dualnost . . . . .  | 10        |
| 2.4. Aproksimacijski algoritmi . . . . .                               | 14        |
| <b>3. Problem osvjetljenja scene</b>                                   | <b>17</b> |
| 3.1. Fizikalni model . . . . .   | 17        |
| 3.2. Smanjenje broja izvora svjetla . . . . .                          | 18        |
| 3.3. Formulacija problema . . . . .                                    | 20        |
| 3.4. $(1 + \epsilon)$ -aproksimacijska shema . . . . .                 | 21        |
| 3.5. Jednostavni $O(1)$ -aproksimacijski algoritam . . . . .           | 27        |
| 3.5.1. $\alpha$ - reduciranje skupa $G_V$ . . . . .                    | 30        |
| 3.5.2. Određivanje aproksimacijskog omjera . . . . .                   | 33        |
| <b>4. Implementacija aproksimacijskih algoritama</b>                   | <b>37</b> |
| 4.1. Wolframove demonstracije . . . . .                                | 37        |
| 4.2. Opis projekta . . . . .   | 37        |
| 4.2.1. Moduli . . . . .  | 38        |
| 4.2.2. Kontrolno sučelje . . . . .                                     | 43        |
| 4.2.3. Grafičko sučelje . . . . .                                      | 44        |
| <b>A Dodaci</b>  | <b>46</b> |
| A1. Modul <code>LPSol</code> . . . . .                                 | 46        |
| A2. Modul <code>lightsourceimage</code> . . . . .                      | 49        |
| A3. Manipulate okruženje . . . . .                                     | 50        |

# 1. Uvod

*Optimizacija* ili *matematičko programiranje* javlja se u većini područja primijenjene matematike. Pojam optimizacija podrazumijeva minimizaciju ili maksimizaciju realne funkcije pronalaženjem najbolje realne ili cjelobrojne vrijednosti varijable na zadanom dopustivom skupu.

Algoritmi za rješavanje optimizacijskog problema ovise ne samo o tipu minimizirajuće (maksimizirajuće) funkcije, već i o strukturi uvjeta koje svako rješenje mora zadovoljavati. U tom pogledu, prvo mjesto zauzima *linearno programiranje*.

Problem linearnog programiranja javlja se tijekom drugog svjetskog rata u planiranju troškova za opremanje vojne sile. Riječ programiranje u ovom slučaju nema nikakve veze s pisanjem programskog koda u nekom od programskih jezika. Pojam *linearno programiranje* označava *minimizaciju* ili *maksimizaciju linearne funkcije* prema zadanom skupu uvjeta koji se sastoji od *linearnih jednakosti* i/ili *nejednakosti*.

U povijesti razvoja linearnog programiranja posebnu ulogu pripisujemo nekolicini svjetski poznatih matematičara. Ruski matematičar *Kantorovich*<sup>1</sup> prvi put je 1939. godine uveo pojam linearnog programiranja u rješavanju problema optimalne potrošnje resursa. 1947. godine *Dantzig*<sup>2</sup> objavljuje simplex metodu za rješavanje problema linearnog programiranja a iste godine *Neumann*<sup>3</sup> razvija teoriju dualnosti. 1979. godine *Khachiyan*<sup>4</sup> objavljuje dokaz o rješivosti problema linearnog programiranja u polinomijalnom vremenu, dok je širu teoretsku i praktičnu pozadinu dao *Karmarkar*<sup>5</sup> 1984. godine uvodeći *metodu unutarnje točke*.

U prvom djelu rada definirani su osnovni pojmovi teorije linearnog programiranja. Na primjeru optimalne potrošnje energije u proizvodnji uveden je problem linearnog programiranja u općenitom obliku, opisana je geometrijska predodžba dopustivog skupa i optimalnog rješenja. Na istom primjeru uveden je dualni problem i osnovni rezultati teorije dualnosti. Kako je osnovni zadatak rada prezentirati aproksimacijske algoritme za specifični problem, definirali smo aproksimacijski omjer za minimizacijske i maksimizacijske probleme.

U drugom djelu rada objašnjen je fizikalni model širenja svjetla iz točkastog izvora što je jedna od osnovnih pretpostavki problema optimalnog osvjetljenja scene. Originalni problem formuliran je kao problem konveksnog programiranja. U radu su analizirani  $(1 + \epsilon)$ -aproksimacijska shema i 4-aproksimacijski algoritam koji se temelji na LP formulaciji. Opisan je i kombinatorijalni  $O(1)$ -aproksimacijski algoritam čija je analiza provedena primal-dual fitovanjem.

U trećem dijelu rada opisana je implementacija  $(1 + \epsilon)$ -aproksimacijske sheme i 4-aproksimacijskog algoritma u projektu *Wolframove demonstracije* pod nazivom *Stage Illumination Project*. Komentirani su osnovni dijelovi koda. Ilustrirane su prednosti dinamičkog okruženja *Manipulate* i načini interakcije korisnika s programom.

---

<sup>1</sup>Леонид Витальевич Канторович (1912.-1986.), ruski matematičar i ekonomist

<sup>2</sup>George Bernard Dantzig (1914.-2005.), američki matematičar

<sup>3</sup>John von Neumann (mađarski: margittai Neumann János Lajos) (1903.-1957.) mađarski i američki matematičar

<sup>4</sup>Леонид Генрихович Хачиян (1952.-2005.), ruski matematičar armenskog podrijetla

<sup>5</sup>Narendra K. Karmarkar (1957. -), indijski matematičar

## 2. Osnove linearnog programiranja

### 2.1. Osnovni pojmovi

Prije općenite definicije problema linearnog programiranja<sup>6</sup> promotrimo sljedeći minimizacijski problem.

**Primjer 2.1** (Problem proizvodnje). *Pretpostavimo da tvornica računalnih komponenti raspolaže s 2 proizvodna procesa koji proizvode matične ploče te grafičke i zvučne kartice. Svaki od procesa troši određenu razinu energije u proizvodnji navedenih komponenti. Neka je  $x_1$  ukupna potrošnja energije prvog procesa, a  $x_2$  ukupna potrošnja energije drugog procesa. Jedinična potrošnja energije na prvom procesu košta 500 kn, pri čemu se proizvede 1 matična ploča, 3 grafičke kartice i 3 zvučne kartice. Jedinična potrošnja energije na drugom procesu košta 800 kn, pri čemu se proizvede 2 matična ploča, 1 grafička kartica i 3 zvučne kartice. Treba odrediti optimalnu razinu potrošnje energije pri kojoj se proizvede najmanje 20 matičnih ploča, najmanje 40 grafičkih kartica i najmanje 50 zvučnih kartica.*

Ukupna cijena potrošene energije dana je izrazom  $500x_1 + 800x_2$ . Želimo odrediti  $x_1$  i  $x_2$  tako da su zadovoljeni svi uvjeti na količinu računalnih komponenti a da je ukupna cijena potrošene energije **minimalna**. Pretpostavljamo da je  $x_1, x_2 \geq 0$ . Specijalno, ako je  $x_i = 0$  za  $i = 1, 2$ , smatramo da je  $i$ -ti proces deaktiviran.

Koristeći oba procesa možemo proizvesti ukupno  $x_1 + 2x_2$  matičnih ploča,  $3x_1 + x_2$  grafičkih kartica i  $3x_1 + 3x_2$  zvučnih kartica. Očito su uvjeti proizvodnje iskazani sljedećim skupom nejednakosti

$$\begin{aligned}x_1 + 2x_2 &\geq 20, \\3x_1 + x_2 &\geq 30, \\3x_1 + 3x_2 &\geq 40,\end{aligned}$$

a naš problem možemo iskazati na sljedeći način

$$\begin{aligned}\min \quad & 500x_1 + 800x_2 \\ \text{uz uvjete:} \quad & \\ & x_1 + 2x_2 \geq 20, \\ & 3x_1 + x_2 \geq 30, \\ & 3x_1 + 3x_2 \geq 40, \\ & x_1, x_2 \geq 0.\end{aligned} \tag{2.1}$$

Vidimo da su  $x_1$  i  $x_2$  varijable čije vrijednosti treba odabrati tako da linearna funkcija  $500x_1 + 800x_2$  poprimi minimalnu vrijednost. Općenito, neka je  $\mathbf{x} \in \mathbb{R}^n$  vektor varijabli  $x_1, x_2, \dots, x_n$ . Uvjeti mogu biti zadani kao linearne jednakosti ili nejednakosti oblika  $\mathbf{a}_i^T \mathbf{x} \geq b_i$ ,  $\mathbf{a}_i^T \mathbf{x} = b_i$  i  $\mathbf{a}_i^T \mathbf{x} \leq b_i$ , gdje su  $\mathbf{a}_i \in \mathbb{R}^n$  za  $1 \leq i \leq m$  i  $\mathbf{b} = [b_1 \ b_2 \ \dots \ b_m]^T \in \mathbb{R}^m$ .

Sada ćemo generalizirati prethodni LP problem. Za dani vektor  $\mathbf{c} = [c_1 \ c_2 \ \dots \ c_n]^T \in \mathbb{R}^n$  i vektor varijabli  $\mathbf{x} \in \mathbb{R}^n$ , linearnu funkciju  $\mathbf{c}^T \mathbf{x} = \sum_{i=1}^n c_i x_i$  zovemo **funkcija cilja**. U Primjeru 2.1 vidimo da je  $\mathbf{c} = [500 \ 800]^T$  i  $\mathbf{x} = [x_1 \ x_2]^T$ .

---

<sup>6</sup>dalje u tekstu PLP

Pretpostavimo da je u PLP-u zadano ukupno  $m$  uvjeta. Neka su  $M_1$ ,  $M_2$  i  $M_3$  podskupovi skupa indeksa  $\{1, 2, \dots, m\}$  takvi da indeksima  $i$  u skupu  $M_1$  odgovaraju uvjeti oblika  $\mathbf{a}_i^T \mathbf{x} \geq b_i$ , indeksima  $i$  u skupu  $M_2$  odgovaraju uvjeti oblika  $\mathbf{a}_i^T \mathbf{x} \leq b_i$ , a indeksima  $i$  u skupu  $M_3$  odgovaraju uvjeti oblika  $\mathbf{a}_i^T \mathbf{x} = b_i$ . Neka su  $N_1$  i  $N_2$  podskupovi skupa indeksa  $\{1, 2, \dots, n\}$  takvi da indeksima  $i$  u skupu  $N_1$  odgovaraju uvjeti nenegativnosti  $x_i \geq 0$ , a indeksima  $i$  u skupu  $N_2$  odgovaraju uvjeti nepozitivnosti  $x_i \leq 0$ . Tada PLP problem možemo prikazati na sljedeći način

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} \\ \text{uz uvjete} \quad & \mathbf{a}_i^T \mathbf{x} \geq b_i, \quad i \in M_1, \\ & \mathbf{a}_i^T \mathbf{x} \leq b_i, \quad i \in M_2, \\ & \mathbf{a}_i^T \mathbf{x} = b_i, \quad i \in M_3, \\ & x_j \geq 0, \quad j \in N_1, \\ & x_j \leq 0, \quad j \in N_2. \end{aligned} \tag{2.2}$$

Vektor  $\mathbf{x} \in \mathbb{R}^n$  koji zadovoljava sve uvjete u (2.2) naziva se **dopustivo rješenje**<sup>7</sup>. Skup svih dopustivih rješenja nazivamo **dopustivi skup**. Varijable  $x_j$  koje nemaju ograničenje na predznak nazivamo **slobodne varijable**. Dopustivo rješenje  $\mathbf{x}^*$  za koje funkcija cilja  $\mathbf{c}^T \mathbf{x}$  poprima minimalnu (maksimalnu) vrijednost zovemo **optimalno dopustivo rješenje** ili samo **optimalno rješenje**. Vrijednost  $\mathbf{c}^T \mathbf{x}^*$  zovemo **optimalna vrijednost funkcije cilja**.

Ako za svaki  $K \in \mathbb{R}$  postoji dopustivo rješenje  $\mathbf{x}$  takvo da je  $\mathbf{c}^T \mathbf{x} < K$  [ $\mathbf{c}^T \mathbf{x} > K$ ] kažemo da je PLP **neograničen odozdo** [odozgo]. Tada je optimalna vrijednost funkcije cilja  $-\infty$  [ $+\infty$ ].

Zbog činjenice da je problem maksimizacije funkcije  $\mathbf{c}^T \mathbf{x}$  ekvivalentan problemu minimizacije funkcije  $-\mathbf{c}^T \mathbf{x}$  i obrnuto, nema potrebe posebno razmatrati maksimizacijski problem.

Uvjet oblika  $\mathbf{a}_i^T \mathbf{x} = b_i$  ekvivalentan je uvjetima  $\mathbf{a}_i^T \mathbf{x} \leq b_i$  i  $\mathbf{a}_i^T \mathbf{x} \geq b_i$ . Nadalje, svaki uvjet oblika  $\mathbf{a}_i^T \mathbf{x} \leq b_i$  može se zamijeniti uvjetom  $-\mathbf{a}_i^T \mathbf{x} \geq -b_i$ . Svaki uvjet oblika  $x_j \leq 0$  ili  $x_j \geq 0$  može se prikazati u obliku  $\mathbf{a}_i^T \mathbf{x} \geq b_i$  gdje je  $\mathbf{a}_i$   $i$ -ti jedinični vektor i  $b_i = 0$ . Drugim riječima, svi uvjeti u (2.2) mogu se prikazati u obliku  $\mathbf{a}_i^T \mathbf{x} \geq b_i$ . Neka je  $\mathbf{A} \in \mathbb{R}^{m \times n}$  matrica kojoj su  $\mathbf{a}_1^T, \mathbf{a}_2^T, \dots, \mathbf{a}_m^T \in \mathbb{R}^n$  vektori redci, tj.

$$\mathbf{A} = \begin{bmatrix} \text{---} & \mathbf{a}_1^T & \text{---} \\ & \vdots & \\ \text{---} & \mathbf{a}_m^T & \text{---} \end{bmatrix}.$$

Sada uvjete  $\mathbf{a}_i^T \mathbf{x} \geq b_i$  za  $i = 1, 2, \dots, m$  možemo kraće pisati kao  $\mathbf{A}\mathbf{x} \geq \mathbf{b}$ , a PLP u (2.2) možemo napisati na sljedeći način

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} \\ \text{uz uvjete} \quad & \mathbf{A}\mathbf{x} \geq \mathbf{b}. \end{aligned} \tag{2.3}$$

---

<sup>7</sup>eng. feasible solution

**Primjer 2.2.** PLP problem iz Primjera 2.1 možemo zapisati kao u (2.3) gdje je  $\mathbf{c} = [500 \ 800]^T$ ,

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 3 & 3 \\ 1 & 0 \\ 0 & 1 \end{bmatrix},$$

i  $\mathbf{b} = [20 \ 30 \ 40]^T$ .

Skup uvjeta može sadržavati samo jednakosti i uvjet nenegativnosti. PLP problem koji ima sljedeći oblik

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} \\ \text{uz uvjete} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

zovemo **LP u standardnom obliku**.

Ovaj oblik PLP problema možemo tumačiti na sljedeći način. Pretpostavimo da je  $\mathbf{x} \in \mathbb{R}^n$  i da su  $\mathbf{A}_1, \dots, \mathbf{A}_n$  stupci matrice  $\mathbf{A}$ . Tada uvjet  $\mathbf{A}\mathbf{x} = \mathbf{b}$  možemo pisati na sljedeći način

$$\sum_{i=1}^n x_i \mathbf{A}_i = \mathbf{b}.$$

Uzmimo da komponente  $b_i$  vektora  $\mathbf{b}$  predstavljaju željene količine  $i$ -tog proizvoda koji se može proizvesti od  $n$  različitih resursa. Komponente  $a_{kj}$  vektora  $\mathbf{A}_j$  za  $1 \leq j \leq n$  i  $1 \leq k \leq m$  govore koliko jedna jedinica  $j$ -tog resursa daje jedinica  $k$ -tog proizvoda. Treba odrediti nenegativne količine  $x_i$  svakog od  $n$  resursa tako da je suma  $\sum_{j=1}^n c_j x_j$  minimalna, gdje  $c_j$  predstavlja cijenu jedinične količine  $j$ -tog resursa.

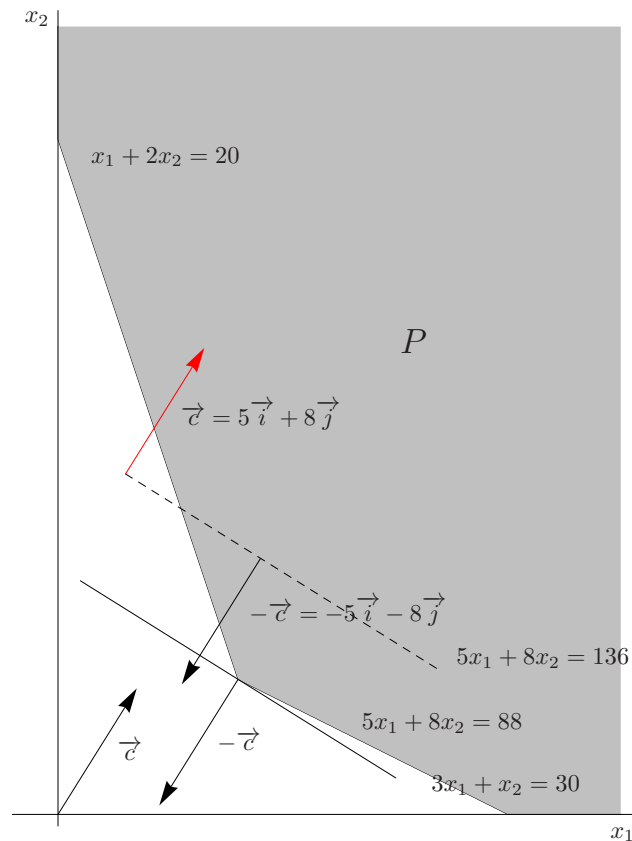
## 2.2. Geometrijska reprezentacija dopustivog skupa i rješenja

U Primjeru 2.1 funkcija cilja ima dvije varijable  $x_1$  i  $x_2$  što nam omogućava grafički prikaz dopustivog skupa. Bez smanjenja općenitosti pretpostavimo da funkcija cilja ima sljedeći oblik

$$5x_1 + 8x_2$$

Dopustivi skup rješenja je osjenčan poliedar na Slici 2.1. U cilju pronalaženja optimalnog rješenja promatrat ćemo sve točke  $(x_1, x_2)$  u kojima funkcija cilja ima vrijednost  $K \in \mathbb{R}$ . To su točke koje pripadaju dopustivom skupu i leže na pravcu  $5x_1 + 8x_2 = K$ . Uočimo da je taj pravac okomit na vektor  $\mathbf{c} = [5 \ 8]^T$ . Svakom od paralelnih pravaca odgovara određena konstanta  $K$ . Paralelnim pomakom u smjeru vektora  $\mathbf{c}$  odgovara povećanje  $K$  (problem maksimizacije funkcije cilja), a paralelnom pomaku u smjeru suprotnom od vektora  $\mathbf{c}$  (tj.

u smjeru vektora  $-\mathbf{c}$ ) odgovara smanjenje  $K$  (problem minimizacije funkcije cilja). Budući da se u našem slučaju radi o minimizaciji funkcije cilja, nastojat ćemo pomaknuti pravac u smjeru vektora  $-\mathbf{c}$  što je više moguće ali tako da ostanemo u dopustivom skupu. Najbolje što možemo učiniti jeste slučaj  $K = 88$  kada je  $\mathbf{x} = [8 \ 6]^T$  (vidi Sliku 2.1).



Slika 2.1: Dopustivi skup iz Primjera 2.1

Sličan pristup možemo napraviti i u trodimenzionalnom slučaju. Skup točaka u kojima funkcija cilja ima istu vrijednost  $K$  predstavljen je ravninom koja je okomita na vektor  $\mathbf{c}$ . Kako bismo minimizirali funkciju cilja, pomaknut ćemo ravninu što je više moguće u smjeru vektora  $-\mathbf{c}$  tako da ostanemo u dopustivom skupu.

Vidimo da je optimalno rješenje *vrh poliedra*. U dvodimenzionalnom i trodimenzionalnom slučaju pojam vrha ima geometrijsko značenje, ali moguće je definirati vrh poliedra u općenitom slučaju (vidi [4]). Ukoliko je optimizacijski problem ograničen, optimalno rješenje je dovoljno tražiti među vrhovima poliedra (vidi [16], [4]). Na ovoj pretpostavci počiva *simplex metoda*, jedna od najčešće korištenih metoda za rješavanje problema linearnog programiranja.

## 2.3. Dualnost

U narednom poglavlju uvodimo pojam dualnosti. Svakom PLP problemu, kojeg ćemo od sada zvati *primal*, pridružujemo njegov dualni problem kojeg ćemo zvati *dual*. U okviru ovog rada najvažnije je razumijevanje dualnosti i odnosa između primala i duala, kao i poznavanje osnovnih rezultata teorije dualnosti.

