Preconditioners של וואידיה

ברצוננו לפתור $Ax=b$ כאשר המטריצה A היא מטריצה סימטרית חיובית (כל הערכים העצמיים שלה חיוביים) ודלילה (רוב הערכים בה הם אפס). דרך אחת לפתור מערכת לינארית כזאת היא הדרך הישירה: מציאת L משולשית עבורה $A=LL^T$, ואח"כ פתרון $Ly=b$ ו – $L^Tx=y$. אולם דרך זאת אינה תמיד מעשית: גם אם המטריצה A היא דלילה, ייתכן כי הפירוק שלה L הוא מאוד לא דליל, ולכן זמן הפירוק יהיה גדול. כמו כן ייתכן ולא יהיה די מקום בזכרון להכיל את L.

לכן נחפש איזשהו קירוב ל – L, לו נקרא preconditioner. קירוב כזה יועיל לנו בפתרון איטרטיבי לפתרון משוואות לינאריות.

דרך אחת לבנות preconditioners, הוא incomplete factorization: במהלך תהליך הפירוק המשולשי, נזרקים האיברים הקטנים של L, מתוך תקוה שהאיברים הקטנים ב – L משפיעים פחות על "מהות", וכך נקבל קירוב טוב. אנחנו נתאר סוג אחר של preconditioners, מסוג complete factorization of incomplete matrices. אנחנו נבצע פירוק מלא, אבל לא למטריצה A, כי אם למטריצה A מדוללת. נסיר (בצורה סימטרית) איברים מתוך A, ונבצע למטריצה המדוללת הזאת פירוק מלא. בהמשך נתאר את האלגוריתם של Viadya, שהציע איזה איברים להסיר מ – A.

בהינתן preconditioner, מספר האיטרציות הדרושות להתכנסות האלגוריתם האיטרטיבי, הוא לכל היותר סדר גודל של שורש ה – condition number של המטריצה $M^{-\frac{1}{2}}AM^{-\frac{1}{2}}$.
ערך זה שווה גם ל – condition number של זוג המטריצות A ו – M שהוא היחס בין הערכים העצמיים המוכללים של זוג המטריצות A ו – M, המוגדר להלן:

קבוצת הערכים העצמיים המוכללים של זוג מטריצות מוגדרת להיות:
$\lambda_f(A,M)=\{\lambda \mid \exists x \;\; Ax=\lambda Mx \;\; Bx\neq 0\}$. קל לראות כי אם M הפיכה, אלו בדיוק הערכים העצמיים של $M^{-1}A$. זאת משום ש: $Ax=\lambda Mx \Leftrightarrow M^{-1}Ax=\lambda x$.

$$cond(M^{-\frac{1}{2}}AM^{-\frac{1}{2}})=cond(M^{-1}A)=cond(A,M)=\frac{\lambda_{\max}(A,M)}{\lambda_{\min}(A,M)}.$$

לכן נחפש preconditioner M, כך שהיחס $\frac{\lambda_{\max}(A,M)}{\lambda_{\min}(A,M)}$ קטן.

בהינתן preconditioner, ננסה לחסום את $\frac{\lambda_{\max}(A,M)}{\lambda_{\min}(A,M)}$ כדי להעריך את קצב ההתכנסות בעזרתו. $\frac{\lambda_{\max}(A,M)}{\lambda_{\min}(A,M)}=\lambda_{\max}(A,M)\cdot\lambda_{\max}(M,A)$ שכן הערכים העצמיים של (M,A) הם אחד חלקי הערכים העצמיים של (A,M). לכן כל מה שדרוש לנו הוא טכניקה למציאת חסם על ערכים עצמיים של זוג מטריצות. לשם כך נשתמש ב - Lemma הבאה.

Support Lemma: תהי M מטריצה positive-definite. הערכים העצמיים $\lambda_f(A,M)$ חסומים על ידי $\sigma(A,M)$ המוגדר כך:

הגדרה: $\sigma(A,M)=\min\{\tau \mid \tau M - A \ \ \textit{is positive semi-definite}\}$. ערך זה מכונה **התמיכה של *M* במטריצה *A*.** (מטריצה מוגדרת להיות positive semi-definite אם הערכים העצמיים שלה אי-שליליים. הגדרה אחרת: מטריצה היא positive semi-definite אם $\forall x \ \ x^T Ax \ge 0$).

