קביעת גדלי תאים באמצעות
Simulated Annealing

עבודת גמר - תוכנית אמירים.

מוגשת ע"י יובל ירום.

בהדרכת: פרופ' דניאל להמן.

תודה

למדריכי פרופ׳ דניאל להמן על ההדרכה והעזרה בכל מהלך העבודה

ולארז נאורי, עזי עטייה וכל קבוצת הCAD בנשיונל-סמיקונדקטורס.

כללי

במסמך זה מתוארת בעיית קביעת גדלי התאים בתכנון מעגלי VLSI בסביבת תאים סטנדרטיים, מוכח כי בעיה זו היא NP-שלמה, ומתוארת תוכנית המנסה למצא קירוב לפתרון האופטימלי.

בעיית קביעת גדלי התאים הינה בעיית אופטימיזציה קומבינטורית. הקלט של הבעיה הוא מעגל לוגי ומגבלות על צורת האותות במעגל, והפלט הוא אוסף התאים המשמשים ליישום כל אחד מהשערים הלוגיים במעגל, כך שהשטח הכולל של התאים יהיה מינימלי מבלי לעבור על המגבלות הנתונות.

התוכנית משתמשת ב-Simulated-Annealing כדי לפתור את הבעיה. Simulated-Annealing הינו אלגוריתם הסתברותי המשמש לפתרון מספר רב של בעיות אופטימיזציה קומבינטוריות, והוא הופעל בהצלחה לצורך פתרון בעיות בתחום התיב״מ של מעגלי VLSI.

תוכן

1. הקדמה.

השלב המרכזי בתכנון מעגלי VLSI הוא תרגום התיאור הפונקציונאלי של מעגלים אלקטרוניים מורכבים לתיאור הפיזי שלהם. בתיאור הפונקציונאלי מאופיין המעגל באמצעות אוסף משוואות המגדירות את הקשר בין הקלטים, הפלטים ומצב הזכרון הפנימי של המעגל - מה המעגל עושה. בתיאור הפיזי מאופיין המעגל באמצעות ההרכב והגיאומטריה של השכבות השונות המונחות על שבב הסיליקון - כיצד יש לבנות את המעגל.

במהלך התרגום עובר המעגל צורת תיאור חשובה נוספת - התיאור הלוגי. בצורת תיאור זו מאופיין המעגל על ידי אוסף של שערים לוגיים והקשרים ביניהם - כיצד המעגל עובד. התרגום מהתיאור הלוגי לתיאור הפיזי של המעגל מתבצע בשלושה שלבים:

1. קביעת הגדלים (Sizing) - בשלב זה בוחרים את היישום המדוייק של כל אחד מהשערים הלוגיים המרכיבים את המעגל. כל שער יכול להיות מיושם במספר דרכים הנבדלות בתכונותיהן החשמליות, ובגודל היישום.
2. מיקום (Placement) - בשלב זה נקבע מיקומם היחסי של התאים.
3. חיוט (Routing) - בשלב זה מעבירים את המוליכים המקשרים בין התאים.

בכדי להקל על האוטומציה של תהליך התרגום פותחו מספר סביבות תכנון. אחת הסביבות החשובות היא של ספריות תאים סטנדרטיים (Standard Cells Libraries). בסביבה זו עובדים עם ספריות המכילות ישומים של השערים הלוגיים השונים. כל ישום נקרא **תא** (cell). תאים אלו הם מלבניים, וגובהם אחיד. מבנה זה של תאים מפשט את תהליכי המיקום והחיוט.

ההתאמה בין שער לבין התא המיישם אותו אינה חד-ערכית. כל שער יכול להיות מיושם באמצעות מספר תאים בעלי גדלים שונים, ובעלי תכונות חשמליות שונות. ככלל תאים גדולים מהירים יותר מתאים קטניים המיישמים את אותו שער, אולם, בשל השפעות משניות הנובעות מקיבול הכניסה של התאים, בחירת תא גדול תגרום להאטת התאים שמאחוריו.

בעיית קביעת גדלי התאים בסביבה זו היא, אם כן, בהנתן תאור לוגי של מעגל למצא הצבה של תאי ספריה במקום השערים במעגל, כך ששטחם הכולל של התאים יהיה מינימלי והמעגל יעמוד במגבלות הזמנים הנתונות.

קיימות שלוש גישות עקריות לפתרון בעיית קביעת הגדלים[1]:

1. חיפוש מקומי - בפתרונות המבצעים חיפוש מקומי (*5,12*) מתחילים מהגודל המינימלי של תאים, ובכל איטרציה מגדילים תא יחיד עד שהמעגל עומד בדרישות. שיפור קל על שיטה זו נעשה ב-(*20*) כאשר לאחר מציאת הפתרון הראשוני מבצעים פאזה של הקטנת תאים, על מנת למצא את השטח המינימלי.
2. מעבר למודל רציף - פתרונות הנובעים מגישה זו (*1,3,7,11,13,14*) מניחים קיום גדלים רציפים של תאים, ואז משתמשים בשיטות לאופטימיזציה רציפה (סימפלקס, ירידה לאורך הגרדיאנט, quasi-Newton ועוד) לצורך מציאת הפתרון. שיטה זו מתאימה לפתרון בעיית קביעת הגדלים של טרנזיסטורים מכיוון שניתן להגדילם בצורה רציפה (או כמעט רציפה). בסביבת תאים סטנדרטיים מספר התאים המשמשים לישום השערים הוא קטן, ולכן יש צורך לעגל את הגדלים המתקבלים מהאופטימיזציה לגדלים הקיימים. עיגול זה של גדלים גורר הפסד שטח, וגורם לכך שהמעגל לא תמיד עומד בדרישות הזמנים.
3. Simulated-Annealing - גישה זו נוסתה ב-(*17*) לצורך פתרון בעיית קביעת הגדלים של טרנזיסטורים. הפתרון המוצע היה נחות לעומת פתרונות הנובעים מגישה 2, אולם נראה כי הסיבה לכך היא העובדה שבעיית קביעת הגדלים של טרנזיסטורים מתאימה מאד למודל הרציף. לגישה זו יתרון ברור על גישה 1 מכיוון שאלגוריתם הSimulated-Annealing הוא הרחבה של אלגוריתם החיפוש המקומי.

Timopt הינה תוכנית מחשב שנכתבה בנשיונל-סמיקונדקטורס (ישראל) לצורך פתרון בעיית קביעת גדלי התאים בסביבה של ספריות תאים סטנדרטיים. הגישה שנבחרה הינה Simulated Annealing וזאת בגלל האיכות הירודה של פתרונות הנובעים משתי הגישות האחרות.

החידושים העיקריים ב Timopt הם:

א. השימוש ב Simulated Annealing לצורך פתרון בעית קביעת גדלי התאים בסביבה של ספריות תאים סטנדרטיים.

א. כל הפתרונות הקודמים מבצעים אופטימיזציה על מעגלים של לוגיקה צירופית, או מרשים אלמנטי סינכרון אך עם המגבלה שהאותות יצאו בזמנים ידועים מתאי הסנכרון. Timopt אינו דורש דרישות אלו (אולם הדרישה שבכל מקרה של Feed-Back יהיו אלמנטי סינכרון נשארת).