Motivacija za uvođenje dualnog problema dolazi od metode *Lagrangeovih multiplikatora* za uvjetovanu optimizaciju diferencijabilne funkcije (vidi [10], str. 374). Analizirajmo sljedeći jednostavan primjer.

**Primjer 2.3.** *Odredimo minimum sljedeće funkcije*

$$\begin{aligned} \min \quad & x^2 + 2y^2 \\ \text{uz uvjet} \quad & x + y = 2. \end{aligned} \tag{2.4}$$

Problem minimizacije funkcije  $f(x, y) = x^2 + 2y^2$  možemo riješiti uvođenjem *Lagrangeove funkcije*  $L(x, y, p)$  na sljedeći način

$$L(x, y, p) = x^2 + 2y^2 - p(2 - x - y). \tag{2.5}$$

Iz (2.5) vidimo da funkciju  $L$  dobivamo relaksacijom uvjeta. Uočimo da funkcije  $L$  i  $f$  imaju iste vrijednosti u točkama koje leže na pravcu  $x + y = 2$ . Iz toga je jasno da možemo tražiti minimum funkcije  $L$  koji zadovoljava uvjet  $x + y = 2$ . Rješavanjem jednadžbi iz nužnih uvjeta  $\frac{\partial L}{\partial x}(x, y) = 0$  i  $\frac{\partial L}{\partial y}(x, y) = 0$  (vidi [10], str. 75) dobivamo da je  $x = \frac{p}{2}$  i  $y = \frac{p}{4}$ . Zbog  $x + y = 2$  imamo da je  $p = \frac{8}{3}$ , odnosno  $(x^*, y^*) = (\frac{4}{3}, \frac{2}{3})$ .

Prethodnim primjerom ilustrirana je sljedeća ideja. Umjesto da smo zaista tražili zadovoljenje uvjeta  $x + y = 2$ , dozvolili smo odstupanje. Lagrangeov multiplikator  $p$  možemo tumačiti kao *cijenu* jediničnog odstupanja od vrijednosti funkcije  $f(x, y) = x^2 + 2y^2$ . U našem slučaju odstupanje iznosi  $2 - x - y$  što nas dovodi do neuvjetovane minimizacije funkcije  $x^2 + 2y^2 + p(2 - x - y)$ . U gornjem primjeru smo vidjeli da je moguće odabrati određenu cijenu  $p$  (u našem slučaju  $p = \frac{8}{3}$ ) tako da je optimalno rješenje neuvjetovanog problema ujedno i optimalno rješenje uvjetovanog problema. Drugim riječima, *za određenu cijenu  $p$  zadovoljavanje ili odsustvo uvjeta ne utječe na optimalnu cijenu*.

Koristeći prethodnu ideju rješavanja uvjetovanog optimizacijskog problema razvijamo mnogo općenitiju klasu problema kojom se bavi *teorija dualnosti*. Svakom uvjetu pridružujemo cijenu njegovog zadovoljenja ili narušavanja i pitamo se za koje cijene  $\mathbf{p}$  zadovoljavanje ili odstupanje od uvjeta ne utječe na optimalnu vrijednost funkcije cilja. Cijene  $\mathbf{p}$  dobivamo rješavanjem novog LP problema koji se naziva *dualni LP*<sup>8</sup>.

Sada ćemo odrediti DLP za problem opisan u Primjeru 2.1. Objasniti ćemo najprije značenje dualnih varijabli i novih uvjeta, a zatim pristupiti formulaciji općenitog DLP problema. Ovdje će biti razmatran samo PLP sa uvjetima  $\mathbf{Ax} \geq \mathbf{b}$  i njemu odgovarajući DLP problem.

---

<sup>8</sup>dalje u tekstu DLP

Pretpostavimo da je  $(x_1^*, x_2^*)^T$  optimalno rješenje iz Primjera 2.1. Kao u prethodnim razmatranjima želimo relaksirati problem na sljedeći način

$$\begin{aligned} \min \quad & 500x_1 + 800x_2 + p_1(20 - x_1 - 2x_2) + p_2(30 - 3x_1 - x_2) + p_3(40 - 3x_1 - 3x_2) \\ \text{uz uvjete:} \quad & x_1, x_2 \geq 0. \end{aligned} \tag{2.6}$$

Izrazi  $20 - x_1 - 2x_2$ ,  $30 - 3x_1 - x_2$  i  $40 - 3x_1 - 3x_2$  označavaju razliku u proizvedenoj i minimalno potraživanoj količini matičnih ploča te grafičkih i zvučnih kartica. Svaku od razlika plaćamo po cijenama  $p_1$ ,  $p_2$  i  $p_3$ , redom. I nakon relaksacije želimo minimizirati ukupnu cijenu potrošene energije uvećanu za cijenu količinskih razlika u proizvodnji. Međutim, želimo cijene  $p_1$ ,  $p_2$  i  $p_3$  takve da optimalna vrijednost minimizirajućeg izraza u (2.6) ne bude veća od optimalne cijene energije  $500x_1^* + 800x_2^*$  prije relaksacije. Takav pristup ima i svoje ekonomsko značenje. Odredit ćemo cijene proizvoda na način da novi scenarij proizvodnje (koji možda i nije dopustiv) ne utječe na optimalnu cijenu potrošene energije.

Uzmimo da funkcija  $g(p_1, p_2, p_3)$  predstavlja optimalnu vrijednost funkcije cilja u (2.6). Sada imamo da je

$$\begin{aligned} g(p_1, p_2, p_3) &= \min_{x_1, x_2 \geq 0} 500x_1 + 800x_2 + p_1(20 - x_1 - 2x_2) \\ &\quad + p_2(30 - 3x_1 - x_2) + p_3(40 - 3x_1 - 3x_2) \\ &\leq 500x_1^* + 800x_2^* + p_1(20 - x_1^* - 2x_2^*) \\ &\quad + p_2(30 - 3x_1^* - x_2^*) + p_3(40 - 3x_1^* - 3x_2^*) \\ &\leq 500x_1^* + 800x_2^*. \end{aligned} \tag{2.7}$$

Prva nejednakost u (2.7) slijedi iz zahtjeva da ukupna cijena novog scenarija proizvodnje ne bude skuplja od starog scenarija koji ima uvjete na količinu proizvoda. Druga nejednakost slijedi iz činjenice da je optimalna razina energije  $(x_1^*, x_2^*)$  dopustiva u starom scenariju (primalnom LP-u) te da su količinske razlike nepozitivne (u početnom scenariju proizvodimo najmanje 20 matičnih ploča, 30 grafičkih i 40 zvučnih kartica). Da bi posljednja nejednakost bila istinita potrebno je zahtijevati da cijene  $p_1$ ,  $p_2$  i  $p_3$  budu nenegativne. Iz prethodnih razmatranja vidimo da je vrijednost funkcije  $g$  za  $p_1, p_2, p_3 \geq 0$  donja međa optimalne vrijednosti funkcije cilja u primalnom LP-u. DLP problem je određivanje cijena  $p_1$ ,  $p_2$  i  $p_3$  tako da donja međa (odnosno vrijednost funkcije  $g$ ) bude maksimalna. Drugim riječima, pitamo se da li je moguće odrediti cijene  $p_1$ ,  $p_2$  i  $p_3$  tako da nova cijena čak i minimalno zahtijevane proizvedene količine bude što bliža cijeni potrošene energije u prvom scenariju. Cijene  $p_1$ ,  $p_2$  i  $p_3$  dobivamo rješavanjem sljedećeg DLP problema

$$\begin{aligned} \max \quad & g(p_1, p_2, p_3) \\ \text{uz uvjete} \quad & p_1, p_2, p_3 \geq 0. \end{aligned}$$

Transformiranjem funkcije  $g$  dobivamo

$$\begin{aligned}
g(p_1, p_2, p_3) &= \min_{x_1, x_2 \geq 0} 500x_1 + 800x_2 + p_1(20 - x_1 - 2x_2) \\
&\quad + p_2(30 - 3x_1 - x_2) + p_3(40 - 3x_1 - 3x_2) \\
&= 20p_1 + 30p_2 + 40p_3 \\
&\quad + \min_{x_1, x_2 \geq 0} \{(500 - p_1 - 3p_2 - 3p_3)x_1 + (800 - 2p_1 - p_2 - 3p_3)x_2\}.
\end{aligned} \tag{2.8}$$

Minimum posljednjeg izraza u (2.8) je konačan samo kad su razlike  $500 - p_1 - 3p_2 - 3p_3$  i  $800 - 2p_1 - p_2 - 3p_3$  nenegativne, tj. kada cijena svih proizvoda koji su dobiveni jediničnom potrošnjom energije u jednom od proizvodnih procesa ne premaši cijenu jedne jedinice energije u tom procesu. Konačno, dolazimo do DLP problema

$$\begin{aligned}
&\max \quad 20p_1 + 30p_2 + 40p_3 \\
&\text{uz uvjet} \quad p_1 + 3p_2 + 3p_3 \leq 500, \\
&\quad \quad \quad 2p_1 + p_2 + 3p_3 \leq 800, \\
&\quad \quad \quad p_1, p_2, p_3 \geq 0.
\end{aligned}$$

Osnovni teorem u teoriji dualnosti tvrdi da je optimalna vrijednost funkcije cilja u PLP-u jednaka optimalnoj vrijednosti funkcije cilja u DLP-u, tj. možemo odrediti cijene  $p_1^*$ ,  $p_2^*$  i  $p_3^*$  tako da početna relaksacija ne utječe na optimalnu vrijednost funkcije cilja u PLP-u. Sada smo uvjereni da su  $p_1$ ,  $p_2$  i  $p_3$  cijene proizvoda koje čak i na minimalno zahtijevanoj razini proizvodnje omogućuju da se "financijski pokrije" optimalno potrošena energija (razmatrana u početnom PLP problemu).

Prethodni postupak možemo ponoviti i na općenitom problemu.

$$\begin{aligned}
&\min \quad \mathbf{c}^T \mathbf{x} \\
&\text{uz uvjete} \quad \mathbf{Ax} \geq \mathbf{b}, \\
&\quad \quad \quad \mathbf{x} \geq \mathbf{0},
\end{aligned}$$

koji ima optimalno rješenje  $\mathbf{x}^*$ . Uvodimo relaksaciju tako da izostavimo uvjet  $\mathbf{Ax} \geq \mathbf{b}$  a u funkciju cilja dodamo izraz  $\mathbf{p}^T(\mathbf{b} - \mathbf{Ax})$ , gdje je  $\mathbf{p}$  vektor cijena iste dimenzije kao i  $\mathbf{b}$ . Sada imamo sljedeći relaksirani problem

$$\begin{aligned}
&\min \quad \mathbf{c}^T \mathbf{x} + \mathbf{p}^T(\mathbf{b} - \mathbf{Ax}) \\
&\text{uz uvjete} \quad \mathbf{x} \geq \mathbf{0}.
\end{aligned}$$

Pretpostavimo da je  $g(\mathbf{p})$  funkcija koja predstavlja optimalnu vrijednost funkcije cilja u relaksiranom problemu u ovisnosti o vektoru cijena  $\mathbf{p}$ . Budući da smo se riješili uvjeta, relaksirani problem dopušta zahtjev da je  $g(\mathbf{p})$  bude uvijek manja ili jednaka optimalnoj vrijednosti funkcije cilja  $\mathbf{c}^T \mathbf{x}^*$  u PLP problemu, odnosno

$$g(\mathbf{p}) = \min_{\mathbf{x} \geq \mathbf{0}} \mathbf{c}^T \mathbf{x} + \mathbf{p}^T(\mathbf{b} - \mathbf{Ax}) \leq \mathbf{c}^T \mathbf{x}^* + \mathbf{p}^T(\mathbf{b} - \mathbf{Ax}^*) \leq \mathbf{c}^T \mathbf{x}^*,$$

gdje posljednja nejednakost slijedi iz činjenice da je  $\mathbf{x}^*$  dopustivo rješenje PLP problema, ali uz dodatni uvjet nenegativnosti  $\mathbf{p} \geq \mathbf{0}$ . Dakle, svaki dopustivi  $\mathbf{p}$  daje donju među  $g(\mathbf{p})$  za optimalnu vrijednost funkcije cilja  $\mathbf{c}^T \mathbf{x}^*$  u PLP problemu. Postavimo sljedeći problem

$$\begin{aligned} & \max g(\mathbf{p}) \\ & \text{uz uvjete } \mathbf{p} \geq \mathbf{0}, \end{aligned}$$

koji možemo tumačiti kao pronalaženje najbolje donje međe na optimalnu vrijednost funkcije cilja u PLP problemu. Taj problem se naziva **dualni problem**. Osnovni rezultat teorije dualnosti tvrdi da je optimalna vrijednost funkcije cilja dualnog problema jednaka optimalnoj vrijednosti funkcije cilja primalnog problema. Drugim riječima, kada je vektor cijena  $p$  izabran na odgovarajući način, relaksacija uvjeta ne utječe na optimalnu vrijednost funkcije cilja u PLP problemu.

Koristeći se definicijom funkcije  $g(\mathbf{p})$ , dobivamo da je

$$\begin{aligned} g(\mathbf{p}) &= \min_{\mathbf{x} \geq \mathbf{0}} \mathbf{c}^T \mathbf{x} + \mathbf{p}^T (\mathbf{b} - \mathbf{A} \mathbf{x}) \\ &= \mathbf{p}^T \mathbf{b} + \min_{\mathbf{x} \geq \mathbf{0}} (\mathbf{c}^T - \mathbf{p}^T \mathbf{A}) \mathbf{x}. \end{aligned} \quad (2.9)$$

Uočimo da je

$$\min_{\mathbf{x} \geq \mathbf{0}} (\mathbf{c}^T - \mathbf{p}^T \mathbf{A}) \mathbf{x} = \begin{cases} 0 & \text{ako je } \mathbf{c}^T - \mathbf{p}^T \mathbf{A} \geq \mathbf{0}^T, \\ -\infty & \text{inače.} \end{cases} \quad (2.10)$$

Prilikom maksimiziranja funkcije  $g(\mathbf{p})$  treba uzimati u obzir samo one vektore  $\mathbf{p}$  za koje je  $g(\mathbf{p})$  različito od  $-\infty$ . Iz (2.9) i (2.10) zaključujemo da je DLP problem ekvivalentan sljedećem LP problemu

$$\begin{aligned} & \min \mathbf{p}^T \mathbf{b} \\ & \text{uz uvjete } \mathbf{p}^T \mathbf{A} \leq \mathbf{c}^T, \\ & \mathbf{p} \geq \mathbf{0}. \end{aligned}$$

Na sličan način možemo formulirati DLP probleme i za ostale oblike PLP problema (vidi [4]). Neka je  $\mathbf{A} \in \mathbb{R}^{m \times n}$  matrica sa redcima  $\mathbf{a}_i^T \in \mathbb{R}^{1 \times n}$  i stupcima  $\mathbf{A}_j^T \in \mathbb{R}^{m \times 1}$ . Za PLP problem sa uvjetima na lijevoj strani dan je ekvivalentan DLP problem sa odgovarajućim uvjetima na desnoj strani.

$$\begin{array}{ll} \min \mathbf{c}^T \mathbf{x} & \max \mathbf{p}^T \mathbf{b} \\ \text{uz uvjete } \mathbf{a}_i^T \mathbf{x} \geq b_i, & i \in M_1, \quad \text{uz uvjete } p_i \geq 0, \quad i \in M_1, \\ \mathbf{a}_i^T \mathbf{x} \leq b_i, & i \in M_2, \quad p_i \leq 0, \quad i \in M_2, \\ \mathbf{a}_i^T \mathbf{x} = b_i, & i \in M_3, \quad p_i \text{ slobodna, } i \in M_3, \\ x_j \geq 0, & j \in N_1, \quad \mathbf{p}^T \mathbf{A}_j \leq c_j, \quad j \in N_1, \\ x_j \leq 0, & j \in N_2, \quad \mathbf{p}^T \mathbf{A}_j \geq c_j, \quad j \in N_2, \\ x_j \text{ slobodna, } & j \in N_3, \quad \mathbf{p}^T \mathbf{A} = c_j, \quad j \in N_3. \end{array}$$

Uočili smo da minimizacijski PLP problem uvijek ima maksimizacijski DLP problem. Vrijedi i obrnuto, tj. svaki maksimizacijski PLP problem ima minimizacijski DLP problem. Lako se može pokazati da transformacijom DLP problema u ekvivalentni minimizacijski problem i formulacijom njemu odgovarajućeg DLP problema dolazimo do početnog PLP problema. Na osnovu početnih razmatranja vrlo jednostavno dolazimo do teorema o slaboj dualnosti i njegovih korolara.

**Teorem 2.1** (Teorem o slaboj dualnosti). *Ako je  $\mathbf{x}$  dopustivo rješenje u minimizacijskom PLP problemu i  $\mathbf{p}$  dopustivo rješenje u njemu odgovarajućem DLP problemu, onda je*

$$\mathbf{p}^T \mathbf{b} \leq \mathbf{c}^T \mathbf{x}.$$

**Dokaz.** Vidi [4], str. 146. □

Posljedica teorema o slaboj dualnosti je sljedeći korolar o slučajevima kada je optimalna vrijednost funkcije cilja u PLP i DLP problemu  $-\infty$  i  $+\infty$ , redom.

**Korolar 2.1.** *a) Ako je optimalna vrijednost funkcije cilja u PLP problemu  $-\infty$ , onda je DLP problem nedopustiv (skup dualnih dopustivih rješenja je prazan).*

*b) Ako je optimalna vrijednost funkcije cilja u DLP problemu  $+\infty$ , onda je PLP problem nedopustiv.*

Druga važna posljedica teorema o slaboj dualnosti je sljedeći korolar.

**Korolar 2.2.** *Neka su  $\mathbf{x}$  i  $\mathbf{p}$  redom dopustiva rješenja u PLP i DLP problemu, takva da je  $\mathbf{p}^T \mathbf{b} = \mathbf{c}^T \mathbf{x}$ . Tada su  $\mathbf{x}$  i  $\mathbf{p}$  redom optimalna rješenja u PLP i DLP problemu.*

Sljedeći teorem predstavlja osnovni rezultat u teoriji dualnosti za linearnu optimizaciju.

**Teorem 2.2** (Teorem o jakoj dualnosti). *Ako PLP problem ima optimalno rješenje, ima ga i njemu odgovarajući DLP problem, a optimalne vrijednosti funkcija cilja u oba problema su međusobno jednake.*

**Dokaz.** Vidi [4], str. 148. □

## 2.4. Aproksimacijski algoritmi

Prije opisa i definicije aproksimacijskih algoritama potrebno je objasniti motivaciju njihova nastanka.

Tijekom razvoja računalnih znanosti posebna pažnja posvećena je **polinomijalnim algoritmima**. To su algoritmi koji se na ulazu (inputu) veličine  $n$  izvršavaju u  $O(n^k)$  vremenu gdje je  $k \in \mathbb{N}$ . Prirodno je pitati se da li su svi problemi rješivi u polinomijalnom vremenu. To je jedno od najvećih otvorenih pitanja u teoriji algoritama. Zbog toga je bilo neophodno klasificirati probleme s obzirom na mogućnosti njihova rješavanja u polinomijalnom vremenu. Nadalje ćemo razmatrati 4 klase problema:

- **Polinomijalni problemi (P-problemi)** - problemi koji su rješivi u polinomijalnom vremenu.
- **Nepolinomijalni problemi (NP-problemi)** - nisu poznati algoritmi koji rješavaju ove probleme u polinomijalnom vremenu. Za njih se u polinomijalnom vremenu može **samo** provjeriti da li je određena instanca problema ujedno i njegovo rješenje. Npr. za graf  $G = (V, E)$  u polinomijalnom vremenu možemo provjeriti da li je niz od  $|V|$  vrhova  $v_1, v_2, \dots, v_{|V|}$  Hamiltonov ciklus, tj. da li je  $(v_i, v_{i+1}) \in E$ , za  $1 \leq i \leq |V| - 1$  i  $(v_{|V|}, v_1) \in E$ .
- **Potpuno nepolinomijalni problemi (NPC-problemi)** - klasa NP-problema za koje se zna da su međusobno "jednako teški". Ako se bilo koji NPC-problem može riješiti u polinomijalnom vremenu onda za svaki NP problem postoji polinomijalni algoritam, tj. vrijedi da je  $P = NP$ . Teoretičari računalnih znanosti smatraju da je  $P \neq NP$  jer do sada nijedan polinomijalni algoritam nije konstruiran za neki NPC problem.
- **Nepolinomijalno teški problemi (NP-teški problemi)** - problemi za koje nismo sigurni da su u NP klasi, tj. ne možemo u polinomijalnom vremenu provjeriti da li je određena instanca problema ujedno i njegovo rješenje.

