הטכניון – מכון טכנולוגי לישראל
Technion – Israel Institute of Technology

ספריות הטכניון
The Technion Libraries

בית הספר ללימודי מוסמכים ע"ש ארווין וג'ואן ג'ייקובס
Irwin and Joan Jacobs Graduate School

©
All rights reserved

This work, in whole or in part, may not be copied (in any media), printed, translated, stored in a retrieval system, transmitted via the internet or other electronic means, except for "fair use" of brief quotations for academic instruction, criticism, or research purposes only.
Commercial use of this material is completely prohibited.

©
כל הזכויות שמורות

אין להעתיק (במדיה כלשהי), להדפיס, לתרגם, לאחסן במאגר מידע, להפיץ באינטרנט, חיבור זה או כל חלק ממנו, למעט "שימוש הוגן" בקטעים קצרים מן החיבור למטרות לימוד, הוראה, ביקורת או מחקר. שימוש מסחרי בחומר הכלול בחיבור זה אסור בהחלט.

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

שפור ביצועי אלגוריתם לדחיסה אוניברסלית באמצעות חזוי

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת תואר
מגיסטר למדעים
בהנדסת חשמל

עמית אורן

H 948

טכניון
פקולטה להנדסת חשמל
פקולטה למדעי מחשב
ספריה מס' ....

הוגש לסנט הטכניון - מכון טכנולוגי לישראל
כסלו תשמ"ח חיפה דצמבר 1987

2061876

000000872058

7.02.89

המחקר נעשה בהנחיית פרופ' דוד מלאך בפקולטה להנדסת חשמל בטכניון.

תודתי נתונה לפרופסור מלאך על הנחייתו ועל עזרתו בכל שלבי
העבודה, ולצוות המעבדה לעיבוד אותות ובמיוחד לזיוה אבני.

© Technion - Israel Institute of Technology, Elyachar Central Library

תוכן העניינים

תקציר 1

רשימת סמלים וקיצורים 3

פרק 1: מבוא ומבנה העבודה 4

1.1 מבוא 4

1.2 מבנה העבודה 6

פרק 2: אלגוריתם זיו-למפל 7

2.1 מבוא 7

2.2 האלגוריתם המקורי 7

2.3 תאור מימושים, הרחבות ותוצאות קודמים 13

פרק 3: אלגוריתם זיו-למפל עם חיזוי (PZL) 19

3.1 תאור האלגוריתם 19

3.2 סיבוכיות האלגוריתם 28

3.3 תוצאות הסימולציות 29

פרק 4: אלגוריתם עץ אוניברסלי 37

4.1 מבוא 37

4.2 אלגוריתמי Variable to Block 39

4.3 נתוח אלגוריתם עץ אוניברסלי 49

4.4 בניית העץ האוניברסלי בעזרת PZL 64

4.5 שפור העץ האוניברסלי במהלך הקידוד 72

פרק 5: סכום ומסקנות 76

נספח A1: אלגוריתם פענוח לאלגוריתם זיו-למפל 78

נספח A3: תוצאות הניסויים ל-PZL 80

נספח A4: תאור גראפי של החסמים 82

© Technion - Israel Institute of Technology, Elyachar Central Library

נספח A5: תוצאות הניסוים לעץ אוניברסלי עם PZL 87
מקורות 90

© Technion - Israel Institute of Technology, Elyachar Central Library

<u>תקציר</u>

בעבודה זו מוצע שיפור לאלגוריתם הדחיסה האוניברסלית זיו-למפל, שמטרתו לגרום להתכנסות מהירה יותר של תהליך הדחיסה, מבלי להכניס עוות לתוצר הדחיסה.

השיפור מושג על-ידי פתוח עץ הקידוד באותם מקומות שיגרמו לירידה ביתירות העץ, וזאת על-ידי ביצוע חיזוי סטטיסטי שלהם, המבוסס על סדרת הקלט שכבר עברה קידוד. באופן זה מושגת דחיסה טובה יותר של הקלט למקודד. האלגוריתם נבדק על מקורות סינטטיים: חסר זכרון ומרקובי, ועל מקורות "אמיתיים" - מקדמי התמרות של תמונות.

השיפור המשמעותי ביותר מושג, כצפוי, באותם מקורות שעבורם מודל השערוך של סטטיסטיקות הקלט בחזאי, תואם את אופי המקור, אך גם עבור המקורות הלא סינטטיים מושג שיפור.

בעבודה מבוצע ניתוח של אלגוריתם דחיסה המבוסס על עץ אוניברסלי, למקרה שבו הוא נבנה על-ידי סדרת לימוד ממקור חסר זכרון ואחר-כך משמש לדחיסת אותו מקור. למקרה זה מפותח חסם תחתון על יחס הדחיסה הממוצע המושג על-ידי אנסמבל עצים בגובה נתון.

באותה טכניקת חסימה, מחושב חסם תחתון נוסף ליחס הדחיסה למקור חסר זכרון שמשיג אלגוריתם זיו-למפל. כאשר מתירים לעץ הקידוד להתפתח עד גודל קבוע, מבצעים איפוס ומתחילים מחדש. בהמשך נבדקת האפשרות לשימוש באלגוריתם המשופר שמוצג בתחילת העבודה לבניית עץ אוניברסלי.

© Technion - Israel Institute of Technology, Elyachar Central Library

הניסויים מבוצעים על סוגי המקורות שפורטו לעיל. מושג שיפור בביצועי העץ האוניברסלי כאשר הוא נבנה בצורה זו, גם מבחינת היתירות בעץ וגם מבחינת אורך סדרת הקלט הדרושה על מנת לבנות אותו. בעבודה נבדקה האפשרות של ביצוע שיפור לעץ האוניברסלי תוך כדי השימוש בו לקידוד סדרת הקלט. הסתבר שבמקרים שנבדקו, השיפור הינו מזערי, ונופל בביצועים מעץ אוניברסלי שנבנה על-ידי אלגוריתם זיו-למפל המשופר, ללא שינוי במהלך קידוד.

© Technion - Israel Institute of Technology, Elyachar Central Library

רשימת סמלים וקיצורים

ZL - אלגוריתם זיו-למפל

PZL - אלגוריתם זיו-למפל עם זזוי

N, n - אורך בלוק

A - א"ב

X_1^n - מחרוזת באורך n

x_i - אות בא"ב

α - גודל הא"ב

I_A - מיפוי מא"ב A לשלמים $0,\cdots,\alpha-1$

$P()$ - הסתברות

$P^*()$ - הסתברות אמפירית

$H()$ - אנטרופיה

$H^*()$ - אנטרופיה אמפירית

$r()$ - יתירות

$R()$ - קצב עץ קידוד

$\rho()$ - יחס דחיסה

$E\{\}$ - תוחלת סטטיסטית

c - מספר פסקאות בפיסוק אינקרמנטלי

$n_0()$ - מספר אפסים במחרוזת בינרית

$n_1()$ - מספר אחדים במחרוזת בינרית

© Technion - Israel Institute of Technology, Elyachar Central Library

פרק 1.

מבוא ומבנה העבודה

1.1 מבוא

בעבודה זו אנו דנים במספר נושאים הקשורים באלגוריתם זיו למפל לדחיסה אוניברסלית ללא עוות [1].

הבעייה המעשית שביסוד דחיסת מידע היא הצורך לשדר או לאחסן אינפורמציה המופקת על-ידי מקור מסויים. מכיוון ששידור ואיחסון כרוכים במחיר, הרי שיש עניין בהצפנה חסכונית בכל שניתן של אינפורמציה זו, כלומר ייצוגה בכמות מינימאלית של סיביות. אם נדרש שהאינפורמציה המקורית תהיה ניתנת לשחזור במלואה מתוך האינפורמציה המוצפנת, הרי שמדובר בצפינה ללא עוות. אם מותרת רמה מסויימת של עוות, הרי שהבעיה היא בעיה של קידוד עם קריטריון טיב, ומטופלת בתורת קצב-עוות.

המודל למקור האינפורמציה בבעיות אלה הוא תהליך אקראי בדיד בזמן. פלט המקור בזמן i הוא המשתנה האקראי x_i המקור נקרא סטציונרי אם התהליך האקראי הינו סטציונרי. צופן הינו פונקציה ממחרוזות של אותיות מקור למחרוזות בינאריות הנקראות מילות קוד. קצב הקוד הוא כמות הסיביות הממוצעת המייצגת את המקור. אם המקור סטציורי ניתן להגדיר את האנטרופיה שלו. האנטרופיה היא החסם התחתון לקצב של כל צופן ללא עוות. ההפרש בין הקצב לאנטרופיה נקרא היתירות של הצופן.

אם הסטטיסטיקה של המקור ידועה, ניתן לתכנן עבורו צופן ללא עוות באמצעות אלגוריתם Huffman. האלגוריתם מייצר צופן

block to variable כלומר מחרוזות מקור באורך קבוע n, ממופות למחרוזות בינאריות באורך משתנה. היתירות בקוד כזה היא לכל היותר n^{-1}.

במקרים מעשיים הסטטיסטיקות של המקור לא ידועות ולכן לא ניתן להשתמש באלגוריתם Huffman. צפינה אוניברסלית מתייחסת לבעיה זו. בצפינה כזו נהוג להניח שסטטיסטיקות המקור שייכות למחלקה מסויימת (למשל חסרי זכרון) והמטרה היא למצוא קוד בעל יתירות נמוכה לכל המקורות במחלקה. סדרה של צפנים בעלי אורך בלוק עולה n נקראת אוניברסלית אם היתירות שואפת לאפס לכל מקור במחלקה.

בעבודה זו אנו משתמשים גם במקדמי התמרות של תמונות כמקור אינפורמציה לצורך הניסויים, וזאת כדי לבדוק את ביצועי האלגוריתמים השונים על מקור לא סטציונרי עם סטטיסטיקה לא ידועה.

מטרות עבודה זו הם הצעת שיפור לאלגוריתם הדחיסה האוניברסלית של זיו-למפל, על-מנת לשפר את מהירות ההתכנסות של תהליך הדחיסה, כמו-כן לנתח את אלגוריתם העץ האוניברסלי, ולבדוק אפשרות בניית עצים אוניברסליים באמצעות אלגוריתם זיו-למפל המשופר.

© Technion - Israel Institute of Technology, Elyachar Central Library

1.2 מבנה העבודה.

א. בפרק 2 מוצג האלגוריתם המקורי של זיו ולמפל [1] לדחיסה אוניברסלית ללא עוות, ומפורטים מספר מימושים ותוצאות שלו שהוצגו ב-[5] ו-[6].

ב. בפרק 3 מוצגת ורסיה של אלגוריתם זיו למפל שפותחה במסגרת עבודה זו ומטרתה לנסות לשפר את מהירות ההתכנסות שלו.

ג. פרק 4 עוסק באחד המימושים שהוצגו ב-[5], אלגוריתם עץ אוניברסלי, ומפותחים שני חסמים הקשורים בו. כמו-כן נבדקת אפשרות לשיפור העץ האוניברסלי במהלך הקידוד.

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

פרק 2.

אלגוריתם זיו למפל

2.1 מבוא

במאמר [1] הוצע אלגוריתם אוניברסלי לדחיסה ללא עוות. אלגוריתם זיו-למפל (ZL). ב-[5] מתוארים מספר מימושים של האלגוריתם, ומתואר יישום שלו בשילוב עם אלגוריתם עם עוות, לדחיסת תמונות לשם אגירה במסד נתונים.

בפרק זה מוצג האלגוריתם המקורי ונסקרים מספר מימושים שלו.

2.2 האלגוריתם המקורי.

המקודד E מוגדר על-ידי החמישיה (S,A,B,g,f)

כאשר: S- אוסף מצבים סופי של המקודד.

A- א"ב בקלט. א"ב סופי $|A|=\alpha$.

B- אוסף סופי של מילות קוד מעל א"ב סופי.

$g:S \times A \rightarrow S$ - פונקצית המצב הבא שנקבעת על-ידי המצב הנוכחי ואות הקלט למקודד.

$f:S \times A \rightarrow B$ - פונקצית הפלט.

המקודד מתחלק לשני חלקים. המקודד החיצוני שמקבל בכניסתו בלוקים ארוכים מאד באורך N, ופולט סדרת מילות קוד באורך לא קבוע מעל B. מילות קוד אלה מופקות על-ידי המקודד הפנימי. המקודד החיצוני מתאפס וחוזר למצב התחלתי לאחר הטיפול בכל בלוק באורך N, וכך למעשה שוכח את ההיסטוריה של הקידוד.

החלק השני הוא המקודד הפנימי הקולט מחרוזות באורך משתנה שאורכן הולך וגדל. המחרוזות מקודדות באופן עוקב, וקידודן תלוי במצב המקודד עם קבלת המחרוזת. המחרוזות נגזרות מתוך סדרת הכניסה למקודד החיצוני על-ידי תהליך פיסוק הקרוי פיסוק אינקרמנטלי.

תהליך הפיסוק הוא סדרתי ומייצר מחרוזת חדשה ברגע שהקידומת של החלק הלא מפוסק של סדרת הקלט שונה מכל המחרוזות הקודמות.

2.2.1 תאור האלגוריתם וניתוחו.

תהיה נתונה מחרוזת X_1^N מעל א"ב סופי A. הפיסוק שלה:

$$X_1^N = X_{n_0+1}^{n_1} \cdots X_{n_p+1}^{n_{p+1}} \quad , \quad n_0=0 \quad , \quad n_{p+1}=N$$

יקרא אינקרמנטלי אם p המחרוזות $X_{n_{j-1}+1}^{n_j}$ $1 \le j \le p$ הן כולן שונות, ואם לכל $1 \le j \le p+1$, אם $n_j - n_{j-1} > 1$ אזי קיים $i < j$ כך ש: $X_{n_{i-1}+1}^{n_i} = X_{n_{j-1}+1}^{n_j}$.

על מנת לפסק סדרה בפסוק אינקרמנטלי נניח שקיימת המחרוזת באורך 0 שנסמנה λ. $\lambda = X_{n-1}^{n_0}$, ואז ניתן לרשום $X_1^N = \lambda X_1^N$ כדי לקבוע את המחרוזת ה j-ית $1 \le j \le p+1$, נבחר n_j כמספר השלם הגדול ביותר, לא גדול מ-N כך ש:

$$X_{n_{i-1}+1}^{n_i} = X_{n_{j-1}+1}^{n_j-1} \qquad (2.1)$$

© Technion - Israel Institute of Technology, Elyachar Central Library

כלומר אם למחרוזת ה-j יש k סימבולים אזי k-1 הסימבולים הראשונים כבר הופיעו במחרוזת קודמת (נבית i). הסימבול האחרון הוא זה שגורם לכך שהמחרוזת ה-j שונה מקודמותיה.

הקידוד של הפסקה ה-j $X_{n_{j-1}+1}^{n_j}$ הינו הייצוג הבינארי של המספר $I = i\cdot\alpha + I_A(X_{n_j})$, $\alpha = |A|$.

I_A הוא מיפוי של א"ב הכניסה A לקבוצת השלמים 0 עד $\alpha-1$.
כיוון ש $0 \le i \le j-1$. נקבל ש:

$$0 \le I'(X_{n_{j-1}+1}^{n_j}) \le (j-1)\alpha + \alpha - 1 = j\alpha - 1 \tag{2.2}$$

מספר הסיביות הדרוש לקידוד המילה ה j-ית הוא
$L_j = \lceil \log(j\alpha) \rceil$ (ה log לפי בסיס 2). $\lceil x \rceil$ ערך שלם עליון.
בהנחה שבבלוק X_1^N יש p+1 מחרוזות שונות, נקבל שהמספר הכולל של סיביות הדרוש לקידוד הבלוק הוא:

$$L = \sum_{j=1}^{p+1} L_j = \sum_{j=1}^{p+1} \lceil \log(j\alpha) \rceil \tag{2.3}$$

כיוון ש: $\lceil \log x \rceil \le 1 + \log x = \log 2x$.

$$L \le \sum_{j=1}^{p+1} \log(2j\alpha) < (p+1)\, \log\,(2\alpha(p+1)) \tag{2.4}$$

לכן יחס הדחיסה חסום על-ידי:

$$\rho_E(X_1^N) \le \frac{p+1}{N \log \alpha}\, \log\,(2\alpha(p+1)) \tag{2.5}$$

© Technion - Israel Institute of Technology, Elyachar Central Library

במאמר [1] מופיעות התוצאות הבאות:

א. יחס הדחיסה של סדרה אינסופית X מעל A בעזרת המקודד E בבלוקים בגודל N כל אחד נתון על-ידי:

$$\rho(X)=\limsup_{N\to\infty}\limsup_{k\to\infty}\frac{1}{kN\log\alpha}\sum_{i=0}^{k}P(X_{iN+1}^{(i+1)N})\log P(X_{iN+1}^{(i+1)N}) \qquad (2.6)$$

כאשר $P(X_a^b)$ הוא מספר הפסקאות בפסוק האינקרמנטלי של המחרוזת X_a^b

ב. הוכח שאם הסדרה X נלקחה ממקור ארגודי סטציונרי בעל אנטרופיה H אזי $Pr\{\rho(X)=H\}=1$.

ג. עבור מקור סטציונרי בעל אנטרופיה H, $E\rho(X)=H$.
E – תוחלת סטטיסטית.

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

תמצית האלגוריתם

(0) איתחול. הכנס את המחרוזת בעלת אורך אפס λ למילון.
קבע $j=1$.

(1) החל מהמקום הנוכחי של המחוון לסדרת הכניסה, מצא את המחרוזת הארוכה ביותר ששייכת למילון.

$$X^{n_j}_{n_{j-1}+1} = X^{n_i}_{n_{i-1}+1}$$

(2) פלוט את הקידוד של המספר המזהה i יחד עם הקידוד של הסימבול הראשון המופיע לאחר המחרוזת בסדרת הכניסה.

(3) הוסף מחרוזת חדשה למילון המורכב מהמחרוזת שנמצאה בשלב (1) ומהסימבול שלאחריה. קבע $j=j+1$. הקצב למחרוזת החדשה ערך מזהה j.

(4) קדם את המכוון של סדרת הכניסה מעבר למחרוזת החדשה (הכוללת את הסימבול הנוסף). חזור ל (1). אם המכוון לא מצביע לסוף הבלוק

(5) בסוף הבלוק חזור לשלב (0).

<u>סיבוכיות</u>

הזמן ליצירת מילת קוד תלוי למעשה בשלב מספר 1 באלגוריתם.
זמן החיפוש במילון יחסי ישר לאורך המחרוזת שעוברת קידוד ולאורך סדרת הכניסה שעברה קידוד.
שלב 3 קובע את כמות הזכרון הדרושה, והוא יחסי ישר לאורך סדרת הכניסה שעברה קידוד.

© Technion - Israel Institute of Technology, Elyachar Central Library

2.3 תאור מימושים, הרחבות ותוצאות קודמים

ב [5] מתוארים מספר מימושים של האלגוריתם שמטרתם להקל על המיכון של האלגוריתם במחשב ולהשיג שיפור בסיבוכיות זמן ומקום שלו.

המימושים הרלוונטיים לענייננו הם:

א. מימוש בעזרת עץ פיסוק.

ב. אלגוריתם עץ אוניברסלי, שידון ביתר הרחבה בפרק 4.

2.3.1 מימוש אלגוריתם זיו למפל (ZL) בעזרת עץ פיסוק

במימוש זה המילון מוחזק בתוך עץ חיפוש מדרגה α α היא עוצמת א"ב הקלט. התאור הפורמלי של האלגוריתם הוא:

(0) התחל את עץ הפיסוק עם α ענפים כך שיכלול את כל הסימבולים של הא"ב. לעלים מוקצים מילות הקוד $0\cdots\alpha-1$. הצב $j=\alpha$.