ב. מודל הזמנים שבו עובד Timopt מתחשב בצורת האותות. כל הפתרונות הקודמים מלבד (*11*) מתייחסים רק להשהייה של האותות בתוך תאים, ולא לצורת האותות.

---

1 בשל הדמיון לבעיה הנדונה מוזכרים גם פתרונות לבעיות קביעת גדלים בהן פונקצית המטרה אינה מינימיזציה של השטח תוך עמידה בדרישות זמנים.

2. Simulated Annealing

אופטימיזציה קומבינטורית מהווה תחום רחב של בעיות במדעי המחשב. אחת הדרכים הכלליות לפתרון[2] בעיות אלו היא חיפוש מקומי (Local Search) בשיטה זו מתחילים מפתרון כלשהוא במרחב הפתרונות ומשפרים אותו עד שמקבלים פתרון שלא ניתן לשפרו. חסרונותיו העקריים של האלגוריתם הם העובדה שכפי הנראה לא ניתן להגיע באמצעותו לפתרון האופטימלי בזמן פולינומי בגודל הבעייה (עבור בעיות שהן NP-קשות, ובהנחה ש$RP \neq NP$[3]), והעובדה ששאלת הסיבוכיות של האלגוריתם עדיין פתוחה (8).

אלגוריתם ה Simulated Annealing הוא הרחבה של החיפוש המקומי. אלגוריתם זה מקבל באופן אקראי (אך מוגבל) גם שינויים שאינם משפרים את הפתרון הנוכחי, מתוך תקוה לקבל בסופו של דבר פתרון טוב יותר. הרעיון סביבו בנוי אלגוריתם זה נובע מתוך האנלוגיה הקיימת בין פתרון של בעיות אופטימיזציה קומבינטורית גדולות, לבין בעיות רבות המופיעות במכניקה סטטיסטית (אנלוגיה זו היא גם המקור לשם של האלגוריתם).

אם כי התיאוריה שמאחורי האלגוריתם אינה מפותחת (ידוע רק כי בתנאים מסויימים על הבעיה ישנה התכנסות אסימפטוטית לפתרון האופטימלי), הרי שהתוצאות המעשיות מראות כי עבור מספר בעיות האלגוריתם מוצא פתרונות טובים יותר, ובזמן קצר יותר, מאשר היוריסטיקות "תפורות" לבעיה.

2.1 אופטימיזציה קומבינטורית

אופטימיזציה הינו התחום העוסק במציאת הפתרון הטוב ביותר במרחב הפתרונות של בעיה מסוימת. אחד הענפים החשובים בתחום זה הוא האופטימיזציה הקומבינטורית. ענף זה עוסק בבעיות אופטימיזציה בהן מרחב הפתרונות הוא בדיד.

לבעיות אופטימיזציה מסוימות - לדוגמא בעית ההתאמה המקסימלית ולגירסא הבדידה של בעית הזרימה

2 למלה 'פתרון' שתי משמעויות - האחת כשם כללי לאיבר במרחב הפתרונות של בעיית אופטימיזציה קומבינטורי, והשניה האלגוריתם שתפקידו למצא את הפתרון הטוב ביותר (או קירוב שלו) של בעיית אופטימיזציה.

3 בכל המשך הדיון אני מקבל את ההנחה ש$RP \neq NP$ ז.א. קיימות בעיות בNP שלא ניתן לפתרן באמצעות אלגוריתם הסתברותי בזמן פולינומיאלי בגודל הבעיה, ובפרט אני מניח כי לבעיות NP-קשות לא קיים אלגוריתם כזה. שאלה זו עדיין פתוחה אך ההנחה המקובלת כיום היא כי היא נכונה.

המקסימלית - נמצאו פתרונות טובים למדי, אך יש מספר רב של בעיות אופטימיזציה קומבינטורית, והידועה שבהן היא בעיית הסוכן הנוסע, שהן NP-קשות, דבר המעמיד בספק את האפשרות למצא שיטה מהירה לפתרונן. מאחר ובעיות אלו הן בעלות חשיבות רבה הרי שלרוב מוכנים להתפשר על טיב הפתרון בכדי לקבל אלגוריתם קירוב לבעיה המוצא פתרון קרוב למדי לפתרון האופטימלי, ובזמן סביר. שתיים מהמשפחות העיקריות של אלגוריתמי קירוב אלו הן חיפוש מקומי ו-Simulated Annealing.

בהמשך הפרק נשתמש בסימונים הבאים: מופע של בעית אופטימיזציה קומבינטורית מיוצג על ידי הזוג הסדור (R,C) כאשר R קבוצה סופית הנקראת **מרחב הפתרונות** (Solution Space) ו C היא פונקציה המתאימה לכל פתרון $i \in R$ מספר ממשי $C(i) \in \Re$ הנקראת **פונקצית המחיר** (Cost Function). המטרה היא למצא $i_0 \in R$ כך שלכל $j \in R$ יתקיים $C(i_0) \leq C(j)$.

2.2 חיפוש מקומי

אלגוריתם החיפוש המקומי (*15*) הוא, כאמור, אלגוריתם קירוב לפתרון בעיות אופטימיזציה קומבינטוריות. לצורך הפעלת האלגוריתם מניחים כי בנוסף למרחב הפתרונות ופונקצית המחיר נתון גם **מנגנון ייצור** (Generation Mechanism). מנגנון זה הינו אלגוריתם הסתברותי המקבל פתרון כלשהוא של הבעיה, ומחזיר פתרון אחר של הבעיה שאינו שונה בהרבה מהפתרון שהוכנס.

מנגנון הייצור מגדיר לכל פתרון $i \in R$ את **הסביבה** (Neighborhood) שלו - R_i שהיא קבוצת כל הפתרונות שניתן לקבל מהפעלה אחת של מנגנון הייצור על i. פתרונות אלו נקראים **השכנים** של i.

פתרון $i \in R$ נקרא מינימום מקומי ביחס למנגנון הייצור או בקיצור **מינימום מקומי** (Local Minimum) אם המחיר שלו קטן או שווה למחיר כל אחד מהשכנים שלו. $\forall j \in R_i : C(i) \leq C(j)$

בהנתן מנגנון הייצור אלגוריתם החיפוש המקומי הוא:

```
S := starting solution;
while S is not a local minimum do
   begin
      S' := a neighbor of S;
      if (C(S') < C(S)) then
         S := S';
   end;
return S;
```

אלגוריתם 1 חיפוש מקומי

לאלגוריתם זה, המשמש לפתרון מקורב של מספר בעיות אופטימיזציה קומבינטורית, יש שלושה חסרונות בולטים:

א. הפתרון המתקבל אינו בהכרח האופטימלי (ולבעיות שהן NP-קשות ההסתברות לקבלת הפתרון האופטימלי אפסית - מתוך ההנחה ש $RP \neq NP$), וכן לא ידוע הריחוק של הפתרון המתקבל מהפתרון האופטימלי.

ב. הפתרון המתקבל תלוי מאד בפתרון ההתחלתי.

ג. מספר האיטרציות עד להגעה למינימום מקומי עלול להיות גדול מאד. במקרים מסויימים (כמו אלגוריתם ה-2-opt לפתרון בעית הסוכן הנוסע) ידוע כי צריך לפעמים מספר מעריכי בגודל הבעיה של איטרציות. ההנחה הרווחת היום היא כי לא ניתן למצא מינימום מקומי ביחס לחלק ממנגנוני הייצור בזמן פולינומיאלי בגודל הבעיה (8).