Jedan od osnovnih pristupa u rješavanju NPC-problema je konstruiranje aproksimacijskih algoritama za približna rješenja koja su se u praksi pokazala vrlo korisnim. Od sada pa nadalje prilikom razmatranja aproksimacijskih algoritama uzimat ćemo u obzir samo optimizacijske probleme čije funkcije cilja poprimaju isključivo pozitivne vrijednosti.

**Definicija 2.1.** *Neka je  $\Pi$  optimizacijski problem s optimalnom vrijednošću funkcije cilja  $C^*$  te neka postoji algoritam koji na ulaznim podacima veličine  $n$  daje rješenje  $\mathbf{x}$  u kojem funkcija cilja ima vrijednost  $C$  sa svojstvom da je*

$$\max \left\{ \frac{C}{C^*}, \frac{C^*}{C} \right\} \leq \rho(n),$$

gdje je  $\rho : \mathbb{N} \rightarrow \mathbb{R}^+$ . Funkciju  $\rho(n)$  zovemo **aproksimacijski omjer**, a algoritam koji ima  $\rho(n)$  aproksimacijski omjer zovemo  **$\rho(n)$ -aproksimacijski algoritam**.

U slučaju minimizacijskog problema, očekujemo da će približno rješenje biti nešto veće od egzaktnog, tj.  $0 < C^* \leq C$ . Tada aproksimacijski omjer iznosi  $\frac{C}{C^*}$ . Slično, kod maksimizacijskog problema je  $0 < C \leq C^*$  gdje aproksimacijski omjer iznosi  $\frac{C^*}{C}$ . Uočimo da je uvijek  $\rho(n) \geq 1$ . Kada je  $\rho(n) = 1$ , algoritma daje egzaktno (optimalno) rješenje.

U posebnu grupu ulaze aproksimacijski algoritmi koji postižu vrlo niske aproksimacijske omjere koristeći više vremena za računanje.

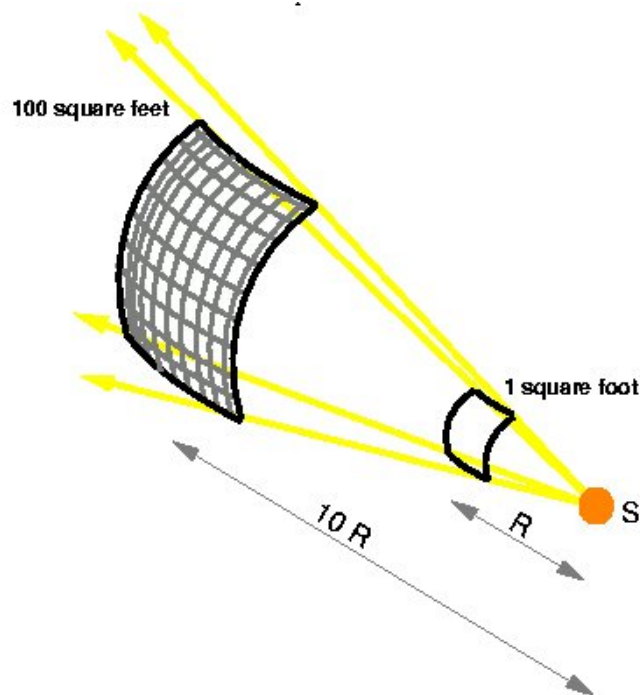
**Definicija 2.2.** *Neka je  $\epsilon > 0$ .  $(1 + \epsilon)$ -aproksimacijski algoritam koji se na veličini ulaza  $n$  izvršava u polinomijalnom vremenu naziva se  **$(1 + \epsilon)$ -aproksimacijska shema**.*

Vrijeme izvršavanja algoritma u većini slučajeva raste kada se  $\epsilon$  smanjuje. Primjerice, vrijeme izvršavanja polinomijalnog algoritma jedne  $(1 + \epsilon)$ -aproksimacijske sheme može biti  $O(\frac{1}{\epsilon} \log n)$ .

### 3. Problem osvjetljenja scene

#### 3.1. Fizikalni model

Problem optimalnog osvjetljenja scene podrazumijeva određene fizikalne pretpostavke. Ovdje ćemo objasniti fizikalni model širenja svjetla iz točkastog izvora u tri dimenzije. Svjetlosna energija koja pogađa koncentrične sfere uvijek je ista ali se gustoća svjetlosnih zraka smanjuje s povećanjem udaljenosti (polumjera koncentričnih sfera). Kao i u mnogim fizikalnim problemima, ovdje možemo primijeniti inverzni kvadratni zakon. Bez smanjenja općenitosti možemo pretpostaviti da je intenzitet svjetlosnog izvora upravo jednak primljenoj energiji. U svakoj točki na udaljenosti  $R$  gustoća svjetla iznosi  $\frac{E}{R^2}$  gdje je  $E$  intenzitet točkastog izvora. Ovu pretpostavku možemo opravdati na sljedeći način: Pretpostavimo da je snop svjetla jednoliko raspoređen na malom djeliću sfere koji se nalazi na udaljenosti  $R$  od točkastog izvora svjetla. Promotrimo sada dio sfere na udaljenosti  $10R$  od točkastog izvora kroz koji prolazi isti snop svjetla. Budući da je površina ovog djela  $10^2 = 100$  puta veća, gustoća svjetla je 100 puta manja (ista količina svjetlosne energije raspodijelila se na 100 puta veću površinu). Odatle zaključujemo da gustoća svjetla obrnuto razmjerna kvadratu udaljenosti od točkastog izvora (vidi Sliku 3.1 )



Slika 3.1: Snop svjetlosnih zraka kroz djelove koncentričnih sfera

U našem slučaju promatramo intenzitet svjetla u proizvoljnoj točki. Bit ćemo dovoljno slobodni pretpostaviti da intenzitet svjetla podliježe inverznom kvadratnom zakonu jer se u okolini svake točke na udaljenosti  $R$  nalazi  $\Theta(R^2)$  manje svjetlosnih zraka.

Sada ćemo opisati odgovarajući dvodimenzionalan problem kojim ćemo se baviti do kraja rada. Pretpostavimo da je dana dužina (kao zatvoreni segment na pravcu u ravnini)  $L \subset \mathbb{R}^2$  i skup točaka  $S \subset \mathbb{R}^2$ ,  $|S| = n$ .  $L$  označava *scenu*, a  $S$  skup izvora svjetla. Naš cilj je odrediti količine energije  $x_s$  za svaki izvor svjetla  $s \in S$  tako da svaka točka na sceni  $L$  prima dovoljnu količinu svjetla (o tome što znači "dovoljno" bit će riječ prilikom formulacije problema) a ukupna energija  $\sum_{s \in S} x_s$  bude minimalna.

### 3.2. Smanjenje broja izvora svjetla

Pod pretpostavkom da svaki izvor svjetla može trošiti proizvoljno mnogo energije, pokazat ćemo da je dovoljno promatrati izvore svjetla čije Voronoijeve ćelije sijeku scenu  $L$ . Izvore svjetla čije Voronoijeve ćelije ne sijeku scenu  $L$  možemo isključiti iz razmatranja i njihovu energiju postaviti na 0 bez bojazni da će optimalna potrošnja energije biti povećana ili da određeni dijelovi scene neće dobiti dovoljno svjetla. O tome govori sljedeća lema, no prije toga definirajmo određene pojmove.

**Definicija 3.1.** *Neka su  $s, q \in S$  izvori svjetla i  $s', q' \geq 0$  duljine njihovih ortogonalnih projekcija na scenu  $L$ . Kažemo da je  $s$  **lijevo [desno]** od  $q$  i pišemo  $s \preceq q$  [ $s \succeq q$ ] ako je  $s' \leq q'$  [ $s' \geq q'$ ].*

**Lema 3.1.** *Neka je  $s \in S$  izvor svjetla čija Voronoijeva ćelija ne siječe scenu  $L$ , te neka su  $s_L \in S$  prvi izvor svjetla lijevo od  $s$  i  $s_D \in S$  prvi izvor svjetla desno od  $s$  čije Voronoijeve ćelije sijeku scenu  $L$ .*

*Tada za svaku količinu energije<sup>9</sup>  $x_s$  i točku  $p \in L$  postoje količine energije  $x_{s_L}$  i  $x_{s_D}$  sa svojstvom  $x_{s_L} + x_{s_D} \leq x_s$  takve da je*

$$\frac{x_{s_L}}{|s_L p|^2} + \frac{x_{s_D}}{|s_D p|^2} \geq \frac{x_s}{|s p|^2}. \quad (3.1)$$

**Dokaz.** Pokazat ćemo da za svaku točku  $p \in L$  i količinu energije  $x_s$  postoje količine energije  $x_{s_L}$  i  $x_{s_D}$  takve da je  $x_{s_L} + x_{s_D} = x_s$  i vrijedi (3.1). To je i dovoljno pokazati jer ako želimo pronaći  $x_{s_L}$  i  $x_{s_D}$  takve da je  $x_{s_L} + x_{s_D} < x_s$ , postupak koji slijedi možemo provesti za  $x'_s$  gdje je  $x_{s_L} + x_{s_D} = x'_s < x_s$ .

Ako je  $x_{s_L} + x_{s_D} = x_s$ , postoji  $\alpha \in \mathbb{R}$ ,  $0 \leq \alpha \leq 1$  takav da je  $x_{s_L} = \alpha \cdot x_s$  i  $x_{s_D} = (1 - \alpha) \cdot x_s$ . Tada (3.1) možemo zapisati u obliku

$$\frac{\alpha}{|s_L p|^2} + \frac{1 - \alpha}{|s_D p|^2} \geq \frac{1}{|s p|^2}, \quad 0 \leq \alpha \leq 1, \quad (3.2)$$

i naš problem svesti na pronalaženje broja  $\alpha$  za kojeg vrijedi (3.2).

Neka je  $v$  točka presjeka Voronoijevog brida između  $s_L$  i  $s_D$  i scene  $L$ . Tada je  $|s_L v| = |s_D v|$ . Zbog pretpostavke da Voronoijeva ćelija od  $s$  ne siječe scenu  $L$ , očito je  $|s v| > |s_L v|$ . Pretpostavimo da smo točku  $s'$  dobili pomicanjem izvora svjetla  $s$  po okomici na scenu  $L$ ,

<sup>9</sup>od sada pa nadalje za svaki  $s \in S$  količinu energije na  $s$  označavamo sa  $x_s$

tako da je  $|s_L v| = |s_D v| = |s' v|$  (vidi Sliku 3.2). Ukoliko smo dokazali egzistenciju od  $\alpha$  za novi izvor svjetla  $s'$  za koje vrijedi

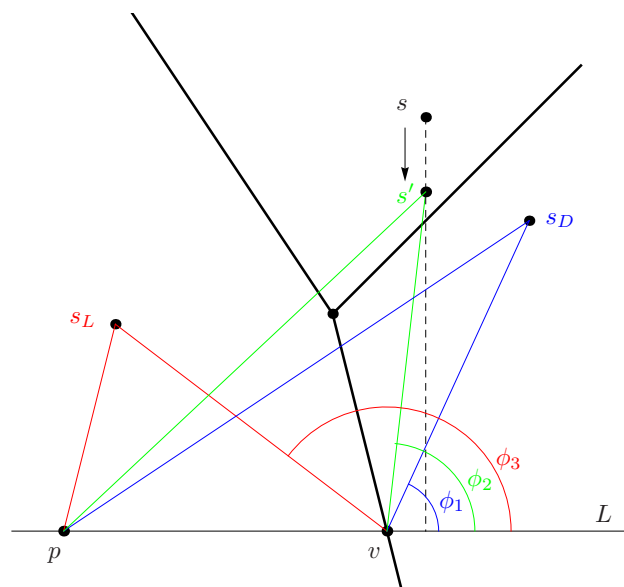
$$\frac{\alpha}{|s_L p|^2} + \frac{1 - \alpha}{|s_D p|^2} \geq \frac{1}{|s' p|^2},$$

zbog činjenice da je

$$\frac{1}{|s' p|^2} > \frac{1}{|s p|^2},$$

vidimo da naš  $\alpha$  zadovoljava (3.2). Nadalje, bez smanjenja općenitosti možemo pretpostaviti da je  $|s_L v| = |s_D v| = |s' v| = 1$ , jer se u suprotnom račun provodi analogno. Postoji dva slučaja s obzirom na položaj točke  $p$ :

1.  $p$  leži **lijevo** od  $v$ , tada je  $|s_L p| < |s' p| < |s_D p|$
2.  $p$  leži **desno** od  $v$ , tada je  $|s_L p| > |s' p| > |s_D p|$



Slika 3.2: Pomicanje izvora svjetla  $s$  tako da je  $|s_L v| = |s_D v| = |s' v|$

Mi ćemo detaljno promotriti prvi slučaj kada  $p$  leži **lijevo** od  $v$  a za drugi slučaj ćemo komentirati razlike u odnosu na prvi. Neka je  $|vp| = d$ . Prema oznakama sa Slike 3.2 vidimo da je  $\angle p v s_D = \pi - \phi_1$ ,  $\angle p v s' = \pi - \phi_2$  i  $\angle p v s_L = \pi - \phi_3$ . U drugom slučaju (kada je  $p$  desno od  $v$ ) je  $\angle p v s_D = \phi_1$ ,  $\angle p v s' = \phi_2$  i  $\angle p v s_L = \phi_3$ . Iz poučka o kosinusima slijedi da je

$$\begin{aligned} |s_D p|^2 &= 1 + d^2 \pm 2d \cos \phi_1, \\ |s' p|^2 &= 1 + d^2 \pm 2d \cos \phi_2, \\ |s_L p|^2 &= 1 + d^2 \pm 2d \cos \phi_3, \end{aligned}$$

gdje '+' stoji kada je  $p$  lijevo od  $v$ , a '-' kada je  $p$  desno od  $v$ . Iz (3.2) dobivamo da je

$$\frac{\alpha}{1 + d^2 \pm 2d \cos \phi_3} + \frac{1 - \alpha}{1 + d^2 \pm 2d \cos \phi_1} \geq \frac{1}{1 + d^2 \pm 2d \cos \phi_2}$$

što sređivanjem daje

$$\alpha \cdot \left( \frac{\pm(\cos \phi_1 - \cos \phi_3)}{|s_L p|^2} \right) \geq \frac{\pm(\cos \phi_1 - \cos \phi_2)}{|s p|^2}. \quad (3.3)$$

Sada ćemo rješenje nejednakosti (3.3) komentirati u dva prethodno opisana slučaja.

1. Kada je  $p$  lijevo od  $v$ , u (3.3) uzimamo '+', a kako je  $0 \leq \phi_1 < \phi_2 < \phi_3 \leq \pi$  dobivamo da je

$$\alpha \geq \frac{\cos \phi_1 - \cos \phi_2}{\cos \phi_1 - \cos \phi_3} \frac{|s_L p|^2}{|s p|^2}.$$

Budući je  $\frac{\cos \phi_1 - \cos \phi_2}{\cos \phi_1 - \cos \phi_3} \leq 1$  i  $\frac{|s_L p|^2}{|s p|^2} \leq 1$  dovoljno je uzeti  $\alpha = \frac{\cos \phi_1 - \cos \phi_2}{\cos \phi_1 - \cos \phi_3}$ .

2. Kada je  $p$  desno od  $v$ , u (3.3) uzimamo '-', a kako je  $0 \leq \phi_1 < \phi_2 < \phi_3 \leq \pi$  dobivamo da je

$$\alpha \leq \frac{\cos \phi_1 - \cos \phi_2}{\cos \phi_1 - \cos \phi_3} \frac{|s_L p|^2}{|s p|^2}.$$

Budući je  $\frac{\cos \phi_1 - \cos \phi_2}{\cos \phi_1 - \cos \phi_3} \leq 1$  i  $\frac{|s_L p|^2}{|s p|^2} \geq 1$  dovoljno je uzeti  $\alpha = \frac{\cos \phi_1 - \cos \phi_2}{\cos \phi_1 - \cos \phi_3}$ .

□

### 3.3. Formulacija problema

Prethodno opisani problem možemo formulirati na sljedeći način

$$\begin{aligned} & \min \sum_{s \in S} x_s \\ \text{uz uvjet } \forall p \in L : & \sum_{s \in S} \frac{x_s}{d^\sigma(p, s)} \geq 1, \\ & x_s \geq 0. \end{aligned} \quad (3.4)$$

Prvi uvjet u (3.4) zahtjeva da u svaku točku na sceni  $L$  stigne dovoljno svjetla<sup>10</sup>. Za određenu točku  $p \in L$  sumiramo ukupnu količinu svjetla koja stigne u  $p$  iz svih izvora svjetla. Količina svjetla koja iz  $s$  dolazi u  $p$  je obrnuto proporcionalna nekoj potenciji udaljenosti  $p$  od  $s$  (vidi Poglavlje 3.1., str. 17). Sa izvora svjetla energije  $x_s$  u točku  $p \in L$  stiže  $\frac{x_s}{d^\sigma(p, s)}$

<sup>10</sup>pod pojmom *dovoljno svjetla* podrazumjevamo uvjet u kojem postavljamo donju granicu na ukupnu količinu svjetla koja je stigla u određenu točku na sceni. U našem slučaju donja granica iznosi 1.

svjetla, gdje uzimamo da je  $\sigma = 2$ . Drugi uvjet pokazuje da pretpostavljamo da na svaki izvor svjetla možemo poslati proizvoljno mnogo energije.

Problem u (3.4) **nije** LP. Scena  $L$  sadrži neprebrojivo mnogo točaka zbog čega optimizacijski problem ima neprebrojivo mnogo uvjeta. Formulacija (3.4) odgovara problemu konveksnog programiranja (vidi [5]). Postoji aproksimacijski algoritam koji rješava problem (3.4) do na aditivnu konstantu (vidi [13], str. 7). Jedna od osnovnih zadaća rada jeste predstaviti aproksimacijske algoritme koji se zasnivaju na LP formulaciji.

### 3.4. $(1 + \epsilon)$ -aproksimacijska shema

Jedna od osnovnih ideja u konstruiranju aproksimacijskih algoritama za problem (3.4) jeste diskretizacija scene na prebrojivi konačan skup točaka  $G$  koji će zamijeniti scenu  $L$ . Umjesto prvog uvjeta u (3.4) zahtijevat ćemo da svaka točka iz skupa  $G$  primi dovoljno svjetla. Na taj način dolazimo do LP formulacije za problem (3.4). Rješenje koje smo dobili ne mora nužno biti dopustivo u originalnom problemu iz (3.4). Odredit ćemo konstantni faktor kojim je dovoljno pomnožiti rješenje LP problema i dobiti dopustivo rješenje originalnog problema. Pokazat ćemo da je to rješenje ujedno  $(1 + \epsilon)$ -aproksimacija.

Slijedi opis algoritma za konstrukciju skupa  $G$ .

**Definicija 3.2** (Udaljenost do najbližeg izvora svjetla). *Za svaku točku  $p \in L$  definiramo funkciju  $ens(p) : L \rightarrow [0, +\infty)$  na sljedeći način*

$$ens(p) = \min_{s \in S} d(p, s).$$

što predstavlja *udaljenost od  $p$  do najbližeg izvora svjetla*.

**Primjedba 3.1.** *Funkcija  $ens(\cdot)$  je 1-Lipschitz neprekidna, odnosno  $ens(p) \leq ens(q) + 1 \cdot |pq|$ .*

Koristeći nejednakost trokuta lako je dokazati prethodnu primjedbu. Neka su  $s_p, s_q \in S$  takvi da je  $ens(p) = |ps_p|$  i  $ens(q) = |qs_q|$ . Tada je

$$ens(p) = \min_{s \in S} |ps| \leq |ps_q| \leq |pq| + |qs_q| = ens(q) + 1 \cdot |pq|.$$

Diskretizacija scene provodit će se na osnovu udaljenosti do najbližeg izvora svjetla. Na područjima scene gdje je  $ens(\cdot)$  mali<sup>11</sup> nalazit će se više točaka, a na mjestima gdje je  $ens(\cdot)$  veliki nalazit će se manje točaka. U tom slučaju jasno je da udaljenosti među točkama na sceni moraju ovisiti o funkciji  $ens(\cdot)$ . U nastavku dajemo definiciju takvog skupa.