(1) החל מהמקום הנוכחי של המכוון לסדרת הכניסה, מצא את המחרוזת שחיפוש שלה במילון מגיע לעלה.

(2) פלוט את הקוד של המספר המזהה של העלה.

(3) הוסף α עלים חדשים לעץ בצומת האחרון שאליו מגיעים בתהליך

הפיסוק (ועל-ידי כך הוסף $\alpha-1$ עלים לעץ כולו!)
הקצה לעלים החדשים את המספרים המזהים $j,j+1,\cdots$, חוץ מהעלה
הראשון שיורש את מילת הקוד מאביו.
הצב $j=j+\alpha-1$.

(4) הזז את המכוון לסדרת הכניסה מעבר לפסקה החדשה, חזור ל (1).

האינדקס j משמש להקצאת מספרים מזהים לעלים.

להלן דוגמא להבהרת תהליך הקידוד.

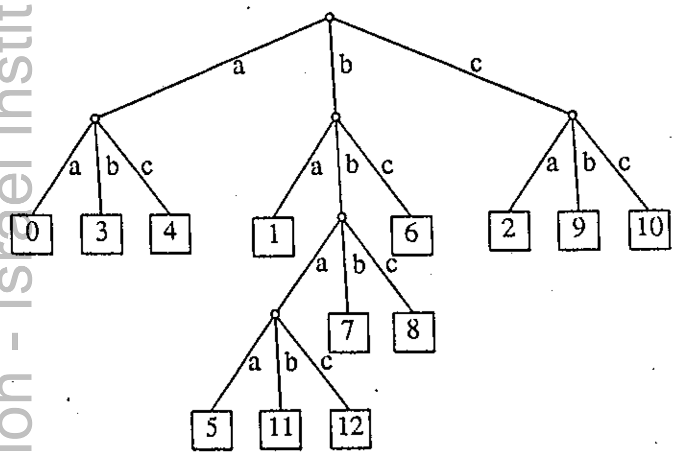
תהא X הסדרה מעל הא"ב הטרנרי {a,b,c}: X=abbbcbbba .

לאחר תהליך הקידוד נקבל:

1 . פיסוק הסידרה a,b,bb,c,bba

2 . סדרת מילות הקוד 0,1,5,2,5

3 . עץ הפיסוק שנוצר הינו:



יש לשים לב שלמרות שלשתי מחרוזות שונות הוקצב אותו ערך מספרי, לא נוצרת בעיה בפענוח משום שהמפענח יפרש אותם כשתי מילות קוד שונות.

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

הקצבת סיביות

כאשר נכנסת הפיסקה ה j לקידוד יש בעץ $1+(\alpha-1)j= (\alpha-1)(j-1)+\alpha$ עלים. (כל פיסקה חדשה מוסיפה לעץ $\alpha-1$ עלים). לכן צריך $\lceil \log j(\alpha-1)+1 \rceil$ סיביות למילת הקוד עבור הפיסקה ה j. מכאן שעבור קידוד של p+1 פיסקאות של הבלוק X_1^N דרושות

$$L=\sum_{j=1}^{p+1}\lceil \log j(\alpha-1)+1 \rceil \tag{2.7}$$

סיביות.

זמן ליצירת מילת קוד.

קל לראות שהזמן ליצירת מילת קוד (השהיה) יחסי ישר לאורך המחרוזת העוברת קידוד. מכאן שההשהיה הממוצעת שווה לאורך הפיסקה הממוצעת.

הזמן לעדכון העץ הינו קבוע ופרופורציוני ל- α.

הזכרון לאחסון המילון

כמות הזכרון לאיחסון המילון גדלה ביחס ישר למספר הפסקאות ולעצמת ה-א"ב α, ועל כן פרופורציוני ל- $(p+1)\alpha$. מכאן רואים מדוע אין שיטה זו טובה ל-א"ב גדול, לדוגמא א"ב של 256 אותיות. אחרי פיסוק של 256 מחרוזות נקבל זכרון השווה בגודלו לתמונה המקורית. מכאן שרק עבור $\alpha=2$ האלגוריתם הזה פרקטי.

© Technion - Israel Institute of Technology, Elyachar Central Library

פענוח

אלגוריתם הפענוח מופיע בנספח A1.
המקלט מחזיק את אותו העץ של המשדר ונשאר מסונכרן איתו.

2.3.2 אלגוריתם עץ אוניברסלי.

ב [5] הוצע אלגוריתם המבוסס על אלגוריתם זיו-למפל הקרוי "עץ פיסוק אוניברסלי". האלגוריתם כולל שני שלבים מרכזיים. האחד בניית עץ פיסוק על-ידי סדרת לימוד אופיינית, והשני שימוש בעץ הזה בקוד Variable to Block (VB), עבור מחלקת מקורות שאותם רוצים לדחוס,באופן שהמסלולים משורש לעלה מגדירים את מילות הקלט לקוד. הייחוד של שיטה זו הוא באופן בניית קוד ה VB, וזאת בעזרת אלגוריתם זיו-למפל. העץ המתקבל בשיטה זו הוא עץ אקראי ולכן מתעוררות שאלות לגבי מידת התאמת עץ אקראי זה למקורות אותם רוצים לדחוס באמצעותו. מקרה פרטי של הנ"ל הוא המקרה בו רוצים לדחוס את אותו המקור ששימש לבניית העץ. מכיוון שאלגוריתם זיו-למפל לומד את המבנה ההסתברותי של המקור, נצפה שההתאמה של העץ האוניברסלי לדחיסת המקור תהיה טובה יותר ככל שיגדל העץ. בפרק 4 נדון ביתר פירוט בשיטת דחיסה זו.

2.3.3 חסם ליתירות הנקודתית.

ב [6] פותח חסם ליתירות הנקודתית המתקבלת מהפעלת אלגוריתם זיו-למפל על סדרות באורך קבוע. ב [6] הוגדרה היתירות הנקודתית $r(X_1^N)$ עבור סדרת קלט X_1^N באורך N באופן הבא:

$$r(X_1^N) \equiv L(X_1^N) - \log \frac{1}{P^*(X_1^N)} \qquad (2.8)$$

כאשר: $L(X_1^N)$ - אורך בסיביות של מחרוזת הפלט שמקצה האלגוריתם לסדרת הכניסה X_1^N.

$P^*(X_1^N)$ - ההסתברות האמפירית של הסדרה X_1^N. בחישוב ההסתברות האמפירית נלקחות בחשבון השכיחויות היחסיות של אותיות הקלט בתור ההסתברות שלהם. לדוגמא: עבור מקור בינארי חסר זכרון

$$P^*(X_1^N) = \left[\frac{N(1)}{N}\right]^{N(1)} \cdot \left[\frac{N(0)}{N}\right]^{N(0)} \qquad (2.9)$$

כאשר N(1) - מספר ה"1" ב X_1^N, ו N(0) מספר ה"0" ב X_1^N הסיבה שהיתירות הוגדרה כך היא ש — $\log \frac{1}{P^*(X_1^N)}$ הוא החסם התחתון לאורך מילת הקוד שמקצה קוד Huffman ל X_1^N. קוד Huffman הוא הקוד האופטימאלי כאשר ההסתברויות של אותיות המקור ידועות. מכאן ש $r(X_1^N)$ מודד את המרחק מהחסם התחתון הזה.

ב [6] התקבלו התוצאות הבאות:

$$\frac{r(X_1^N)}{N} \le \frac{\log\left[\frac{4e}{\log e}\cdot(1-\varepsilon_n)\cdot\log N\right]}{(1-\varepsilon_N)\cdot\log N} \qquad (2.10)$$

© Technion - Israel Institute of Technology, Elyachar Central Library

כאשר

$$\varepsilon_N = 2\cdot\frac{1+\log\log 2N}{\log N} \qquad (2.11)$$

זהו חסם עליון ליתירות הנקודתית לסיבית בהפעלת אלגוריתם זיו-למפל על סדרות באורך N הנפלטות ממקור חסר זכרון.

ב [6] פותח חסם דומה למקורות בינאריים בעלי מספר סופי של מצבים FSM מסוג unifilar, כלומר כאלה שידיעת המצב הנוכחי של המקור, וידיעת האות שנפלט, קובעים את המצב הבא שאליו יגיע המקור. התשלום הנוסף ביתירות בגין אי ידיעת הפרמטרים ההסתברותיים של מקור כזה בעל k מצבים הוא:

$$\frac{\log k^2}{(1-\varepsilon_N)\log N} \qquad (2.12)$$

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

פרק 3

אלגוריתם זיו למפל עם חיזוי (PZL)

גרסה זו לאלגוריתם זיו-למפל פותחה במסגרת העבודה ובאה להציע שינוי באלגוריתם במטרה לשפר את מהירות ההתכנסות של תהליך הדחיסה.

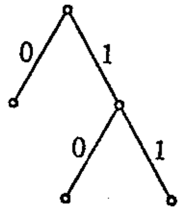
באלגוריתמים לדחיסה שמוכח לגביהם שהם מתכנסים לאנטרופית המקור יש טעם במאמץ לשפר מהירות ההתכנסות רק בקטע ההתחלתי שבו לומד האלגוריתם את סטטיסטיקות המקור, ואז הוא לא יעיל.

החל מאותו שלב שהאלגוריתם קרוב עד כדי סטיה קטנה מהאנטרופיה, אין טעם בנסיון לשפר מכיוון שהרווח יהיה קטן מאד.

3.1 תאור האלגוריתם

המוטיבציה מאחרי הצעת השיפור לאלגוריתם מצריה באופן הפעולה של אלגוריתם אופטימלי לתכנון קודי VARIABLE TO BLOCK,
עבור מקור חסר זכרון, אלגוריתם TUNSTALL [4].
האלגוריתם נדון ביתר פירוט בסעיף 4.2 אולם נציג כאן את עקריו.
קוד ה VB מיוצג על ידי עץ α-ארי, כאשר α גודל א"ב המקור.
מילות הקלט לקוד מוגדרות על ידי המסלולים שורש - עלה של העץ.

לדוגמא ל א"ב בינארי



העץ הנ'ל מדגיר קוד VB שמילות הקלט שלו הן {"0","10","11"}. מחרוזות הקלט למקודד מפוסקות לתת מחרוזות השייכות לקבוצת מילות קלט אלו.

האלגוריתם מתחיל בעץ עם שורש ו α עלים,ובכל עלה מחזיקים את הסתברות המקור לאות הא"ב שמיוצגת על ידי הקשת שורש - עלה.
האלגוריתם עובר מעץ אופטימלי זה בגודל α ,לעץ אופטימלי בגודל $2\alpha-1$, על ידי הפיכת העלה בעל ההסתברות הגבוהה ביותר לצומת פנימי ואב ל α עלים חדשים,כל אחד מהם מתאים לאות מקור.
בעלים החדשים מחזיקים את ההסתברות של מחרוזות הקלט המוגדרות על ידי המסלולים שורש - עלה, כלומר אם X_1X_2 מחרוזת קלט לקוד

$$P(X_1X_2)=P(X_1)\cdot P(X_2)$$

כאשר $P()$ חוק ההסתברות של המקור.
באותו אופן ממשיכים להגדיל את העץ כל פעם על ידי הרחבת העלה הסביר ביותר עד לקבלת קוד עם יתירות רצויה.
לסדרת העצים יתירות היורדת לפי $O\left(\frac{1}{n}\right)$ כאשר n אורך מילת היציאה של הקוד.
קיימת הרחבה של אלגוריתם זה למקורות מרקוביים [4].

© Technion - Israel Institute of Technology, Elyachar Central Library

אם נתבונן על אופן פעולת אלגוריתם זיו-למפל נראה שהאלגוריתם עובר מעץ לעץ גם על ידי הרחבת עלה מסוים בכל פעם,אלה שעלה זה נבחר על ידי סדרת הקלט.

נוסיף לאלגוריתם זיו-למפל מנגנון נוסף לגידול העץ,מנגנון שינסה לשערך מי הוא העלה הסביר ביותר.
לצורך זה נניח על המקור מודל הסתברותי P_θ כאשר $\underline{\theta}$ פרמטרי המודל,את $\underline{\theta}$ משערכים באופן שוטף על ידי משערך $\underline{\hat{\theta}}$ מתוך סדרת הקלט שעברה קידוד. וזאת כדי שלמפענח תהיה אותה אינפורמציה לעדכון $\underline{\hat{\theta}}$.

נסמן ב- α_N^i את העלה ה i בעץ הקידוד זיו- למפל בשלב שיש לו N עלים, ונסמן ב X_N^i את מילת הקלט המתאימה לעלה זה.
נחשב:

$$I = \underset{i}{argmax}(P_{\underline{\hat{\theta}}}(X_N^i) \Big| \text{ לכל ה- } X_N^i \text{ ששיכים לעץ } \}$$

ונרחיב את העלה α_N^I באותו אופן המורחב על ידי אלגוריתם זיו-למפל . α_N^I הוא העלה הסביר ביותר בעץ בהנתן המודל P_θ ווקטור השערוך $\underline{\hat{\theta}}$
האופן בו משולב מנגנון זה באלגוריתם זיו-למפל הוא כדלהלן:
א. על סמך P_θ ו $\underline{\hat{\theta}}$ בצע הרחבת העץ.
ב. בצע פזת זיו-למפל רגילה כלומר: קריאת פיסקת קלט, שידור מילת קוד והרחבת העץ.
ג. עדכון $\underline{\hat{\theta}}$ על סמך פיסקת הקלט החדשה.
ד. חזרה ל - א.

© Technion - Israel Institute of Technology, Elyachar Central Library

ניתן לראות שלכל פיסקת קלט מורחב העץ פעמיים במקום פעם אחת.
השיפור ביחס הדחיסה יהיה תלוי במידה רבה בהתאמת המודל למבנה ההסתברותי של המקור ולמהירות ההתכנסות של המשערך לפרמטרים.
אולם גם במקרים בהם אין התאמה מושלמת של המודל למקור ניתן להשיג שיפור, כפי שנראה בתוצאות הסימולציות.

הסיבות לכך ששני המנגנונים שולבו יחד ולא נבחר מנגנון המודל (חזוי) בלבד הם כמה.

לא מובטח שאם נשתמש רק בו האלגוריתם יתכנס, ביחוד באותם מקרים שבהם מודל החזוי P_θ לא מאפיין במדויק את המקור.

כמו כן מנגנון החזוי מטיל עומס חישובי שהולך וגדל עם גידול העץ ולכן הופך לא פרקטי בשלב מסוים.

כזכור מאמצים לשיפור יחס הדחיסה אפקטיביים בתחילת תהליך הדחיסה, עובדה זאת משתלבת עם העובדה שהעומס החישובי שמטיל תהליך החזוי גדל עם גודל העץ, ומכאן שיש ענין בעצירת מנגנון החזוי בשלב מסוים. סיבה נוספת לעצור את החזוי ולהמשיך עם הרחבות זיו-למפל בלבד, היא שהרחבות של העץ על סמך החזוי שנעשות בשלב שבו לא נותר הרבה קלט יש סכוי שלא ינוצלו, מאידך גודל מילת היציאה עלול לגדול כתוצאה להרחבות אלו.

קיימת לכן נקודה שמעבר לה לא כדאי להמשיך את תהליך החזוי מכיוון שהרחבות העץ לא מנוצלות אך גודל מילת היציאה גדל.

התנהגות בדיוק כזו נצפתה בסימולציות, התברר שאם מפסיקים את תהליך החזוי לפני סיום הקלט מתקבל יחס דחיסה טוב יותר.

המודל שנבחר לסימולציות הוא מודל F.S.M. UNIFILAR מסדר 10.

פרמטרי המודל, שהם הסתברויות המעבר, משוערכים על ידי השכיחויות היחסיות. בעלי העץ מוחזקת סטטיסטיקה מספקת של סדרות הקלט המתאימות להם, ליעל את חישובי ההסתברויות.

© Technion - Israel Institute of Technology, Elyachar Central Library

בעיה היכולה להתעורר במימוש האלגוריתם במחשב היא של underflow בחישוב של הסתברויות העלים. לפתרון בעיה זו אפשר לעבוד עם לוג-הסתברויות במקום הסתברויות, מכיוון שבמקרים מעשיים, הסתברות העלה תהיה מכפלה של הסתברויות הקשורות במסלול שורש-עלה.

© Technion - Israel Institute of Technology, Elyachar Central Library

משיקולים מעשיים מומש האלגוריתם רק עבור א"ב בינארי, ולהלן תמציתו.

(0) התחל את עץ הפיסוק עם שני ענפים. הקצה להם את מילות הקוד 1, 0. הצב j=2 . התחל הסתברויות אמפירית לאפס.

(1) מצא את העלה הסביר ביותר לפי המודל $P_{\underline{\theta}}$ והפרמטרים המשוערכים $\hat{\underline{\theta}}$.

(2) הוסף שני בנים לאותו עלה. לשמאלי הורש את מילת הקוד של האב, ולימני תן את מילת הקוד j. הצב j=j+1.

(3) החל מהמקום הנוכחי של המכוון לסדרת הכניסה. מצא את המחרוזת שהחיפוש שלה במילון מוביל לעלה.

(4) פלוט את מילת הקוד בעלה. מיוצגת ב $\lceil \log(2j-1) \rceil$ סיביות.

(5) הוסף שני בנים לעלה שסיים את החיפוש. לשמאלי הורש מילת קוד של האב. לימני תן מילת קוד j. הצב j=j+1.

(6) עדכן הסתברויות אמפיריות שיכללו את הפיסקה החדשה.

(7) הזז את המכוון לסדרת הכניסה מעבר לפיסקה החדשה.

(8) חזור ל-(1).

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

תהא נתונה הסידרה הבינארית $Z = 01011...$

נניח מקור חסר זכרון, ונבחר מודל חיזוי מתאים, כלומר

$$P^*(X)=\hat{P}_0^{n_0(X)}\,\hat{P}_1^{n_1(X)}$$

כאשר $\hat{P}_0$ הוא המשערך להסתברות הופעת "0", ו $\hat{P}_1$ הוא המשערך להסתברות הופעת "1".

$n_0(X)$ – מספר האפסים בסדרה X.

$n_1(X)$ – מספר האחדים בסדרה X.

$P^*(X)$ – ההסתברות המשוערכת להופעת הסידרה X.

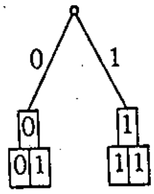
לצורך השיערוך נבחר משערכי שכיחות יחסית. כלומר:

$$\hat{P}_0(Y)=\frac{n_0(Y)}{N} \qquad \hat{P}_1(Y)=\frac{n_1(Y)}{N}$$

כאשר N – אורך הסדרה Y. במקרה שלנו Y היא כל סדרת הקלט שכבר עברה קידוד. בעלי העץ נחזיק סטטיסטי מספיק לחישוב P, במקרה זה מספר ה "1" במסלול שורש עלה, ואורך המסלול שורש עלה.

עלה יסומן $\begin{array}{|c|}\hline a\\\hline\end{array}\begin{array}{|c|c|}\hline b&c\\\hline\end{array}$ כאשר a– מילת הקוד, b–מספר ה "1" במסלול שורש עלה c–אורך המסלול שורש עלה.