הוכחה: נראה כי אם $\lambda \in \lambda_f(A,M)$ וגם $\tau M - A$ היא positive-semidefinite, אז $\lambda \le \tau$.

יהי λ ערך עצמי. כלומר קיים *y* כך ש: $Ay=\lambda My$. לכן $(\lambda M - A)y = 0$ וכן $y^T(\lambda M - A)y = 0$. נניח בשלילה $\lambda > \tau$. נסמן $\varepsilon = \lambda - \tau > 0$. מכאן:

$$y^T(\tau M - A)y = y^T[(\lambda-\varepsilon)M - A]y =$$

$$y^T(\lambda M - A)y - \varepsilon y^T My = -\varepsilon y^T My < 0$$

הגענו לסתירה.

מסקנה: $\sigma(A,M)$ נותן לנו חסם עליון על הערכים העצמיים של $M^{-\frac{1}{2}}AM^{-\frac{1}{2}}$, ובפרט על הערך העצמי העליון.

מסקנה: $\text{cond}(A,M) \le \sigma(A,M)\cdot\sigma(M,A)$.

כלי נוסף בו נעזר לצורך מציאת חסם על ה – condition number, הוא ה – Symmetric Product Support Lemma.

Symmetric Product Support Lemma: תהי $U \in R^{n\times k}$ בטווח של $V \in R^{n\times p}$. אזי $\sigma(UU^T, VV^T) = \min_{VW=U} \|W\|_2^2$.

נשתמש במשפט זה על-ידי הצגת המטריצה *A* שלנו כ – UU^T, והצגת מטריצת ה – preconditioner שלנו כ – VV^T ומציאת *W* המקיימת: $VW=U$. מכך נסיק $\sigma(UU^T, VV^T) \le \|W\|_2^2$.

הגדרה: נורמה של מטריצה: $\|W\|_2 = \max_x \frac{\|Wx\|_2}{\|x\|_2}$. או $\|W\|_2 = \max_{\|x\|_2=1} \|Wx\|_2$.

בהמשך ננצל את העובדה ש: $\|W\|_2^2 \le \|W\|_F^2$ וגם $\|W\|_2^2 \le \|W\|_1 \|W\|_\infty$.

Vaidya הציע preconditioner עבור מטריצות Stieltjes, שהן Diagonally Dominant (כלומר בכל שורה הערך על האלכסון גדול או שווה מסכום הערכים המוחלטים של האיברים מחוץ לאלכסון), סימטריות וכל האיברים מחוץ לאלכסון אי-חיוביים.

נראה כעת כיצד ניתן להציג מטריצות Steitljes כ – UU^T.

הגדרה: edge vector <*ij*> מוגדר להיות הוקטור שכולו אפסים חוץ מבמקומות *i* ו – *j* בהם יש 1 ו - 1- בהתאמה. vertex vector <*i*> מוגדר להיות הוקטור שכולו אפסים חוץ מבמקום ה - *i* בו יש 1.

משפט: מטריצה Seiltjes ניתן לפרק ל - UU^T כאשר העמודות של *U* הן edge-vectors או vertex vectors מוכפלים בקבועים.

הוכחה: נניח תחילה כי סכום האיברים בכל שורה הוא בדיוק 0.

במטריצה $\sum_{i<j} a_{ij} \langle ij \rangle \langle ij \rangle^T$ האיבר ה – ij הוא a_{ij} כמו ב – A, כיון שכל מחובר תורם בדיוק לשני איברים מחוץ לאלכסון: a_{ji} , a_{ij}, ולשני איברים על האלכסון: a_{jj} , a_{ii}, וכל איבר מחוץ לאלכסון של A מקבל תרומה ממחובר אחד בלבד. לכן האיברים מחוץ לאלכסון ב – A וב - $\sum_{i<j} a_{ij} \langle ij \rangle \langle ij \rangle^T$ זהים. האיברים על האלכסון מסתכמים כך שסכום האיברים בכל שורה הוא 0, כמו ב – A. לכן האיברים על האלכסון ב – A וב - $\sum_{i<j} a_{ij} \langle ij \rangle \langle ij \rangle^T$ זהים.