כפי הנראה נובעות שלש הבעיות הללו מכך שאלגוריתם החיפוש המקומי מקבל רק שינויים המשפרים את הפתרון הנוכחי. מסתבר (לפחות מבחינה נסיונית) כי באמצעות שינוי האלגוריתם כך שיקבל (בצורה מוגבלת) גם שינויים המרעים את הפתרון הנוכחי ניתן להתגבר על הבעיות המוזכרות. אחד האלגוריתמים המבוססים על שינוי זה הוא Simulated Annealing.

2.3 Simulated Annealing

הבסיס לאלגוריתם ה Simulated Annealing הוא האנלוגיה הקיימת בין פתרון של בעיות אופטימיזציה גדולות לבין מציאת ה Ground State של מערכת פיסיקלית.

מכניקה סטטיסטית היא הענף בפיסיקה העוסק בחקר ההתנהגות של מערכות פיסיקליות מורכבות בתנאי שיווי משקל טרמו-דינמי (thermal equilibrium) ובטמפרטורה קבועה. בתנאים אלו מתקיים כי ההסתברות

שמערכת כזו תהיה במצב i בעל רמת אנרגיה E_i נתונה על ידי ההתפלגות:

$$P\{E = E_i\} = \frac{1}{Z(T)} e^{-\frac{E_i}{K_B T}}$$

כאשר T הינה הטמפרטורה של המערכת, K_B הוא קבוע בולצמן ו $Z(T)$ הוא קבוע נרמול הנתון על ידי:

$$Z(T) = \sum_i e^{-\frac{E_i}{K_B T}}$$

כשהסכום עובר על כל המצבים האפשריים של המערכת.

אחת השאלות המרכזיות במכניקה סטטיסטית היא מה קורה למערכת כאשר הטמפרטורה יורדת - לדוגמא האם המערכת קופאת או לא, ובמדה והמערכת קופאת האם יווצר גביש או זכוכית. מתוך התפלגות בולצמן מתקבל כי אם נקרר מערכת הנמצאת בשיווי משקל טרמו-דינמי היא תגיע בסופו של דבר למצב בעל האנרגיה הנמוכה ביותר (Ground state). על מנת לשמור על המערכת במצב של שיווי משקל הקירור חייב להיות איטי. אם הקירור יהיה מהיר מדי הרי שהמערכת תצא משיווי משקל טרמו-דינמי, ולכן לאחר הקפיאה לא נקבל גביש אלא זכוכית.

בשנת 1983 הבחינו Kirkpatrick, Gelatt ו Vecchi (9) באנלוגיה הקיימת בין פתרון בעיות אופטימיזציה קומבינטוריות גדולות לבין בעית מציאת ה Ground State של מערכות פיסיקליות גדולות. אנלוגיה זו (המסוכמת בטבלה 1) הובילה אותם לרעיון של הכנסת משתנה בקרה המדמה את הטמפרטורה לתהליך פתרון בעיות אופטימיזציה. האלגוריתם המקבל ממסקנה זו הוא ה Simulated Annealing (אלגוריתם 2).

אופטימיזציה קומבינטורית	מכניקה סטטיסטית
מרחב הפתרונות	מצבים אפשריים של המערכת
פונקצית מחיר	אנרגיה
פתרון אופטימלי	Ground State
מינימום מקומי	זכוכית
חיפוש מקומי	קירור מהיר
Simulated Annealing	קירור איטי

טבלה 1 אנלוגיה בין אופטימיזציה קומבינטורית למכניקה סטטיסטית.

```
S := initial solution;
T := Initial temperature;
repeat
   while (not in equilibrium) do
      begin
         S' := a random neighbor of S;
         Δ := C(S') - C(S);
         P := min(1, exp(-Δ/T));
         if (random(0,1) ≤ P) then
            S := S';
      end;
   update T;
until (Stop criterion is satisfied);
return S;
```

אלגוריתם 2 Simulated-Annealing

2.4 מודל מתימטי

המודל המתימטי המתאר את אלגוריתם ה Simulated Annealing בצורה הטובה ביותר הוא שרשראות מרקוב (*4*). שרשראות מרקוב משמשות להצגת הסתברויות למאורעות כך שהתוצאה של כל ניסוי תלויה את ורק בתוצאת הניסוי הקודם. שרשראות מרקוב מיוצגות באמצעות קבוצה של הסתברויות מותנות $P_{i,j}(k-1, k)$ כאשר לכל זוג (i,j) של תוצאות $P_{i,j}(k-1, k)$ היא ההסתברות שהתוצאה של הניסוי ה-k תהיה j כאשר נתון שהתוצאה של הניסוי ה-$k-1$ היתה i. אם $P_{i,j}(k-1, k) = P_{i,j}(m-1, m)$ לכל k ו m השרשרת נקראת **שרשרת מרקוב הומוגנית** אחרת היא נקראת **שרשרת מרקוב לא הומוגנית**.

עבור אלגוריתם ה-Simulated Annealing נסמן ב-$G_{i,j}(T)$ את ההסתברות שמנגנון הייצור יוציא את הפתרון j כאשר הוא מקבל כקלט את הפתרון i בטמפרטורה T, וב-$A_{i,j}(T)$ את ההסתברות שהאלגוריתם יקבל שינוי מפתרון i לפתרון j כאשר הטמפרטורה היא T. ונקבל:

$$P_{i,j}(T) = \begin{cases} G_{i,j}(T)A_{i,j}(T) & i \neq j \\ 1-\sum_{i \neq j} G_{i,j}(T)A_{i,j}(T) & i = j \end{cases}$$

ניתן להבחין בין שתי צורות התיחסות לאלגוריתם ה-Simulated Annealing:

א. אלגוריתם הומוגני - האלגוריתם מיוצג על ידי סדרה של שרשראות מרקוב הומוגניות (בטמפרטורה קבועה) כאשר הטמפרטורה משתנה בין שרשרת לשרשרת.

ב. אלגוריתם לא הומוגני - האלגוריתם מיוצג על ידי שרשרת מרקוב לא הומוגנית. הטמפרטורה מתעדכנת לאחר כל קבלה, או דחיה, של שינוי בפתרון הנוכחי.

ב (*18*) מראים כי באלגוריתם ההומוגני, בהנחה ש $G_{i,j}(T)$ אינו תלוי ב T (ולכן יסומן כ-$G_{i,j}$) אזי התנאים:

(a) $\forall i,j \in R: G_{i,j}=G_{j,i}$

(b) $\forall i,j,k \in R: C(i) \le C(j) \le C(k) \Rightarrow A_{i,k}(T)=A_{i,j}(T)A_{j,k}(T)$

(c) $\forall i,j \in R: C(i) \ge C(j) \Rightarrow A_{i,j}(T)=1$

(d) $\forall i,j \in R: (T>0 \wedge C(i)<C(j)) \Rightarrow 0<A_{i,j}(T)<1$

(e) $\forall i,j \in R: C(i)<C(j) \Rightarrow \lim_{T\to 0}(A_{i,j}(T)) = 0$

מספיקים בשביל להבטיח התכנסות אסימפטוטית של האלגוריתם למינימום הגלובלי.