העץ ההתחלתי הוא:

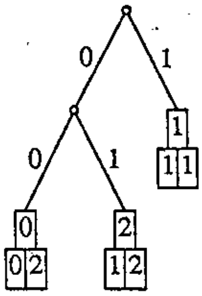


כמו-כן, בהתחלה $\hat{P}_0=0$ ו $\hat{P}_1=0$.

בשלב זה באלגוריתם מתבצעת פרדיקציה.

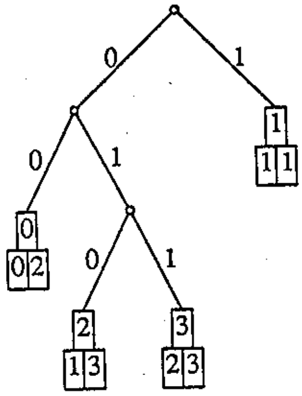
ההסתברות של העלה 0 היא $P^*(0)=0$ ההסתברות של העלה 1 היא $P^*(1)=1$

נבחר במקסימום את העלה 0. ומתקבל העץ:



בשלב זה נכנסת מילת הקלט "01". ומתעדכן העץ. ומשודרת מילת הקוד "2". ב $\lceil \log_2 3 \rceil = 2$ סיביות.

העץ החדש:



כעת נעדכן את ההסתברויות האמפיריות בסדרת הקלט: $\hat{P}(0)=1/2$, $\hat{P}_1=1/2$.

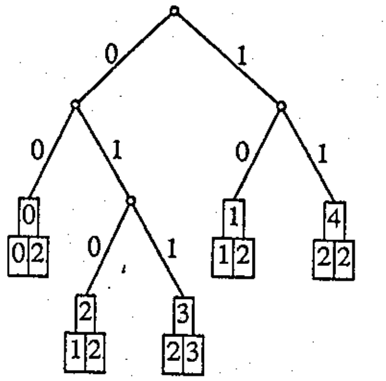
נשים לב שעל סמך מילת הקוד "2" ששודרה, יודע המקלט לעדכן את העץ ואת ההסתברויות האמפיריות לאותו מצב של המשדר.

© Technion - Israel Institute of Technology, Elyachar Central Library

בשלב זה שוב מתבצע חיזוי. נסמן ב $Q^*(n)$ את ההסתברות האמפירית של העלה ה n ונקבל:

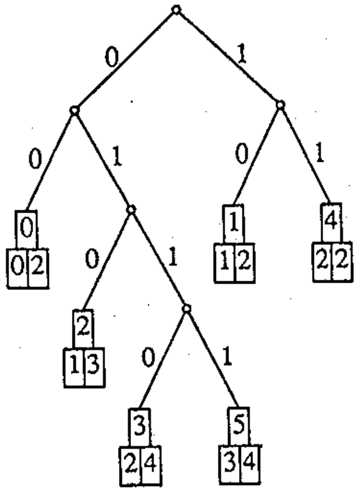
$$Q^*(0)=(\hat{P}_0)^2=1/4$$
$$Q^*(1)=\hat{P}_1=1/2$$
$$Q^*(2)=(\hat{P}_0)^2\cdot\hat{P}_1=1/2$$
$$Q^*(3)=\hat{P}_0(\hat{P}_1)^2=1/8$$

לאחר בחירת המקסימום נקבל שיש להרחיב את העץ בעלה מספר "1". ונקבל את העץ הבא:



בשלב זה נכנסת סדרת הקלט 011 ומשודרת מילת הקוד "3" ב $\lceil \log_2 5 \rceil = 3$ סיביות.

העץ לאחר העדכון יהיה:



© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

באותו אופן ממשיך האלגוריתם על יתר סדרת הקלט.

3.2 סיבוכיות האלגוריתם.

א. סיבוכיות זמן

התוספת בסיבוכיות הזמן על פני האלגוריתם המקורי היא כדלקמן:

חישובי פרמטרי המודל דורשים עדכון לכל תו קלט, עבור כל אחד מפרמטרי המודל.

חישוב ההסתברויות האמפיריות של העלים דורשים לכל פיסקת קלט מעבר אחד על כל העלים (כאשר מספר העלים שווה למספר הפיסקאות שנקלטו כבר). חישוב המקסימום נעשה, בד בבד עם חישובי ההסתברויות. מכאן שבשלב שבו יש j עלים בעץ, יש להשקיע כמות עבודה נוספת בחישוב ההסתברויות בעלים, היחסית ל-j ולמספר פרמטרי המודל.

מכאן שאם בסך הכל ישנם C פיסקאות בסדרת הקלט, נקבל כמות עבודה נוספת היחסית ל- $\sum_{i=1}^{c} i$ וזה יחסי ל-C^2.

ניתן להוריד את הסיבוכיות על-ידי זה שמפסיקים לשערך את ההסתברויות האמפיריות אחרי שנקלטו מספר מסויים של תווים, כלומר להקפיא את המשוערך. אם מחזיקים את העלים ברשימה ממויינת, צריך לעדכן רק עדכון אחד עבור כל פיסקת קלט. בסדרות ארוכות החסכון מאד משמעותי.

ב. סיבוכיות מקום

האלגוריתם דורש להחזיק רשימת מצביעים לעלים, וכמו כן לכל עלה יש להחזיק סטטיסטיקה מספקת עבור המסלול משורש אליו. כיוון שמספר העלים שווה בסופו של דבר ל- c מספר פיסקאות הקלט, הרי שכמות הזכרון הנוספת יחסית ל- c .

3.3 תוצאות הסימולציות

בוצעו מספר סימולציות במגמה להשוות בין ביצועי שני האלגוריתמים. מקורות האינפורמציה שנבחרו לצורך הסימולציות היו משני סוגים. מקורות סינטטיים המבוססים על מחולל סדרות אקראיות לכאורה במבנה LFSR (Linear Feedback Shift Register). ראה [8]. הסוג השני היו מקורות "אמיתיים", מקדמי התמרות של תמונות.

נתחיל בתוצאות עבור מקור חסר זכרון.
נוסו 3 מקורות בינאריים בעלי אנטרופיות 0.1, 0.5 ו-0.9 .

בטבלה 3.3.1 מופיעות התוצאות עבור המקור בעל אנטרופיות תכנון 0.5 . העמודה הראשונה היא אורך סדרת הקלט. העמודה מימינה היא האנטרופיה האמפירית, או $-\frac{1}{N}\log P^*(X_1^n)$ כאשר n הוא אורך הסדרה ו $P^*(X_1^n)$ ההסתברות האמפירית שלה.

© Technion - Israel Institute of Technology, Elyachar Central Library

כלומר $P^*(X_1^n)=\left(\frac{n_1}{n}\right)^{n_1}\left(\frac{n_0}{n}\right)^{n_0}$ באשר n_1 הוא המספר ה "1"
ב X_1^n ו n_0 מספר האפסים ב X_1^n. האנטרופיה האמפירית מהוה חסם תחתון על אורך מילת הקוד לאות מקור עבור קוד Huffman, שהוא האופטימלי במקרה שהסתברויות המקור ידועות. נשתמש במרחק מחסם זה למדידת היתירות שמשיגים האלגוריתמים השונים.

העמודה השלישית בטבלה (ZL) היא תוצאות הפעלת אלגוריתם זיו-למפל המקורי על הסדרות, באשר יחס הדחיסה מחושב לפי

$$\frac{\sum_{i=1}^{c}\lceil \log(i+1)\rceil}{n}$$

n – אורך הסדרה
C – מספר הפסקאות

וזאת כיוון שהפיסקה ה j מקודדת בעץ בעל j+1 עלים.

העמודה הרביעית (PZL) היא תוצאות הדחיסה של אלגוריתם זיו-למפל עם פרדיקציה, באשר ממשיכים עם הפרדיקציה עד סוף הסדרה. יחס הדחיסה מחושב לפי

$$\frac{\sum_{i=1}^{c'}\lceil \log(2i+1)\rceil}{n}$$

n – אורך הסדרה
c' – מספר הפיסקאות בסדרת הקלט

קל לראות שהפסקה ב-i מקודדת על-ידי עץ בעל 2i+1 עלים.

העמודה החמישית (FPZL) היא אלגוריתם זיו למפל עם פרדיקציה עד אמצע הסדרה. יחס הדחיסה מחושב לפי

$$\frac{\sum_{i=1}^{c''}\lceil \log l(i)\rceil}{n}$$

n – אורך הסדרה
c'' – מספר הפיסקאות בסדרת הקלט
$l(i)$ – מספר העלים בעץ בעת קידוד הפסקה מ- i

© Technion - Israel Institute of Technology, Elyachar Central Library

כאמור, כאן מופיעה הטבלה עבור מקור חסר זכרון בעל אנטרופיה 0.5 והגרף המשווה בין העמודה השלישית לעמודה החמישית מבחינת היתירות הנקודתית. כלומר, משניהם מחסירים את העמודה השניה.

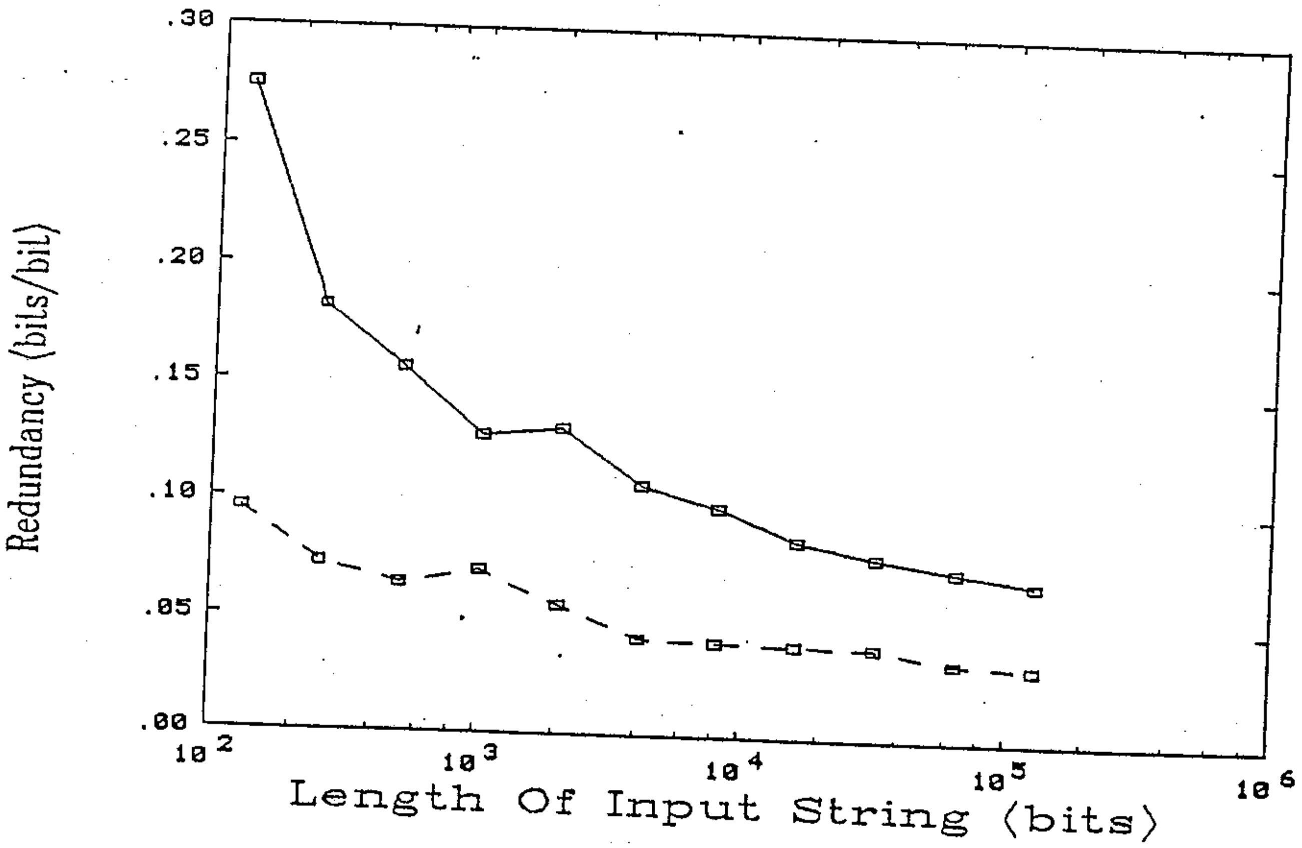
הגרפים עבור אנטרופיות 0.1 ו- 0.9 מופיעים בנספח A3.

© Technion - Israel Institute of Technology, Elyachar Central Library

אורך	אנטרופיה	ZL	PZL	FPZL
128	0.498029	0.773438	0.648438	0.593750
251	0.532497	0.714844	0.628906	0.605469
512	0.498029	0.654297	0.587891	0.562500
1024	0.452053	0.580078	0.540039	0.522461
2048	0.478519	0.609863	0.552246	0.534668
4096	0.488350	0.595947	0.546631	0.530518
8192	0.50024	0.598999	0.555298	0.541138
16324	0.505134	0.590909	0.561995	0.545455
32768	0.502627	0.581726	0.555328	0.542572
65536	0.499688	0.573441	0.548874	0.533829
131072	0.498420	0.567413	0.545372	0.530731

טבלה 3.3.1
השוואה בין האלגוריתמים השונים על מקור בינארי
חסר זכרון עם אנטרופית תכנון 0.5
table 3.3.1
comparison of the various algorithms on a
binary memoryless source of entropy 0.5

© Technion - Israel Institute of Technology, Elyachar Central Library



גרף 1.3.3

השוואה בין היתירותיות הנקודתיות ב-ZL וב-FPZL

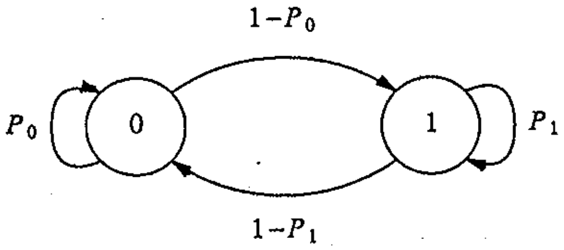
עבור מקור בינרי חסר זיכרון עם אנטרופיית תכנון 0.5

Comparison of redundancy in ZL and FPZL

for Bernoulli Source with $H=0.5$

© Technion - Israel Institute of Technology, Elyachar Central Library

נעבור כעת לתוצאות עבור מקור unifilar FSM בעל שני מצבים. את המקור ניתן לתאר כך:



במצב 0 המקור פולט "0" , ובמצב 1 המקור פולט "1".

בסימולציות נבחר $P_0=P_1$ על מנת לקבל אנטרופיה מסדר 0 השווה ל-1. האנטרופיה המותנית מסדר ראשון תוכננה להיות 0.5.

בטבלה 3.3.2 מופיעות התוצאות באותה מתכונת כמו בטבלה 3.3.1

בנספח A3, מופיעות התוצאות עבור דחיסה של מקדמי התמרות של תמונות. כאן לא ניתן למדוד את היתירות מכיוון שלא ניתן לחשב את האנטרופיה. נזכור שהמודל שמניח האלגוריתם לצורך החידוי, הוא מרקובי מסדר 10.

לסכום התוצאות המספריות, ניתן לראות שעבור מקורות חסרי זכרון במחרוזות הארוכות ביותר, FPZL הוריד את היתירות לסיבית ב 32% עבור H=0.1, ב 53% עבור H=0.5 וב 66% עבור H=0.9.

במקור unifilar FSM ירדה היתירות לסיבית במחרוזת הארוכה ביותר ב 40%.

במקרה של מקדמי התמרות של תמונות גם התקבל שיפור, אולם לא ניתן לחשב את היתירות עצמה. אך ההבדלים האבסולוטיים בין תוצאות יחסי הדחיסה די דומים למתקבל עבור מקור ה FSM.

© Technion Israel Institute of Technology, Elyachar Central Library

מתוך התוצאות אמנם מתברר שהפרדיקציה משפרת את יחס הדחיסה גם עבור המקרה של מקורות לא סטציונריים עם סטטיסטיקה לא ידועה, וגם עבור המקרה של מקורות סינטטיים.

יש לציין שנוסו שיטות אחרות לביצוע הפרדיקציה, למשל הרחבת העלה העמוק ביותר וכו'. אך הם לא הניבו שום שיפורים משמעותיים, וברוב המקרים, אף פגעו.

© Technion - Israel Institute of Technology, Elyachar Central Library

אורך סדרת הקלט	אנטרופיה אמפירית	ZL	PZL	FPZL
128	0.546315	0.929688	0.789063	0.75
256	0.499323	0.832031	0.683594	0.636719
512	0.504540	0.75	0.642578	0.611328
1024	0.482060	0.703125	0.625977	0.593750
2048	0.484098	0.672363	0.622559	0.586426
4096	0.486175	0.648682	0.605225	0.586914
8192	0.496100	0.638062	0.604248	0.579346
16304	0.491184	0.615842	0.584232	0.566038
38768	0.498281	0.610931	0.582428	0.567017

טבלה 3.3.2
השוואה בין האלגוריתמים השונים על מקור
unifilar FSM, עם אנטרופיות תכנון מסדר 0
$H_0=1$, ואנטרופית תכנון מותנית מסדר ראשון 0.5.

Table 3.3.2
comparison of various algorithms on binary unifilar F.S.M source of entropy $H_0 = 1$ and conditional entropy $H_1 = 0.5$

© Technion - Israel Institute of Technology, Elyachar Central Library

פרק 4.

אלגוריתם עץ אוניברסלי

4.1 מבוא.

אלגוריתם העץ האוניברסלי כפי שהוצג ב [5], מבוסס על הרעיון של שימוש בסדרת לימוד על מנת לבנות עץ קידוד, ושימוש בעץ לקידוד סדרות אחרות. הרעיון הוא שאם סדרת הלימוד מאפיינת טוב את קבוצת הסדרות שרוצים לדחוס, הרי שהביצועים של העץ האוניברסלי יהיו טובים.

עובדה שלא צויינה במפורש ב [5] היא ששיטת העץ האוניברסלי היא בעצם שיטת קידוד ידועה, המקובלת יותר בשם: קידוד VB (Variable to Block). צפינה מאורך משתנה לקבוע.

סדרות הקלט של הקוד מוגדרות על-ידי המסלולים מן השורש לכל אחד מהעלים. מספר העלים הוא כמובן מספר מילות הקוד. אורך מילת הקוד ביציאה הוא m, – מספר העלים בעץ.

המיוחד בשיטת העץ האוניברסלי הוא כמובן אופן בניית עץ הקידוד. בספרות מתוארת שיטה לבניית עץ קידוד אופטימלי עבור מקור חסר זכרון, בעל סטטיסטיקה ידועה, ועבור מקורות מרקוביים unifilar [4]. כמו-כן מתוארות שיטות לאדפטציה של עצים למקורות בעלי סטטיסטיקה המשתנה לאט [3].

באלגוריתם העץ האוניברסלי, העץ שנבנה על-ידי סדרת הלימוד, הוא פונקציה של סדרת הלימוד, ולכן אקראי.

מזה נובע שכל קבוצת העצים בעלי מספר קבוע של עלים מהווה

© Technion - Israel Institute of Technology, Elyachar Central Library

אנסמבל הסתברותי. לכל עץ באנסמבל יש הסתברות שדווקא הוא יתקבל מסדרת הלימוד. לכל עץ כזה באנסמבל ניתן .גם לחשב במקרים מסויימים את יחס הדחיסה שלו, באם הוא יופעל על מקור מסויים. מכאן נוכל לחשב את יחס הדחיסה הממוצע על כל האנסמבל, ולשאול כיצד משתנה ממוצע זה כפונקציה של מספר העלים של עצי האנסמבל.