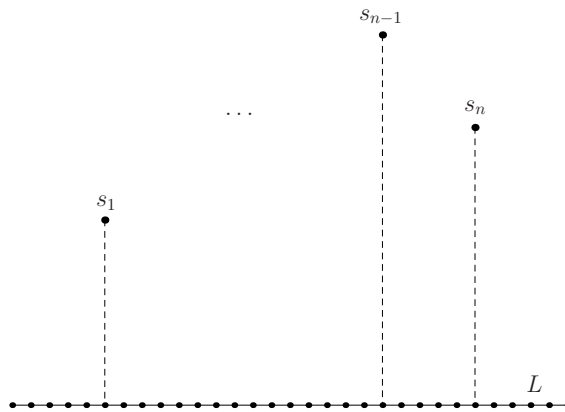
**Definicija 3.3** ( $\epsilon$ -dobar skup točaka). *Skup točaka  $G \subset L$  koji ima sljedeće svojstvo*

$$\forall p \in L \exists g \in G : d(p, g) \leq \epsilon \cdot ens(p),$$

*naziva se  $\epsilon$ -dobar skup točaka.*

<sup>11</sup>tj. na mjestima gdje je "najbliži izvor svjetla veoma blizu"

Pretpostavimo da je najmanja udaljenost svakog izvora svjetla do scene  $L$  jednaka 1 te da je duljina scene jednaka  $D$ .  $\epsilon$ -dobar skup točaka možemo dobiti tako da počevši od neke točke  $p \in L$  u skup  $G$  dodajemo točke na udaljenosti  $\epsilon$ , a to činimo duž cijele scene  $L$  (vidi Sliku 3.3). Na taj način konstruirali smo skup veličine  $O(\frac{D}{\epsilon})$ . Pokazat ćemo da  $\epsilon$ -dobar skup možemo konstruirati i sa **manje** točaka u asimptotskom smislu.



Slika 3.3: Trivijalan  $\epsilon$ -dobar skup točaka

Pretpostavimo da postoji samo jedan izvor svjetla  $s^{12}$ . Počnimo od krajnje lijeve točke  $p_l$  na sceni  $L^{13}$ . Dodajmo  $p_l$  u skup  $G_s$ . Nastavljamo s konstrukcijom skup  $G_s$  tako da desno od  $p_l$  dodamo točku na udaljenosti  $2\epsilon \text{ens}(p_l)$ . Postupak nastavljamo iterativno sve dok je moguće dodati točku koja je na sceni  $L$ . Prethodno opisani postupak formaliziran je u Algoritmu 3.1.

---

**Algoritam 3.1:**  $\epsilon$ -dobar skup točaka

---

**Ulaz:** duljina scene  $D$ , izvor svjetla  $s$ , krajnja lijeva točka  $p_0$ ,  $\epsilon$

**Izlaz:** skup točaka  $G_s$

$G_s \leftarrow G_s \cup \{p_0\};$

$p \leftarrow$  točka na sceni  $L$  desno od  $p_0$  na udaljenosti  $2\epsilon \text{ens}(p_0);$

**while**  $p$  je na sceni  $L$  **do**

$G_s \leftarrow G_s \cup \{p\};$

$p \leftarrow$  točka na sceni  $L$  desno od  $p$  na udaljenosti  $2\epsilon \text{ens}(p);$

---

Sljedeća lema tvrdi da je skup  $G_s$  konstruiran u Algoritmu 3.1  $\epsilon$ -dobar skup točaka dajući asimptotsku ocjenu za njegovu veličinu.

**Lema 3.2.** *Neka je  $s \in S$  i  $\epsilon > 0$ . Tada za skup  $G_s$  dobiven Algoritmom 3.1 vrijedi:*

---

<sup>12</sup>Ova pretpostavka ne umanjuje općenitost budući da smo u prethodnom poglavlju pokazali kako pomoću Voronoijevog dijagrama možemo smanjiti broj izvora svjetla. Naime, svaka Voronoijeva ćelija  $\mathcal{V}(s)$  koja siječe scenu  $L$  odredit će segment kojem će pripadati samo *jedan* izvor svjetla  $s$

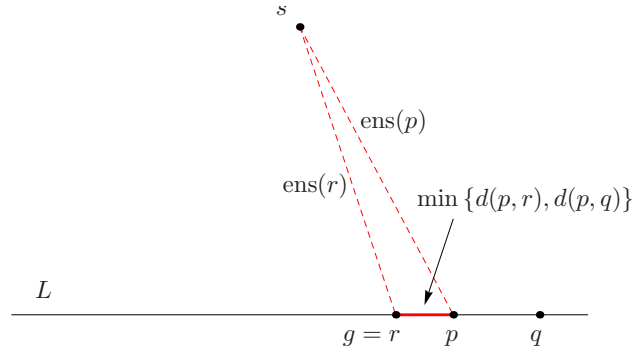
<sup>13</sup>možemo početi i od bilo koje druge točke sa malom modifikacijom algoritma, vidi [13], str. 10

- a)  $G_s$  je  $\epsilon$ -dobar skup točaka za jedan izvor svjetla  $s \in S$
- b)  $|G_s| = O\left(\frac{\log D}{\epsilon}\right)$ , gdje je  $D$  duljina scene  $L$ .

**Dokaz.**

- a) Neka je  $p \in L$ . Treba pokazati da postoji točka  $g \in G_s$  takva da je  $d(p, g) \leq \epsilon \text{ens}(p)$ . Bez smanjenja općenitosti pretpostavimo da je  $r \preceq p \preceq q$  gdje su  $r, q \in G_s$  takve da je  $r, q \succ s$ , te da je  $r$  prva točka lijevo od  $p$ , a  $q$  prva točka desno od  $p$  (dokaz se provodi analogno ako pretpostavimo da je  $p$  između dvije točke iz  $G_s$  koje se nalaze lijevo od  $s$ ). Zbog pretpostavke  $r \preceq p$  vrijedi da je  $\text{ens}(r) \leq \text{ens}(p)$  (vidi Sliku 3.4). Očito je

$$\min \{d(p, r), d(p, q)\} \leq \frac{|rq|}{2}.$$



Slika 3.4:  $\epsilon$ -dobar skup točaka  $G_s$

Kako je  $|rq| \leq 2\epsilon \text{ens}(r)$  (prema konstrukciji iz Algoritma 3.1), dobivamo da je

$$\min \{d(p, r), d(p, q)\} \leq \epsilon \text{ens}(r) \leq \epsilon \text{ens}(p).$$

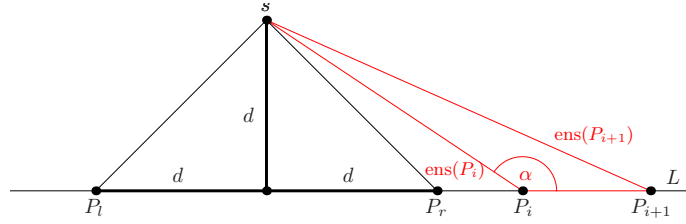
Točku  $g$  biramo među točkama  $r$  i  $q$  tako da uzmemo onu koja je bliža točki  $p$ .

- b) Sada ćemo odrediti asimptotsku ocjenu za kardinalitet skupa  $G_s$ . Neka je  $d$  udaljenost izvora svjetla  $s$  od scene  $L$ , te neka su točke  $P_l, P_r \in L$  kao na Slici 3.5.

Tada je  $\text{ens}(p) \geq d$  za svaku točku  $p \in \overline{P_l P_r}$ , odnosno  $|uv| = 2\epsilon \text{ens}(u) \geq 2\epsilon d$ . Zaključujemo da je unutar  $\overline{P_l P_r}$  sadržano  $O(1/\epsilon)$  točaka.

Sada ćemo ocjeniti broj točaka izvan segmenta  $\overline{P_l P_r}$ . Neka su  $P_i, P_{i+1} \succ P_r$ . Prema poučku o kosinusima slijedi da je

$$\begin{aligned} \text{ens}(P_{i+1})^2 &= |P_i P_{i+1}|^2 + \text{ens}(P_i)^2 - 2\text{ens}(P_i)|P_i P_{i+1}| \cos \alpha \\ &= 4\epsilon^2 \text{ens}(P_i)^2 + \text{ens}(P_i)^2 - 4\epsilon \text{ens}(P_i)^2 \cos \alpha \\ &= \text{ens}(P_i)^2 (1 + 4\epsilon^2 - 4\epsilon \cos \alpha) \geq \text{ens}(P_i)^2 (1 + \epsilon)^2 \end{aligned}$$


 Slika 3.5: Ocjena broja točaka u skupu  $G_s$ 

gdje posljednja nejednakost slijedi iz činjenice da je  $\frac{3\pi}{4} \leq \alpha < \pi$ , odnosno  $-1 < \cos \alpha \leq -\frac{1}{2}$ . Odatle dobivamo da je  $\text{ens}(P_{i+1}) \geq \text{ens}(P_i)(1 + \epsilon)$  što znači da udaljenost između točaka unutar  $\overline{P_l P_r}$  raste najmanje s faktorom  $a = 1 + \epsilon$  u svakoj iteraciji. Sada ćemo odrediti gornju među na broj točaka izvan intervala  $\overline{P_l P_r}$ .

$$\begin{aligned} D &\geq 2\epsilon \sum_{i=0}^{|G_s|-1} |p_i p_{i+1}| = 2\epsilon \sum_{i=0}^{|G_s|-1} \text{ens}(p_i) \\ &\geq 2\epsilon \text{ens}(p_0) \sum_{i=0}^{|G_s|-1} a^i \geq 2\epsilon \frac{a^{|G_s|} - 1}{a - 1}, \end{aligned}$$

što transformacijom daje

$$\begin{aligned} |G_s| &\leq \log_a \left( \frac{D}{2} + 1 \right) = \frac{\log(D + 2) - 1}{\log(1 + \epsilon)} \\ &\leq \frac{\log(D + 2)}{\log(1 + \epsilon)} = O\left(\frac{\log D}{\epsilon}\right). \end{aligned}$$

jer je  $\log(1 + \epsilon) \approx \epsilon$  za  $\epsilon \approx 0$ .

Budući da broj točaka unutar  $\overline{P_l P_r}$  ima ocjenu  $O(1/\epsilon)$  možemo zaključiti da je  $O(\frac{\log D}{\epsilon})$  ocjena za ukupan broj točaka generiranih Algoritmom 3.1.  $\square$

Ukoliko duljinu scene  $D$  smatramo konstantom, ocjena iz Leme 3.2 je ujedno asimptotski donja ocjena. Koristeći pretpostavku da je izvor svjetla  $s$  udaljen za bar 1 od scene  $L$ , veličina praznog susjedstva  $\text{ens}(\cdot)$  može biti najviše  $\sqrt{1 + D^2}$ . Tada je broj točaka unutar  $G_s$  najmanje  $\frac{D}{d(p,q)} = \frac{D}{\epsilon\sqrt{1+D^2}} = \Omega(\frac{1}{\epsilon})$ , odnosno  $|G_s| = \Theta(\frac{1}{\epsilon})$  (vidi [11], str. 42).

U Algoritmu 3.1 pokazali smo kako konstruirati  $\epsilon$ -dobar skup točaka za **jedan** izvor svjetla  $s \in S$ . Neka je  $G = \bigcup_{s \in S} G_s$ . Nije se teško uvjeriti da je  $G$   $\epsilon$ -dobar skup za cijeli  $S$  (odnosno za sve izvore svjetla). Za svaku točku  $p \in L$  očito postoji točka koja je najviše udaljena za  $\epsilon \cdot \text{ens}(p)$  jer je  $\text{ens}(p) = d(p, s_p)$  za neki  $s_p \in S$  a  $G_{s_p}$  je  $\epsilon$ -dobar skup unutar kojeg možemo pronaći takvu točku. Asimptotski gornja ocjena za broj točaka u  $G$  je  $O(\frac{n}{\epsilon} \log D)$ .

Može se postaviti pitanje da li smo "pretrpali" skup  $G$  s nepotrebnim točkama dodajući svaki put novi  $G_s$  po cijeloj duljini scene. Sljedeća lema pokazuje da je  $G$  "pretrpan" i da je korištenjem Voronoijevog dijagrama moguće konstruirati asimptotski manji  $\epsilon$ -dobar skup  $G$ .

**Lema 3.3.** *Postoji  $\epsilon$ -dobar skup točaka  $G$  čiji kardinalitet ima gornju asimptotsku ocjenu  $O(\frac{n}{\epsilon} \log(1 + \frac{D}{n}))$ .*

**Dokaz.** Kao i u prethodnom poglavlju, uzmimo u obzir samo Voronoijeve ćelije koje si-  
jeku scenu  $L$ . One mogu podijeliti scenu  $L$  u najviše  $n$  dijelova koje ćemo označiti sa  $D_1, D_2, \dots, D_n$ . Svakom dijelu  $D_i$  odgovara određeni izvor svjetla  $s_i \in S$ . Iz definicije Voronoijevog dijagrama slijedi da je  $\text{ens}(p) = d(p, s_i)$ , za svaku točku  $p \in D_i$ . Neka su  $d_1, d_2, \dots, d_n$  redom duljine dijelova scene  $D_1, D_2, \dots, D_n$ . Na svaki dio  $D_i$  možemo primijeniti Algoritam 3.1. Dobiveni  $\epsilon$ -dobar skup označimo sa  $G_i$ . Neka je  $G = \bigcup_i^n G_i$ . Tada je prema Lemi 3.2

$$|G| = \sum_{i=1}^n |G_i| = O\left(\frac{1}{\epsilon} \sum_{i=1}^n \log(1 + d_i)\right).$$

Ova suma postiže maksimum kada su svi dijelovi jednake duljine, tj. kada je  $d_i = \frac{D}{n}$ .  $\square$

**Lema 3.4.** *Neka je  $G$   $\epsilon$ -dobar skup točaka, te neka su  $p$  i  $q$  dvije susjedne točke iz  $G$ . Tada je*

$$|pq| \leq \frac{2\epsilon}{1 - \epsilon} \text{ens}(p).$$

**Dokaz.** Bez smanjenja općenitosti pretpostavimo da je  $p \preceq q$ . Neka je  $z \in L$  točka takva da je  $|pz| = |zq|$ . Prema Definiciji 3.3 znamo da je  $|zq| \leq \epsilon \text{ens}(z)$ . Koristeći Primjedbu 3.1 dobivamo da je

$$|pq| = 2|zq| \leq 2\epsilon \text{ens}(z) \leq 2\epsilon(\text{ens}(p) + |pz|) \leq 2\epsilon(\text{ens}(p) + \frac{1}{2}|pq|),$$

odnosno

$$(1 - \epsilon)|pq| \leq 2\epsilon \text{ens}(p).$$

Sređivanjem gornje nejednakosti dobivamo tvrdnju leme.  $\square$

**Lema 3.5.** *Neka je  $\mathbf{x}_s$  rješenje LP-a u (3.4) s obzirom na  $\epsilon$ -dobar skup točaka  $G$ , gdje je  $\epsilon < c$  za neki  $0 < c < 1$ . Tada je  $(1 + \frac{2-c}{(1-c)^2} \cdot \epsilon)\mathbf{x}_s$  dopustivo rješenje originalnog problema u (3.4), tj. množenje rješenja  $\mathbf{x}_s$  s faktorom  $1 + \frac{2-c}{(1-c)^2}\epsilon$  osigurava dovoljno osvjetljenje svih točaka na sceni  $L$ .*

**Dokaz.** Rješenje  $\mathbf{x}_s$  zadovoljava uvjet da je svaka točka  $p' \in G$  dovoljno osvjetljena. Pretpostavimo da je  $p \in L$  točka koja **ne pripada** skupu  $G$ . Želimo pokazati da množenjem rješenja  $\mathbf{x}_s$  s faktorom  $\psi = 1 + O(\epsilon)$  postizemo da je

$$\sum_{s \in S} \frac{\psi \cdot x_s}{|sp|^2} \geq 1, \tag{3.5}$$

tj. da točka  $p$  prima dovoljno svjetla. Budući je  $|sp| \leq |sp'| + |p'p| \forall s \in S$ , iz Leme 3.5 slijedi da je

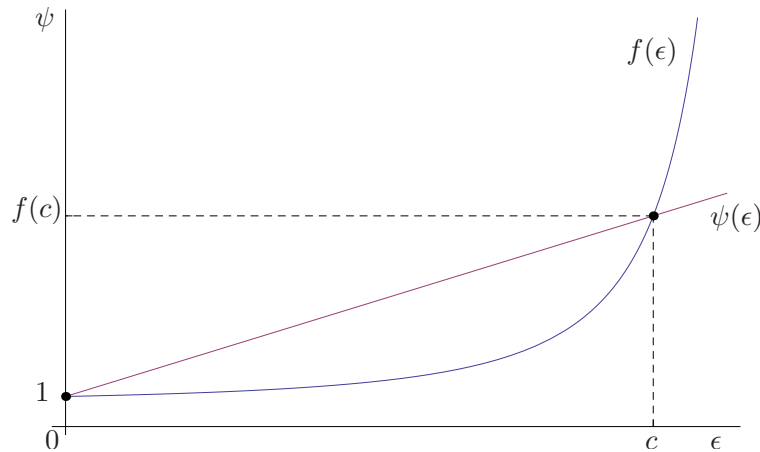
$$\sum_{s \in S} \frac{\psi \cdot x_s}{|sp|^2} \geq \sum_{s \in S} \frac{\psi \cdot x_s}{(|sp'| + |p'p|)^2} \geq \sum_{s \in S} \frac{\psi \cdot x_s}{|sp'|^2 \left(1 + \frac{\epsilon}{1-\epsilon} \cdot \frac{\text{ens}(p')}{|sp'|}\right)^2} \quad (3.6)$$

Prema Definiciji 3.2 slijedi da je  $\text{ens}(p') \leq |p's|$  odnosno  $\frac{\text{ens}(p')}{|p's|} \leq 1$  za svaki  $s \in S$ . Zbog pretpostavke da  $p' \in G$  prima dovoljno svjetla, dobivamo da je

$$\sum_{s \in S} \frac{\psi \cdot x_s}{|sp'|^2 \left(1 + \frac{\epsilon}{1-\epsilon} \cdot \frac{\text{ens}(p')}{|sp'|}\right)^2} \geq \frac{\psi}{\left(1 + \frac{\epsilon}{1-\epsilon}\right)^2}. \quad (3.7)$$

Iz (3.6) i (3.7) vidimo da je dovoljno zahtijevati

$$\frac{\psi}{\left(1 + \frac{\epsilon}{1-\epsilon}\right)^2} = \psi(1 - \epsilon)^2 \geq 1.$$



Slika 3.6: Određivanje broja  $\psi$

odnosno  $\psi \geq \frac{1}{(1-\epsilon)^2}$ . Neka su  $\psi = \psi(\epsilon) = k \cdot \epsilon + l$  i  $f(\epsilon) = \frac{1}{(1-\epsilon)^2}$  (vidi Sliku 3.6) takvi da je  $\psi(0) = f(0)$  i  $\psi(c) = f(c)$ . Konstante  $k$  i  $l$  dobivamo iz traženih uvjeta

$$\psi = \psi(\epsilon) = 1 + \frac{2 - c}{(1 - c)^2} \epsilon.$$

Sada vidimo da je

$$\begin{aligned}
\psi &= 1 + \frac{2-c}{(1-c)^2} \epsilon = 1 + \left( \frac{1}{(1-c)^2} + \frac{1}{1-c} \right) \epsilon \\
&\geq 1 + \left( \frac{1}{(1-\epsilon)^2} + \frac{1}{1-\epsilon} \right) \epsilon \\
&= \frac{1}{(1-\epsilon)^2}.
\end{aligned}$$

□

Specijalno za  $c = \frac{1}{2}$  dobivamo da je  $\psi = 1 + 6\epsilon$ .

Rezultati Leme 3.3 i Leme 3.5 objedinjeni su u sljedećem ključnom teoremu za ovo poglavlje.

**Teorem 3.1** (( $1 + \epsilon$ )-aproksimacijska shema). *Neka je  $D$  duljina scene  $L$  i  $S$  skup izvora svjetla od kojih je svaki barem na jediničnoj udaljenosti od scene  $L$ . Tada je moguće izračunati  $\mathbf{x}_S$  kao količine energije za svaki izvor svjetla  $s \in S$  tako da svaka točka na sceni  $L$  primi najmanje jediničnu količinu svjetla gdje je*

$$\sum_{s \in S} x_s \leq (1 + \epsilon) \sum_{s \in S} x_s^{OPT},$$

gdje je  $x_s^{OPT}$  optimalna količina energije (tj. rješenje problema iz (3.4)).  $\mathbf{x}_S$  je moguće izračunati u polinomijalnom vremenu rješavajući LP sa  $O(\frac{n}{\epsilon} \log \frac{D}{n})$  uvjeta i  $n$  varijabli.