ב-Simulated Annealing נבחר $A_{i,j}(T) = \min\left(1,\exp\left(-\frac{C(j)-C(i)}{T}\right)\right)$, ולכן מתקיימים תנאים (b), (c), (d) ו-(e). ולכן על מנת להבטיח התכנסות אסימפטוטית לאופטימום מספיק שתנאי (a) יתקיים.

באותו מקום הם מראים כי עבור המודל הלא-הומוגני אחד התנאים ההכרחיים להתכנסות אסימפטוטית הוא שקצב ירידת הטמפרטורה $T(k)$ בזמן k לא יהיה מהיר מ-$O(|\log k|^{-1})$. ולכן יש צורך במספר אקספוננציאלי של צעדים, והאלגוריתם אינו מהווה פתח לפתרון השאלה האם NP=RP.

2.5 תוצאות ישומיות

Simulated Annealing מהווה אלגוריתם בעל פוטנציאל לפתרון קל לישום, ובעל איכות גבוהה, של בעיות אופטימיזציה קומבינטורית. האלגוריתם נוסה רבות לפתרון בעיות העולות בתחום התיב"ם של מעגלי VLSI, ובעיקר עבור Placement ו-Routing (*9,19*). האלגוריתם הראה שיפור ניכר בטיב התוצאות ובזמן הריצה, לעומת אלגוריתמים קודמים ששימשו לפתרון בעיות אלו. האלגוריתם נוסה, בהצלחה לא מבוטלת, גם עבור בעיות של PLA-folding, Logic minimization ועוד. בנוסף לבעיות מתחום זה משתמשים באלגוריתם ,בין השאר, גם בעבוד

תמונות וברשתות נוירונים.

האלגוריתם הושווה גם להיוריסטיקות קיימות לפתרון בעיות NP-שלמות קלאסיות (10), כמו בעיית הסוכן הנוסע (Traveling Salesman Problem) ובעיית פיצול גרף[4] (Graph Partitioning Problem). בשני המקרים נבחר מנגנון הייצור באמצעות ההגדרה הטבעית של קרבה בין פתרונות הקיימת עבור בעיות אלו (החלפת קצוות של שתי קשתות במסלול בבעית הסוכן הנוסע, והחלפה בין קודקודים משני צדי החתך עבור בעית פיצול הגרף). המסקנה העיקרית מההשואה היא כי עבור בעיות סדורות (בעיית הסוכן הנוסע עם ערים על סריג, או בעית פיצול גרף גיאומטרי) היו תוצאות ההיוריסטיקות הקיימות טובות מאלו של Simulated Annealing. עבור בעיות לא סדורות מצא Simulated Annealing פתרונות טובים יותר, ובזמן קצר יותר, מההיוריסטיקות הללו.

4 בשני המקרים ההשואה היא עם האלגוריתמים של Kernighan ו Linn לפתרון בעיות אלו.

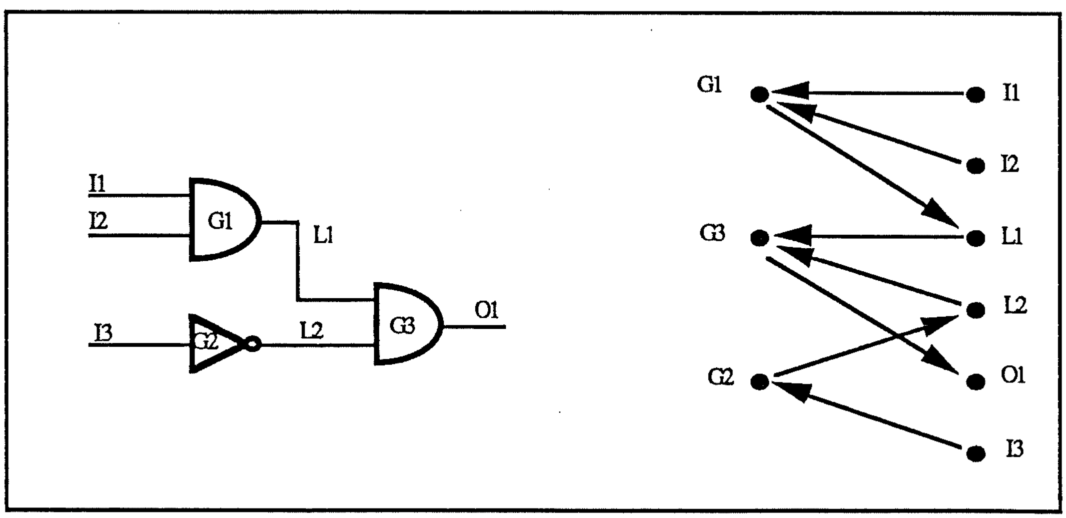
3. בעית קביעת הגדלים.

קביעת הגדלים הינו השלב הראשון בתרגום התיאור הלוגי של מעגל לתיאור הפיזי שלו. בשלב זה נבחר היישום של כל אחד מהשערים הלוגיים המרכיבים את המעגל. הבחירה נעשית מתוך מטרה לצמצם את שטחו של המעגל מבלי לעבור על דרישות זמנים נתונות.

הפרק מציג את בעיית קביעת הגדלים הכללית. בנוסף לכך מתאר הפרק את הבעיה כפי שהיא עולה בנשיונל סמיקונדקטורס ומוכיח כי היא NP-שלמה.

3.1 בעיית קביעת הגדלים הכללית.

התיאור הלוגי של מעגל VLSI הוא גרף דו צדדי מכוון $L = (C, N, P)$, כאשר C היא קבוצה של פונקציות לוגיות בסיסיות[5] - **שערים** (Gates), N היא קבוצה של **רשתות** (Nets) - כאשר כל רשת היא אוסף של נקודות אקוויולנטיות מבחינה חשמלית, ו-P (קשתות הגרף) היא קבוצה של **נמלים** (Ports) - כניסות ויציאות של השערים (שרטוט 1).



שרטוט 1 תיאור לוגי של מעגל, והמעגל המתואר.

5 ההגדרה של פונקציה לוגית בסיסית תלויה במתודולוגית התכנון, ויכולה לנוע בין פונקציה של טרנזיסטור יחיד לבין תת-מעגלים מורכבים (כגון מונים)

לכל שער נתונה קבוצה של **תאים** (Cells) - ישום אפשרי של שער - המיישמים אותו. כל התאים המיישמים את אותו שער מבצעים את אותה פונקציה לוגית, אך הם נבדלים זה מזה בשטחם, ובתכונותיהם החשמליות.

הקלט לבעיית קביעת גדלי התאים הוא תאור לוגי של מעגל, הגדרה של צורת האותות במעגל ואוסף של מגבלות על צורת האות בכל אחת מהרשתות במעגל. הפלט הוא בחירה של תא עבור כל שער במעגל כך שלכל שער מותאם אחד התאים המיישמים אותו, סכום שטחי התאים הוא מינימלי והצבת התאים במקום השערים לא תגרום לחריגה מהמגבלות הנתונות.