© Technion - Israel Institute of Technology, Elyachar Central Library

4.2 אלגוריתמי VB

נסקור תחילה מספר תוצאות מהספרות על קודי VB.

נתחיל במינוח והגדרות:

יהא A – א"ב הקלט

נסמן A^* – קבוצת כל המחרוזות בעלות אורך סופי מעל ה-א"ב A.

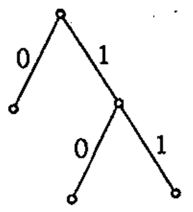
קבוצה $\Gamma\subset A^*$ תיקרא <u>שלמה</u> אם לכל סדרה אינסופית מעל A יש קידומת (prefix) ב- Γ

קבוצה $\Gamma\subset A^*$ תיקרא <u>מתאימה</u> אם אף מילה ב- Γ אינה קידומת של מילה אחרת ב- Γ.

קוד VB הינו מיפוי חח"ע מקבוצה שלמה ומתאימה $\Gamma\subset A^*$ לקבוצת הסדרות הבינאריות באורך $\lceil \log_2|\Gamma| \rceil$.

דוגמא לקבוצה שלמה ומתאימה מעל ה-א"ב הבינארי היא הקבוצה {0,10,11}.

נשים לב שקבוצה זו מתאימה לעץ הקידוד



קל לראות שכל קבוצה שלמה ומתאימה, מגדירה עץ קידוד α-ארי מלא. (כלומר, לכל צומת פנימי יש α בנים. כזכור α – גודל א"ב הקלט).

© Technion - Israel Institute of Technology, Elyachar Central Library

ב [2] מופיעות הלמות הבאות:

נסמן ב W(C) קבוצה של מלים השייכות ל A^* והיא בעלת C איברים.

<u>למה 1</u> : הקבוצה W(α)=A היא שלמה ומתאימה, והיא הקבוצה השלמה והמתאימה היחידה בגודל α.

<u>למה 2</u> : אם W(T) שלמה ומתאימה ו $w \in W(T)$,

אזי הקבוצה $W(T+\alpha-1)=\{\{W(T)-w\}\bigcup_{z\in A} wz\}$

(כאשר wz הוא שרשור של המילה w עם האות z.), היא קבוצה שלמה ומתאימה בגודל T+α-1.

במקרה זה נאמר ש W(T+α-1) הוא הרחבה של W(T) ו w היא המילה המרחיבה.

<u>למה 3</u> : תהא W(T) קבוצה שלמה ומתאימה בגודל T. אז קיימת סידרה $\{W_n\}$ כך שלכל n, W_n קבוצה שלמה ומתאימה בגודל $\alpha+(n-1)(\alpha-1)$, וכך ש $W_1=A$ ו W_{n+1} הרחבה של W_n ו W(T) איבר בסדרה.

<u>למה 4</u> : אם W(T) קבוצה שלמה ומתאימה בגודל T, אזי קיים m שלם חיובי כך ש: $T=\alpha+m(\alpha-1)$.

ניתן לבנות על בסיס ה-א"ב A מקור חסר זכרון, על-ידי הגדרת פונקציית הסתברות חד מימדית Q על A.

הפילוג Q, משרה פילוג על קבוצה שלמה ומתאימה W, באופן הבא:

תהא $\underline{w}=w_1\cdots w_m$, $\underline{w}\in W$, ו $1\le i\le m, w_i\in A$.

נגדיר

$$P(\underline{w})=\prod_{i=1}^{m}Q(w_i)$$

© Technion - Israel Institute of Technology, Elyachar Central Library

קל להוכח ש $\sum_{w \in W} P(w)=1$ עבור W שלמה ומתאימה, מתוך זה שעבור $W(\alpha)=A$; $\sum_{w \in A} P(w)=1$ באופן טריוויאלי. ומלמה 3, ובאינדוקציה, נקבל את כל היתר.

נגדיר $L(w)$, $w \in A^*$ כאורך באותיות מקור של המילה w. עבור קבוצה שלמה ומתאימה W, נגדיר:

$$\bar{L}(W) \equiv \sum_{w \in W} P(w)L(w) \qquad (4.1)$$

האורך הממוצע של אותיות הקלט. (שווה להשהיה של המקור).

נגדיר:

$$H(W) \equiv -\sum_{w \in W} P(w)\log_2 P(W) \qquad (4.2)$$

– אנטרופית W

יחס הדחיסה הממוצע של הקוד הוא $\dfrac{\lceil \log_2 |W| \rceil}{\bar{L}(W)}$ בסיביות לאות מקור. מגדירים:

$$R_{\min}(T) = \min_{W(T)} \frac{\log_2 T}{\bar{L}(W(T))} \qquad (4.3)$$

כאשר המינימום נלקח על כל הקבוצות השלמות והמתאימות $W(T)$ בעלות T איברים.

ב [2] מופיעה הוכחה ש $\lim_{T\to\infty} R_{\min}(T) = H(A)$ כאשר $H(A)$ אנטרופית המקור. המשפט מוכח על-ידי הצגה של אלגוריתם בניה לסדרה של עצים אופטימליים (כלומר שמגשימים את $R_{\min}$) ומציאת חסמים על הקצב שלהם. האלגוריתם דורש ידיעת הסתברויות המקור.

בהקשר של דחיסה אוניברסלית, מקובל להגדיר מספר צורות של יתירויות והתכנסויות של קודים.

תהא Λ מחלקה של מקורות סטציונריים.
לכל מקור $\theta \in \Lambda$ יש פילוג הסתברות P_θ שנותן את ההסתברות של המחרוזות הנפלטות.

תהא W קבוצה שלמה ומתאימה מעל א"ב הקלט. נגדיר:

$$n \equiv \lceil \log_2 |W| \rceil \quad , \quad R_n(W,\theta) = \frac{n}{\bar{L}_\theta(W)} \tag{4.4}$$

כאשר:

$$\bar{L}_\theta(W) \equiv \sum_{\underline{x} \in W} P_\theta(\underline{x}) L(\underline{x}) \quad , \tag{4.5}$$

כלומר $\bar{L}_\theta(W)$ הוא האורך הממוצע של מילות הקלט לקוד בהינתן שהמקור הוא θ . $R_n(W,\theta)$ הוא הקצב של הקוד W בהינתן שהמקור θ, ובהינתן שמספר הביטים למילת יציאה n.

חסם תחתון על קצב הקוד VB המוגדר על-ידי W הוא [4].

$$K(W,\theta) = \frac{-\sum_{\underline{x} \in W} P_\theta(\underline{x}) \log P_\theta(\underline{x})}{\bar{L}_\theta(W)} \tag{4.6}$$

היתירות של הקוד W יחסית למקור θ היא:

$$r_n(W,\theta) = R_n(W,\theta) - K(W,\theta) \tag{4.7}$$

© Technion - Israel Institute of Technology, Elyachar Central Library

היתירות המקסימאלית תוגדר כ :

$$\hat{r}_n(W) \equiv \sup_{\theta \in \Lambda} \{r_n(W,\theta)\} \qquad (4.8)$$

אם Y קבוצת כל קודי ה VB בעלי מילת יציאה באורך n נגדיר:

$$R_{VB}(n) = \inf_{W \in Y} \{\hat{r}_n(W)\} \qquad (4.9)$$

יתירות המינימקס.

נאמר שסדרה $w_1, w_2, \cdots$ של קודים היא אוניברסלית במובן החלש אם:

$$R_n(W_n,\theta) \underset{N\to\infty}{\rightarrow} H(\theta) \ \ \forall\theta \in \Lambda$$

(מילת היציאה של w_n היא באורך n)

כאשר $H(\theta) = \lim_{n\to\infty} H_n(\theta)$ האנטרופיה של המקור θ.

אם ההתכנסות היא במידה שווה, נאמר שהסדרה היא אוניברסלית במובן החזק.

אם $\hat{r}_n(W_n) \underset{N\to\infty}{\rightarrow} \theta$ נאמר שהסדרה אוניברסלית במובן מינימקס.

ב [4] מופיעה התוצאה הבאה.

עבור מחלקת המקורות חסרי הזכרון

$$R_{VB}(n) \le \frac{\log n}{2n} + O\left(\frac{1}{n}\right) \qquad (4.10)$$

© Technion - Israel Institute of Technology, Elyachar Central Library

ב [4] מוזכר אלגוריתם לתכנון עץ קידוד אופטימלי עבור מקור בעל סטטיסטיקה ידועה, הידוע בשם אלגוריתם Tunstall.

האלגוריתם מייצר עץ קידוד שהעלים שלו מתאימים למילות הקוד. מכל צומת פנימי יוצאים α ענפים המסומנים במספרים α...1 (בזכור α עוצמת א"ב הקלט). תהליך הקידוד בעזרת עץ זה מתחיל בשורש, עם קבלת תו x מהמקור מתקדמים אל הצומת הבא בעץ דרך הקשת עם סימון x. כאשר מגיעים לעלה פולטי הקוד המתאימה לעלה וחוזר חלילה.

מכאן נובע שלכל עלה מתאימה מחרוזת קלט יחידה $x_1 \cdots x_k$ של אותיות מקור והסתברותו של העלה היא:

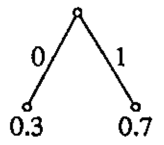
$$P_\theta(\underline{x}) = \prod_{i=1}^{k} P_\theta(x_i) \qquad (4.11)$$

<u>פירוט האלגוריתם</u>

האלגוריתם מתחיל מעץ שבו יש רק שורש עם α בנים. הוא מיצר עץ אופטימלי גדול יותר מעץ אופטימלי נתון, על ידי הרחבת העלה בעל ההסתברות הגבוהה ביותר, ב α עלים נוספים.

דוגמא: נבית מקור בינארי $A=\{0,1\}$ חסר זיכרון עם $P(0)=0.3$ ו $P(1) = 0.7$.

העץ ההתחלתי הוא:

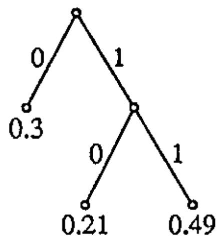


© Technion - Israel Institute of Technology, Elyachar Central Library

קל לראות ש: $\bar{L}=1$ – אורך ממוצע של מילות הקלט.

$R=\frac{\log_2 2}{\bar{L}}=1$ – קצב הקוד.

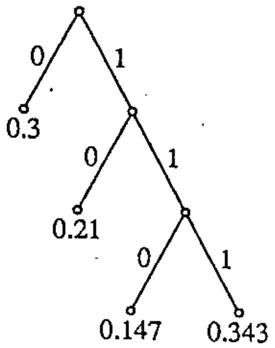
על ידי הרחבת העלה הסביר ביותר נקבל:



$$\bar{L}=0.3+2*0.7=1.7$$

$$R=\frac{\lceil \log_2 3 \rceil}{1.7}=1.176$$

על ידי הרחבת העלה הסביר ביותר נקבל:



$$\bar{L}=0.3+2*0.21+3*0.49=2.19$$

$$R=\frac{\log_2 4}{2.19}=0.913$$

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

נבנה שנתון לנו עץ בעל n עלים עם אורכי מילות קלט
$l_1,\cdots,l_n$ וההסתברויות המתאימות $p_1,\cdots,p_n$.
האורך הממוצע של מילות הקלט הוא:

$$\bar{L}_\theta=\sum_{i=1}^{n} l_i\cdot p_i \qquad (4.12)$$

נבנה בלי הגבלת הכלליות, ש p_n הוא המקסימלי מבין $p_1,\cdots,p_n$.

אם נרחיב את העץ דרך p_n נקבל ש:

$$\bar{L}_{n+1}=\sum_{i=1}^{n-1}l_i\cdot p_i+(l_n+1)\sum_{i=1}^{\alpha}p_n\cdot q_i=\bar{L}_n+p_n \qquad (4.13)$$

של העץ החדש.

כאשר q_i ההסתברות של הסימבול ה-i,
ו-α הוא גודל ה-א"ב.

מכאן ברור אינטואיטיבית מדוע כדאי לבחור תמיד בעלה בעל ההסתברות הגבוהה ביותר. בביטוי עבור קצב קוד ה- VB (ראה(4.4)), המונה תמיד עולה כפונקציה של n, ללא תלות בסטטיסטיקות המקור. לכן, אם רוצים קצב נמוך ככל האפשר, יש לשאוף ל $\bar{L}_n$ גדול בכל האפשר. על-ידי בחירת העלה בעל ההסתברות המקסימלית בכל שלב, מובטח לנו ש $\bar{L}_n$ יהיה הגדול ביותר האפשרי בכל שלב.

ב [2] מוכחת הנוסחה

$$H(W)=\bar{L}(W)H(A) \qquad (4.14)$$

כאשר H(A) אנטרופית המקור חסר הזכרון.
$\bar{L}(W)$ - האורך הממוצע של מילות הקלט.
$H(W)$ - האנטרופיה של המילון.
כמו-כן, אם לעץ יש n עלים הרי ש $H(W)\le\log_2 n$.

מהנוסחה הנ"ל ברור שרק כאשר כל העלים בעץ יהיו שווי הסתברות, (כלומר $H(W)=\log_2 n$, וכאשר לעץ יהיו 2^k עלים

©Technion - Israel Institute of Technology, Elyachar Central Library

עבור k שלם, ישתווה המצב של הקוד לאנטרופיית המקור. מכאן ברורה השאיפה לגרום לכך שכל העלים בעץ, יהיו שווי הסתברות. פיצול בכל פעם של העלה הסביר ביותר להסתברויות קטנות יותר; יש לו האפקט של הפיכת הסתברויות העלים ל"שוות יותר".

אם נתבונן כעת כיצד אלגוריתם זיו למפל בונה את עץ הקידוד, נגלה שלא מובטח לנו שבכל שלב יורחב העלה הסביר ביותר, אלא לכל עלה יש סיכוי להיות מפוצל לפי ההסתברות שלו עצמו.

לדוגמא עבור מקור בינארי עם $P(0) = 0.7$
ויהא נתון העץ ההתחלתי

יש הסתברות של 0.7 שיפוצל העלה הסביר ביותר, והסתברות של 0.3 שיפוצל העלה הלא נכון. מבחינה אינטואיטיבית, גם כאן נראה שלבסוף נקבל עץ טוב מכיוון שלעלים הכי סבירים, יש הסיכוי הגדול ביותר להתפצל.

מהשיקולים הנ"ל רואים מהי המוטיבציה לאלגוריתם זיו-למפל עם פרדיקציה. המודל של המקור משמש אותנו לשערך איזהו העלה בעל ההסתברות הגבוהה ביותר, ואותו מרחיבים בכל שלב. מובן שאם המודל לא מתאים, נרחיב עלים לא נכונים, ואז נפסיד.

© Technion - Israel Institute of Technology, Elyachar Central Library

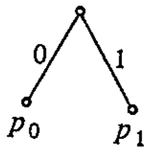
4.3 ניתוח אלגוריתם עץ אוניברסלי

ראינו בסעיף הקודם שהעץ שבונה האלגוריתם מתוך סדרת הלימוד הינו אקראי. אם נחליט שאנו רוצים עץ עם n עלים, נקבל אחד מתוך העצים בעלי n עלים מעל א"ב המקור.

פילוג ההסתברות של המקור משרה על כל עץ בקבוצת העצים בני n עלים שני גדלים. האחד הוא ההסתברות שאותו העץ יתקבל בתהליך הבניה, והגודל השני הוא הקצב שמשיג העץ יחסית למקור.

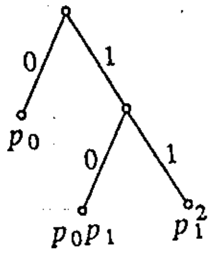
לצורך הניתוח, נניח ש-א"ב הקלט הוא בינארי ושהמקור חסר זכרון. נסמן p_0 ההסתברות שמקור יפלוט "0", ו $p_1=1-p_0$.

תהליך בניית העץ מתחיל מעץ התחלתי בעל שני עלים:

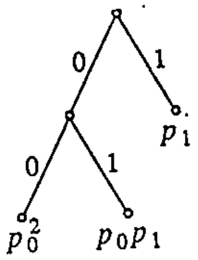


בשלב זה ישנן שתי אפשרויות.

בהסתברות p_1 יגיע "1", ואז העץ יתפתח לעץ הבא:



או שבהסתברות p_0 יגיע "0", ואז העץ יתפתח לעץ:



כלומר, ניתן להסתכל על תהליך הבניה כמו על משחק הגרלה שבו בשלב ה n ישנם n+1 תוצאות אפשריות, כל אחת עם ההסתברויות שלה. לאחר קבלת התוצאה, מתעדכן סט התוצאות האפשריות וסט ההסתברויות המתאימות.

ההסתברות לקבלת סדרת עצים מסויימת, כאשר כל עץ הוא הרחבה דרך עלה מסויים של העץ שקדם לו, היא מכפלת ההסתברויות שהיו רשומות בעלים דרכם בוצעו ההרחבות.

ייתכן, כמובן, ששתי סדרות שונות של עצים יסתיימו באותו עץ, בסופו של דבר.

נגדיר עומק ממוצע L של עץ נתון כאורך הממוצע של מילות הקלט של הקוד שהוא מגדיר. כלומר, אם לעץ מסוים n עלים והסתברויותיהם $q_1,\cdots,q_n$, והעומקים שלהם (מספר קשתות משורש לעלה) הם $l_1,\cdots,l_n$ אזי:

$$L \equiv \sum_{i=1}^{n} q_i \cdot l_i \qquad (4.15)$$

נגדיר קצב R של עץ בעל n עלים

$$R \equiv \frac{\lceil \log_2 n \rceil}{L} \qquad (4.16)$$

יש לשים לב ש R איננו יחס הדחיסה של העץ. יחס הדחיסה הממוצע של העץ הינו

$$\rho = \sum_{i=1}^{n} q_i \frac{\lceil \log_2 n \rceil}{l_i} = \lceil \log_2 n \rceil \sum_{i=1}^{n} \frac{q_i}{l_i}$$

אך מאי שוויון ינסן (Jensen) מתקבל מקעירות הפונקציה $f(x)=\frac{1}{x}$ ש:

$\rho \geq R$.

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

בציור 4.2.1 אנו רואים את המסלולים השונים האפשריים של התפתחות תהליך הבניה של העץ, עד עץ בגודל 4 עלים.

ליד כל עץ רשומים ה L ו R שלו, וכמו-כן, ההסתברות שהוא יתקבל P.

לכל אנסמבל מופיע הגודל $\bar{R}$ שהוא הקצב הממוצע שלו.

מכל עץ מופיעים חצים אל העצים שיכולים להתקבל ממנו, ועל החצים מופיעות ההסתברויות שעץ אחר יתפתח לאחר.

בהנתן אנסמבל עצים בעלי n עלים, נרשום:

$$\bar{\rho} \equiv E\{\rho\} \geq \bar{R} \equiv E\{R\} = E\,\frac{\lceil \log_2 n \rceil}{L} \geq \frac{\lceil \log_2 n \rceil}{E\{L\}} \equiv \frac{\lceil \log_2 n \rceil}{\bar{L}} \qquad (4.17)$$

הערה: אי השוויון מתקבל מאי שוויון ינסן (Jensen)- $f(x)=\frac{1}{x}$, פונקציה קעורה.

וכמו-כן כיוון ש $\rho \geq R$ לכל עץ באנסמבל. כאשר $\bar{\rho}$ — יחס הדחיסה הממוצע לאנסמבל העצים.

ניתן למצוא קשרים רקורסיביים המאפשרים לחשב את $\bar{L}$ ועל-ידי כך למצוא חסם תחתון על $\bar{R}$.