### 3.5. Jednostavni $O(1)$ -aproksimacijski algoritam

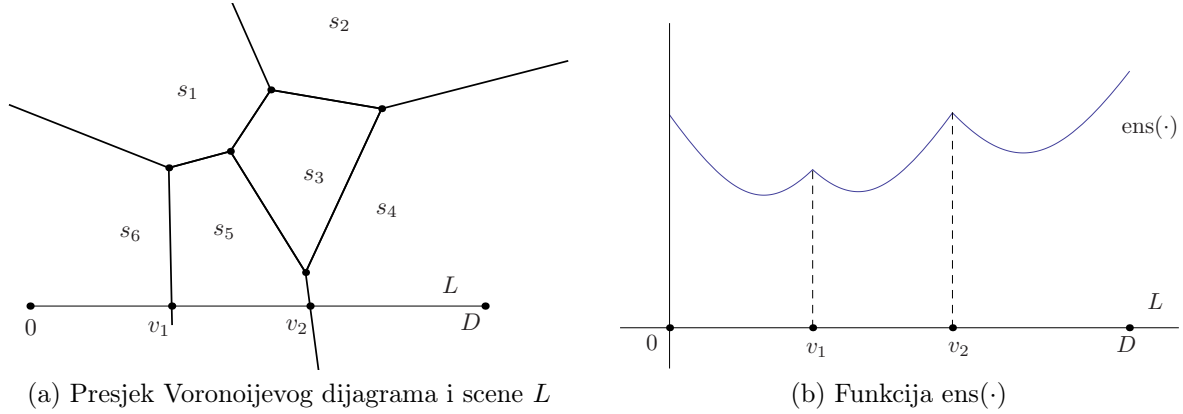
( $1+\epsilon$ )-aproksimacijski algoritam za problem osvjetljenja scene koji smo konstruirali u prethodnom poglavlju zasniva se na rješavanju LP-a koji i pored toga što je polinomijalan zahtjeva dosta vremena. Pored toga, postoji i logaritamska zavisnost o duljini scene  $D$ . U ovom poglavlju konstruirat ćemo, između ostalog, jednostavni kombinatorijalni algoritam koji se izvršava u  $O(n^2)$  vremenu. Skup točaka  $G_s$  iz prethodnog poglavlja dodatno ćemo relaksirati na veličinu reda  $O(n)$ . Rješenje takvog reduciranog problema razlikovat će se od rješenja originalnog problema u faktoru reda veličine  $O(1)$ .

Sada ćemo detaljnije opisati funkciju  $\text{ens}(p)$ . Pretpostavimo li da je scena  $L$  skup točaka

$$L = \{(p, 0) : 0 \leq p \leq D\} \subset \mathbb{R}^2,$$

gdje je  $D$  duljina scene  $L$ , funkciju  $\text{ens}(\cdot)$  možemo definirati kao funkciju jedne varijable.

**Propozicija 3.1.** *Neka je  $\mathcal{V}(S)$  Voronoijev dijagram skupa  $S$  i  $l_i = L \cap \mathcal{V}(s_i)$  dijelovi scene  $L$  unutar Voronoijevih ćelija  $\mathcal{V}(s_i)$  koje sijeku scenu  $L$  i za  $1 \leq i \leq m$ . Neka su  $v_j$  točke presjeka Voronoijevih bridova sa scenom  $L$  (vidi Sliku 3.7) i  $\text{ens}(\cdot) : [0, D] \rightarrow \mathbb{R}$ , gdje je  $D$  duljina scene  $L$ . Tada vrijede sljedeće tvrdnje:*


 Slika 3.7: Analiza funkcije  $\text{ens}(\cdot)$ 

- a) Funkcija  $\text{ens}(\cdot)$  je diferencijabilna u svim točkama osim u  $v_j$ .
- b) U točkama  $v_j$  funkcija  $\text{ens}(\cdot)$  postiže lokalne maksimume.

**Dokaz.**

- a) Neka je  $v_k$  točka presjeka scene  $L$  i Voronoijevog brida između ćelija  $\mathcal{V}(s_k)$  i  $\mathcal{V}(s_{k+1})$ , gdje je  $1 \leq k \leq m - 1$  (vidi Sliku 3.7). Očito je  $l_i = [v_{i-1}, v_i]$  gdje su  $v_0 = 0$  i  $v_m = D$ . Vidimo da je svakoj točki  $p \in l_i$  (tj.  $p \in \mathcal{V}(s_i)$ ) najbliži izvor svjetla  $s_i$ . To znači da je

$$\text{ens}(p) = \sum_{i=1}^m I_i(p) \cdot d(p, s_i). \quad (3.8)$$

gdje je  $I_i(\cdot)$  indikator intervala  $l_i$ . Neka je  $s_k = (s_k^x, s_k^y)$  i  $s_{k+1} = (s_{k+1}^x, s_{k+1}^y)$ . Sada ćemo ispitati diferencijabilnost funkcije  $\text{ens}(\cdot)$  u točki  $v_k$ . Derivacija s lijeva (vidi [15]) u točki  $v_k$  je zadana sljedećim izrazom:

$$\begin{aligned} \text{ens}(v_k)'_{-} &= \lim_{v \rightarrow v_k^-} \frac{\text{ens}(v) - \text{ens}(v_k)}{v - v_k} = \lim_{v \rightarrow v_k^-} \frac{d(v, s_k) - d(v_k, s_k)}{v - v_k} \\ &= \lim_{v \rightarrow v_k^-} \frac{d(v, s_k) - d(v_k, s_k)}{v - v_k} \cdot \frac{d(v, s_k) + d(v_k, s_k)}{d(v, s_k) + d(v_k, s_k)} \\ &= \lim_{v \rightarrow v_k^-} \frac{d^2(v, s_k) - d^2(v_k, s_k)}{(v - v_k)(d(v, s_k) + d(v_k, s_k))} \\ &= \lim_{v \rightarrow v_k^-} \frac{(s_k^x - v)^2 + (s_k^y)^2 - ((s_k^x - v_k)^2 + (s_k^y)^2)}{d(v, s_k) + d(v_k, s_k)} \\ &= \lim_{v \rightarrow v_k^-} \frac{v + v_k - 2s_k^x}{d(v, s_k) + d(v_k, s_k)} = \frac{v_k - s_k^x}{d(v_k, s_k)}. \end{aligned} \quad (3.9)$$

Na sličan način se može pokazati da je

$$\text{ens}(v_k)'_+ = \frac{v_k - s_{k+1}^x}{d(v_k, s_{k+1})}. \quad (3.10)$$

Kako je  $s_k^x \neq s_{k+1}^x$  i  $d(v_k, s_k) = d(v_k, s_{k+1}) = \text{ens}(v_k)$ , iz (3.9) i (3.10) slijedi da je

$$\text{ens}(v_k)'_- = \text{ens}(v_k)'_+,$$

odnosno, funkcija  $\text{ens}(\cdot)$  nije diferencijabilna u točkama  $v_k$ ,  $1 \leq k \leq m$ . Ukoliko je  $v \neq v_k$ , izrazi u (3.9) i (3.10) imaju jednake vrijednosti.

b) Dokaz jednostavan (geometrijski).

□

**Lema 3.6.** *Neka je  $G_V$  skup točaka  $v_k \in [0, D]$  u kojima funkcija  $\text{ens}(\cdot)$  ima lokalne maksimume (točke iz Propozicije 3.1). Nadalje, neka je  $\mathbf{x}_V^*$  dopustivo rješenje problema u (3.4) u odnosu na skup točaka  $G_V$ . Tada je  $\mathbf{x}^* = 4 \cdot \mathbf{x}_V^*$  dopustivo rješenje u odnosu na sve točke scene  $L$ .*

**Dokaz.** Neka je  $p \in L$  i  $p \notin G_V$ . Neka je točka  $p_V \in G_V$  takva da je  $p$  sadržana u krugu polumjera  $\text{ens}(p_V)$ . Takva točka  $p_V$  uvijek postoji jer je svaka točka na sceni  $L$  sadržana u najmanje jednom krugu s centrom u prvoj lijevoj ili prvoj desnoj susjednoj točki u  $G_V$ . Jasno je da vrijedi  $\text{ens}(p_V) \leq d(p_V, s)$  za svaki  $s \in S$ , tj. svi izvori svjetla su najmanje  $\text{ens}(p_V)$  udaljeni od točke  $p_V$ . Prema izboru točke  $p_V$  imamo da je  $|pp_V| \leq \text{ens}(p_V)$ . Nadalje, vrijedi i sljedeće

$$|ps| \leq |pp_V| + |p_Vs| \leq \text{ens}(p_V) + |p_Vs| \leq 2|p_Vs|.$$

Sada ćemo odrediti faktor  $\psi$  takav da je  $\sum_{s \in S} \frac{\psi \cdot x_s^*}{|ps|^2} \geq 1$ .

$$\sum_{s \in S} \frac{\psi \cdot (x_V^*)_s}{|ps|^2} \geq \sum_{s \in S} \frac{\psi \cdot (x_V^*)_s}{4|p_Vs|^2} \geq \frac{\psi}{4} \sum_{s \in S} \frac{(x_V^*)_s}{|p_Vs|^2} \geq 1.$$

Vidimo da je dovoljno uzeti  $\psi = 4$ , odnosno da je  $\mathbf{x}^* = 4 \cdot \mathbf{x}_V^*$  dopustivo rješenje za problem u (3.4). □

Direktna posljedica prethodne leme je sljedeći korolar.

**Korolar 3.1.** *Rješavanje LP-a s najviše  $n + 1$  uvjeta (kada zahtjevamo da svaka točka u  $G_V$  dobije dovoljno svjetla) predstavlja 4-aproksimacijski algoritam za problem u (3.4).*

Drugim riječima, ako želimo  $O(1)$ -aproksimaciju rješenja, to možemo postići u vremenu neovisnom o duljini  $D$  scene  $L$ . Nasuprot tome, kod  $(1 + \epsilon)$ -aproksimacijske sheme imali smo logaritamsku ovisnost o duljini scene.

**Algoritam 3.2:**  $\alpha$ -reduciranje skupa  $G_V$ **Ulaz:** skup točaka  $G_V$ , konstanta  $\alpha > 0$ **Izlaz:** skup točaka  $G_P$  $G_P = G_V$ ;Izračunati  $\text{ens}(p_i)$  za svaki  $p_i \in G_P$ ;Sortirati  $p_i$  u padajućem poretku prema vrijednosti funkcije  $\text{ens}(\cdot)$ , tj. tako da je $\text{ens}(p_1) \geq \text{ens}(p_2) \geq \dots \geq \text{ens}(p_m)$ ;**for**  $i = 1, \dots, n$  **do**    **if**  $p_i \in G_P$  **then**        **for**  $j = i + 1, \dots, n$  **do**            **if**  $|p_i p_j| \leq \alpha \text{ens}(p_i)$  **and**  $p_j \in G_P$  **then**                 $G_P = G_P \setminus \{p_j\}$ **return**  $G_P$ ;**3.5.1.  $\alpha$  - reduciranje skupa  $G_V$** 

U nastavku želimo konstruirati skup točaka  $G_V$  koji mora imati svojstvo da je  $|p_i p_j| \geq C \cdot \max \{\text{ens}(p_i), \text{ens}(p_j)\}$  za neku konstantu  $C > 0$ . Budući da ovo svojstvo ne mora uvijek biti zadovoljeno, konstruirat ćemo algoritam koji će ukloniti nepotrebne točke.

U nastavku ćemo pokazati da za konstantu  $\alpha$  čak i nakon uklanjanja točaka (tj. za skup  $G_P$ ) možemo konstruirati  $O(1)$ -aproksimacijski algoritam koristeći skup  $G_P$ . Prvo ćemo odrediti konstantu kojom trebamo pomnožiti dopustivo rješenje problema (3.4) u odnosu na skup  $G_V$ .

**Lema 3.7.** *Neka je  $x_P^*$  dopustivo rješenje problema (3.4) takvo da svaka točka iz  $G_P$  primi dovoljno svijetla. Tada je  $x_V^* = (1 + \alpha)^2 \cdot x_P^*$  dopustivo rješenje za skup točaka  $G_V \supseteq G_P$ .*

**Dokaz.** Neka je  $p_i \in G_V$  i  $p_i \notin G_P$ . Pretpostavimo da je  $p_j \in G_P$  točka zbog koje je  $p_i$  izbačena iz skupa  $G_P$ . Prema Algoritmu 3.2 slijedi da je bio ispunje uvjet  $|p_i p_j| \leq \alpha \text{ens}(p_i)$ . Sada imamo da je

$$\begin{aligned} |p_i s| &\leq |p_i p_j| + |p_j s| \leq \alpha \text{ens}(p_j) + |p_j s| \\ &\leq \alpha |p_j s| + |p_j s| = (1 + \alpha) |p_j s|. \end{aligned}$$

Slično kao i u dokazu Leme 3.6 zaključujemo da je dopustivo rješenje  $x_P^*$  dovoljno pomnožiti faktorom  $(1 + \alpha)^2$ , tj. da je  $(1 + \alpha)^2 \cdot x_P^*$  dopustivo rješenje problema (3.4) s obzirom na skup točaka  $G_V$ .  $\square$

**Lema 3.8.** *Za bilo koje dvije točke  $p_i, p_j \in G_P$  vrijedi sljedeće svojstvo:*

$$|p_i p_j| \geq \alpha \max \{\text{ens}(p_i), \text{ens}(p_j)\}.$$

**Dokaz.** Dokaz slijedi iz Algoritma 3.2. Kada ovaj uvjet ne bi bio ispunjen mogli bismo razmotriti dva slučaja koji vode u kontradikciju. U prvom slučaju kada je  $\text{ens}(p_i) \geq \text{ens}(p_j)$ , točka  $p_j$  bi bila izbačena iz skupa  $G_P$ , inače bi bila izbačena točka  $p_i$ .  $\square$

Direktna posljedica Leme 3.7 i Korolara 3.1 je sljedeći korolar.

**Korolar 3.2.** *Rješavanje LP-a s uvjetima da svaka točka u  $G_P$  primi dovoljno svjetla predstavlja  $4(1 + \alpha)^2$ -aproksimacijski algoritam za problem u (3.4).*

Sada ćemo opisati kombinatorijalni algoritam koji služi za pridruživanje energije izvorima svjetla iz skupa  $S$ . Pretpostavimo da je  $|G_P| = k$ , te da svakoj točki  $p_i \in G_P$  odgovara njoj najbliži izvor svjetla  $s_i$  kojem ćemo pridružiti količinu energije  $x_i$ .

---

**Algoritam 3.3:** Kombinatorijalni  $O(1)$ -algoritam

---

**Ulaz:** skup točaka  $G_V$ , konstanta  $\alpha > 0$

**Izlaz:** rješenje  $\mathbf{x}_P^*$

Izračunati skup točaka  $G_V$  koristeći Voronoijev dijagram skupa  $S$ ;

Koristeći Algoritam 3.2 odrediti skup  $G_P$  s konstantom  $\alpha$ ,  $|G_P| = k$ ;

Sortirati  $p_i \in G_P$  u padajućem poretku prema vrijednosti funkcije  $\text{ens}(\cdot)$ , tj. tako da je  $\text{ens}(p_1) \geq \text{ens}(p_2) \geq \dots \geq \text{ens}(p_k)$ ;

**for**  $i = 1, \dots, k$  **do**

$$(x_P^*)_i = \max \left\{ 0, |p_i s_i|^2 \cdot \left( 1 - \sum_{j=1}^{i-1} \frac{x_j}{|s_j p_i|^2} \right) \right\}$$

**return**  $\mathbf{x}_P^*$ ;

---

Osnovna ideja Algoritma 3.3 je pridruživanje energije izvoru svjetla koji odgovara točki iz  $G_P$  i to samo onoliko koliko je potrebno da ta točka primi dovoljno energije (uzimajući u obzir i količinu svjetla koja dolazi sa ostalih izvora). Pridruživanje se odvija u padajućem poretku s obzirom na vrijednost funkcije  $\text{ens}(\cdot)$ . Vrijeme izvršavanja algoritma je reda veličine  $O(n^2)$ .

Sljedeći cilj je dokazati da energija koju prima bilo koji izvor svjetla neće premašiti određenu konstantu.

Neka je  $P_i^<$  svjetlo primljeno od svih izvora svjetla  $s_j$  gdje je  $j < i$  (sa izvora svjetla lijevo od  $s_i$ ),  $P_i^=$  svjetlo primljeno od izvora svjetla  $s_i$ ,  $P_i^>$  svjetlo primljeno od svih izvora svjetla  $s_j$  gdje je  $j > i$  (sa izvora svjetla desno od  $s_i$ ).

**Primjedba 3.2.** *Iz Algoritma 3.3 vidimo da će izvor svjetla  $s_i$  biti korišten ako i samo ako točka  $p_i$  nije primila dovoljno svjetla sa izvora kojima je energija već pridružena tijekom algoritma. U terminima prethodnih oznaka vrijedi da je*

$$P_i^= > 0 \Leftrightarrow P_i^< < 1.$$

**Lema 3.9.**  $P_i^> < 4$

**Dokaz.** Bez smanjenja općenitosti pretpostavimo da sve točke  $p_j$  za  $j > i$  leže desno od  $p_i$  (ovo moramo pretpostaviti jer smo u Algoritmu 3.2 vršili sortiranje). Iz Leme 3.8 i činjenice  $|p_i s_j| + |s_j p_j| \geq |p_i p_j|$  za svaki  $1 \leq i, j \leq k$  slijedi

$$\begin{aligned}
|p_i s_j| &\geq |p_i p_j| - |s_j p_j| = |p_i p_j| - \text{ens}(p_j) \\
&= \left( \sum_{l=i}^{j-1} |p_l p_{l+1}| \right) - \text{ens}(p_j) \\
&\geq \left( \sum_{l=i}^{j-1} \alpha \cdot \max \{ \text{ens}(p_l), \text{ens}(p_{l+1}) \} \right) - \text{ens}(p_j) \\
&= \alpha \left( \sum_{l=i}^{j-1} \text{ens}(p_l) \right) - \text{ens}(p_j). \tag{3.11}
\end{aligned}$$

Pretpostavimo da je  $\alpha \geq 2$  te da je  $x = \sum_{l=i}^{j-1} \text{ens}(p_l)$  i  $y = \text{ens}(p_j)$ . Očigledno je  $x \geq y$  budući je  $\text{ens}(p_j) \leq \text{ens}(p_l)$  za  $l \geq j$ . Sada je očigledno

$$\begin{aligned}
(\alpha \cdot x - y)^2 &\geq (2x - y)^2 = (x + (x - y))^2 \\
&= x^2 + 2x(x - y) + y^2 \geq x^2
\end{aligned}$$

odakle slijedi da je  $\left( \alpha \left( \sum_{l=i}^{j-1} \text{ens}(p_l) \right) - \text{ens}(p_j) \right)^2 \geq \left( \sum_{l=i}^{j-1} \text{ens}(p_l) \right)^2$ . Iz (3.11) slijedi da je

$$|p_i s_j|^2 \geq \left( \sum_{l=i}^{j-1} \text{ens}(p_l) \right)^2. \tag{3.12}$$

Sada ćemo odrediti ocjenu za  $P_i^>$ . Iz Algoritma 3.3 vidimo da je svakom izvoru svjetla  $s_i$  pridruženo najviše  $\text{ens}(p_i)^2$  energije (jer je  $s_i$  izvor svjetla najbliži točki  $p_i$ ), tj.  $x_i \leq \text{ens}(p_i)^2$ . Kako je  $\text{ens}(p_l) \geq \text{ens}(p_j)$  za  $i \leq l \leq j - 1$ , imamo da je

$$\begin{aligned}
P_i^> &= \sum_{j=i+1}^k \frac{x_j}{|p_i s_j|^2} \leq \sum_{j=i+1}^k \frac{\text{ens}(p_j)^2}{\left( \sum_{l=i}^{j-1} \text{ens}(p_l) \right)^2} \\
&\leq \sum_{j=i+1}^k \frac{\text{ens}(p_j)^2}{(j-i)^2 \text{ens}(p_j)^2} = \sum_{j=i+1}^k \frac{1}{(j-i)^2} \\
&= \sum_{j=1}^{k-i} \frac{1}{j^2} \leq \sum_{j=1}^{\infty} \frac{1}{j^2} = \frac{\pi^2}{6} < 2 \tag{3.13}
\end{aligned}$$

Posljednja jednakost u literaturi poznata je kao rješenje *Baselovog problema*. Postoji više načina rješavanja ovog problema a svi oni koriste različite pristupe i različite matematičke

alate. Prvi je L. Euler<sup>14</sup> dao rješenje ovog problema 1735. godine. Pored dokaza koji koriste napredna znanja matematičke analize i analitičke teorije brojeva, postoje i dokazi koji ne koriste ništa više od elementarne matematike. Budući da ovo izlazi iz opsega rada više pogledati u [1], str. 35.

Množenjem ocjene iz (3.13) s 2 dobivamo tvdnju leme za sve  $p_j$ . □

Slično kao i u prethodnoj lemi, u nastavku dajemo gornju među za ukupnu energiju koja u točki  $p_i \in G_P$  dolazi iz svih izvora svjetla koji se nalaze lijevo od  $s_i$ .