מודל הזמנים הוא בדרך כלל יותר פשוט מאשר ההתנהגות האמיתית של המעגל וזאת משתי סיבות:

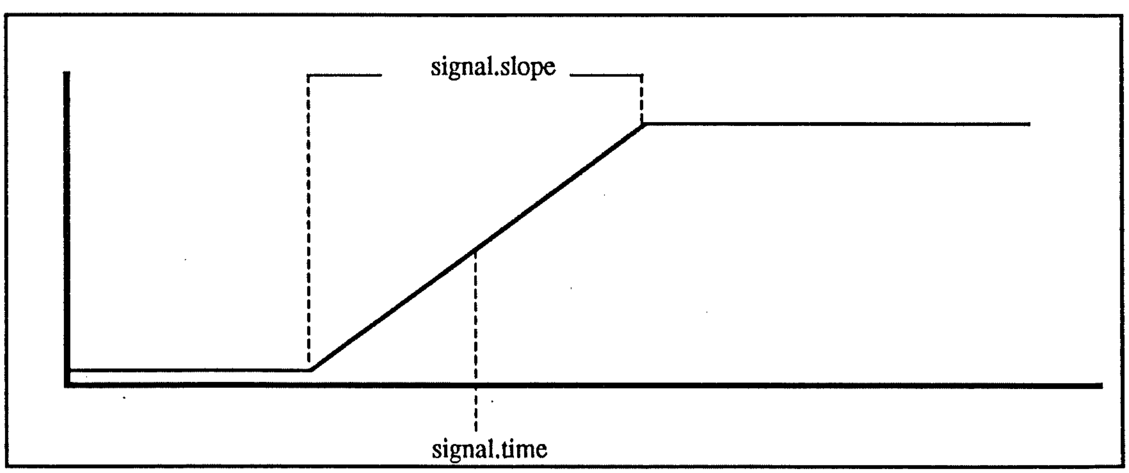
א. מאחר ולא כל פרטי היישום ידועים בזמן קביעת גדלי התאים (המיקום היחסי ואורכי המוליכים לא נקבעו עדיין) לא ניתן לתת תיאור מדוייק של התנהגות המעגל.

ב. תהליך האופטימיזציה דורש עוצמת חישוב רבה. שימוש במודל זמנים מסובך ייקר את החישוב מבלי לתרום בצורה משמעותית לכדאיות.

ברוב הפתרונות המוצעים מתיחסים למודל זמנים המניח כי השינוי באות נעשה מיידית ולכן כל אות מתואר על ידי הזמן בו עשוי להתרחש שינוי כזה.

3.2 מודל הזמנים בנשיונל-סמיקונדקטורס.

כל אות מתואר באמצעות שני פרמטרים - signal.time הוא הזמן של אמצע השינוי באות, signal.slope הוא משך השינוי. ההנחה היא כי השינוי הוא פחות או יותר לינארי (שרטוט 2).



שרטוט 2 תיאור של אות

לכל רשת יש קיבול פרזיטי המסומן בCself. לכל כניסה של תא יש קיבול Cin. הקיבול של רשת מוגדר כסכום של הקיבול הפרזיטי שלה, ושל הקיבולים של הכניסות המחוברות אליה - $C = Cself + \sum Cin$.

ההשהיה של אותות בתוך תאים עולה לינארית יחד עם השפוע של האות בכניסה, והקיבול של הרשת ביציאה של התא. גם השיפוע של האות ביציאה של התא עולה לינארית עם השיפוע של האות בכניסה, ועם הקיבול ביציאה.

המגבלות על צורת האותות במעגל הן משלושה סוגים:

א. מגבלות שיפוע - השיפוע של אותות על רשתות חסום מלמעלה.

ב. מגבלות זמנים מרומזות - מגבלות על הזמנים של האותות המוגדרות מעצם מבנה המעגל. דוגמאות למגבלות כאלו הן כניסות של D-latch, אליהן צריך האות להגיע לפני שהשער נסגר, או מכפלי פאזה (Phase multipliers) אליהם צריכים האותות להגיע לפני אות השעון.

ג. מגבלות זמנים מוצהרות - על כל רשת ניתן להגדיר זמן שעד אליו האותות חייבים להגיע לרשת זו.

כאשר נתונים האותות בכל רשתות הכניסה של המעגל ניתן לחשב את צורת האות בכל רשת במעגל בצורה הבאה: עבור כל רשת n נסמן ב in(n) את קבוצת השערים שלפחות אחת מיציאותיהם מחוברת ל n. עבור כל שער c נסמן ב in(c) את כל הרשתות המחוברות לאחת או יותר מהכניסות של c. האותות על רשתות המעגל שאינן כניסות מקיימים את הנוסחאות:

$$n.time = \min(n.time\_limit, \max_{c \in in(n)}(\max_{m \in in(c)} m.time + delay(c)))$$

$$nslope = \min(n.slope\_limit, \max_{c \in in(n)}(out\_slope(c))$$

תקינות ההגדרה נובעת מתוך הדרישה שבכל לולאה בגרף יהיו לפחות שתי רשתות עליהן יש מגבלות זמנים מרומזות, ומתוך כך שעל כל רשת יש מגבלת שיפוע.

3.3 סיבוכיות הבעיה.

בסעיף זה נראה כי בעיית ההכרעה של קביעת גדלי התאים (SIZING) היא בעיה NP-קשה, ובמקרה שהדיוק הנדרש בחישוב צורת האותות קבוע וסופי הבעיה היא שלמה ב NP. מופע של בעיה זו הוא: בהנתן תאור לוגי של מעגל חשמלי, שיפועי האותות בכניסות של המעגל, אוסף מגבלות על שיפועי האותות ברשתות של המעגל, ספריית תאים המיישמת את שערי המעגל ומספר k האם קיימת בחירה של תאים מהספריה עבור השערים במעגל כך

שמגבלות השיפוע אינן נפגעות, וסכום שטחי התאים קטן או שווה ל k. ההוכחה תהיה באמצעות רדוקציה ל FES. ברור כי אם בעיה זו שלמה בNP אז גם הבעיה בה יש לעמוד במגבלות הזמנים שלמה בNP.

בעית ה Feedback Edge Set (FES) היא הבעיה של מציאת תת גרף חסר מעגלים מכוונים מקסימאלי בתוך גרף מכוון נתון. מופע של הבעיה הוא: בהנתן זוג (G,k) כאשר G הוא גרף מכוון, ו k הוא מספר טבעי, האם קיימת תת קבוצה בגודל קטן או שווה לk של קשתות שלG כך שהגרף המתקבל מG לאחר מחיקת קשתות אלו הוא חסר מעגלים. בעייה זו היא NP-שלמה (*6*).

לצורך הרדוקציה צריך להראות תרגום של מופע (G,k) של FES למופע $f(G,k)$ של SIZING הניתן לחישוב בזמן פולינומי כך ש $(G,k) \in FES \Leftrightarrow f(G,k) \in SIZING$.

יהיו $G = (V,E)$ גרף מכוון ו k מספר טבעי. $(G,k) \in FES$ אס"ם קיימת קבוצה $U \subseteq E, |U| \le k$ כך ש-ש-$\tilde{G} = (V, E - U)$ הינו גרף חסר מעגלים. יהא $G' = (V',E')$ כאשר $V' = V \cup \{i\}$ ו $E' = E \cup \{(i,v) \mid v \in V\}$.
ברור כי $(G,k) \in FES \Leftrightarrow (G',k) \in FES$ כי i אינו משתתף באף מעגל ב G'.