מסקנה: $A = \sum_{i<j} a_{ij} \langle ij \rangle \langle ij \rangle^T = \sum_{i<j} \left(\sqrt{a_{ij}} \langle ij \rangle\right)\left(\sqrt{a_{ij}} \langle ij \rangle\right)^T = UU^T$ כאשר U היא המטריצה שעמודותיה הם הוקטורים $\sqrt{a_{ij}} \langle ij \rangle$.

ה – Lemma הבאה מראה מדוע ניתן להפטר מההנחה שסכום האיברים בכל שורה הוא אפס.

למה: תהי A מטריצה Stieltjes. תהי A' המטריצה שבה כל האיברים מחוץ לאלכסון הם כמו אלו של A, אך האיברים על האלכסון הם כאלה שמביאים לסכום אפס בכל שורה. תהי M מטריצה Stieltjes עם אותם row-sums כמו של A, ו – M' המטריצה המתאימה לה. כלומר, $M' = M - (A - A')$. אם $\tau M' - A'$ היא positive semi-definite עבור $\tau \ge 1$ אז $\tau M - A$ גם היא positive semi-definite.
הוכחה: $\tau M - A = \tau(M' + A - A') - A = (\tau M' - A') + (\tau - 1)(A - A')$. סכום שתי מטריצות positive semi-definite היא מטריצה positive semi-definite.

כזכור, ברצוננו להציג את המטריצה A כ - UU^T ואת M כ - VV^T. ברצוננו לבחור את M כך שהנורמה של W, המטריצה שמקיימת $U = VW$ היא קטנה.
מהי המטריצה W?

$$\begin{pmatrix} v_1 & v_2 & \dots & v_p \end{pmatrix} \begin{pmatrix} & w_{1i} & \\ & w_{2i} & \\ & \vdots & \\ & w_{pi} & \end{pmatrix} = \begin{pmatrix} & u_i & \end{pmatrix}$$

$$V_{n\times p} W_{p\times m} = U_{n\times m}$$

כלומר, לכל עמודה של U מתקיים: $u_i = \sum_{k=1}^{p} w_{ki} v_k$.

טענה: אם $A = UU^T$ מטריצה Stieltjes ו - $M = VV^T$ מטריצה שמתארת תת-גרף של A שהוא עץ, אז כל אחת מהעמודות של U נפרש על-ידי העמודות של V. זאת משום שוקטורי הקשת של מסלול מ – i ל – j פורשים את $\langle ij \rangle$. לכן אם M מתאר איזשהו גרף קשיר, ובפרט עץ, אז הוקטורים ב – V פורשים את הוקטורים ב – U.

Vaidya הציע לבנות Preconditioner שהגרף שלו הוא עץ פורש מקסימלי. לעץ פורש מקסימלי יש את התכונה שלכל קשת ב - G_A, הקשתות במסלול המתאים לו ב - G_M כבדות לפחות כמוהו. לכן, אם *M* המטריצה של העץ הפורש המינימלי, אז קיימת מטריצה *W* שמקיימת $V_{n\times p}W_{p\times m}=U_{n\times m}$, שהערכים של *W* כולם קטנים שווים מ – 1 בערך מוחלט.

$$\lambda_{\max}(A,M)\le\sigma\left(UU^T,VV^T\right)\le\|W\|_2^2\le\|W\|_F^2=\sum_{i,j}w_{ij}^{\;2}\le mp=m(n-1)$$

כמו כן, $\lambda_{\max}(M,A)\le\sigma\left(VV^T,UU^T\right)\le 1$. זאת משום שהמטריצה *W* שעבורה $UW=V$ היא מטריצה שבכל עמודה שלה יש בדיוק ערך אחד השונה מאפס (ערך זה הוא אחד). הנורמה של מטריצה כזאת היא 1.
זמן בניית *M* הוא *O(m*log*n)*. *M* מייצג עץ ,לכן את *M* ניתן לפרק ללא התמלאות. הפעלתה בכל איטרציה לוקחת *O(n)*. אם *m=O(n)*, אז יש *O(n)* איטרציות שכל אחת מהן לוקחת *O(n)* זמן. סכ"ה *O(n²)* זמן.