**Lema 3.10.**  $P_i^< \leq 6$ .

**Dokaz.** Neka je  $p_j$  za  $j < i$  prva točka lijevo od  $p_i$  takva da je  $x_j > 0$ , tj. da je aktiviran  $j$ -ti izvor svjetla. Iz konstrukcija **Algoritma 3.3** slijedi da je  $P_j^= + P_j^< = 1$ . Prema **Lemi 3.9** znamo da  $p_j$  prima najviše 2 jedinice svjetla od svih izvora svjetla koji se nalaze lijevo od  $s_j$  a imaju indekse  $k > j$ . Dakle, najviše 3 jedinice svjetla stigne u  $p_i$  s lijeva a isto toliko i s desna, odakle zaključujemo da je  $P_i^< \leq 6$ . □

Koristeći prethodne dvije leme i činjenicu da je  $P_i^= \leq 1$ , dobivamo sljedeći korolar koji daje gornju i donju među na količinu svjetla koje stiže u bilo koju točku  $p_i \in G_P$ .

**Korolar 3.3.** *Ako su količine energije  $x_i$  pridružene izvorima svjetla iz  $S_P$  u skladu s **Algoritmom 3.3**, onda za svaku točku  $p_i \in G_P$  vrijedi sljedeće*

$$1 \leq P_i = P_i^< + P_i^= + P_i^> \leq 4 + 1 + 6 = 11$$

### 3.5.2. Određivanje aproksimacijskog omjera

U narednoj analizi prethodno opisanog kombinatorijalnog algoritma koristit ćemo se teorijom dualnosti za linearnu optimizaciju koja je u kratkim crtama opisana u poglavlju 2.3. na 10. stranici. Za rješenje  $\mathbf{x}^*$  koje smo dobili kao rezultat **Algoritma 3.3** odredit ćemo koliko je "udaljeno" od optimalnog osvjetljenja točaka u skupu  $G_P$ . Nakon toga se pozivamo na rezultat **Korolara 3.2** koji nam govori da je  $4(1 + \alpha)^2 \mathbf{x}^*$  dopustivo rješenje za originalni problem opisan u (3.4).

*Metoda primal-dual fitovanja*<sup>15</sup> zasniva se na odnosu PLP i DLP problema, te njihovih optimalnih vrijednosti funkcija cilja. Osnovna ideja jeste predstaviti kombinatorijalni algoritam kao iterativno popravljavanje primalnog i dualnog rješenja tako da na kraju algoritma primalno rješenje bude omeđeno sa dualnim rješenjem pomnoženom s određenom konstantom. Budući da dobiveno dualno rješenje nije dopustivo osnovni korak analize jeste skaliranje dualnog rješenja tako da ono postane dopustivo. Koristeći se teoremom o jakoj dualnosti, dobit ćemo da je faktor skaliranja ujedno i aproksimacijski odnos za naš algoritam.

Sada ćemo formulirati PLP problem za skup točaka  $G_P$  i skup izvora svjetla  $S_P$  na sljedeći način

<sup>14</sup>Leonhard Paul Euler (15. travnja 1707. – 18. rujna 1783.), švicarski matematičar

<sup>15</sup>eng. Primal-dual fitting

$$\begin{aligned} \min \quad & \sum_{s \in S_P} x_s \\ \text{uz uvjete} \quad & \sum_{s \in S_P} \frac{x_s}{d^2(p,s)} \geq 1, \quad \forall p \in G_P \\ & x_s \geq 0, \quad \forall s \in S_P. \end{aligned} \quad (3.14)$$

DLP problem koji odgovara problemu u (3.14) ima sljedeći oblik:

$$\begin{aligned} \max \quad & \sum_{p \in G_P} y_p \\ \text{uz uvjete} \quad & \sum_{p \in G_P} \frac{y_p}{d^2(p,s)} \leq 1, \quad \forall s \in S_P \\ & y_p \geq 0, \quad \forall p \in G_P. \end{aligned} \quad (3.15)$$

Slično kao i za Primjer 2.1, dualni problem u (3.15) ima svoju interpretaciju. Zamislimo da su  $y_p$  za  $p \in G_P$  težine točaka u  $G_P$  pomoću kojih možemo izraziti "utjecaj" na svaki izvor svjetla  $s$  u  $S_P$ . Želimo da ukupne težine svih točaka bude maksimalan ali tako da ukupni utjecaj svih točaka na jedan izvor svjetla bude najviše 1.

U sljedećoj lemi dajemo odnos između  $|s_i p_j|$  i  $|s_j p_i|$  nakon  $\alpha$ -reduciranja opisanog u Algoritmu 3.2.

**Lema 3.11.** *Neka su  $s_i \in S_P$  i  $p_j \in G_P$  dobiveni nakon  $\alpha$ -reduciranja. Tada vrijede sljedeće ocjene*

$$|s_j p_i| \left(1 - \frac{2}{\alpha - 1}\right) \leq |s_i p_j| \leq |s_j p_i| \left(1 + \frac{2}{\alpha - 1}\right).$$

**Dokaz.** Iz nejednakosti trokuta slijedi da je

$$\begin{aligned} |s_i p_j| &\leq |s_i p_i| + |p_i p_j| = |p_i p_j| + \text{ens}(p_i) \leq |p_i s_j| + |s_j p_j| + \text{ens}(p_i) \\ &= |p_i s_j| + \text{ens}(p_j) + \text{ens}(p_i) \leq |p_i s_j| + 2E, \end{aligned} \quad (3.16)$$

gdje je  $E = \max\{\text{ens}(p_i), \text{ens}(p_j)\}$ . Iz Leme 3.8 i nejednakosti trokuta slijedi da je

$$\alpha E \leq |p_i p_j| \leq |p_i s_j| + |s_j p_j| \leq |p_i s_j| + E$$

što daje

$$E \leq \frac{|p_i s_j|}{\alpha - 1}. \quad (3.17)$$

Koristeći (3.16) i (3.17) dobivamo da je

$$|s_i p_j| \leq |s_j p_i| \left(1 + \frac{2}{\alpha - 1}\right).$$

Lijevu nejednakost dobivamo analogno.  $\square$

Nije teško uočiti da je odnos  $1 + \frac{2}{\alpha - 1} \leq 2$  za  $\alpha \geq 3$ , tj. da je  $|s_i p_j|$  dulji od  $|s_j p_i|$  najviše dva puta. Uzmemo li da je  $y_i$  količina svjetla koje stiže u točku  $p_i \in G_P$  dobit ćemo nedopustivo rješenje za DLP problem u (3.15). Sljedeća lema pokazuje da skaliranje rješenja  $\mathbf{y}$  dobivamo dopustivo rješenje DLP problema.

**Lema 3.12.** *Neka je  $\mathbf{x}^*$  rješenje PLP problema u (3.14) dobiveno Algoritmom 3.3 te neka je  $c_P^*$  vrijednost funkcije cilja za to rješenje. Tada postoji dopustivo rješenje  $\mathbf{y}^*$  DLP problema u (3.15) s vrijednošću funkcije cilja  $c_D^*$  tako da vrijedi*

$$c_D^* \geq \frac{1}{44}c_P^*.$$

**Dokaz.** Dokaz je konstruktivan. Pretpostavimo da postoji  $\beta > 0$  takav da je  $y_i^* = \beta x_i^*$ . Treba odrediti  $\beta$  tako da  $\mathbf{y}^*$  bude dopustivo rješenje DLP problema u (3.15). Neka je  $s_j \in S_P$  i  $p_i \in G_P$ . Tada je

$$\sum_{i=1}^m \frac{y_i^*}{|s_j p_i|^2} = \beta \sum_{i=1}^m \frac{x_i^*}{|s_j p_i|^2}. \quad (3.18)$$

Iz Leme 3.11 slijedi da je

$$|s_i p_j| \leq |s_j p_i| \left(1 + \frac{2}{\alpha - 1}\right),$$

odnosno

$$|s_j p_i| \geq \frac{|s_i p_j|}{\left(1 + \frac{2}{\alpha - 1}\right)} \geq \frac{|s_i p_j|}{2}. \quad (3.19)$$

Nadalje, u Korolaru 3.3 imali smo da je

$$P_i = \sum_{i=1}^m \frac{x_i^*}{|s_i p_j|^2} \leq 11. \quad (3.20)$$

Koristeći se relacijama (3.19) i (3.20) u (3.18) dobivamo da je

$$\begin{aligned} \sum_{i=1}^m \frac{y_i^*}{|s_j p_i|^2} &= \beta \sum_{i=1}^m \frac{x_i^*}{|s_j p_i|^2} \leq \beta \left(1 + \frac{2}{\alpha - 1}\right) \sum_{i=1}^m \frac{x_i^*}{|s_i p_j|^2} \\ &= \beta \left(1 + \frac{2}{\alpha - 1}\right) P_i \leq 11\beta \left(1 + \frac{2}{\alpha - 1}\right) \leq 1. \end{aligned}$$

Iz posljednje nejednakosti slijedi da je

$$\beta \leq \frac{1}{11 \left(1 + \frac{2}{\alpha - 1}\right)}.$$

Dovoljno je uzeti  $\alpha = 3$ . Tada imamo da je  $\beta = \frac{1}{44}$ , odnosno da je  $\mathbf{y}^* = \frac{1}{44}\mathbf{x}^*$  dopustivo rješenje u DLP problemu. Nadalje, za  $\alpha \geq 3$  imamo da je  $y_i^* \geq \frac{1}{44}x_i^*$  što daje

$$c_D^* = \sum_{i=1}^m y_i^* \geq \frac{1}{44} \sum_{i=1}^m x_i^* = \frac{1}{44}c_P^*.$$

□

Neka je  $c_P^{OPT}$  optimalno rješenje PLP problema i  $c_D^{OPT}$  optimalno rješenje DLP problema. Budući smo u Lemi 3.12 pokazali da je moguće konstruirati dopustivo rješenje  $c_D^*$  u DLP problemu takvo da je  $c_D^* \geq \frac{1}{44}c_P^*$ , iz Teorema 2.2 o jakoj dualnosti slijedi da je

$$\frac{1}{44}c_P^* \leq c_D^* \leq c_D^{OPT} = c_P^{OPT} \leq c_P^*.$$

Odatle vidimo da je  $c_P^* \leq 44c_P^{OPT}$ , odnosno da je Algoritam 3.3 44-aproksimacijski algoritam za reducirani skup točaka  $G_P$ . To znači da  $\mathbf{x}_P^*$  nije "lošije" više od 44 puta u odnosu na optimalno rješenje koje dobivamo rješavanjem PLP problema u (3.14). Iz optimalnog rješenja za reducirani skup točaka  $G_P$  možemo dobiti dopustivo rješenje za originalni problem u (3.4) koje je veće od optimalnog za najviše faktor  $4(1 + \alpha)^2$ . Rezultate prethodne analize iskazat ćemo u sljedećem teoremu.

**Teorem 3.2.** *Neka je  $L$  scena i  $S$  skup od  $n$  izvora svjetla. Postoji kombinatorijalni algoritam koji u  $O(n^2)$  vremenu izračunava količine energije za sve izvore svjetla tako da svaka točka na sceni  $L$  primi najmanje jednu jedinicu svjetla. Aproksimacijski omjer je konstantan, tj. dobiveno rješenje je najviše  $O(1)$  puta veće od optimalnog.*

## 4. Implementacija aproksimacijskih algoritama

### 4.1. Wolframove demonstracije

U ovom poglavlju opisat ćemo vizualizaciju problema optimalnog osvjetljenja scene u okviru projekta *Wolframove demonstracije*<sup>16</sup>. Projekt *Wolframove demonstracije* podrazumijeva izgradnju baze besplatnih *Mathematica* kodova za ilustraciju vrlo širokog spektra različitih problema na području znanosti, tehnologije, matematike, umjetnosti, financija i mnogih drugih polja.

Korisnicima i znalcima programiranja u Mathematici, kao i onima koji se bave znanstveno-istraživačkim radom, pruža se mogućnost interaktivne vizualizacije složenih problema na relativno jednostavan način. Osnovna prednost demonstracija leži u mogućnosti promjene ulaznih parametara i evaluaciji izlaza u stvarnom vremenu. Demonstracije se mogu koristiti kao pomoćno sredstvo u nastavi matematike tako što omogućuju lakše razumijevanje određenih pojmova te brže i efikasnije rješavanje zadataka.

Jednostavno i pregledno programiranje u nekoliko linija koda omogućava izgradnju manjih aplikacija koje se mogu objaviti na web stranici projekta<sup>17</sup>. Aproksimacijski algoritmi za optimalno osvjetljenje scene koji su opisani u prethodnom poglavlju implementirani su u okviru Wolframove demonstracije **Stage Illumination Problem** koja je objavljena na stranici projekta (vidi [14]). U nastavku slijedi detaljan opis demonstracijskog projekta.

### 4.2. Opis projekta

Kao i u prethodnom poglavlju, bavit ćemo se samo dvodimenzionalnim slučajem gdje je scena predstavljena dužinom, a izvori svjetla točkama u ravnini. Naš zadatak je odrediti optimalnu potrošnju energije za svaki izvor svjetla tako da ukupno potrošena energija bude minimalna a svaka točka na sceni dobije dovoljno svjetla. U prethodnoj cjelini, preciznije u Teoremu 3.1 i Korolaru 3.1, vidjeli smo da naš problem možemo riješiti  $(1 + \epsilon)$ -aproksimacijskom shemom i 4-aproksimacijskim algoritmom. Korištenjem ovih algoritama implementirat ćemo problem tako da svakoj točki (izvoru svjetla) možemo mijenjati položaj nakon čega se mijenja i optimalna energija pridružena tom skupu točaka.

Demonstracijski projekt ćemo podijeliti u nekoliko ključnih cjelina:

- Moduli
- Kontrolno sučelje
- Grafičko sučelje

---

<sup>16</sup>eng. Wolfram Demonstrations Project

<sup>17</sup>Wolfram Demonstrations Project - <http://demonstrations.wolfram.com/>

### 4.2.1. Moduli

Mathematica pruža mogućnost definiranja funkcija koje se mogu pozivati iz različitih dijelova koda. Prilikom poziva modula prosljeđujemo proizvoljan broj argumenata koji se unutar modula ponašaju kao lokalne varijable (vidi [7], str. 178). Rezultati računanja u modulu mogu biti pohranjeni u jednu varijablu. Ako želimo da jedan modul vraća više varijabli, moguće je koristiti liste. Elementi liste bit će varijable koje želimo vratiti. Ovaj pristup često je korišten u projektu.

U projektu su korišteni sljedeći moduli

- `intersection`
- `roundToSigDigit`
- `energyOnStagePoint`
- `energyOnStage`
- `LPSol`
- `lightsourceimage`

Modul `intersection` računa priesjek pravca zadanog s dvije točke koje su pohranjene unutar liste `x` s pravcem  $y = 0$  na kojem leži scena. Pravac zadan s točkama u `x` predstavlja brid Voronoijeve ćelije za izvor svjetla `i`. Ukoliko je  $x$ -koordinata presječne točke veća od nule i manja od duljine scene, brid Voronoijeve ćelije siječe scenu  $L$ . U tom slučaju koordinatu  $x$ -koordinatu presjeka stavljamo na listu `iP`, a indeks izvora svjetla `i` stavljamo na listu `sS` (vidi Sliku 4.1).

```
intersection[x_, i_, iPoint_, sStage_] :=
  Module[{xkor, iP = iPoint, sS = sStage},
    xkor = (x[[1, 1]]*x[[2, 2]] - x[[2, 1]]*x[[1, 2]])/(x[[2, 2]] -
      x[[1, 2]]);
    If[0 <= xkor <= 40, iP = Append[iP, xkor];
      sS = Append[sS, i];];
    {iP, sS}];
```

Slika 4.1: Modul `intersection`

Modul `roundToSigDigit` koristimo za zaokruživanje argumenta na određeni broj decimalnih znamenki. Primjerice, poziv `roundToSigDigit[12.543367, 4]` vraća `12.5434`. Potreba za zaokruživanjem pojavila se prilikom formatiranja prikaza određenih numeričkih podataka u grafičkom sučelju.

Modul `energyOnStagePoint` računa ukupnu količinu svjetla koja stiže u točku `stagePoint` sa izvora svjetla na pozicijama `pos` i energijama `optimalEnergies`.

```

energyOnStage[stagePoint_, optimalEnergies_, pos_] :=
  Module[{stagePointEnergy, p = pos},
    stagePointEnergy =
  Table[Total[
    optimalEnergies/(Norm /@ (p -
      Table[stagePoint[[i]], {Length[p]}))2 // N, {i,
    Length[stagePoint]}]];

```

Slika 4.2: Modul energyOnStage

```

LPSol[pos_, voron_, \[Epsilon]_, alg_]

```

Slika 4.3: Potpis modula LPSol

Modul `energyOnStage` ima isti zadatak kao i modul `energyOnStagePoint` s razlikom da računa ukupne količine svjetla na zadanom skupu točaka `stagePoint` (vidi Sliku 4.2).

Modul `LPSol` je osnovni modul u projektu.

Unutar `LPSol` modula riješeni su sljedeći zadaci:

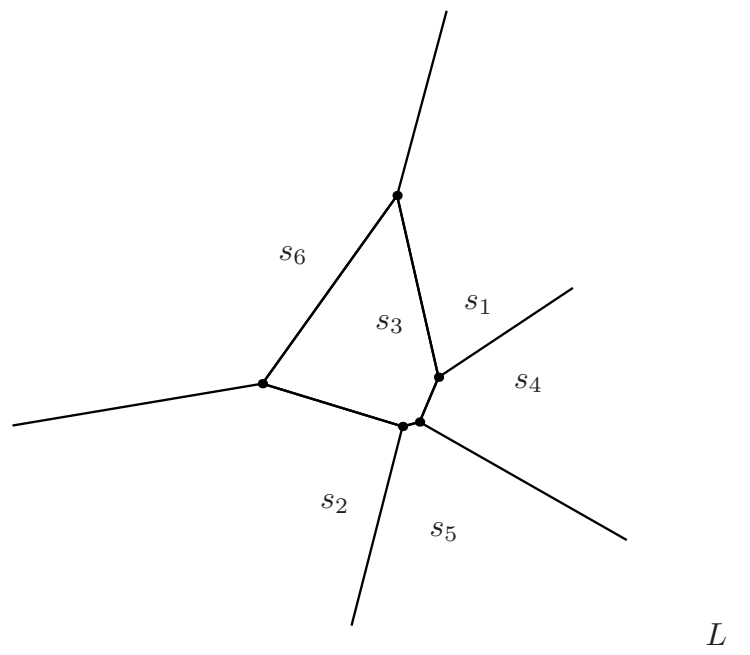
- U skladu s rezultatima Leme 3.1 identificirani su izvori svjetla čije Voronoijeve ćelije sijeku scenu  $L$ .
- Scena  $L$  je podijeljena na segmente i za svaki segment određen je  $\epsilon$ -dobar skup gdje je  $\epsilon > 0$ .
- Nakon konstrukcije  $\epsilon$ -dobrog skupa  $G$  (prema Lemi 3.3), formuliran je LP problem i određene optimalne energije za izvore svjetla tako da točke u skupu  $G$  primaju dovoljno svjetla

U identifikaciji izvora svjetla čije Voronoijeve ćelije ne sijeku scenu  $L$  korišten je paket `ComputationalGeometry`. Modul prima 4 argumenta (vidi Sliku 4.3). Prvi argument `pos` je lista sa koordinatama točaka koje predstavljaju izvore svjetla. Drugi argument `voron` je lista koja sadrži rezultate funkcije `VoronoiDiagram` pozvane na skupu točaka `pos`. Unutar liste `voron` imamo dvije podliste. U prvoj podlisti nalaze se dva tipa objekata. Prvi tip objekata su dvočlane liste koje predstavljaju koordinate vrhova Voronoijevog dijagrama. Drugi tip su `Ray` objekti koji predstavljaju zrake Voronoijevih ćelija. To su polupravci kojima je vrh Voronoijevog dijagrama ishodišna točka. Zrake se pojavljuju u neomeđenim Voronoijevim ćelijama. Objekt `Ray` koji predstavlja zraku sadrži dvije dvočlane liste. Prva dvočlana lista predstavlja ishodište zrake i to je svakako jedna od koordinata Voronoijevih vrhova. Druga točka određuje polupravac (smjer). Prvo su navedene koordinate Voronoijevih vrhova a zatim `Ray` tako da je moguće utvrditi indekse (redne brojeve elementa u listi) vrhova i zraka. Druga lista ima onoliko podlista koliko ima Voronoijevih ćelija. Svaka podlista sadrži indeks izvora

svjetla koji generira Voronoijevu ćeliju i listu u kojoj se nalaze indeksi Voronoijevih vrhova. To su vrhovi i zrake koji su incidentni toj Voronoijevoj ćeliji. Navedeni su u suprotnom smjeru od kazaljke na satu (vidi Sliku 4.4).

```
<< ComputationalGeometry`
pos = RandomReal[{1, 10}, {6, 2}];
{voronVertices, voronCel} = VoronoiDiagram[pos]

{{{1.17383, 6.04237}, {4.11488, 10.151}, {4.23474, 5.10748}, {4.60671,
  5.20522}, {5.01962, 6.18202},
  Ray[{5.01962, 6.18202}, {7.93335, 8.119}],
  Ray[{4.11488, 10.151}, {5.19107, 14.1598}],
  Ray[{4.23474, 5.10748}, {3.126, 0.782938}],
  Ray[{1.17383, 6.04237}, {-4.25855, 5.13202}],
  Ray[{4.60671, 5.20522}, {9.10893, 2.6399}]}}, {{1, {2, 5, 6,
  7}}, {2, {3, 1, 9, 8}}, {3, {1, 3, 4, 5, 2}}, {4, {5, 4, 10,
  6}}, {5, {4, 3, 8, 10}}, {6, {1, 2, 7, 9}}}}
```