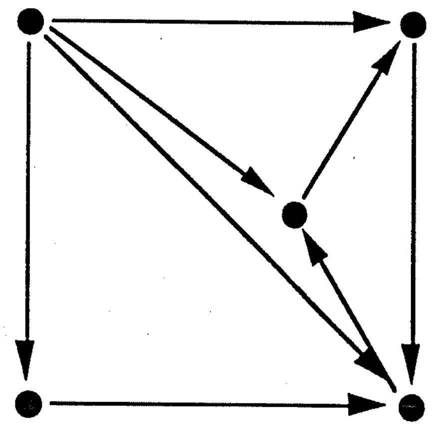
נבנה מ G' תאור לוגי של מעגל VLSI. כזכור, התיאור הלוגי מיוצג באמצעות גרף דו צדדי מכוון $L(C,N,P)$ כאשר C הינה קבוצת השערים, N היא קבוצת הרשתות, ו P היא קבוצת הנמלים - הקישורים בין השערים והרשתות. במעגל שנבנה מ-G נקח:

$$C = E'$$

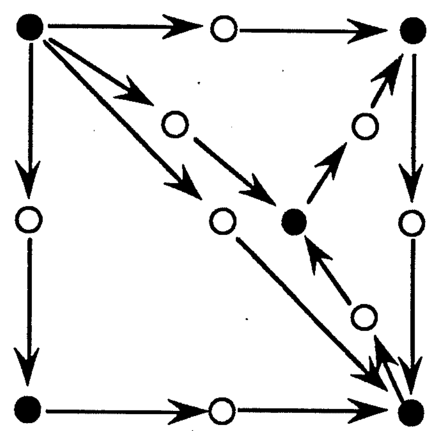
$$N = V'$$

$$P = \{(e,v) \mid e \in E' \wedge v \in V' \wedge \exists u \in V' : e=(u,v)\} \cup \{(v,e) \mid e \in E' \wedge v \in V' \wedge \exists u \in V' : e=(v,u)\}$$

לדוגמא:



יתורגם ל:



רשת הכניסה של המעגל היא זו המתאימה לו.

ספריית התאים תכלול שני תאים שלכל אחד מהם כניסה אחת ויציאה אחת. התא הראשון - C_1 - הוא בעל שטח 0, ופונקצית חישוב השיפוע ביציאה היא: out_slope = (1 + in_slope)/2. התא השני - C_2 - הוא בעל שטח 1, ופונקצית חישוב השיפוע ביציאה היא: out_slope = (1 + in_slope)/4.

השיפוע של האות בכניסה הוא 1/2. המגבלה על השיפועים של האותות ברשתות היא $1 - 2^{-2|C|}$

בצורה זו קיבלנו עבור כל מופע של FES מופע של SIZING. ברור כי ניתן לבצע תרגום זה בזמן פולינומי, ולכן על מנת להראות NP-קשיות של SIZING, נותר רק להראות כי זוהי אכן רדוקציה.

טענה:

נסתכל בהתאמה כלשהי של תאים לשערים ב L, אזי

א. השיפוע על כל רשת חסום בין 1/4 לבין $1 - 2^{-2|C|}$

ב. אם קיימת במעגל לולאה[6] שלכל שער בה מותאם C_1, אזי קיימת בלולאה רשת שהשיפוע עליה הוא $1 - 2^{-2|C|}$

ג. אם בכל לולאה במעגל קיים שער שמותאם לו C_2, אזי השיפוע בכל רשת בלולאה קטן מ $1 - 2^{-2|C|}$

הוכחה:

א. החסם העליון ברור מהגדרת השיפוע על רשת. השיפוע על הרשת i הוא 1/2 ולכן החסם התחתון מתקיים עבור i. תהי n רשת אחרת, אזי קיים תא c ש-i היא כניסה שלו ו n היא היציאה שלו. השיפוע על n חסום מלמטה על ידי $1/4 > 4/(1/2 + 1)$.

ב. נניח בשלילה שלא קיימת בלולאה רשת שהשיפוע שלה הוא $1 - 2^{-2|C|}$. תהא n הרשת בעלת השיפוע המקסימלי בלולאה. $n.slope < 1 - 2^{-2|C|}$. יהא c התא שיוצא מ n בלולאה ותהא 'n הרשת שביציאה של c. $n'.slope \geq \max(1 - 2^{-2|C|}, (1 + n.slope)/2) > n.slope$ - וזו סתירה למקסימליות השיפוע על n.

ג. נגדיר לכל רשת את העומק שלה כמספר השערים במסלול ההפוך (כנגד כיווני הקשתות בגרף) הארוך ביותר שלכל השערים בו מותאם C_1. ברור כי העומק של כל רשת קטן ממספר השערים במעגל, כי אחרת יש לולאה בה לכל השערים מותאם C_1. נראה באינדוקציה על עומק הרשת d שהשיפוע על הרשת חסום על ידי $1 - 2^{-(d+1)}$. עבור $d=0$ - לכל השערים שנכנסים לרשת מותאם C_2, ולכן השיפוע ביציאה שלהם קטן מ 1/2, ולכן השיפוע על הרשת קטן מ 1/2.

נניח כי הטענה מתקיימת עבור $d-1$ אזי עבור d - כל הכניסות של הרשת הן כאלו שלשער שלפניהן מותאם C_2 או שהעומק של הרשת שבכניסת התא ההוא קטן מ d. שערים שמותאם להם השער השני יתרמו שיפוע שקטן מ 1/2, ושאר השערים יתרמו שיפוע של $1 - 2^{-(d+1)} \geq (1 + in\_slope)/2$, ולכן הטענה מתקיימת גם עבור d.

מהטענה נובע מיידית כי $(G,k) \in FES \Leftrightarrow f(G,k) \in SIZING$, ומתוך כך שFES היא NP-שלמה נקבל כי SIZING היא NP-קשה.

העובדה שהבעיה נמצאת ב-NP נובעת מכך שמכונה לא דטרמיניסטית שתנחש את התא המתאים לכל שער, ואת השיפוע של האות על כל רשת תרוץ בזמן פולינומי, והבדיקה כי אכן אין חריגות מהמגבלות וכי השטח קטן מ k גם היא ניתנת לביצוע בזמן פולינומי. הדרישה לדיוק קבוע הכרחית לצורך ההוכחה הזו מכיוון שאחרת הניחוש של

---

6 אני משתמש במלה לולאה לתיאור מעגל בגרף המתאר את מעגל הVLSI על מנת להמנע מבלבול בין המושגים.

השיפוע על הרשת עלול לקחת זמן שאינו פולינומי באורך הקלט. לא ידוע לי כעת אם כאשר הדיוק הוא אחד הקלטים של הבעיה האם היא עדיין בNP או לא.

4. תאור הפתרון.

יישום של אלגוריתם ה Simulated Annealing דורש תכנון זהיר של ארבעה מרכיבים עיקריים:

(1) בחירת מרחב הפתרונות של הבעיה.

(2) בחירת מנגנון הייצור.

(3) קביעת פונקציית המחיר.

(4) קביעת הפרמטרים של האלגוריתם.

פרק זה מתאר את ארבעת המרכיבים הללו כפי שהם באים לידי ביטוי בTimopt.

4.1 מרחב הפתרונות.