נגדיר מספר סימונים:

$P(\tau)$ – ההסתברות שהעץ τ יתקבל בתהליך הבניה.

$L(\tau)$ – העומק הממוצע של העץ τ. (ראה (4.15)).

$\bar{L}_n$ — $E\{L(\tau)\}$ על כל העצים $\tau \in T_n$.

T_r – קבוצת העצים הבינאריים המלאים (לכל צומת פנימי 2 בנים) בעלי r עלים.

$T_{r,n-r}$ – קבוצת העצים הבינאריים המלאים בעלי r עלים בתת העץ השמאלי של השורש, ו n-r עלים בתת עץ הימני של השורש.

$(\bar{L}_n \mid r,n-r)$ — $E\{L(\tau) \mid \tau \in T_{r,n-r}\}$.

כלומר העמק הממוצע רק על עצים השייכים ל $T_{r,n-r}$.

ממשפט ההחלקה נובע:

$$\bar{L}_n = \sum_{r=1}^{n-1} (\bar{L}n \mid r,n-r) \cdot Prob\{T_{r,n-r}\} \tag{4.18}$$

<u>טענה 1</u>

$$Prob\{T_{r,n-r}\} = \binom{n-2}{r-1} p_0^{r-1} p_1^{n-r-1} \tag{4.19}$$

<u>הוכחה:</u> כל עץ $\tau \in T_{r,n-r}$ ניתן לבנות באופן הבא:

1. בוחרים $\tau_1 \in T_r$ להיות תת עץ שמאלי של τ.
2. בוחרים $\tau_2 \in T_{n-r}$ להיות תת עץ ימני של τ.
3. מחליטים על סדר בו מתווספים העלים המרכיבים את τ_1 ו τ_2

לעץ τ .

לביצוע שלב 3 ישנן $\binom{n-2}{r-1}$ אפשרויות, כי צריך להוסיף n-2 צמתים פנימיים לעץ ו r-1 מתוכם הולכים לתת העץ השמאלי. כלומר, אם נסדר את כל ה n-2 צמתים בשורה, יש לבחור r-1 בלי חשיבות לסדר עבור תת העץ השמאלי ומכאן $\binom{n-2}{r-1}$. כיון שאין חשיבות לעץ הספציפי שנבחר בשלבים 1 ו 2, שכן מה שנדרש הוא שיהיה שייך ל- $T_{r,n-r}$ נקבל ש:

$$Prob\,(T_{r,n-r}) = \binom{n-2}{r-1}\cdot \sum_{\tau_1 \in T_r} P_l(\tau_1)\cdot \sum_{\tau_2 \in T_{n-r}} P_r(\tau_2) \tag{4.20}$$

כאשר $P_l(\tau_1)$ זוהי ההסתברות ש τ_1 יתקבל כתת העץ השמאלי, ו $P_r(\tau_2)$ היא ההסתברות ש τ_2 יתקבל כתת העץ הימני.

ניתן לראות ש $P_l(\tau_1)=p_0^{r-1}(\tau_1)$, וזאת כיון שההסתברויות בשלבי בנית τ_1 מוכפלות ב p_0 כאשר הוא הופך מעץ עצמאי לתת העץ השמאלי של τ . כיון שבמהלך בניתו יש r-1 הרחבות, סך הכל מוכפלת הסתברותו ב p_0^{r-1} .

מאחר ש $\sum_{\tau_1 \in T_r} P(\tau_1)=1$, נקבל ש:

$$\sum_{\tau_1 \in T_1} P_l(\tau_1) = p_0^{r-1} \tag{4.21}$$

מאותם שיקולים נקבל ש:

$$\sum_{\tau_2 \in T_{n-r}} P_r(\tau_2) = p_1^{n-r-1} \tag{4.22}$$

ומכאן מ.ש.ל.

טענה 2

$$(\bar{L}_n \mid r,n-r) = 1 + p_0 \bar{L}_r + p_1 \bar{L}_{n-r} \qquad (4.23)$$

נתבונן בעץ $\tau \in T_{r,n-r}$, נסמן את תת העץ השמאלי שלו ב τ_1 ואת הימני ב τ_2 .

נניח שעומקי העלים של τ_1 הם $l_1,\cdots,l_r$ עם הסתברויות $\alpha_1,\cdots,\alpha_r$.

ונניח שעומקי העלים של τ_2 הם $k_1,\cdots k_{n-r}$ עם הסתברויות $\beta_1,\cdots,\beta_{n-r}$ בהתאמה.

כאשר τ_1 הופך בן שמאלי של τ , העומקים גדלים ב-1 וההסתברויות בעלים מוכפלות ב p_0. באותו אופן עבור τ_2 ונקבל:

$$L(\tau) = \sum_{i=1}^{r} p_0 \alpha_i (1+l_i) + \sum_{i=1}^{n-r} p_1 \beta_i (1+k_i) \qquad (4.24)$$

$$L(\tau) = p_0 + p_0 \sum_{i=1}^{r} \alpha_i l_i + p_1 + p_1 \sum_{i=1}^{n-r} \beta_i k_i \qquad (4.25)$$

$$L(\tau) = 1 + p_0 L(\tau_1) + p_1 L(\tau_2) \qquad (4.26)$$

אם נוציא תוחלת על כל ה $\tau_1 \in T_r$ ו $\tau_2 \in T_{n-r}$ נקבל את מ.ש.ל.

מטענה 1 ו 2 נובע ש:

$$\bar{L}_n = \sum_{r=1}^{n-1} (1 + p_0 \bar{L}_r) p_0^{r-1} p_1^{n-r-1} \binom{n-2}{r-1} \qquad (4.27)$$

$+p_1\bar{L}_{n-r}$

$$= \sum_{r=1}^{n-1} p_0^{r-1} p_1^{n-r-1} \binom{n-2}{r-1} + \sum_{r=1}^{n-1} p_0^{r} p_1^{n-r-1} \binom{n-2}{r-1} \bar{L}_r + \sum_{r=1}^{n-1} p_0^{r-1} p_1^{n-r} \binom{n-2}{r-1} \bar{L}_{n-r} \qquad (4.28)$$

© Technion - Israel Institute of Technology, Elyachar Central Library

$$= 1 + \sum_{r=1}^{n-1} p_0^r p_1^{n-r-1} \binom{n-2}{r-1} \bar{L}_r + \sum_{m=1}^{n-1} p_0^{n-m-1} p_1^m \binom{n-2}{n-m-1} \bar{L}_m \qquad (4.29)$$

כיוון ש $\binom{n-2}{r-1} = \binom{n-2}{n-r-1}$ נקבל:

$$\bar{L}_n = 1 + \sum_{r=1}^{n-1} (p_0^r p_1^{n-r-1} + p_1^r p_0^{n-r-1}) \bar{L}_r \binom{n-2}{r-1} \qquad (4.30)$$

בסיס הרקורסיה הוא $\bar{L}_1 = 0$ $\bar{L}_2 = 1$.

אם נסמן $\bar{R}_n = \dfrac{\lceil \log_2 n \rceil}{\bar{L}_2}$

אזי $\bar{R}_n$ הוא החסם התחתון על הקצב הממוצע של אנסמבל העצים בגודל n. (ראה 4.17)

מתוך [2] נקבל
$$\forall \tau \in T_n \quad H(\tau) = L(\tau) H(p_0)$$

כאשר $H(\tau)$ היא אנטרופיית העץ τ , כלומר, אם $q_1, \cdots, q_n$ הם הסתברויות העלים של τ

$$H(\tau) = -\sum_{i=1}^{n} q_i \log q_i \qquad (4.31)$$

מכיוון ש $H(\tau) \le \log_2 n$ (כי לעץ n עלים) נקבל:

$$L(\tau) \le \frac{\log_2 n}{H(p_0)} \qquad (4.32)$$

וזה נכון לכל עץ τ בעל n עלים.

אם נוציא תוחלת, נקבל ש:

© Technion - Israel Institute of Technology, Elyachar Central Library

$$\bar{L}_n \le \frac{\log_2 n}{H(p_0)} \le \frac{\lceil \log_2 n \rceil}{H(p_0)} \qquad (4.33)$$

$$\bar{\rho}_n \ge \bar{R}_n \ge \vec{R}_n \ge \frac{\lceil \log_2 n \rceil H(p_0)}{\lceil \log_2 n \rceil} = H(p_0) \qquad (4.34)$$

כלומר, נקבל שערכו של החסם תמיד גדול מהאנטרופיה, ולכן רלוונטי לכל n.

הערה: ברור ש $\bar{R}_n$ חייב להיות גדול מאנטרופית המקור, וזאת מכיון שעבור כל עץ קידוד בעל מספר כלשהו של עלים, הקצב שלו, R, חייב להיות גדול מהאנטרופיה של המקור, אחרת נסתור את משפט הצפינה ללא עוות של Shannon.

את החסם ניתן לחשב באמצעות תכנית מחשב, שתחשב לכל n את ערכי $\bar{L}_i$ עבור $2 \le i \le n$.

בטכניקה זהה ניתן לקבל תשובה לשאלה אחרת. מה קורה אם עוצרים את אלגוריתם זיו-למפל כאשר העץ מגיע לגודל מסויים. מבצעים איפוס ומתחילים מהתחלה שאלה זו רלוונטית כאשר יש לנו גודל מסויים של זכרון שיכול להכיל עץ פיסוק עד גודל מקסימלי נתון.

למקור בינארי חסר זכרון ניתן למצוא חסם תחתון על יחס הדחיסה כפונקציה של הגודל המקסימלי של העץ.

נניח שאנו מתירים מספר מקסימלי של n עלים. במקרה זה קל לראות שמותר לנו לקלוט n-1 פסקאות קלט. סה"כ כמות הסיביות שתיפלט כתוצאה מקידוד n-1 הפסקאות תהיה $B(n)=\sum_{i=2}^{n}\lceil \log_2 i \rceil$.

מספר הסיבוכיות שיוקלטו הוא מ"א X, לכל עץ בעל n עלים יש X משלו. (כמות הסיבוכיות שבונה אותו).
עבור עץ נתון τ יחס הדחיסה יהיה

$$\rho_\tau=\frac{B(n)}{X_\tau} \quad \tau\in T_n \tag{4.35}$$

אם נמצע על פני אנסמבל העצים ונשתמש באי שוויון ינסן נקבל:

$$\bar{\rho}_n \geq \frac{B(n)}{\bar{X}_n} \tag{4.36}$$

כאשר $\bar{X}_n$ – ממוצע על פני מספר סיבוכיות הקלט הדרוש לבניית עץ בגודל n.
$\bar{\rho}_n$ – ממוצע על יחס הדחיסה על פני אנסמבל העצים בגודל n.

כעת נפתח רקורסיה לחישוב $\bar{X}_n$.
כמו קודם נרשום:

$$\bar{X}_n=\sum_{r=1}^{n-1}(\bar{X}_n \mid r,n-r)\,Prob\,\{T_{r,n-r}\} \tag{4.37}$$

כאשר $(\bar{X}_n \mid r,n-r)=E\,\{X_\tau \mid T_{r,n-r}\}$
X_τ – מספר הסיבוכיות הבונות את העץ τ.

מטענה 1 קבלנו כבר: $Prob\,\{T_{r,n-r}\}=\binom{n-2}{r-1}p_0^{r-1}\,p_1^{n-r-1}$

(4.38) <u>טענה 3</u> $(\bar{X}_n \mid r,n-r)=\bar{X}_r+\bar{X}_{n-r}+n-2$

© Technion - Israel Institute of Technology, Elyachar Central Library

<u>הוכחה</u>

נתבונן בעץ מסויים $\tau \in T_{r,n-r}$. נגדיר $\tau_1 \in T_r$ כתת העץ השמאלי שלו ו $\tau_2 \in T_{n-r}$ תת העץ הימני שלו.

מספר הפסקאות הנדרש על-מנת לבנות את תת העץ השמאלי הינו r-2. כי יש לו r עלים. כאשר העץ הופך מעץ עצמאי לתת עץ של τ כל פיסקה שבונה אותו מתארכת בסיבית אחת, כי נוספת לה הקידומת 0. כמו-כן, מתווספת הפיסקה 0. סה"כ, לכן, צריך r-1 סיביות נוספות.

מאותו שיקול נצטרך עבור תת העץ הימני n-r-1 סיביות נוספות, כאשר הוא הופך מעץ עצמאי לתת עץ ימני של τ. סה"כ נזדקק ל $n-r-1+r-1=n-2$ סיביות נוספות עבור כל העץ τ , מאשר נדרשות לבניית שני תתי העצים שלו כאשר הם עצמאיים. כלומר:

$$X_\tau = n-2+X_{\tau_1}+X_{\tau_2} \tag{4.39}$$

אם נמצע על פני כל אנסמבל העצים, נקבל מ.ש.ל.

מטענה 1 ו-3 נקבל:

$$\bar{X}_n = \sum_{r=1}^{n-1}(\bar{X}_r+\bar{X}_{n-r}+n-2)\,p_0^{r-1}\,p_1^{n-r-1}\binom{n-2}{r-1} \tag{4.40}$$

$$= n-2+\sum_{r=1}^{n-1}(\bar{X}_r+\bar{X}_{n-r})\,p_0^{r-1}\,p_1^{n-r-1}\binom{n-2}{r-1} \tag{4.41}$$

$$= n-2+\sum_{r=1}^{n-1}\bar{X}_r\,p_0^{r-1}\,p_1^{n-r-1}\binom{n-2}{r-1}+\sum_{r=1}^{n-1}\bar{X}_{n-r}\,p_0^{r-1}\,p_1^{n-r-1}\binom{n-2}{r-1} \tag{4.42}$$

© Technion - Israel Institute of Technology, Elyachar Central Library

על-ידי הצבה m=r-n בסכום השני, ומשיקולי סימטריה נקבל

$$\bar{X}_n = n-2+\sum_{r=1}^{n-1}(p_0^{r-1}\, p_1^{n-r-1})\bar{X}_r \binom{n-2}{r-1} \qquad (4.43)$$

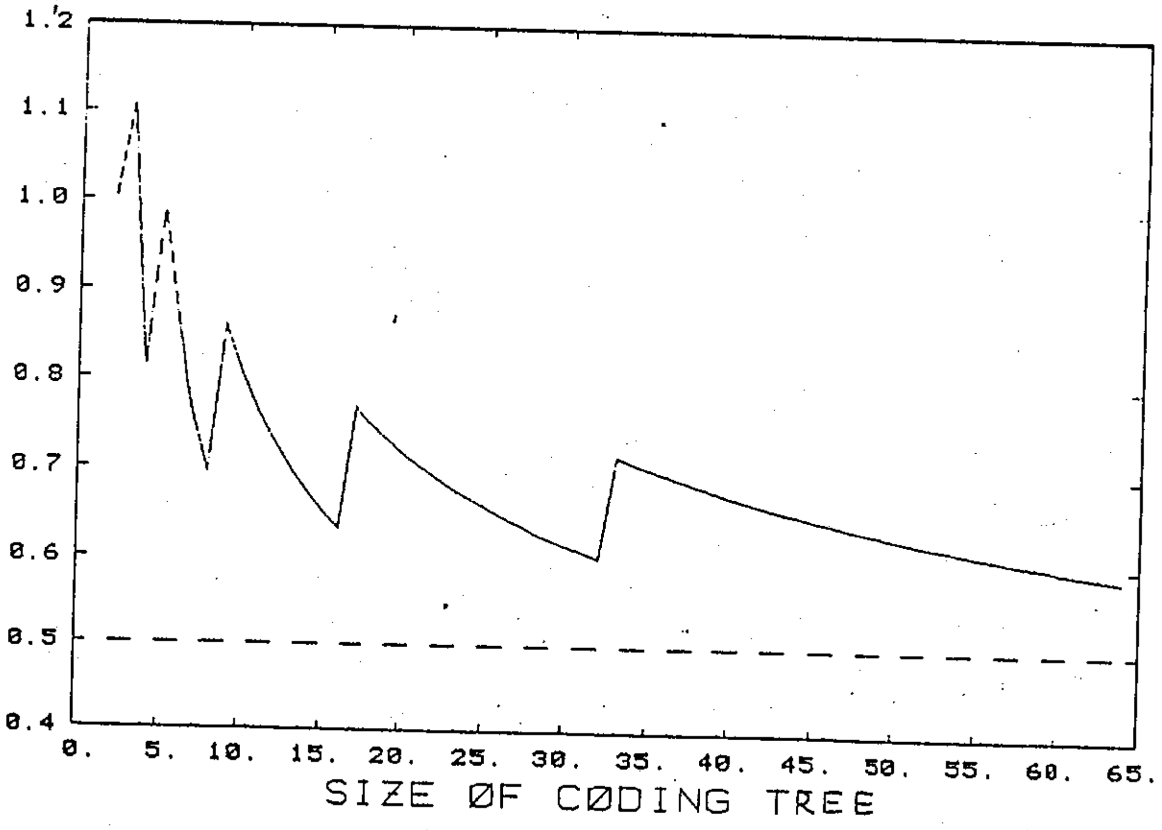
שתי הרקורסיות הורצו במחשב עבור מקורות בינאריים חסרי זכרון בעלי אנטרופיות 0.1 , 0.5 , 0.9 .

להלן הגרפים עבור המקרה של אנטרופיה 0.5 . בנספח A4 מופיעים הגרפים עבור יתר המקרים.

המינימה המקומיים בגרפים הם בנקודות בהם מספר העלים הוא מהצורה 2^k, k שלם, וזאת מכיוון שברגע שיגדל העץ, מספר הסיביות הנדרש לייצוג מילת היציאה יגדל ב-1.