Slika 4.4: Voronoijev dijagram za skup od 6 točaka

Koristeći takvu strukturu Voronoijevog dijagrama moguće je odrediti izvore svjetla čiji Voronoijeve ćelije sijeku scenu  $L$ . Osnovna ideja algoritma je da procesira svaku Voronoijevu ćeliju posebno, da utvrdi da li ona sadrži bridove (generirane s dva susjedna Voronoijeva vrha) i/ili zrake koji eventualno sijeku scenu. Na osnovu krajnjih vrhova jednog brida moguće je utvrditi da li on siječe scenu  $L$ . Ukoliko su  $y$ -koordinate vrhova suprotnog predznaka, pozivamo modul `intersection`. Kod zraka utvrđujemo predznak  $y$ -koordinate vektora koji

određuje smjer zrake i predznak  $y$ -koordinate ishodišne točke. Promatramo samo one zrake kod kojih su ta dva predznaka suprotna. Kada pronađemo takvu zraku, pozivamo modul `intersection` (vidi Sliku 4.5). Kada prethodnu proceduru izvršimo na svakoj Voronoijevoj ćeliji, u listi `sS` nalaze se indeksi izvora svjetla čije Voronoijeve ćelije sijeku scenu  $L$ , a u listi `iP` nalaze se  $x$ -koordinate presječnih točaka ( $y$ -koordinate su 0 jer scena  $L$  leži na pravcu  $y = 0$ ).

```

For[i = 1, i <= numS, i++,
  {For[j = 1, j <= Length[dC[[i, 2]]], j++,
    {
      If[j == Length[dC[[i, 2]]], m = 1, m = j + 1];
      If[dC[[i, 2, j]] <= numberOfVertices,
        If[dC[[i, 2, m]] <= numberOfVertices,

          If[Sign[vr[[dC[[i, 2, j]], 2]]] == -Sign[
            vr[[dC[[i, 2, m]], 2]]],
            {iP, sS} =
              intersection[{vr[[dC[[i, 2, j]]]], vr[[dC[[i, 2, m]]]]}, i,
                iP, sS]],
              { If[vr[[dC[[i, 2, m]], 1, 2]] < 0,

                If[vr[[dC[[i, 2, m]], 2, 2]] > vr[[dC[[i, 2, m]], 1, 2]],
                  {iP, sS} =
                    intersection[vr[[dC[[i, 2, m]]]], i, iP, sS]],
                    If[vr[[dC[[i, 2, m]], 1, 2]] == 0,
                      {iP, sS} =
                        intersection[vr[[dC[[i, 2, m]]]], i, iP, sS],

                    If[vr[[dC[[i, 2, m]], 2, 2]] < vr[[dC[[i, 2, m]], 1, 2]],
                      {iP, sS} =
                        intersection[vr[[dC[[i, 2, m]]]], i, iP, sS]]]]},
                If[vr[[dC[[i, 2, j]], 1, 2]] < 0,

                If[vr[[dC[[i, 2, j]], 2, 2]] > vr[[dC[[i, 2, j]], 1, 2]],
                  {iP, sS} =
                    intersection[vr[[dC[[i, 2, j]]]], i, iP, sS]],
                    If[vr[[dC[[i, 2, j]], 1, 2]] == 0,
                      {iP, sS} =
                        intersection[vr[[dC[[i, 2, j]]]], i, iP, sS],

                    If[vr[[dC[[i, 2, j]], 2, 2]] < vr[[dC[[i, 2, j]], 1, 2]],
                      {iP, sS} =
                        intersection[vr[[dC[[i, 2, j]]]], i, iP, sS]]]]
            ]
          }
        ]
      }
    ]
  }
]

```

Slika 4.5: Algoritam za prepoznavanje izvora svjetla čije Voronoijeve ćelije sijeku scenu  $L$

Implementacija Algoritma 3.1 za konstrukciju  $\epsilon$ -dobrog skupa ostvarena je u modulu `discretizeStage` (vidi Sliku 4.6). Argumenti `a` i `b` označavaju granice segmenta na sceni  $L$  koji se nalazi u Voronoijevoj ćeliji generiranoj izvorom svjetla  $s$  koji ima koordinate `sx` i `sy`.

```
discretizeStage[{a_, b_, sx_, sy_}] :=
Module[{dSt = {}, n}, dSt = Append[dSt, a];
n = First[dSt] +
2*\[Epsilon]*EuclideanDistance[{First[dSt], 0}, {sx, sy}];
While[n <= b,
{ dSt = Append[dSt, n];
n = Last[dSt] +
2*\[Epsilon]*EuclideanDistance[{Last[dSt], 0}, {sx, sy}];
}]; dSt];
```

Slika 4.6: Modul `discretizeStage` za konstrukciju  $\epsilon$ -dobrog skupa

U varijabli `dStage` nalaze se  $x$ -koordinate točaka koje pripadaju  $\epsilon$ -dobrom skupu  $G$  definiranom u Definiciji 3.3. U rješavanju LP problema koristili smo *Mathematica* metodu `LinearProgramming` (iscrpan opis ove metode može se naći u [6], str. 4). Argumenti metode su matrica  $A \in \mathbb{R}^{|G| \times n}$  gdje je  $n$  broj izvora svjetla, vektori  $b$  i  $c$  koji odgovaraju formulaciji LP problema u (2.3) s tim da dodajemo uvjete nenegativnosti  $\mathbf{x} \geq \mathbf{0}$ . Na osnovu (3.4) vidimo da je

$$[A]_{ij} = \frac{1}{d^2(s_j, p_i)},$$

gdje je  $p_i \in G$ ,  $c = [1 \ 1 \dots 1]^T \in \mathbb{R}^n$  i  $b = [1 \ 1 \dots 1]^T \in \mathbb{R}^{|G|}$ .

```
A = Table[1/((pos[[j, 1]] - Flatten[dStage][[i]])^2 +
pos[[j, 2]]^2), {i,
Length[Flatten[dStage]]}, {j, numS}];
b = Table[1, {Length[Flatten[dStage]]}];
c = Table[1, {numS}];
optimalEnergies = LinearProgramming[c, A, b];
```

Slika 4.7: Formulacija i rješavanje LP problema

U modulu `LPSol` posebno je obrađen slučaj kada ne postoji presjek Voronoijevog dijagrama i scene  $L$ , tj. kada se scene  $L$  u potpunosti nalazi u točno jednoj Voronoijevoj ćeliji. Tada postoji samo jedan izvor svjetla koji uzimamo u razmatranje. U tom slučaju izračunavanje optimalne energije možemo provesti i bez rješavanja LP problema (vidi Dodatak A1. na 46. stranici).

Argument `\[Epsilon]` određuje vrijednost parametra  $\epsilon$  kojeg koristimo za konstrukciju skupa  $G$ . Argument `alg` određuje koji aproksimacijski algoritam koristimo za rješavanje problema. Ukoliko je vrijednost ovog argumenta jednaka 1 koristit ćemo  $(1 + \epsilon)$ -aproksimacijsku shemu, u suprotnom koristimo 4-aproksimacijski algoritam.

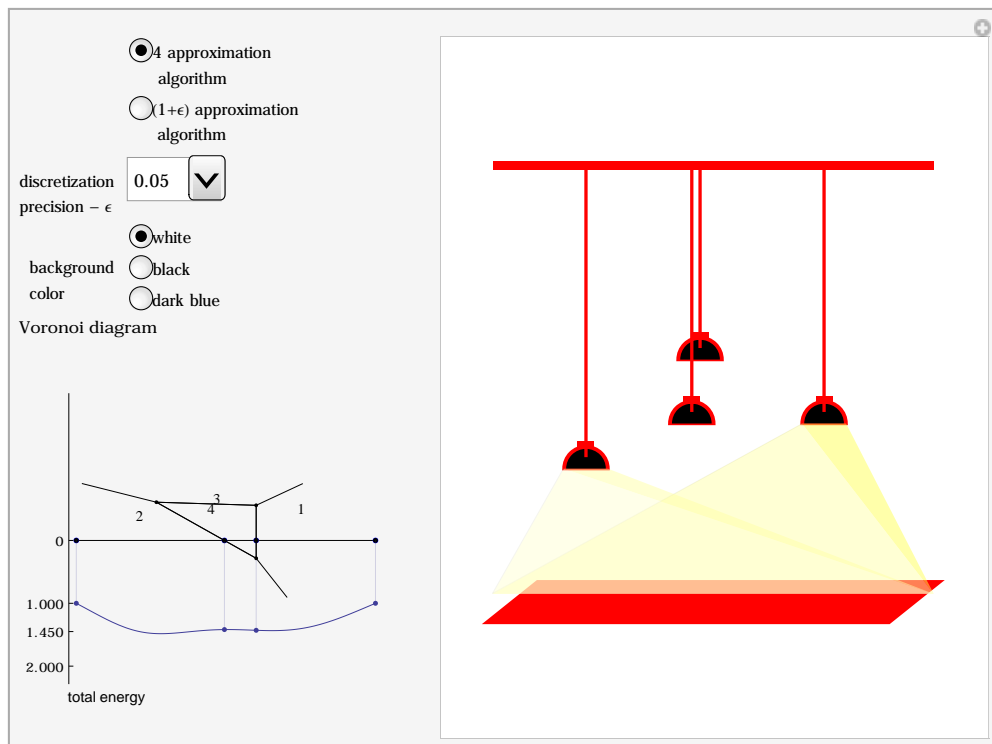
Modul `lightsourceimage` vraća grafičke objekte koji vizualiziraju izvore svjetla sa svjetlosnim zrakama čiji intenzitet ovisi o optimalnim količinama energije koje smo izračunali pomoću modula `LPSol`. Programski kod modula može se pronaći u Dodatku A2. na 49. stranici.

#### 4.2.2. Kontrolno sučelje

Osnovno okruženje koje objedinjuje gore opisane module i omogućava njihovo dinamičko funkcioniranje predstavlja *Mathematica*-metoda `Manipulate`<sup>18</sup> (vidi [8], str. 44). Pomoću padajućih izbornika i kliznih gumbića omogućava dinamičku manipulaciju ulaznim parametrima što pokreće trenutnu reevaluaciju koda koji se nalazi u `Manipulate`-u.

U projektu *Stage Illumination problem* korišteni su radio-gumbovi za odabir željenog aproksimacijskog algoritma. Primjerice, na Slici 4.8 vidimo da je odabran 4-aproksimacijski algoritam.

```
Manipulate[... ,
  {{ctrl, 1, ""}, {0 -> "4 approximation\n algorithm",
    1 -> "(1+\[Epsilon]) approximation\n algorithm"}},
  Appearance -> "Vertical", ControlType -> RadioButton},...]
```



Slika 4.8: Radio-gumbi za odabir algoritma

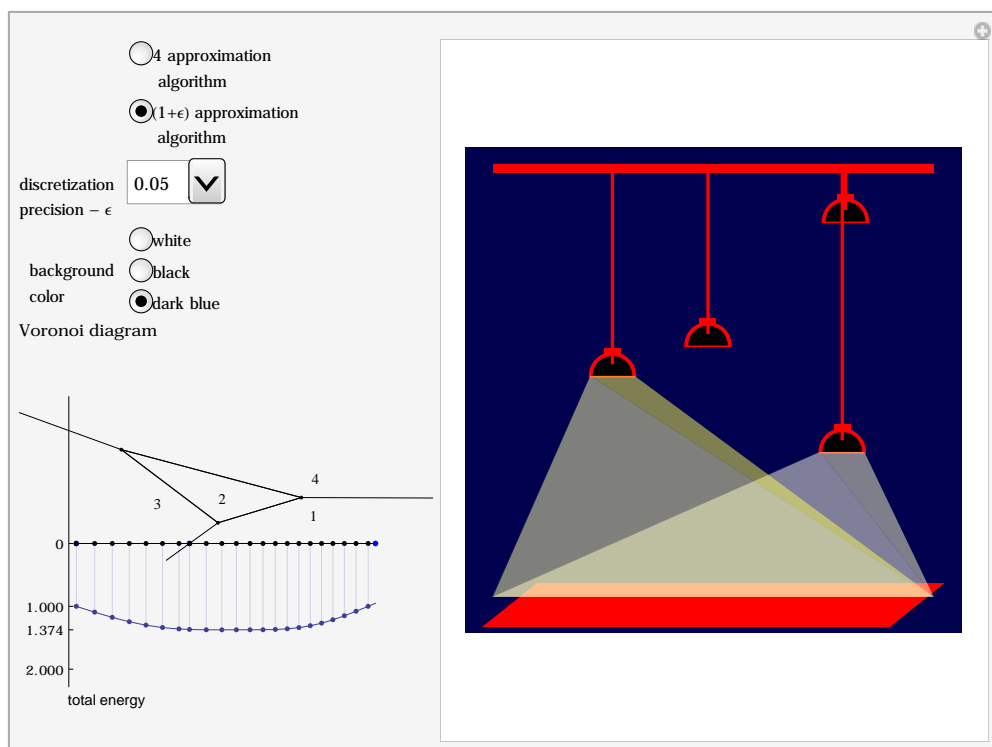
<sup>18</sup>javlja se u verzijama *Mathematica 6.0* i *Mathematica 7.0*

Parametar  $\epsilon$  za određivanje skupa  $G$  možemo odabrati u padajućem izborniku discretization precision (vidi Sliku 4.9). Na Slici 4.8 vidimo da je  $\epsilon = 0.05$ .

```
Manipulate[...,...
  {{\[Epsilon], 0.05, "discretization\nprecision - \[Epsilon]"}, {0.01,
    0.02, 0.03, 0.04, 0.05, 0.1}},...]
```

Slika 4.9: Padajući izbornik za odabir konstante  $\epsilon$

Boja pozadine može se biti bijela, plava i crna. Promjena boje regulirana je radiogumbima (vidi Sliku 4.10).



Slika 4.10: Promjena boje pozadine

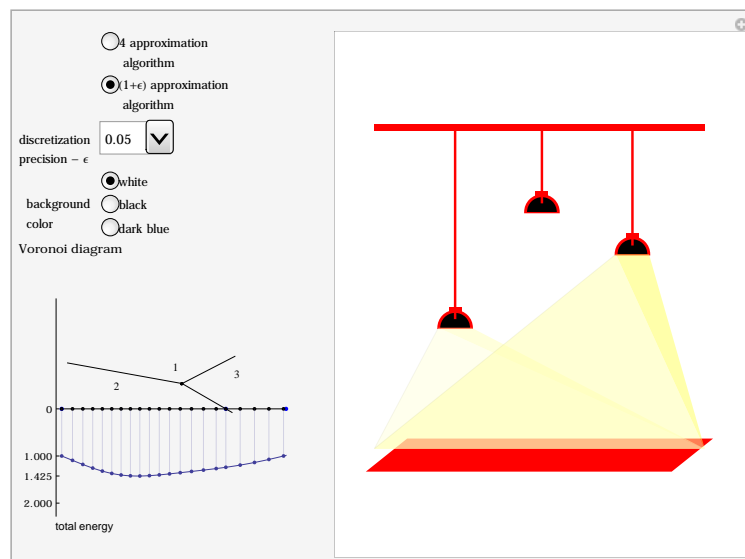
U drugom dijelu kontrolnog sučelja prikazan je Voronoijev dijagram za trenutnu konfiguraciju scene i položaj izvora svjetla. Također su prikazane sve točke iz skupa  $G$  (ili skupa  $G_V$  ako se radi o 4-aproksimacijskom algoritmu, vidi Lemu 3.6) sa pripadnom količinom svjetla, a na  $y$ -osi prikazana je i maksimalna količina svjetla na sceni (vidi Dodatak A3. na 50. stranici). Primjerice, na Slici 4.8 vidimo da je maksimalna količina svjetla 1.45.

### 4.2.3. Grafičko sučelje

Grafičko sučelje omogućava vidljivu predstavu scene i izvora svjetla. Svaki izvor svjetla prikazan je u obliku svjetiljke koja emitira svjetlosne zrake na scenu. Iako fizikalni model

pretpostavlja širenje svjetla u svim smjerovima, ovdje smo radi bolje vizualizacije prikazali samo dio svjetla koje dolazi na scenu. Svakom svjetlu možemo mijenjati položaj. Možemo dodati svjetlo ili ukloniti postojeće. Svako svjetlo predstavljeno je *Locator* objektom (vidi [8], str. 68) čije se koordinate prilikom pomicanja mijenjaju dinamički. Za svaki *Locator* objekt poziva se modul *lightsourceimage* koji na odgovarajućoj poziciji crta svjetiljku i svjetlosne zrake (vidi Sliku 4.11).

```
Manipulate[
  {optEnerg, inten, dStage, discreteEnergies, continEnergies, ticks,
   voronDiag} =
  Join[LPSol[p, ComputationalGeometry`VoronoiDiagram[p], \[Epsilon],
    ctrl], {ComputationalGeometry`DiagramPlot[p]}}];
  intenround =
  Table[If[Round[1000*inten[[i]]] == 0, 1, 1 - inten[[i]]], {i,
    Length[inten]}}];
  Show[Graphics[{Red,
    Polygon[{{-1, -2}, {4, 2}, {41, 2}, {36, -2}, {-1, -2}}]},
  Table[lightsourceimage[intenround[[i]], p[[i]], optEnerg[[i]], {i,
    Length[p]}]
  , Graphics[{Red,
    Polygon[{{0, 0.98*40}, {0, 40}, {40, 40}, {40, 0.98*40}, {0,
      0.98*40}}]}],
  Table[Graphics[{Thick, Red,
    Line[{{p[[i, 1]], 0.99*40}, {p[[i, 1]], p[[i, 2]]}}]}, {i,
    Length[p]}], PlotRange -> {{-1, 41}, {-1, 40}}, ImageSize -> 300,
  ImagePadding -> 10, Background -> backcolor],...]
```



Slika 4.11: Grafičko sučelje i *Locator* objekti

## A Dodaci

### A1. Modul LPSol

```

LPSol[pos_, voron_, \[Epsilon]_, alg_] :=
Module[{vr, dC, rays, numberOfVertices, sS = {}, iP = {},
  numS = Length[pos], i, j, m, dPr, sSCoor, dStage, sInt, numOfP, A,
  b,
  c, minD, minin, pom, maxener, intenz, dStageGraphics,
  optimalEnergies, enerPoint, enmax, discreteEnergies,
  continEnergies, ticks, ceilEnmax, floorEnmax, sig4Enmax,
  nCeilEnmax, nFloorEnmax, dstagePoints},
  {vr, dC} = voron;
rays =
  Map[{-#[[1]], #[[2]]} &,
    Select[Flatten[vr], Dimensions[#] == {2} &]];
numberOfVertices = Dimensions[vr][[1]] - Dimensions[rays][[1]];
For[i = 1, i <= numS, i++,
  {For[j = 1, j <= Length[dC[[i, 2]]], j++,
    {
      If[j == Length[dC[[i, 2]]], m = 1, m = j + 1];
      If[dC[[i, 2, j]] <= numberOfVertices,
        If[dC[[i, 2, m]] <= numberOfVertices,

If[Sign[
  vr[[dC[[i, 2, j]], 2]] == -Sign[vr[[dC[[i, 2, m]], 2]]],
  {iP, sS} =
intersection[{vr[[dC[[i, 2, j]]], vr[[dC[[i, 2, m]]]}], i,
iP, sS]],
  { If[vr[[dC[[i, 2, m]], 1, 2]] < 0,

If[vr[[dC[[i, 2, m]], 2, 2]] > vr[[dC[[i, 2, m]], 1, 2]],
  {iP, sS} =
intersection[vr[[dC[[i, 2, m]]], i, iP, sS]],
  If[vr[[dC[[i, 2, m]], 1, 2]] == 0,
  {iP, sS} =
intersection[vr[[dC[[i, 2, m]]], i, iP, sS],
If[vr[[dC[[i, 2, m]], 2, 2]] < vr[[dC[[i, 2, m]], 1, 2]],
  {iP, sS} =
intersection[vr[[dC[[i, 2, m]]], i, iP, sS]]]]],
  If[vr[[dC[[i, 2, j]], 1, 2]] < 0,

If[vr[[dC[[i, 2, j]], 2, 2]] > vr[[dC[[i, 2, j]], 1, 2]],
  {iP, sS} =
intersection[vr[[dC[[i, 2, j]]], i, iP, sS]],