מרחב הפתרונות שנבחר הינו אוסף כל הישומים האפשריים של המעגל. מרחב זה כולל הרבה פתרונות שאינם עומדים במגבלות הנתונות, אולם יתרונו בכך שכל מאד להגדיר מנגנון ייצור מהיר. את הפתרונות הילא חוקיים ניתן לנפות באמצעות הטלת קנס על פתרונות כאלו בפונקציית המחיר.

4.2 מנגנון הייצור

מנגנון הייצור צריך להבחר כך שתהיה אפשרות להגיע באמצעותו מכל פתרון לכל פתרון אחר, כמו כן, על מנת להבטיח התכנסות אסימפטוטית לפתרון האופטימלי רצוי לבחור מנגנון ייצור בלתי תלוי בטמפרטורה[7], שבו ההסתברות למעבר מפתרון i לפתרון j זהה להסתברות של המעבר ההפוך.

מנגנון הייצור שנבחר ב Timopt בוחר שער אקרעי במעגל, ובוחר לו יישום חדש. מנגנון זה אינו דורש חישוב רב ומבטיח את ההתכנסות האסימפטוטית לפתרון האופטימלי.

---

7 למרות זאת ראה (19)

4.3 פונקצית המחיר.

פונקצית המחיר נבחרה כך שהיא תנפה פתרונות לא חוקיים. הפונקציה היא:

$$C(S) = Area(S) + \alpha \bullet late(S) + \beta \bullet slope\_exc(S)$$

כאשר α ו β קבועים, ו:

$$Area(S) = \sum_{cell} cell.area$$

$$late(S) = \sum_{net} \max(0, net.time - net.time\_limit)$$

$$slope\_exc(S) = \sum_{net} \max(0, net.slope - net.slope\_limit)$$

לצורך חישוב net.time וnet.slope משתמשים באלגוריתם מונע מאורעות (אלגוריתם 3). עבור כל תא במעגל מגדירים את המאורע של 'השתנה האות על אחת או יותר מרשתות הכניסה של התא'. בתחילת החישוב 'מדליקים' מאורע על כל התאים שאחת הכניסות של המעגל היא כניסה שלהם. בכל שלב בוחרים תא שעבורו יש מאורע 'דלוק', ומחשבים את צורת האותות ביציאות של התא. אם האותות ביציאות השתנו מדליקים מאורע על כל התאים שמחוברים ליציאות אלו.

```
initialize();
while there are raised events do
   begin
      C := a cell for which an event is raised;
      lower event on C;
      foreach output net of C do
         begin
            compute net.time and net.slope;
            if changed net.time or net.slope then
               raise event on each output cell of net;
         end;
   end;
```

אלגוריתם 3. חישוב צורת אותות על רשתות.

ברור כי במדה והאלגוריתם מתכנס הרי שמצאנו את צורת האותות על הרשתות.

התכנסות שהזמנים והשיפועים מובטחת באמצעות איתחולם (עבור רשתות שאינן כניסות של המעגל) ל $-\infty$ ואז הזמן והשיפוע המחושבים לכל רשת הם פונקציה מונוטונית עולה של זמן הריצה, ולכן תמיד קטנים או שווים לזמן ולשיפוע על הרשת (כפי שהוגדרו בסעיף 3.2), ומכאן שהאלגוריתם מתכנס.

4.4 הפרמטרים של האלגוריתם.

באלגוריתם ה Simulated-Annealing כפי שהוצג בפרק 2 הושארו ארבעה פרמטרים של האלגוריתם פתוחים. ארבעת הפרמטרים הללו הם הטמפרטורה ההתחלתית, זיהוי שיווי המשקל הטרמודימנמי, דרך עדכון הטמפרטורה וקריטריון הסיום. בספרות מתוארות גישות רבות לקביעת פרמטרים אלו. נראה כי עבור בעיית קביעת הגדלים מספיקות השיטות הפשוטות.

הטמפרטורה ההתחלתית נקבעה כך שההסתברות ששינוי יתקבל תהיה גדולה מ χ - פרמטר בקרה הנקבע על ידי המשתמש. לצורך חישוב הטמפרטורה מחשבים את $\overline{\Delta C}$ - גודל השינוי הממוצע (לרעה) במחיר, ובוחרים $T_0 = \overline{\Delta C}/\ln \chi^{-1}$. ערך ברירת המחדל של χ הוא 0.8.

מניחים כי לאחר שהתקבלו $L \bullet N$ שינויים אנו נמצאים בשיווי משקל טרמודינמי. כאשר N הוא הגודל של הסביבה (גודל זה אינו תלוי במצב הנוכחי), ו L הוא פרמטר הנקבע על ידי המשתמש. בטמפרטורות נמוכות, כאשר המצב הנוכחי הוא מינימום מקומי, ההסתברות לקבלת שינוי היא נמוכה מאד, ולכן על מנת למנוע מצב של ניסיונות כושלים רבים מדי מוגבל גם מספר ניסיונות השינוי המבוצעים.

עדכון הטמפרטורה נעשה על פי הנוסחה $T_k = \alpha(\alpha R + 1 - R)T_{k-1}$, כאשר α הינו פרמטר הנקבע על ידי המשתמש $(\alpha < 1)$, ו-R הינו היחס בין מספר השינויים שהתקבלו לבין מספר השינויים שנוסו בשרשרת[8] ה-k-1.

האלגוריתם עוצר כאשר גודלו של השינוי המקסימלי בשרשרת האחרונה היה שווה להפרש בין מחירי הפתרון הטוב ביותר והפתרון הרע ביותר בשרשרת. מצב זה אינו סביר בטמפרטורות גבוהות, אולם ככל שהטמפרטורה יורדת הוא נעשה סביר יותר ויותר. לצורך מציאת המינימום המקומי הקרוב מבוצע חיפוש מקומי (הנסיון מראה כי מינימום זה נמצא במרחק של הפעלות בודדות של מנגנון הייצור).

8 אני משתמש במודל החומוגני לתיאור התנהגות האלגוריתם.

5. תוצאות ריצה.

5.1 השפעת הפרמטרים על האלגוריתם.

בטבלאות 2 ו 3 מסוכמים זמני הריצה וטיב התוצאות של ריצות Timopt עבור ערכים שונים של הפרמטרים של האלגוריתם. התוצאות מהוות ממוצע של חמש או יותר ריצות עם אותם פרמטרים. כל הריצות נעשו על מחשב Sun3-60. בטבלה 2 עדכון הטמפרטורה נעשה על פי הנוסחה $T_k = \alpha T_{k-1}$, ובטבלה 3 על פי הנוסחה $T_k = \alpha(\alpha R + 1 - R)T_{k-1}$. כל כניסה בטבלה היא מהצורה זמן/שטח כאשר הזמן נמדד בשניות, והשטח במיקרון-מרובע.

α \ L	1	1.5	2
0.9	141285/1163	140983/2136	140983/2091
0.8	141760/697	141264/1215	141220/1255
0.6	142105/549	141544/883	141738/814

טבלה 2.

α \ L	1	1.5	2
0.9	141609/869	141157/1480	141202/1500
0.8	141608/623	141458/973	141651/938
0.6	141911/515	141756/735	141600/684

טבלה 3.