Augmented Vaidya Preconditioners

היתרון ב – Maximum Spanning Tree Preconditioners*,* ולמעשה בכל preconditioner שהגרף שלו הוא עץ, הוא בכך שאין התמלאות בפקטוריזציה של *M*. כלומר הפירוק יהיה זול, וכן הפעלת ה – preconditioner בכל איטרציה. אולם ייתכן שאם נרשה קצת התמלאות, השיפור בקצב ההתכנסות יהיה כזה שיגבר על מחיר הפירוק והגדלת זמן פתרון המערכת *Mz=r* בכל איטרציה.

הרעיון: נמצא עץ פורש מקסימלי. נחלק את הגרף ל – *k* חלקים קשירים שמספר הצמתים בהם בערך $\frac{n}{k}$. בין כל זוג חלקים, מוסיפים את הקשת הכבדה ביותר בינהם (אם קיימת קשת).
אם *k=1*, זהו בדיוק ה – Maximum spanning tree preconditioner הרגיל. אם *k=n* מתקבל *M=A*, כי כל הקשתות שהוסרו מוחזרות.
ככל שנחלק ליותר חלקים, כך פחות קשתות יוסרו, והקירוב למטריצה *A* יהיה טוב יותר. כתוצאה מכך יהיו פחות איטרציות עד להתכנסות. מצד שני הפירוק יארך יותר זמן, והזמן של כל איטרציה גם יגדל. אם נחלק לפחות חלקים, הקירוב פחות טוב, מספר האיטרציות יגדל, אך זמן הפירוק יקטן, והזמן של כל איטרציה יקטן.

נתון גרף שהדרגה המקסימלית בו, *d*, חסומה על-ידי קבוע. נתאר איך לייצג כל וקטור קשת ב – G_A כקומבינציה לינארית של וקטורים ב – G_M.
כל קשת בגרף מנתבים באחת משתי דרכים – אם שני קודקודי הקצה שלה באותו חלק, מנתבים דרך העץ. אם קודקודי הקצה בשני חלקים שונים, מנתבים דרך הקשת הכבדה שנוספה (ודרך העץ). אורך כל מסלול $O\left(\frac{n}{k}\right)$. כל קשת פנימית תומכת לכל היותר בכל הקשתות שצד אחד שלהן באותו החלק. לכן כל קשת משתתפת ב - $O\left(\frac{n}{k}\right)$ מסלולים.

$$\|W\|_2^2\le\|W\|_1\|W\|_\infty=\left(\max_j\sum_i\left|w_{ij}\right|\right)\cdot\left(\max_i\sum_j\left|w_{ij}\right|\right)=O\left(\frac{n^2}{k^2}\right).$$

כאמור, ברצוננו לחלק את הגרף לחלקים קשירים שגודלם פחות או יותר זהה. לשם כך נשתמש באלגוריתם לו קראנו TreePartition.

TreePartition

```
#comment: s_i = number of vertices in the subtree rooted at i

s_i ← 1
for each child j of i
        if (s_j > n/k + 1)
                TreePartition(j)
        if (s_j ≥ n/k)
                form a new subtree rooted at j
                disconnect j from i
        else
                s_i ← s_i + s_j
```

s_i מאותחל בתחילת האלגוריתם למספר הצמתים בתת-העץ ששורשו ב – *i*.
טענה: אלגוריתם TreePartition מחלק את תת-העץ ששורשו *i* לחלקים שגודלם בין *n/t* ל – $dn/t+1$, פרט אולי לחלק ששורשו *i* שיכול להיות קטן יותר (אך לא גדול יותר).
הוכחה: נראה כי העץ מחולק כיאות, וכי כשהאלגוריתם חוזר, s_i מכיל את הערך הנכון.
נוכיח באינדוקציה על גובה תת-העץ.
טענה זו ברורה כאשר *i* הוא עלה. נניח הגובה של תת העץ של *i* הוא *h*, וכי הטענה נכונה לכל *h'<h*. לכל בן *j* של *i*, אם אם נקרא ל – TreePartition(*j*) אז על-פי הנחת האינדוקציה כל תתי-העץ שנוצרים בקריאה הם בגודל המתאים וכן s_j מקבל את הערך של תת-העץ שנשאר ששורשו *j*. אם $s_j \ge n/t$, אז גודלו של תת העץ ששורשו *j* חוקי, והוא הופך לחלק בפני עצמו. אחרת הוא קטן מדי ונשאר מחובר ל – *i* וגודלו מחובר ל – s_i. גודלו של s_i בסוף התהליך הוא לכל היותר *dn/t + 1*. מש"ל.