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

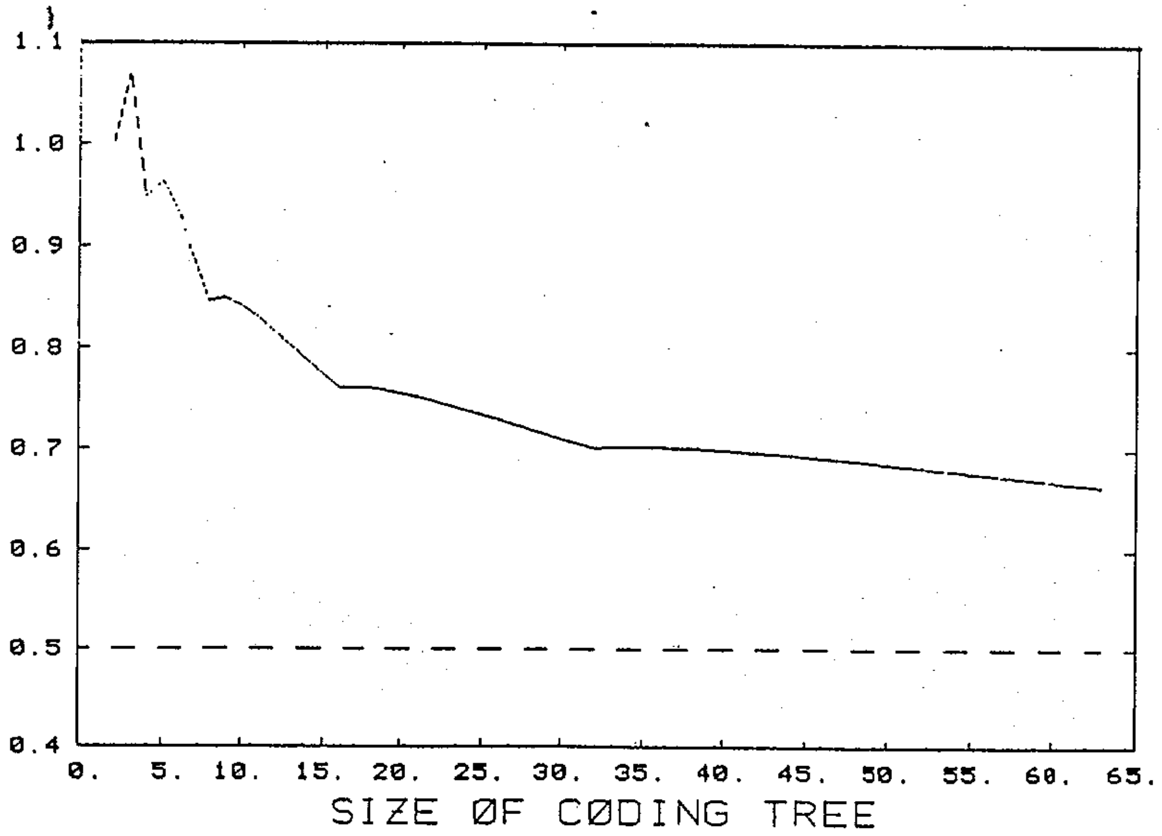


חסם תחתון על קצב ממוצע של אנסמבל עצים הנבנים

ממקור חסר זכרון בעל H=0.5 , לפי גודל העץ.

lower bound on average rate of a universal tree

ensemble built by a binary m. - less source of entropy 0.5



חסם תחתון על יחס הדחיסה הממוצע המושג על מקור
חסר זכרון עם H=0.5 כאשר העץ מוגבל בגדלו המקסימלי.

lower bound on average compression ratio obtained
a binary m - less source of entropy 0.5 when ZL coding
tree is limited in maximum size

© Technion - Israel Institute of Technology, Elyachar Central Library

4.4 בנית העץ האוניברסלי בעזרת זיו-למפל פרדיקטיבי

בסעיף זה נבחן את האפשרות להשתמש באלגוריתם זיו-למפל עם פרדיקציה (P.Z.L) לבנית העץ האוניברסלי.

מבחינה אינטואיטיבית, אם המודל מתאים טוב למקור, נקבל שיפור בקצב עץ הקידוד. ואמנם כך מתברר מתוך הסימולציות.

הסימולציות נערכו על מקורות סינטטיים (חסרי זכרון), ועל מקורות אמיתיים (מקדמי התמרות של תמונות).

בשני המקרים משתמשים בסדרת לימוד על מנת לבנות את עץ הקידוד. במקרה של מקורות חסרי זכרון, ניתן לחשב <u>במדוייק</u> את קצב העץ, כי פרמטרי המקור ידועים. במקרה של התמונות, כיוון שאין אפיון סטטיסטי של המקור, לא ניתן לחשב את יחס הדחיסה, אלא רק למדוד אותו על סדרות שנפלטו מהמקור.

ההשוואה בין ZL ל PZL נעשתה על עצים שווי גודל, בגדלים 128,64,32 וכו' עלים, שנבנו על-ידי אותה סדרת לימוד. ההשוואה מתייחסת לשני פרמטרים: קצב העץ וכמות הביטים הנדרשים על מנת לבנות אותו.

נבדקו מקורות חסרי זכרון בעלי אנטרופיה 0.1, 0.5 ו 0.9.
בציור 4.4.1 מופיעות התוצאות בגרף עבור המקרה של אנטרופיה 0.5

בגרף של היתירות, הקו השלם מתייחס ליתירות בעץ עבור בניה עם ZL. הקו המרוסק עבור בניה עם PZL. הקו המקווקד הוא היתירות

שמשיג האלגוריתם האופטימלי לתכנון העץ, אלגוריתם Tunstall. יש לזכור שאלגוריתם Tunstall דורש ידיעה של פרמטרי המקור, ולכן לא אוניברסלי. הוא מופיע כאן בתור חסם תחתון להערכת הביצועים של ZL ו PZL.

בטבלה 4.4.1 נתונות התוצאות המספריות עבור עצים בגדלים שהם חזקה של 2. ניתן לראות שעבור עצים בגודל 512 היתירות נמוכה בכ-40% , כמו-כן כמות הביטים הדרושה קטנה בבמחצית לטובת PZL.

במקרה של מקדמי הצמרות של תמונות, לא ניתן לחשב את היתירות, כי האנטרופיה לא ידועה. אך ניתן לחשב את האנטרופיות

המותנות ואמנם בטבלה 4.4.2 מופיעות האנטרופיות האמפיריות המותנות ממספר סדרים של סדרת הקלט.

הגדרתה של האנטרופיה האמפירית מסדר m היא:

$$\hat{H}(X_n \mid X_{n-1}\cdots X_{n-m}) = -\sum_{X_n\cdots X_{n-m}\in\{0,1\}^{m+1}} P^*(X_n,\cdots,X_{n-m})\log P^*(X_n \mid X_{n-1}\cdots X_{n-m})$$

כאשר P^* - הסתברות אמפירית.

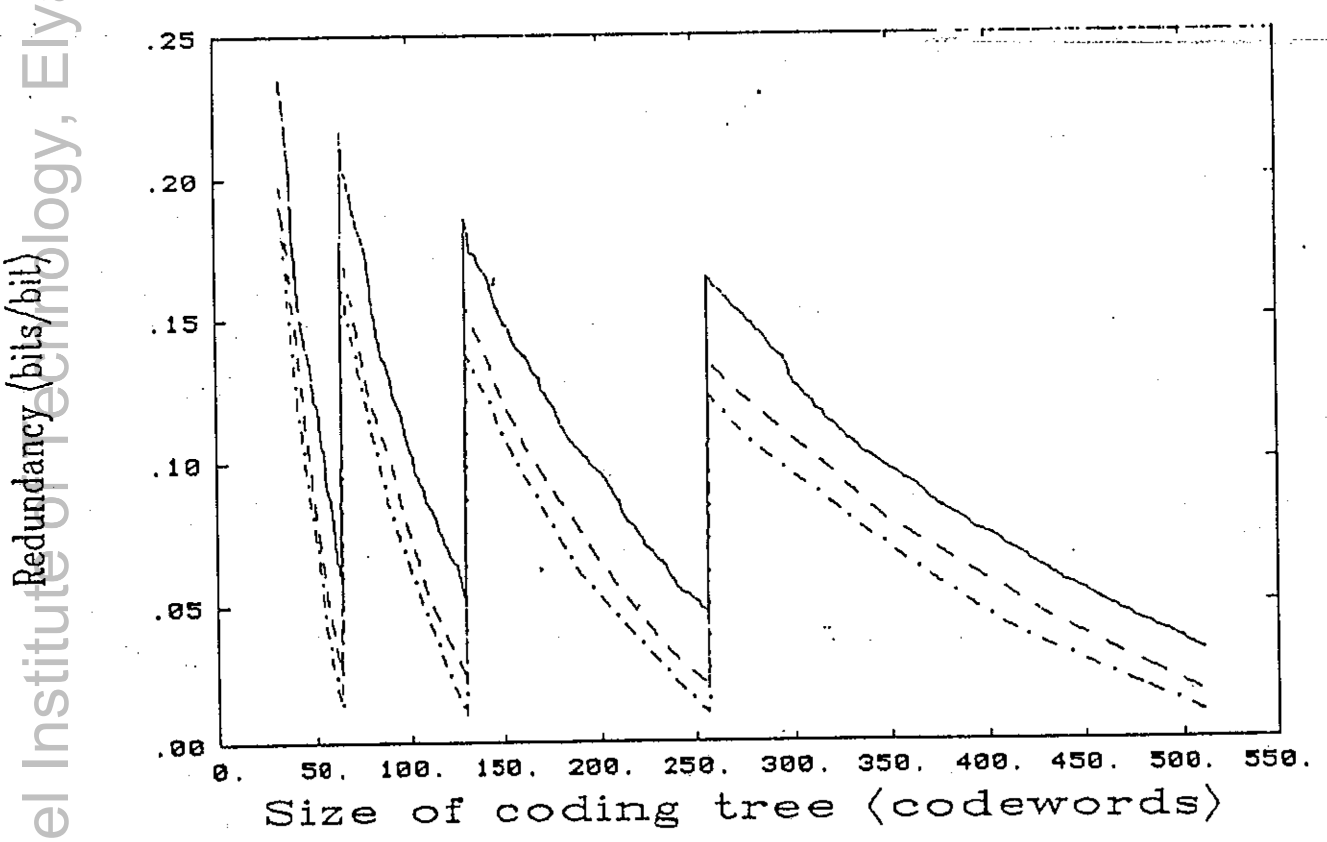
בטבלה 4.4.3 ערוכה ההשוואה בין יחסי הדחיסה שהושגו על-ידי העצים. לצורך בניית העצים על-ידי PZL הונח מודל מרקובי מסדר 10 .

אם ניקח כמדד ליתירות את יחס הדחיסה פחות האנטרופיה המותנית האמפירית מסדר 10, נקבל שעבור PZL התקבלה "יתירות" הנמוכה בכ-15% מאשר ZL בעצים של 512 עלים. הביצועים הפחות טובים נובעים כנראה מהעובדה שהמודל המרקובי מסדר 10 לא מתאים

©Technion - Israel Institute of Technology, Elyachar Central Library

למקור שאינו סטציונרי. יש לזכור שסדרת הלימוד קצרה יותר מאשר סדרת הקלט, ומהווה רישא שלה והסטטיסטיקות המאפיינות את הרישא של סדרת הקלט אינן בהכרח מאפיינות את המשך הסדרה. מאחר שהחזאי משערך פרמטרים על סמך הרישא, אין לצפות לביצועים טובים על פני כל הסדרה.

© Technion - Israel Institute of Technology, Elyachar Central Library



בניית העץ עם ZL	————	results for ZL
בניית העץ עם PZL	– – – –	results for PZL
אלגוריתם Tunstall	– · – ·	Tunstall algorithm

ציור 4.4.1

יתירות עץ הקידוד על מקור חסר זכרון H=0.5

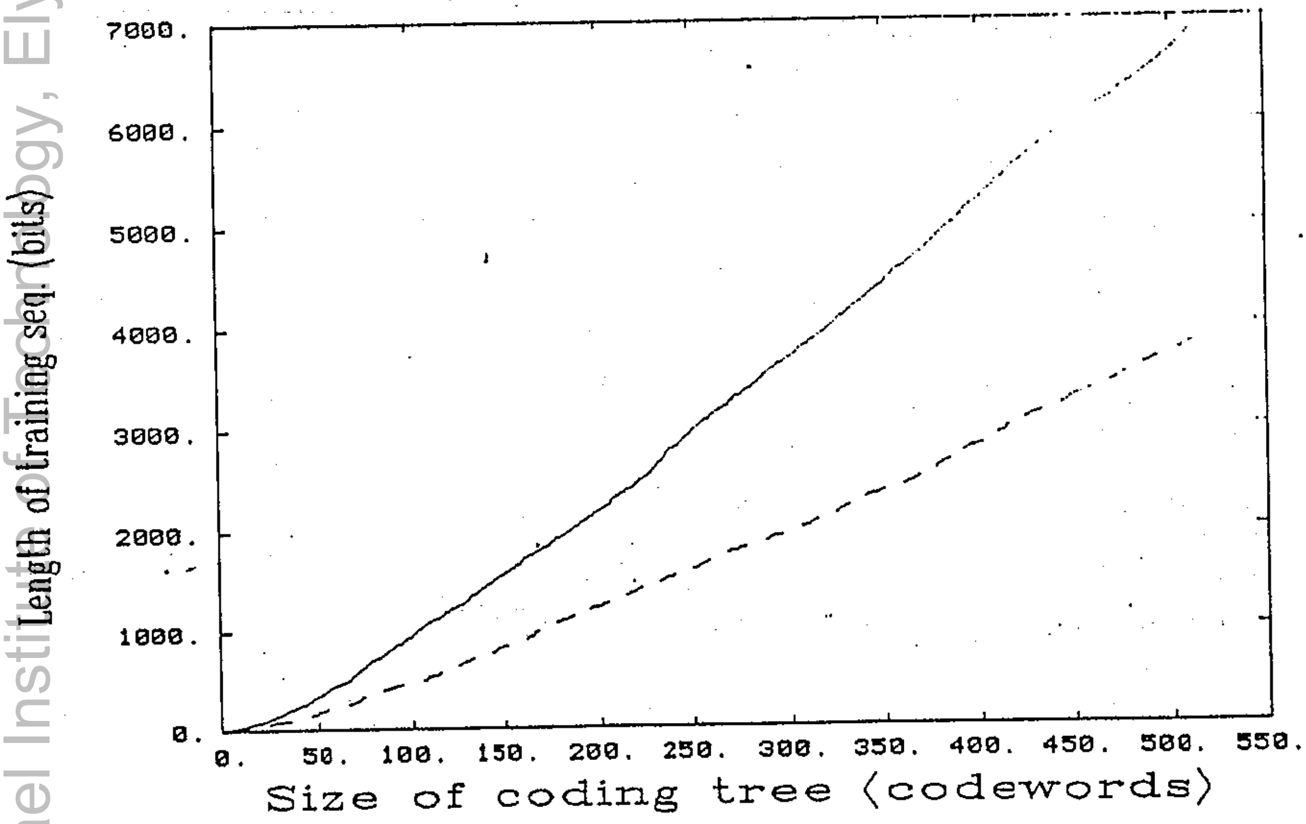
עבור אלגוריתמי בנייה שונים:

Figure 4.4.1

Redundancy of coding trees for different algorithms for binary m-less source of entropy 0.5

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library



ZL בנייה עם ------- results for ZL
PZL בנייה עם - - - - results for PZL
ציור 4.4.2
אורך סדרת הלימוד הדרוש לבניית עץ קידוד
אוניברסלי למקור בינארי חסר זכרון H=0.5
Figure 4.4.2
length of training sequence necessary for
building universal trees. for binary m-less source H=0.5

בנספח A5 מופיעים הגרפים עבור מקורות חסרי זכרון בינארי בעלי אנטרופיה 0.1 ו 0.9 .

לסכום, ניתן לראות כי החסכון המושג על-ידי אלגוריתם PZL לעומת אלגוריתם ZL בבניית עץ אוניברסלי למקורות חסרי זכרון, הוא משמעותי ביותר גם ביחס הדחיסה במושג, וגם באורך סדרת הלימוד הקצרה ביותר. עבור מקדמי התמורות, החסכון ביחס הדחיסה קטן יותר, והחסכון באורך סדרת הלימוד, עדיין קיים.

© Technion - Israel Institute of Technology, Elyachar Central Library

תוצאות עבור PZL						
מספר עלים בעץ	אנטרופיה 0.1		אנטרופיה 0.5		אנטרופיה 0.9	
	אורך	יתירות	אורך	יתירות	אורך	יתירות
32	208	0.105632	85	0.052660	60	0.026655
64	644	0.057522	240	0.06075	157	0.02561
128	2229	0.025315	680	0.046942	389	0.024301
256	6968	0.028464	1658	0.037963	923	0.021195
512	15603	0.016834	3802	0.035944	2131	0.01741

תוצאות עבור ZL						
מספר עלים בעץ	אנטרופיה 0.1		אנטרופיה 0.5		אנטרופיה 0.9	
	אורך	יתירות	אורך	יתירות	אורך	יתירות
32	400	0.124844	166	0.094812	113	0.050586
64	1233	0.07711	407	0.078376	295	0.058624
128	4207	0.04093	1292	0.074811	743	0.052182
256	11642	0.032592	3060	0.062108	1797	0.047299
512	28185	0.028982	6865	0.056927	4119	0.032682

טבלה 4.4.1
השוואה בין ZL ל PZL מקורות חסרי זכרון.
comparison of ZL and PZL for m-less sources.

© Technion - Israel Institute of Technology, Elyachar Central Library

סדר התניה	אנטרופיה אמפירית מותנית
0	0.624549
1	0.61404
5	0.582317
9	0.559579
10	0.490484

טבלה 4.4.2
אנטרופיות אמפיריות מותנות עבור סדרת מקדמי התמורות של תמונות

מספר עלים בעץ	יחס דחיסה עץ PZL	יחס דחיסה עץ ZL
32	0.726196	0.714966
64	0.659766	0.727002
128	0.634375	0.710596
256	0.624023	0.689258
512	0.607324	0.628418
1024	0.0593018	0.598877

טבלה 4.4.3
השוואה בין יחסי דחיסה המושגים על ידי עצים שנבנו ע"י ZL ו PZL עבור סדרה של מקדמי התמורות של תמונה.

table 4.4.3
comparison of compression ratios obtained on trees built by ZL and PZL for image transform coefficients

© Technion - Israel Institute of Technology, Elyachar Central Library

4.5 שיפור העץ האוניברסלי במהלך הקידוד

שיטת העץ האוניברסלי סובלת מבעיה מעצם מהותה. במקרה של מקורות סטציונריים מכיון שהעץ המתקבל מתהליך הלימוד הינו אקראי לא מובטח שהוא הטוב ביותר עבור המקור הנדחס. ראינו שבניית העץ על-ידי PZL עשוייה לשפר את יחס הדחיסה.

גישה אחרת לבעיה היא לשפר את העץ באופן שוטף במהלך הקידוד. כלומר למדוד את סטטיסטיקת המקור, ולהשתמש בהם לשיפור העץ, בלי לשנות את גודלו.

ב [3] מוצע אלגוריתם VB אדפטיבי המבוסס על האלגוריתם האופטימלי של TUNSTALL. האלגוריתם הוצע עבור מקורות לא סטציונריים המשתנים לאט. שיטת העבודה של האלגוריתם היא לספור עבור כל עלה בעץ כמה פעמים הוא מופיע ואת העלים היותר שכיחים להרחיב, על חשבון עלים פחות שכיחים, שמוצמתים מהעץ.

על סמך אלגוריתם זה נוסח אלגוריתם המבצע שפור לעץ אוניברסלי. האלגוריתם נוסה על מקורות סינטטים ומקדמי התמרות של תמונות. השיפור נוסה על עצים שנבנו על-ידי זיו למפל פרדיקטיבי.

פרוט האלגוריתם הוא כדלקמן.

מבנה נתונים : עץ קידוד אוניברסלי, כאשר לכל עלה בעץ מצמידים מונה.

© Technion - Israel Institute of Technology, Elyachar Central library

(0) אפס את כל המונים.

(1) קודד מחרוזת קלט, הגדל ב-1 את המונה של העלה שסיום
פיסוק של מחרוזת קלט נוכחית.

(2) האם המונה של אותו העלה גדול מסף N_{max}? אם לא חזור ל
1. אם כן המשך ל 3.

(3) חפש זוג עלים אחים (כלומר בעלי אב משותף) שסכום המונים
שלהם מינימלי. במקרה של יותר מאפשרות אחת בחר באקראי.

(4) הצמת זוג עלים אלו מהעץ, והפוך את האב שלהם לעלה.

(5) הוסף לעלה שהמונה שלו N_{max} שני בנים והפוך אותו על ידי
כל לצומת פנימי.

(6) לך ל 0.

בניסויים נבחר N_{max} בתחום 2-8 ונלקחו התוצאות הטובות ביותר. ה tradeoff בבחירת N_{max} הוא כזה שככל שהוא גדל נלקחת יותר אינפורמציה בחשבון בקביעת העלה הסביר ביותר וההתחלה תהיה טובה יותר. אולם מתבצעים פחות שינויים בעץ. וכמו כן עבור N_{max} נתון ככל שהעץ גדול יותר הוא יתעדכן ביתר איטיות בגלל שהמסלולים שורש - עליהם ארוכים יותר.