ההשפעה של ערכו של L על טיב התוצאות הולכת וקטנה ככל שקצב ירידת הטמפרטורה קטן. הסיבה לכך היא שכאשר הטמפרטורה יורדת מהר אין משמעות לשאלה האם המערכת הייתה בשיווי משקל טרמו-דינמי לפני השינוי. התברר כי עבור ערכי α גדולים מ 0.9 השיפור בתוצאות קטן מאד, בעוד שזמן הריצה גדל מאד. ערכי ברירת המחדל נבחרו כ $\alpha=0.9$ ו $L=1.5$. המשתמש רשאי, כאמור, לשנות ערכים אלו לאחרים.

5.2 השוואה עם CDA.

CDA הינה התוכנית ששמשה לפתרון בעית קביעת גדלי התאים בנשיונל סמיקונדקטורס לפני כתיבת Timopt. האלגוריתם עליו CDA מתבסס יוצא מההנחה שהגדלים רציפים, ומשתמש באלגוריתם הירידה לאורך הגרדיאנט לביצוע אופטימיזציה לא לינארית. בסיום הריצה CDA מעגל את התוצאות לגדלים קיימים. בטבלה 4 מסוכמות תוצאות השוואה בין Timopt ל CDA. זמן הריצה נתון בשניות, השטח במיקרון-מרובע והחריגות בננו-שניות.

		Timopt				CDA			
מעגל	מספר שערים	זמן ריצה	שטח	חריגות זמן	חריגות שיפוע	זמן ריצה	שטח	חריגות זמן	חריגות שיפוע
xrspc	36	151	28250			294	28250		
xecnt	48	170	45358		0.3	687	45341		0.3
xicni	57	141	38224			519	38361		
xicmp	58	162	50409		0.1	2498	50674	0.2	0.1
liqc	81	248	71536			2723	78404		
lfifc	82	731	87869		0.6	נכשל			
lrdlg	82	400	64281			7780	65448	0.9	0.1
lrwrq	83	669	76377			21835	90590		
lreqg	93	906	81563	0.2		נכשל			
xtptm	193	2349	170424			29851	178525	0.2	

טבלה 4. השוואה בין Timopt ל CDA.

6. סיכום.

מטרת העבודה היתה בדיקת ישימות אלגוריתם ה Simulated-Annealing לפתרון בעית קביעת גדלי התאים בסביבה של ספריות תאים סטנדרטיים. נמצא כי האלגוריתם מתאים לפתרון הבעיה, ונותן תוצאות טובות יותר מאלגוריתמים קודמים, תוך הקטנה משמעותית של זמן הריצה.

לצורך העבודה נכתבה התוכנית Timopt המשתמשת באלגוריתם זה לצורך פתרון הבעיה. Timopt נכתב כהרחבה לתוכנית Timest ששמשה לביצוע הערכות זמנים על מעגלי VLSI. במהלך ההרחבה הותאם Timest למודל הזמנים הדרוש, והוסף לו מנגנון ה Simulated-Annealing.

נראה כי האלגוריתם מתאים לפתרון בעיית קביעת גדלי התאים בסביבה של תאים סטנדרטיים יותר מאשר אלגוריתמים המבוססים על שיטות אופטימיזציה לא לינאריות. כאשר האלגוריתם מושווה לחיפוש מקומי מתברר כי, כצפוי, זמן הריצה של Simulated Annealing גדול בהרבה מזה של חיפוש מקומי, אך הפתרונות המתקבלים טובים יותר. ההחלטה מי מהאלגוריתמים עדיף תלויה מאד בדרישות - אם זמן הריצה קריטי הרי שרצוי לבצע חיפוש מקומי, אך אם טיב הפתרונות חשוב יותר עדיף לבצע Simulated Annealing.

חסרונו העיקרי של האלגוריתם הוא בכך שהבסיס התיאורטי שלו אינו רחב, ואין כל הוכחה (מלבד הוכחה ניסויית) כי יש לו עדיפות על פני אלגוריתמים אחרים. יתרונו העיקרי בכך שהוא מאפשר מציאת פתרונות סבירים בזמן מהיר, ומתאים למגוון רחב של בעיות.

ביבליוגרפיה

[1] Berkelaar, M.R.C.M., and Jess, J.A.G.: "Gate Sizing in MOS Digital Circuits with Linear Programming", *Proc. 1st EDAC* (1990) pp. 217-221.

[2] Chan, P.K.: "Algorithms for Library-Specific Sizing of Combinatorial Logic", *Proc. 27th DAC* (1990) pp. 353-356.

[3] Cirit, M.A.: "Transistor Sizing in CMOS Circuits", *Proc 24th DAC*, (1987) pp. 121-124.

[4] Feller, W.: *An Introduction to Probability Theory and its Aplications vol 1*, 3rd edition, John Willey and Sons Co., 1968.

[5] Fishburn, J.P., and Dunlop, A.E.: "TILOS: A Posynomial Programming Approach to Transistor Sizing", *Proc. ICCAD* (1985) pp. 326-328.

[6] Garey, M.R., and Johnson, D.S.: *Computers and Intractability* , W.H. Freeman and Company, 1979.

[7] Hedlund, K.S.: "Aesop: A Tool for Automated Transistor Sizing", *Proc 24th DAC*, (1987) pp. 114-120.

[8] Johnson, D.S., Papadimitriou, C.H., and Yannakakis M.: "How Easy is Local Search?", *J. Comp. Sci.*, 37(1988) pp. 79-100.

[9] Kirkpatrick, S., Gelatt Jr., C.D. and Vecchi, M.P.: "Optimization by Simulated Annealing", *Science*, 220 (1983) pp. 671-680.

[10] Lam, J., Delosme, J.M.: "Simulated Annealing: a Fast Heuristic for Some Generic Layout Problems", *Proc. ICCAD* (1988) pp. 510-513.

[11] Levin, E.: "C.D.A. - Circuit Designer's Aid", unpublished.

[12] Lin, S., Marek-Sadowska, M., and Kuh, E.S.: "Delay and Area Optimization in Standard-Cell Design", *Proc. 27th DAC*, 1990 pp. 349-352.

[13] Marple, D.P., and El Gamal, A.: "Optimal Selection of Transistor Sizes in Digital VLSI Circuits", *Advanced Research in VLSI Proc. 1987 Stanford Conf.*, pp. 151-172.

[14] Matson, M.D.: "Optimization of Digital MOS VLSI Circuits", *Proc. Chapel Hill conf. on VLSI* (1985) pp. 111-126.

[15] Papadimitriou, C.H., and Steiglitz, K.: *Combinatorial optimization*, Prentice Hall inc., 1982.

[16] Papadimitriou, C.H., Schäffer, A.A., and Yannakakis, M.: "On The Complexity of Local Search", *Proc. 22nd STOC* (1990) pp. 438-445.

[17] Pincus, J.D., and Despain, A.M.: "Delay Reduction Using Simulated Annealing", *Proc. 23rd DAC*, (1986) pp. 690-695.

[18] Van Laarhoven, P.J.M, and Aarts, E.H.L.: *Simulated Annealing: Theory and Applications*, D. Reidel Publishing Co., 1987.

[19] Wong, D.F., Leong, H.W., and Liu, C.L.: *Simulated Annealing for VLSI Design*, Kluwer Academic Publishers, 1988.

[20] Wu, C-H.A, Vander Zanden, N., and Gajski, D.: "A New Algorithm for Transistor Sizing in CMOS Circuits", *Proc. 1st EDAC*, (1990) pp. 589-593.