פירוק של *M* לוקח $O(n+k^6)$ במקרה הכללי, ו – $O(n+k^{1.5})$ אם הגרף של *A* מישורי.
סך כל זמן החישוב במקרה הכללי הוא $O(k^6+\left(\frac{n}{k}\right)(n+k^4))$ (חישוב *M* הוא זניח). האיזון בין שלב האיטרציות לזמן הפירוק מתקבל כאשר $k=O(n^{0.25})$. במקרה זה זמן החישוב הכולל הוא $O(n^{1.75})$.
סך כל זמן החישוב כאשר *A* מישורי, הוא $O(k^{1.5}+\left(\frac{n}{k}\right)(n+k\log k))$ והבחירה הטובה ביותר עבור *k* היא $O(n^{0.8})$. במקרה כזה זמן החישוב הכולל הוא $O(n^{1.2})$.

Augmented MWB Preconditioners

מצאנו preconditioner שמבוסס על עצים עבור מטריצות סימטריות שהערכים מחוץ לאלכסון שלהן שליליים (כלומר הקשתות המתאימות להם חיוביות). מה לגבי מטריצות שהערכים מחוץ לאלכסון שלהן הם מעורבים (חיוביים ושליליים).
מטריצות כאלה ניתן לייצג כ – UU^T, כאשר העמודות של U הן edge-vectors (חיוביים או שליליים) המוכפלים בקבוע.

נגדיר וקטור קשת שלילית להיות $\begin{array}{cl} \begin{pmatrix} \\ 1 \\ 1 \\ \\ \end{pmatrix} & \begin{array}{l} \\ \leftarrow i \\ \leftarrow j \\ \\ \end{array} \end{array} = \; > ij <$. שימו לב כי :

$$> ij <> ij <^T = \begin{array}{cl} \begin{pmatrix} & & & \\ & 1 & 1 & \\ & 1 & 1 & \\ & & & \end{pmatrix} & \begin{array}{l} \\ \leftarrow i \\ \leftarrow j \\ \\ \end{array} \end{array}$$

כאשר המטריצה A היא "בדיוק" diagonally dominant, אז קל לראות כי את A ניתן להציג כ:

$$A = \sum_{\substack{a_{ij}<0 \\ i<j}} \left|a_{ij}\right| \langle ij \rangle \langle ij \rangle^T + \sum_{\substack{a_{ij}>0 \\ i<j}} \left|a_{ij}\right| > ij <> ij <^T =$$

$$\sum_{\substack{a_{ij}<0 \\ i<j}} \left(\sqrt{\left|a_{ij}\right|} \langle ij \rangle\right)\left(\sqrt{\left|a_{ij}\right|} \langle ij \rangle\right)^T + \sum_{\substack{a_{ij}>0 \\ i<j}} \left(\sqrt{\left|a_{ij}\right|} > ij <\right)\left(\sqrt{\left|a_{ij}\right|} > ij <\right)^T$$