בטבלה 4.5.1 מופיעות תוצאות הסימולציות על סדרה בת 130,000 סיבות שנפלטו ממקור בינארי חסר זיכרון בעל אנטרופיה 0.5. המסקנה העיקרית מתוצאות אלו היא ששיפור העץ עבור מקורות חסרי זיכרון משפרת מזערית את הקידוד.

© Technion - Israel Institute of Technology, Elyachar Central Library

בעצים גדולים יחס הדחיסה הסופי תלוי במידה רבה בעץ ההתחלתי ועל כן בעצים שנבנו על ידי PZL התקבל בסופו של דבר יחס דחיסה יותר טוב מאשר בעצים שנבנו על ידי ZL.

במקרה של מקדמי התמרות שיטת שיפור העץ לא שיפרה את יחס הדחיסה באופן משמעותי ובמספר מקרים אף פגעה בו. דבר המצביע על אי יכולת האלגוריתם השבור להסתגל לשינויים בסטטיסטיקות.

המסקנה לגבי מקורות סטציונרים היא ששיפור העץ אינו תחליף ל PZL. כלומר בניית עץ על ידי ZL עד גודל מסוים, ושיפור על ידי אלגוריתם. השיפור נוסף בביצועים מאשר בנייה על ידי PZL ללא שום שיפורים נוספים.

© Technion - Israel Institute of Technology Elyachar Central Library

השיפור על עץ מ ZL	קידוד בעזרת עץ מ ZL	השיפור על עץ מ PZL	קידוד בעזרת עץ מ PZL	מספר עלים בעץ
0.551147	0.593370	0.543709	0.550079	32
0.543961	0.574631	0.543228	0.556061	64
0.537743	0.570213	0.532936	0.544579	128
0.546753	0.559753	0.533142	0.536316	256
0.549866	0.553299	0.532288	0.532562	512
0.547638	0.548706	0.531464	0.531693	1024

טבלה 4.5.1

יחסי דחיסה עבור קידוד

עם ובלי שיפור

table 4.5.1

compression ratios for coding

with and without improvement

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

פרק 5

סכום ומסקנות

בעבודה הוצעה גירסה של אלגוריתם זיו למפל בשם זיו למפל פרדיקטיבי (PZL), חסמים אנליטים לביצועים של אלגוריתם זה לא פותחו, אבל הוא הושווה אמפירית עם האלגוריתם המקורי של זיו למפל (ZL). מקורות האינפורמציה שנבדקו היו, חסר זיכרון, FSM unifilar בעל שני מצבים, ומקדמי התמרות של תמונות. בכל אחד מסוגי מקורות אלו נמצא שביצועי PZL היו טובים יותר מאשר ZL כאשר ההבדל בינהם היה גדול יותר במקורות חסרי זיכרון ו FSM ופחות במקדמי התמרות, וזאת עקב אי הסטציונריות של מקור התמונות, שאינה קונסיסטנטית עם מודל החיזוי באלגוריתם PZL.

המחיר שיש לשלם עבור הביצועים היותר טובים הוא של זמן ומקום. כפי שצוין בסוף פרק 3, הביצועים היו טובים ב 30% - 60% מאשר ZL עבור המקורות הסינטטים.

בהמשך בוצע ניתוח של אלגוריתם עץ אוניברסלי עבור מקורות חסרי זיכרון ופותחו שני חסמים תחתונים. האחד, חסם תחתון על הקצב של עץ אוניברסלי שנבנה על-ידי מקור חסר זיכרון כפונקציה של מספר העלים בו. ובשני חסם תחתון על יחס הדחיסה למקור חסר זיכרון שמשיג אלגוריתם זיו למפל כאשר יש מגבלה על הגודל המכסימלי של הזיכרון למילון, והאלגוריתם מבצע איפוס עצמי לאחר שמתמלא הזיכרון ומתחיל מחדש.

נבדקה האפשרות להשתמש באלגוריתם PZL במקום ZL לבניית העץ האוניברסלי. נמצא בניסויים לשלושת סוגי המקורות שפורטו קודם שבניית העץ האוניברסלי על ידי PZL נותנת ביצועים טובים יותר בדחיסה, ומשתמשת בסדרת לימוד קצרה יותר מאשר ZL.

כמו כן נבדק בעבודה אלגוריתם אדפטציה לעצי קידוד הלקוח מן הספרות, לצורך שיפור העץ האוניברסלי במהלך הקידוד עימו.

המסקנה מניסויים אלה היתה ששיפור העץ משפר במעט מאוד את יחס הדחיסה הסופי והסתבר שעץ התחלתי טוב יותר, עדיף על פני שיפור במהלך הקידוד, כלומר העץ שנבנה עם PZL הוא עדיף על פני עץ שנבנה עם ZL במהלך הקידוד.

בעיה תכנונית העומדת בפני המשתמש באלגוריתם זיו - למפל לבנית עץ אוניברסלי, היא איזה סדר גודל של עץ יבטיל לו ברמת סמך נתונה, יתירות הנמוכה מסף מסוים.

לצורך פתרון בעיה זו נדרשת ידיעה של התוחלת והשונות של קצב העצים. מידיעת גדלים אלו ושימוש באי-שיוויון דוגמת אי-שיוויון צ'בישב, האומר שלמשתנה אקראי X:

$$Prob\{\,|X-\bar{X}|<a\}\geq 1-\frac{\sigma_X^2}{a^2}$$

$$\sigma_X^2=E\{(X-\bar{X})^2\}$$

$$\bar{X}=E\{X\}$$

ניתן בעקרון למצוא חסם על גודל העץ n שיבטיח את הביצועים הנדרשים. בהנחה שהחסם שפותח עבור התוחלת מספיק קרוב לתוחלת עצמה. הרי שכיוון אפשרי להמשך המחקר הוא למצוא ביטוי דומה לשונות.

© Technion - Israel Institute of Technology, Elyachar Central Library

נספח A1

פענוח אלגוריתם זיו-למפל במימוש עץ פיסוק

אלגוריתם הפיענוח מופיע ב [7] נספח ד', ומפורט כאן לנוחיות הקורא:

המפענח משתמש בטבלה הנבנית בצורה אנלוגית לבניה של העץ במקודד. מפתח החיפוש בטבלה הוא מילת הקוד המגיעה מהמקודד.

מבנה הטבלה

L	SEQUENCE . . .
1	0
1	1
.	
.	
.	

העמודה L מבטאת את האורך של המחרוזת שמשמאלה בעמודה SEQUENCE.

הערך המספרי של מילת הקוד המגיעה מהמקודד מהווה את מפתח הכניסה לטבלה, וליתר דיוק, מציין את מספר השורה בטבלה המתאימה למילת הקוד.

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

שלבי הפיענוח

בעזר בסימנים הבאים:

RCW - מילת הקוד שהתקבלה.

NCW - מילת הקוד החדשה שנוצרה בעדכון הטבלה.

FPT - השורה הפנוייה הבאה בטבלה.

(0) אתחול אתחל את הטבלה עם התוכן המתאים למילות קוד 0 ו 1 (ראה ציור). קבע FPT=2.

(1) קרא מילת קוד RCW. אורכה מתקבל מראש על-ידי חישוב

(2) פלוט את המחרוזת המתאימה למילת הקוד שהתקבלה.

(3) שרשר סיבית 0 במקום בטבלה המתאים ל RCW - מילת הקוד שהתקבלה קדם את שדה ה L המתאים ב 1.

(4) הוסף מילת קוד חדשה NCW לטבלה על-ידי:

א. העתקת המחרוזת המתאימה ל RCW למקום המוצבע על-ידי FPT.

ב. עדכון שדה L של השורה המוצבעת על-ידי FPT (ה L שמה לשדה L של RCW)

ג. הפוך את הסיבית האחרונה של NCW.

ד. קדם FPT ב 1.

(5) חזור ל (1).

נספח A3

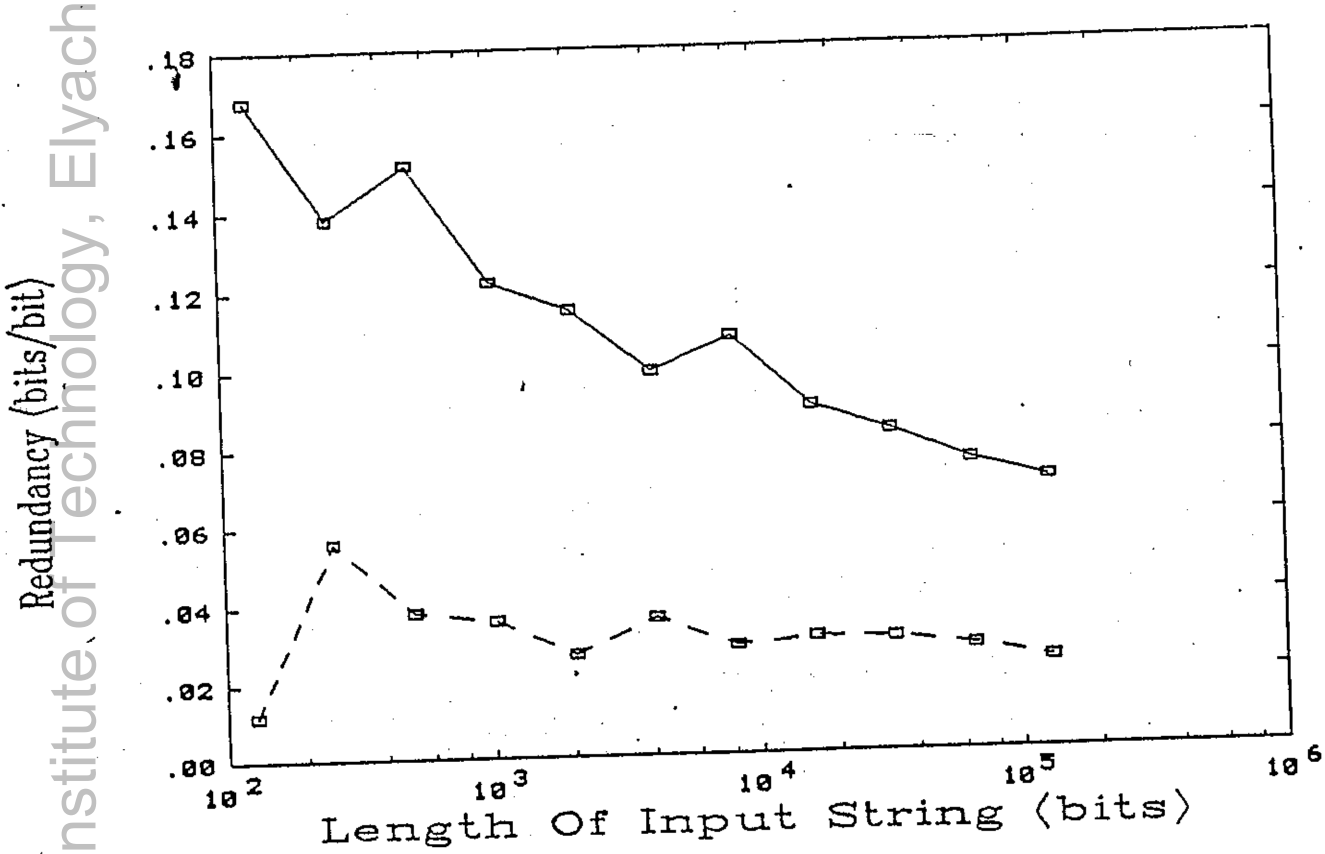
תוצאות סימולציות לאלגוריתם זיו-למפל עם חיזוי (PZL)

בנספח זה מופיעות תוצאות הסימולציות שנערכו להשוואת ביצועי FPZL (עם הפסקת PZL) חיזוי באמצע הסדרה) מול ביצועי זיו-למפל עבור מקורות חסרי זכרון, במתכונית שהוסברה בסעיף 3.3 .

להלן התוצאות המספריות והגרפיות למקורות חסרי זכרון בעלי אנטרופיות תכבון H=0.1 ו H=0.9.

בהמשך לנ"ל מופיעה טבלת התוצאות עבור ההשוואה בין האלגוריתמים, כאשר הפעם מקור האינפורמציה הוא מקדמי התמרות של התמונות.

© Technion - Israel Institute of Technology, Elyachar Central Library



השוואה בין היתירות הנקודתית ב ZL ו FPZL
עבור מקור בינארי חסר זכרון עם אנטרופית תכנון 0.9 .

comparison of redundancy for ZL and FPZL

for Bernoulli source of H = 0.9

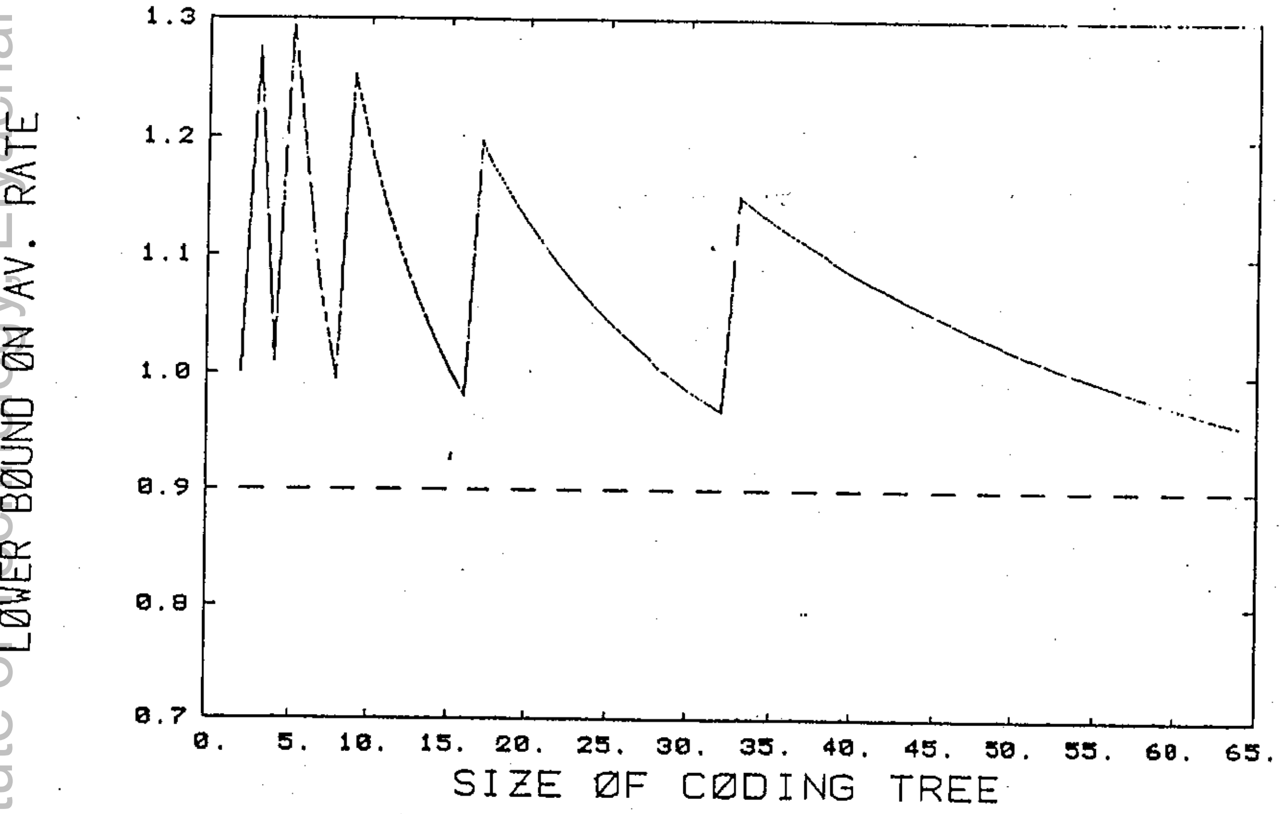
© Technion - Israel Institute of Technology, Elyachar Central Library

נספח A4

הצגה גרפית של החסמים למקורות חסרי זכרון

להלן יופיעו הגרפים של החסמים שהוצגו בסעיף 4.3 למקרים של מקורות חסרי זכרון בעלי אנטרופיה $H=0.1$ ו $H=0.9$.

© Technion - Israel Institute of Technology, Elyachar Central Library



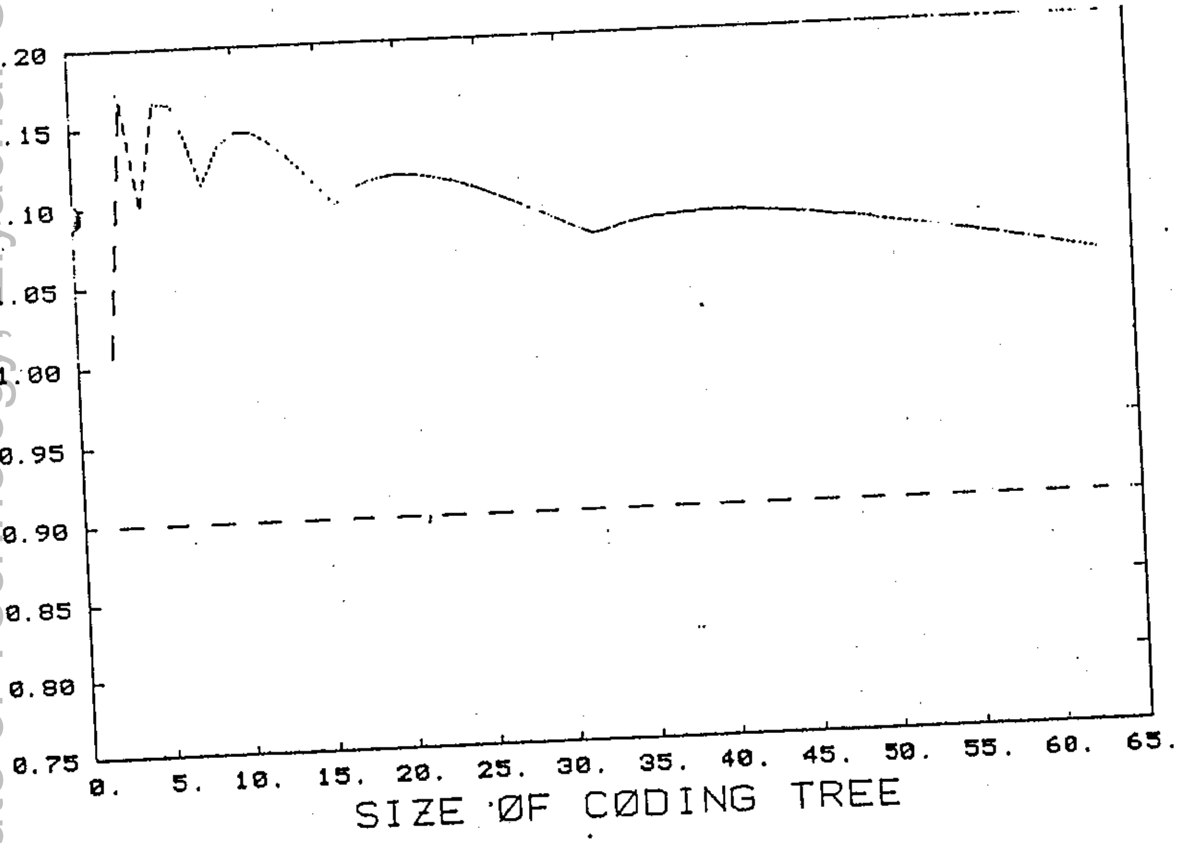
חסם תחתון על הקצב הממוצע של אנסמבל עצים אוניברסליים