```

```

        If[vr[[dC[[i, 2, j]], 1, 2]] == 0,
            {iP, sS} =
            intersection[vr[[dC[[i, 2, j]]]], i, iP, sS],
        If[vr[[dC[[i, 2, j]], 2, 2]] < vr[[dC[[i, 2, j]], 1, 2]],
            {iP, sS} =
            intersection[vr[[dC[[i, 2, j]]]], i, iP, sS]]]
    ]
}
]
}];
If[Length[iP] == 0, minD = EuclideanDistance[pos[[1]], {0, 0}];
minin = 1;
For[i = 2, i <= numS, i++,
    pom = EuclideanDistance[pos[[i]], {0, 0}];
    If[pom < minD, minD = pom; minin = i]];
optimalEnergies = ConstantArray[0, numS];
optimalEnergies[[minin]] =
    Max[{minD^2, EuclideanDistance[pos[[minin]], {40, 0}]^2} //
    N;
intenz = optimalEnergies/optimalEnergies[[minin]];
sInt = {0, 40};
dStage = sInt;
enerPoint =
energyOnStage[dstagePoints = {{0, 0}, {40, 0}}, optimalEnergies,
pos];
enmax = optimalEnergies[[minin]]/pos[[minin, 2]]^2;
,
If[alg == 1, sS = Union[sS];
sS = Sort[sS, pos[[#1, 1]] < pos[[#2, 1]] &];
sSCoor = pos[[sS]];
discretizeStage[{a_, b_, sx_, sy_}] :=
Module[{dSt = {}, n}, dSt = Append[dSt, a];
n = First[dSt] +
2*\[Epsilon]*EuclideanDistance[{First[dSt], 0}, {sx, sy}];
While[n <= b,
{ dSt = Append[dSt, n];
n = Last[dSt] +
2*\[Epsilon]*EuclideanDistance[{Last[dSt], 0}, {sx, sy}];
}]; dSt];
sInt = Sort[Union[Flatten[Join[{0}, iP // N, {40}]]]];
dPr = Join[ Partition[sInt, 2, 1], sSCoor, 2];
dStage = discretizeStage /@ dPr;

A = Table[

```

```

1/((pos[[j, 1]] - Flatten[dStage][[i]])^2 + pos[[j, 2]]^2), {i,
  Length[Flatten[dStage]]}, {j, numS}];
  b = Table[1, {Length[Flatten[dStage]]}];
  c = Table[1, {numS}];
  optimalEnergies = LinearProgramming[c, A, b];
  enerPoint =
energyOnStage[
  dstagePoints =
  Partition[Riffle[Union[Flatten[dStage]], 0, {2, -1, 2}], 2],
  optimalEnergies, pos];
  enmax = Max[enerPoint];,
  sS = Union[sS];
  sS = Sort[sS, pos[[#1, 1]] < pos[[#2, 1]] &];
  sSCoor = pos[[sS]];
dStage = Union[Join[{0}, iP // N, {40}]];
  sInt = dStage;
  A =

Table[1/((pos[[j, 1]] - Flatten[dStage][[i]])^2 +
  pos[[j, 2]]^2), {i,
  Length[Flatten[dStage]]}, {j, numS}];
  b = Table[1, {Length[Flatten[dStage]]}];
  c = Table[1, {numS}];
  optimalEnergies = LinearProgramming[c, A, b];
  enerPoint =

energyOnStage[
  dstagePoints =
  Partition[Riffle[Union[Flatten[dStage]], 0, {2, -1, 2}], 2],
  optimalEnergies, pos];

enmax = (*Partition[Riffle[sSCoor[[All, 1]], 0, {2, -1, 2}],
  2]*)
Max[energyOnStage[
  Partition[Riffle[sSCoor[[All, 1]], 0, {2, -1, 2}], 2],
  optimalEnergies, pos]]];
  maxener = Max[optimalEnergies];
  intenz = optimalEnergies/maxener;
];
ceilEnmax = Ceiling[enmax];
floorEnmax = Floor[enmax];
sig4Enmax = roundToSigDigit[enmax, 4];
nCeilEnmax = N[ceilEnmax, 4];
nFloorEnmax = N[floorEnmax, 4];
dStageGraphics =

```

```

Graphics[{{Black, Line[{{0, 0}, {40, 0}]}, {Blue,
  PointSize[0.018],
  Point[Table[{{sInt[[i]], 0}, {i, Length[sInt]}]}]},
  Point[dstagePoints]}, PlotRange -> {{-1, 41}, {-80, 2*40}},
  ImageSize -> 350];
discreteEnergies =
ListPlot[
  Table[{{Flatten[dStage][[i]], -(70/ceilEnmax)*enerPoint[[i]]}, {i,
    Length[Flatten[dStage]]}], PlotRange -> {{-1, 41}, {-80, 0}},
  Filling -> Axis, Axes -> {False, True}];
continEnergies =
Plot[-(70/ceilEnmax)*
  energyOnStagePoint[{x, 0}, optimalEnergies, pos], {x, 0, 40}];
ticks = {},
If[70/ceilEnmax <= 8,
  Join[{{0, 0}},
    Table[If[
      Mod[i, 2] == 0, {-i*(70/ceilEnmax), N[i, 4]}, {(70/ceilEnmax),
        Null}], {i, ceilEnmax}]],
  Join[{{0, 0}},
    Table[{-i*(70/ceilEnmax), N[i, 4]}, {i, floorEnmax - 1}],
    If[70 - (70/ceilEnmax)*enmax <= 10,
      If[(70/ceilEnmax)*enmax - (70/ceilEnmax)*floorEnmax >=
        10, {{-floorEnmax*(70/ceilEnmax),
          nFloorEnmax}, {-(70/ceilEnmax)*enmax,
            sig4Enmax}}, {{-floorEnmax*(70/ceilEnmax),
              nFloorEnmax}, {-70, nCeilEnmax}}]],
      If[(70/ceilEnmax)*enmax - (70/ceilEnmax)*floorEnmax >=
        10, {{-floorEnmax*(70/ceilEnmax),
          nFloorEnmax}, {-(70/ceilEnmax)*enmax, sig4Enmax}, {-70,
            nCeilEnmax}}, {{-(70/ceilEnmax)*enmax, sig4Enmax}, {-70,
              nCeilEnmax}}]]]]];
{optimalEnergies, intenz, dStageGraphics, discreteEnergies,
  continEnergies, ticks}
]

```

## A2. Modul lightsourceimage

```

lightsourceimage[intenz_, pos_, energ_] :=
Module[{e = energ, r = 40/20},
  If[intenz ==
    1, {Graphics[{{EdgeForm[Thick, Red]}, FaceForm[Red]},
      Disk[{pos[[1]], pos[[2]] - 0.6*r},
        r, {0, Pi}], {FaceForm[Black]},
      Tooltip[Disk[{pos[[1]], pos[[2]] - 0.6*r}, 0.9*r, {0, Pi}],

```

```

Row[{"Energy on this lamp: ", e}], {Thick, Red,
Polygon[{{pos[[1]] + r*Cos[5 \[Pi]/8],
pos[[2]] - 0.6*r + r*Sin[5 \[Pi]/8]}, {pos[[1]] +
r*Cos[5 \[Pi]/8],
pos[[2]] - 0.6*r + 1.25*r}, {pos[[1]] + r*Cos[3 \[Pi]/8],
pos[[2]] - 0.6*r + 1.25*r}, {pos[[1]] + r*Cos[3 \[Pi]/8],
pos[[2]] - 0.6*r + r*Sin[3 \[Pi]/8]}, {pos[[1]] +
r*Cos[5 \[Pi]/8],
pos[[2]] - 0.6*r +
r*Sin[3 \[Pi]/8]}]}], {Graphics[{{EdgeForm[{Thick,
Red}], FaceForm[{Red}],
Disk[{pos[[1]], pos[[2]] - 0.6*r},
r, {0, Pi}], {FaceForm[{Black}],
Tooltip[Disk[{pos[[1]], pos[[2]] - 0.6*r}, 0.9*r, {0, Pi}],
Row[{"Energy on this lamp: ", e}], {Thick, Red,
Polygon[{{pos[[1]] + r*Cos[5 \[Pi]/8],
pos[[2]] - 0.6*r + r*Sin[5 \[Pi]/8]}, {pos[[1]] +
r*Cos[5 \[Pi]/8],
pos[[2]] - 0.6*r + 1.25*r}, {pos[[1]] + r*Cos[3 \[Pi]/8],
pos[[2]] - 0.6*r + 1.25*r}, {pos[[1]] + r*Cos[3 \[Pi]/8],
pos[[2]] - 0.6*r + r*Sin[3 \[Pi]/8]}, {pos[[1]] +
r*Cos[5 \[Pi]/8],
pos[[2]] - 0.6*r + r*Sin[3 \[Pi]/8]}]}],
Graphics[Opacity[0.5],
Polygon[{{pos[[1]] - r, pos[[2]] - 0.6*r}, {pos[[1]] + r,
pos[[2]] - 0.6*r}, {40, 0.75}, {0, 0.75}, {pos[[1]] - r,
pos[[2]] - 0.6*r}},
VertexColors -> {RGBColor[1, 1, intenz],
RGBColor[1, 1, intenz], White, White}}]}]]

```

### A3. Manipulate okruženje

```

Manipulate[
{optEnerg, inten, dStage, discreteEnergies, continEnergies, ticks,
voronDiag} =
Join[LPSol[p, ComputationalGeometry[VoronoiDiagram[p], \[Epsilon],
ctrl], {ComputationalGeometry[DiagramPlot[p]}]];
intendiround =
Table[If[Round[1000*inten[[i]]] == 0, 1, 1 - inten[[i]], {i,
Length[inten]}];
Show[Graphics[{Red,
Polygon[{{-1, -2}, {4, 2}, {41, 2}, {36, -2}, {-1, -2}}]],
Table[lightsourcimage[intendiround[[i]], p[[i]], optEnerg[[i]], {i,
Length[p]}]
, Graphics[{Red,

```

```

Polygon[{{0, 0.98*40}, {0, 40}, {40, 40}, {40, 0.98*40}, {0,
  0.98*40}}]],
Table[Graphics[{{Thick, Red,
  Line[{{p[[i, 1]], 0.99*40}, {p[[i, 1]], p[[i, 2]]}}]}], {i,
  Length[p]}], PlotRange -> {{-1, 41}, {-1, 40}}, ImageSize -> 300,
ImagePadding -> 10, Background -> backcolor],
{{ctrl1, 1, ""}, {0 -> "4 approximation\n algorithm",
  1 -> "(1+\[Epsilon]) approximation\n algorithm"},
Appearance -> "Vertical", ControlType -> RadioButton},
{{\[Epsilon], 0.05, "discretization\nprecision - \[Epsilon]"}, {0.01,
  0.02, 0.03, 0.04, 0.05,
  0.1}}, {{p, {{25.81944658952817',
  21.157741196748425'}, {33.99719138743964',
  31.539576852477573'}, {26.822098747653776',
  6.275976066503482'}, {10.779151746677044',
  21.839736383556897'}}}}, {4, 5}, {36, 36}, Locator,
Appearance ->
Graphics[{{White, PointSize[0.02], Opacity[0], Point[{0, 0]}],
  ImageSize -> 30],
LocatorAutoCreate -> {1, 6}}, {{backcolor, White,
  "background\ncolor"}, {White -> "white", Black -> "black",
  RGBColor[0, 0, 0.309804] -> "dark blue"}, Appearance -> "Vertical",
ControlType -> RadioButton},
{inten, ControlType -> None},
{discreteEnergies, ControlType -> None},
{dStage, ControlType -> None},
{continEnergies, ControlType -> None},
{ticks, ControlType -> None},
{intenround, ControlType -> None},
{voronDiag, ControlType -> None},
{optEnerg, ControlType -> None},
Delimiter,
Style["Voronoi diagram", Bold],
Dynamic[
Show[dStage, discreteEnergies, continEnergies, voronDiag,
Graphics[
Text[Style["total energy", FontFamily -> "Helvetica"], {4, -88}]],
PlotRange -> {{-1, 41}, {-80, 2*41}}, PlotRangeClipping -> False,
ImageSize -> 250, Axes -> {False, True}, AxesOrigin -> {-1, 0},
Ticks -> ticks, TicksStyle -> Directive[FontSize -> 8],
LabelStyle -> ("Label"), AspectRatio -> 1/1.08,
ImagePadding -> 30], SynchronousUpdating -> False],
ControlPlacement -> Left, AutorunSequencing -> {1, 2, 3, 4},
SynchronousUpdating -> True,
SynchronousInitialization -> False, Initialization :> (...)]

```

## Literatura

- [1] M. Aigner, G. M. Ziegler, Nicolas Puech, and Jean-Marie Morvan. *Proofs from the book*. Springer, third edition, 2000.
- [2] Sanjeev Arora. Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. In *Journal of the ACM*, pages 2–11, 1996.
- [3] Marshall Bern and David Eppstein. Approximation algorithms for geometric problems. pages 296–345. PWS Publishing Company, 1995.
- [4] Dimitris Bertsimas and John. N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, Belmont, Massachusetts, 1997.
- [5] S. Boyd and L. Vanden. *Convex optimization*. Cambridge University Press, 2004.
- [6] Wolfram Mathematica Documentation Center. *Wolfram Mathematica Tutorial Collection - CONSTRAINED OPTIMIZATION*. Wolfram Research, 2008.
- [7] Wolfram Mathematica Documentation Center. *Wolfram Mathematica Tutorial Collection - CORE LANGUAGE*. Wolfram Research, 2008.
- [8] Wolfram Mathematica Documentation Center. *Wolfram Mathematica Tutorial Collection - DYNAMIC INTERACTIVITY*. Wolfram Research, 2008.
- [9] Wolfram Mathematica Documentation Center. *Wolfram Mathematica Tutorial Collection - VIZUALIZATION AND GRAPHICS*. Wolfram Research, 2008.
- [10] Edwin K. P. Chong and Stanislaw H. Zak. *An Introduction to Optimization*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley and Sons, 2001.
- [11] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to algorithms*. MIT Press, second edition, 2001.
- [12] Mark De Berg, Marc Van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational Geometry - Algorithms and Applications*. Springer, second revised edition, 2000.
- [13] Friedrich Eisenbrand, Stefan Funke, Andreas Karrenbauer, and Domagoj Matijevic. Energy-aware stage illumination. *International Journal of Computational Geometry and Applications*, 18(1/2):107–129, April 2005.
- [14] S. Jelić and D. Matijević. Stage illumination problem. Wolfram Demonstrations Project - <http://demonstrations.wolfram.com/StageIlluminationProblem>, 2009.
- [15] Dragan Jukić and Rudolf Scitovski. *Matematika I*. Sveučilište Josipa Jurja Strossmayera u Osijeku, Osijek, 2000.
- [16] Bernhard Korte and Jens Vygen. *Combinatorial Optimization*. Number 21 in Algorithms and combinatorics. Springer, second edition, 2002.
- [17] For The Terminal, Doratha E. Drake, and Stefan Hougardy. On approximation algorithms. *Information Processing Letters*, 89:15–18.

- [18] Jorge Urrutia. Art gallery and illumination problems. In *Handbook of Computational Geometry*, pages 973–1027. North-Holland, 2000.

**Sažetak.** U radu je proučavan sljedeći problem: Pretpostavimo da dužina  $L$  predstavlja scenu i  $S \subset \mathbb{R}^2$  skup točkastih izvora svjetla. Intenzitet emitiranog svjetla je obrnuto srazmjernan kvadratu udaljenosti do izvora. Treba odrediti količine energije  $x_s \geq 0$  za svaki  $s \in S$  tako da svaka točka  $p \in L$  primi dovoljno svjetla (primjerice 1 jedinicu svjetla) pri čemu je ukupna energija  $\sum_{s \in S} x_s$  minimalna. Nakon diskretizacijom scene moguće je odrediti približno optimalno rješenje problema koristeći  $(1 + \epsilon)$ -aproksimacijsku shemu koja se bazira na LP formulaciji. Ako uzmemo samo skup  $G_V$  točaka presjeka Voronoijevog dijagrama nad  $S$  i scene  $L$ , moguće je formulirati LP problem koji daje 4-aproksimaciju originalnog rješenja. Konstruiran je i jednostavni  $O(1)$ -aproksimacijski algoritam koji se izvršava u  $O(n^2)$  vremenu gdje je  $n = |S|$  a čija se analiza temelji na dualnom fitanju. Na koncu je ilustrirana implementacija prethodno opisanih algoritama u programskom paketu *Mathematica* pomoću dinamičkog okruženja *Manipulate*.

**Ključne riječi:** aproksimacijski algoritmi, linearno programiranje,  $(1 + \epsilon)$ -aproksimacijska shema, kombinatorijalni algoritam, jaka dualnost, slaba dualnost, aproksimacijski omjer.

**Summary.** In this paper the following problem is considered: Assume that stage is represented by line segment  $L$  and light sources are represented by set of points  $S \subset \mathbb{R}^2$  in the plane. Intensity of emitted light is inversely proportional to squared distance from the light source. Our goal is to assign powers  $x_s \geq 0$  to all light sources  $s \in S$  such that each point  $p \in L$  receives sufficient amount of light (e.g. one unit) while overall power consumption is minimized.

After discretization of stage  $L$  it's possible to determine approximate solution by using  $(1 + \epsilon)$ -approximation scheme which is based on LP formulation. If we consider only a set of intersection points  $G_V$  between Voronoi diagram over  $S$  and stage  $L$ , LP formulation that gives 4-approximate solution is obtainable. Simple combinatorial  $O(1)$ -approximation algorithm is also constructed whose analysis relies on primal-dual fitting method. It runs in  $O(n^2)$  time ( $n = |S|$ ). Finally, we illustrated an implementation of mentioned algorithms in *Mathematica* software by using dynamical environment *Manipulate*.

**Keywords:** approximation algorithms, linear programming,  $(1 + \epsilon)$ -approximation scheme, combinatorial algorithm, strong duality, weak duality, approximation ratio.

## Životopis

Slobodan Jelić je rođen 13. veljače 1986. godine u Osijeku. Osnovnu školu s odličnim uspjehom završava 2000. godine u Dalju, a iste godine upisuje *Gimnaziju Vukovar* u Vukovaru. Osnovnu Glazbenu školu u Vukovaru završava 2002. godine, odsjek harmonika i klavir. 2004. godine osvaja prvo mjesto na županijskom natjecanju iz fizike u Vukovarsko-srijemskoj županiji i drugo mjesto na regionalnom natjecanju iz filozofije. Gimnaziju završava 2004. godine s odličnim uspjehom kao đak generacije. Iste godine upisuje preddiplomski studij matematike na Odjelu za matematiku pri Sveučilištu J. J. Strossmayera u Osijeku. Nakon završenog preddiplomskog studija, 2007. godine nastavlja sveučilišni diplomski studij financijske i poslovne matematike. Odlukom Senata Sveučilišta Josipa Jurja Strossmayera u travnju 2008. godine prima **Rektorovu nagradu** za seminarski rad pod nazivom *Karakteristične funkcije i njihove primjene u statistici* pod mentorstvom prof. dr. sc. Mirte Benšić. U srpnju 2008. godine pohđa stručnu praksu na Odjelu za matematiku pri Sveučilištu u Oviedu (Španjolska). Radi u laboratoriji za numeričke simulacije i obradu signala na temu *Edge detection in MATLAB* gdje implementira Sobelov i Cannyjev detektor ruba slike. U zimskom semestru akademske 2008./09. godine kao demonstrator sudjeluje u provođenju nastave iz kolegija Matematika u okviru preddiplomskog studija na Ekonomskom fakultetu u Osijeku. U siječnju 2009. godine prima nagradu i priznanje za postignut izuzetan uspjeh (5.00) na 1. godini sveučilišnog diplomskog studija. Sudjeluje u realizaciji projekta pod nazivom *Matematika s po muke* koji ima za cilj uklanjanje teškoća u praćenju nastave matematike na svim sastavnicama sveučilišta. Pod vodstvom doc. dr. sc. Domagoja Matijevića realizira projekt Wolframovih demonstracija *Stage Illumination Problem*. U rujnu 2009. pohđa ljetnu školu *Optimization under Uncertainty* na Max Planck institutu u Saarbrückenu (Njemačka). Od 2006. godine dobitnik je državne stipendije a od 2007. godine dobitnik je stipendije Osječko-Baranjske županije. U srpnju 2009. godine prima *Top stipendiju* tjednika Nacional.