הנבנים על-ידי מקור חסר זכרון בעל H=0.9.

lower bound for everage rate of an ensemble of universal

trees built by a Bernouli source of H = 0.9

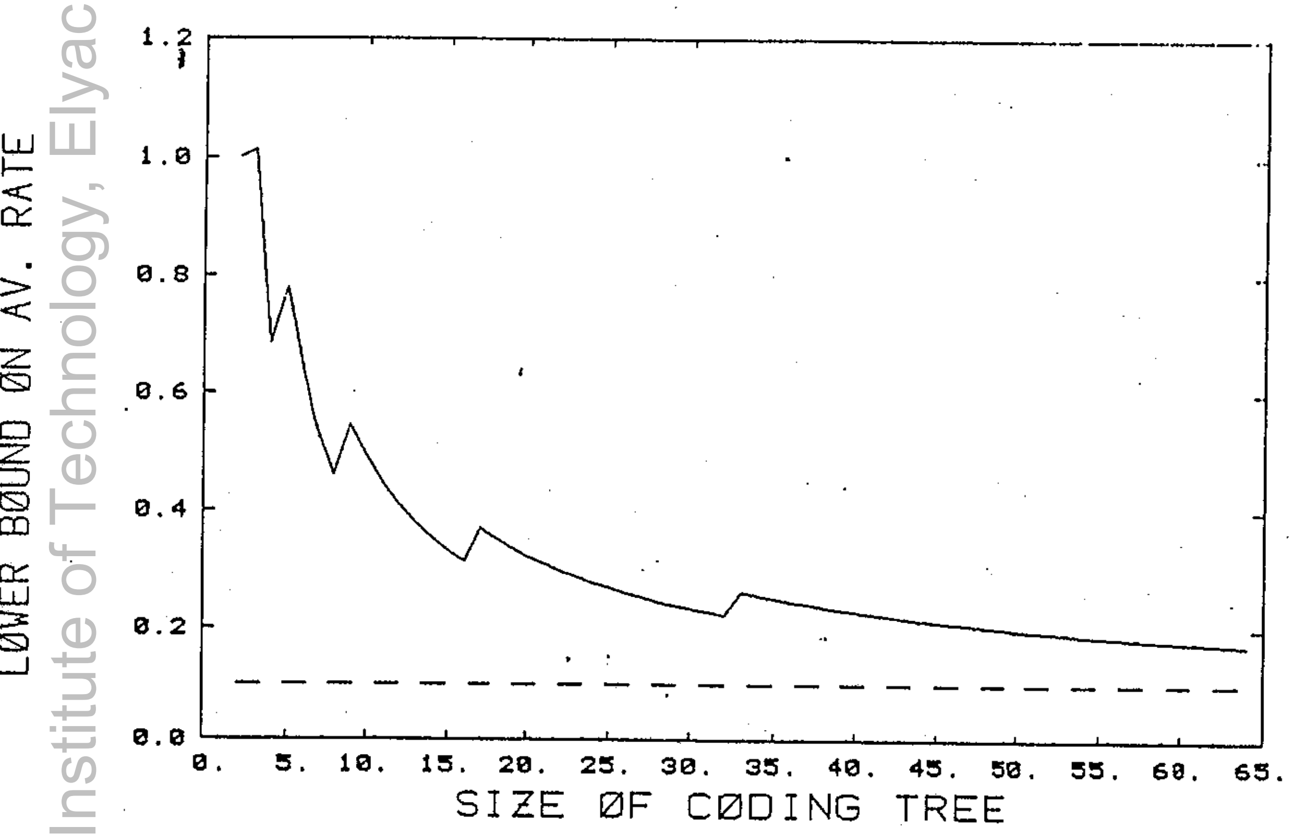
© Technion - Israel Institute of Technology, Elyachar Central Library



חסם תחתון על יחס הדחיסה המושג על-ידי אלגוריתם זיו-למפל
למקור חסר זכרון על-ידי H=0.9 כאשר עץ הקידוד מוגבל
בגודלו.

lower bound of av. compression ratio obtained by ZL algorithm for Bernoulli source of H = 0.9 when coding is limited in maximum size

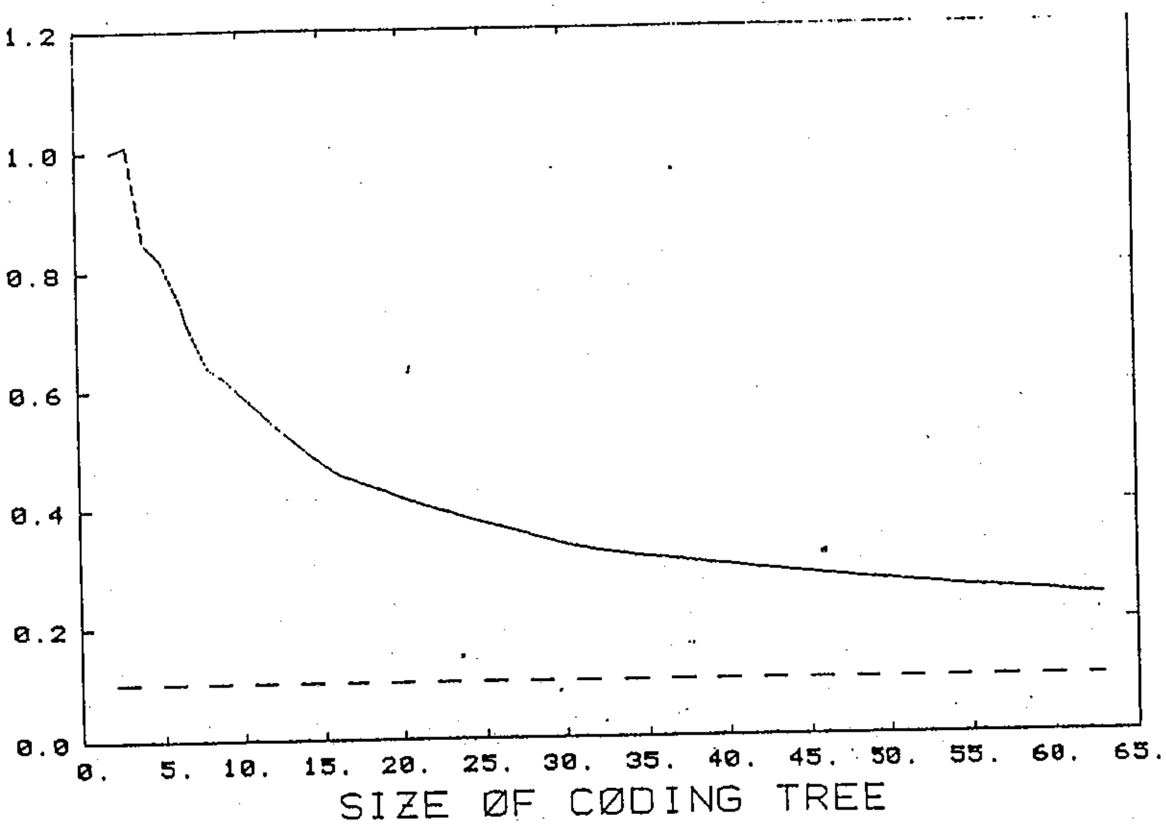
© Technion - Israel Institute of Technology, Elyachar Central Library



חסם תחתון על קצב ממוצע של אנסמבל עצים אוניברסליים הנבנים על-ידי מקור חסר זכרון בעל H=0.1.

lower bound on average rate of an ensemble of universal trees built by a Bernoulli source of H=0.1

© Technion - Israel Institute of Technology, Elyachar Central Library



חסם תחתון על יחס הדחיסה המושג על-ידי אלגוריתם זיו-למפל
על מקור חסר זכרון בעל H=0.1 כאשר עץ הקידוד מוגבל
בגודלו.

lower bound on av. compression ratio obtained by ZL algorithm for Bernoulli source of H = 0.1 when coding tree is limited in maximum size.

© Technion - Israel Institute of Technology, Elyachar Central Library

נספח A5

תוצאות גרפיות של השוואת שיטות לבניית עץ אוניברסלי

בנספח זה מופיעות התוצאות שהוזכרו בסעיף 4.4 של השוואת שיטות לבניית עץ אוניברסלי עבור מקורות חסרי זכרון.

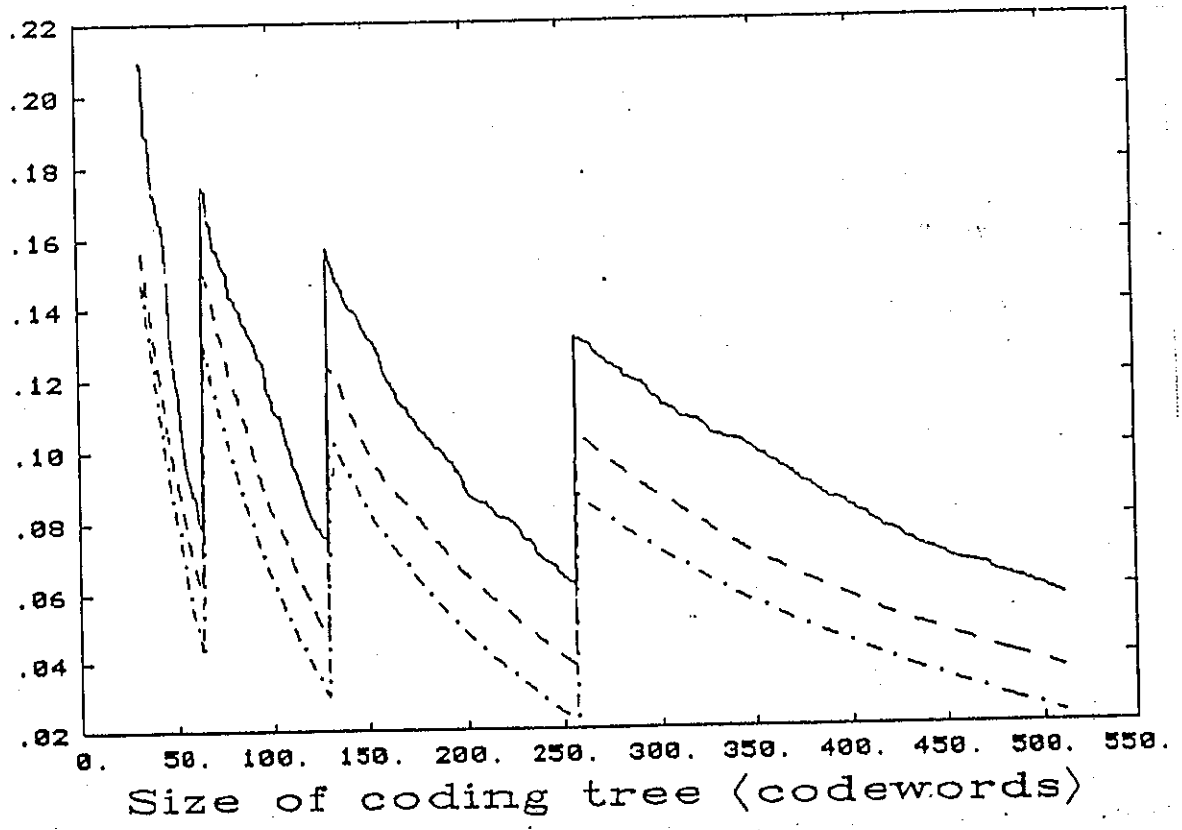
בשני הגרפים הראשונים מופיעה היתירות בעץ כפונקציה של גודלו, עבור 3 שיטות בנייה. הקו המקוקד (-.-.-.-) מציג את ביצועי אלגוריתם Tunstall לתכנון עץ כאשר הסטטיסטיקות ידועות, ומופיע כאן רק כחסם להערכת הביצועים.

הקו המקווקו (— — — —) מציג את ביצועי PZL בבניית עץ אוניברסלי. הקו השלם מציג את ביצועי זיו-למפל רגיל לבניית עץ אוניברסלי.

בזוג הגרפים השני מודגרת השוואה בין אורכי סדרות הלימוד הדרושות על מנת לבנות עץ אוניברסלי למקור חסר זכרון. הקו המקווקו (— — — —) הוא לאלגוריתם PZL, והקו השלם הוא לאלגוריתם זיו-למפל רגיל.

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

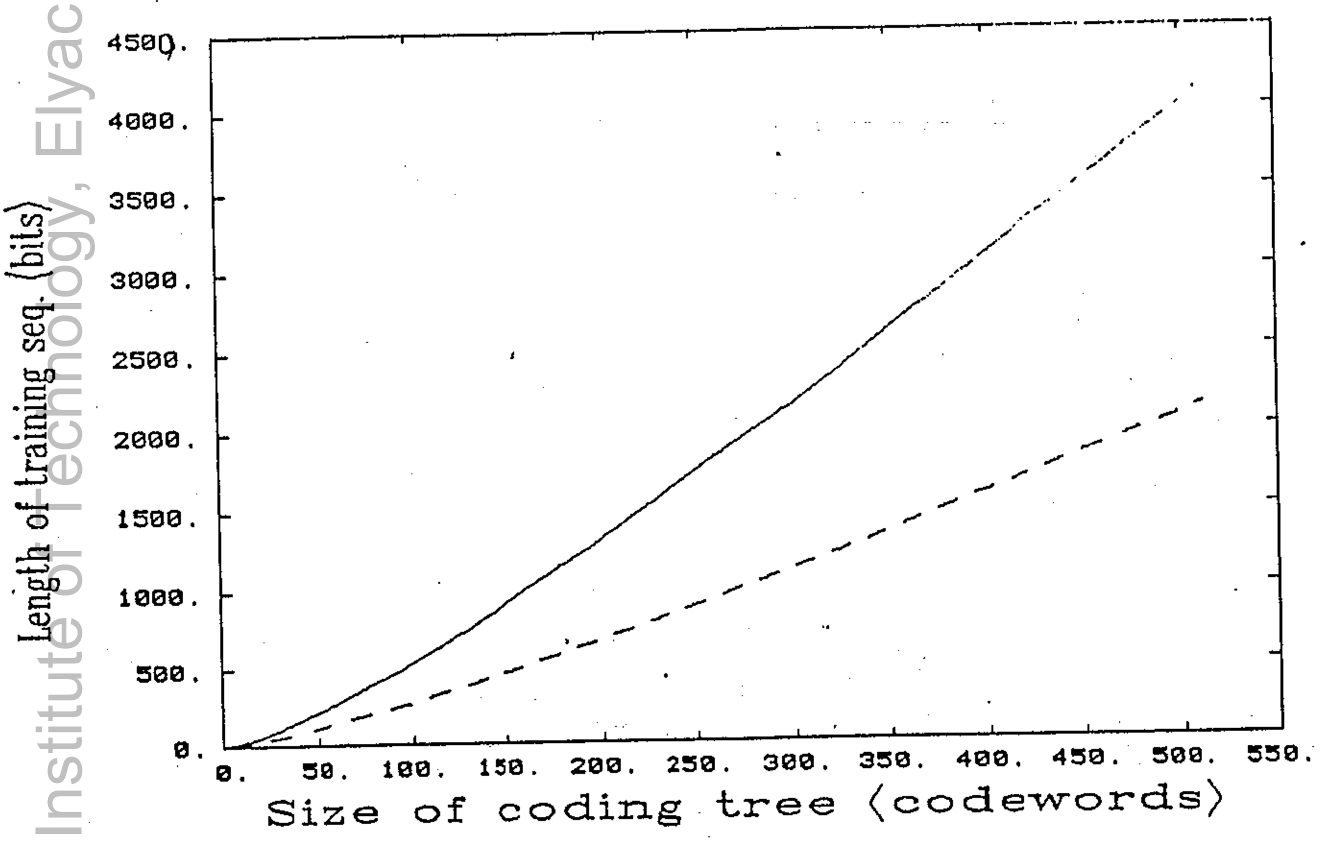


השוואת אלגוריתמים לבניית עץ אוניברסלי למקור

חסר זכרון עם 0.9

Comparison of algorithms for building universal tree

for Bernuolli source H = 0.9



השוואת גודל סדרת הלימוד הנחוצה לבניית

עץ אוניברסלי למקור חסר זכרון H=0.9

comparison of length of training sequence for building

universal trees for Bernoulli source H = 0.9

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

REFERENCES   מקורות ספרות

[1] J. Ziv and A. lempel,"Compression of Individual Sequences via Variable Rate Coding" IEEE Trans. Inform theory Vol. 24 pp. 530-536, sep. 1978

[2] F. Jelinek and K.S Schneider, "On Variable-Length-to-Block Coding" IEEE Trans. Inform. theory Vol. 18 pp 765-774, nov. 1972

[3] T. Leonardus, N.M. Mulder, D.L. Cohn, "An Adaptive Noiseless Variable-To-Fixed Encoding Algorithm for Discrete Ergodic Sources" Proc. 16th Allerton Conf. on Commun. , Control and Computing pp. 972-981, 1978

[4] M.S. Wallace, "Some Techniques in Universal Source Coding for Composite Sources". Ph.d.Thesis, University of Illinois 1982

[5] A. Lejtman, "Efficient Data-Base Storage of Images By Compression Techniques", M.Sc. Thesis Technion, I.I.T., nov. 1984

[6] E. Plotnik, "Topics in Universal Coding", M.Sc. Thesis, Technion I.I.T., feb. 1986.

[7] L.D. Davisson, "Universal Noiseless Coding", IEEE Trans. on Inform. Theory, Vol. 19 pp. 783-795, nov.1973

[8] Documantation of function "Random", on Unix 4.2 Bsd system

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

IMPROVING THE PERFORMANCE OF A UNIVERSAL CODING ALGORITHM USING PREDICTION

RESARCH THESIS

submitted is partial fulfillment of the requirements
for the degree of Master of Science
in Electrical Engineering

Amit Oren

Submitted to the Senate of the Technion - Israel Institute
of Technology

Kislev 5748 Haifa December 1987

This research was carried out in the Faculty of Electrical Engineering under the supervision of Professor David Malah

I would like to thank Prof. Malah for his dedicated guidance and help throughout the period of research. Thanks also to the Signal Processing Laboratory staff, especially to Ziva Avni.

© Technion - Israel Institute of Technology, Elyachar Central Library

Table of Contents

ABSTRACT 1
List of symbols and abbreviations 3
Chapter 1: PREFACE, STRUCTURE OF THE THESIS 4
1.1 Preface 4
1.2 Structure of the thesis 6
Chapter 2: ZIV - LEMPEL ALGORITHM 7
2.1 Preface 7
2.2 The original algorithm 7
2.3 Description of previous work 13
Chapter 3: ZIV LEMPEL ALG. WITH PREDICTION (PZL) 19
3.1 Description of the algorithm 19
3.2 Complexity of the algorithm 28
3.3 Simulations results 29
Chapter 4: UNIVERSAL TREE ALGORITHM 37
4.1 Preface 37
4.2 Variable to Block algorithms 39
4.3 Analysis of universal tree algorithm 49
4.4 Construction of the universal tree with PZL 64
4.5 Improving the universal tree while coding 72
Chapter 5: SUMMARY AND CONCLUSIONS 76
APPENDIX A1: DECODING THE ZIV - LEMPEL ALGORITHM 78
APPENDIX A3: SIMULATIONS RESULTS FOR PZL 80
APPENDIX A4: GRAPHS OF BOUNDS 82
APPENDIX A5: SIMULATIONS RESULTS FOR UNIVERSAL TREE WITH PZL 87
REFERENCES 90

© Technion - Israel Institute of Technology, Elyachar Central Library