כלומר $A=UU^T$ כאשר העמודות של U הן edge vectors חיוביים ושליליים המוכפלים בקבוע.
מסתבר שבמקרה זה, יש למצוא בסיס מקסימלי של עמודות U, במקום עץ פורש מקסימלי. נביט על U כעל אוסף של וקטורים. ברצוננו למצוא V תת-קבוצה של U, שעבורה קיימת מטריצה W שמקיימת $U=VW$. אנו רוצים שכל וקטור ב – U ניתן יהיה לייצג כקומבינציה לינארית של וקטורי V. כלומר V צריך להיות לפחות בסיס. אנו נבחר את V להיות בסיס מקסימלי – בסיס שסכום משקלות הוקטורים בו מקסימלי. מסתבר, שכאשר בגרף של A יש רק קשתות חיוביות, זה שקול למציאת עץ פורש מקסימלי. בהמשך נזכיר כי בחירה זו של בסיס מקסימלי מבטיחה כי כל האיברים ב – W קטנים או שווים ל – 2 בערכם המוחלט. עבור עץ פורש מקסימלי הראינו שכל קשת e נתמכת על-ידי קשתות שמשקלן גדול לפחות כמו המשקל של e. תכונה זאת נשמרת כאשר בוחרים את V להיות בסיס מקסימלי.

טענה: יהי $u_1, u_2, \ldots, u_\ell$ בסיס מקסימלי של הוקטורים $u_1, u_2, \ldots, u_m$ שמשקליהם $\alpha_1, \alpha_2, \ldots, \alpha_m$. יהי $u_k = \beta_1 u_1 + \beta_2 u_2 + \ldots + \beta_\ell u_\ell$. אזי, אם $\beta_i \neq 0$ אז $\alpha_i \geq \alpha_k$.
הוכחה: נניח בשלילה כי עבור i כלשהו $\alpha_i < \alpha_k$ וגם $\beta_i \neq 0$. נראה כי אם נסיר את u_i מהבסיס ונציב במקומו את u_k, נקבל בסיס שמשקלו רב יותר.

$$u_i = \frac{1}{\beta_i}\left(u_k - \beta_1 u_1 - \cdots - \beta_{i-1} u_{i-1} - \beta_{i+1} u_{i+1} - \cdots - \beta_\ell u_\ell\right).$$

מכאן שהבסיס החדש פורש גם הוא. כמו כן משקלו גדול יותר מזה של הבסיס המקסימלי. זוהי סתירה.

מציאת בסיס מקסימלי היא מקרה פרטי של מציאת קבוצה בלתי-תלוייה מקסימלית במטרואיד.
מטרואיד הוא זוג $M=(S,\ell)$ המקיים את התנאים הבאים:

- S קבוצה סופית ולא ריקה
- ℓ היא קבוצה לא ריקה של תתי קבוצות של S, שנקראות תתי-הקבוצות הבלתי תלויות של S, המקיים שאם $B\in\ell$ וגם $A\subseteq B$ אז $A\in\ell$.
- אם $A\in\ell$, $B\in\ell$ וגם $|A|<|B|$, אז קיים איבר $x\in B\setminus A$ כך ש: $A\cup\{x\}\in\ell$

קל לראות כי בהנתן S קבוצה של וקטורים, ו - ℓ קבוצה של קבוצות בלתי תלויות של וקטורים ב – S, אז $M=(S,\ell)$ הוא מטרואיד.

נניח לכל איבר ב – S יש משקל (במקרה שלנו נצמיד לוקטור שמייצג קשת מ – i ל – j, את הערך $\sqrt{|a_{ij}|}$). קיים אלגוריתם גנרי למציאת קבוצה בלתי-תלויה מקסימלית במטרואיד.
אלגוריתם זה ממיין את האיברים ב – S לפי משקלם, ומוסיף כל אחד מהם, אלא אם כן הוספת האיבר תיצור קבוצה שאינה בלתי תלויה.
במקרה שלנו, עלינו להוסיף וקטור-וקטור, אלא אם הוקטור שאנו בודקים תלוי לינארית בוקטורים שכבר בקבוצה שלנו. לשם מציאת בסיס מקסימלי ביעילות עלינו למצוא אלגוריתם יעיל שבודק האם קבוצה של edge-vectors תלויים לינארית. ניעזר בטענה הבאה:

טענה: וקטורי הקשתות של גרף בלתי מכוון הם בלתי-תלויים לינארית, אם"ם כל רכיב קשירות לא כולל אף מעגל חיובי וכולל לכל היותר מעגל שלילי אחד.

הגדרות:
מעגל חיובי – מעגל שמספר הקשתות השליליות בו הוא זוגי.
מעגל שלילי – מעגל שמספר הקשתות השליליות בו הוא אי-זוגי.

לכן כל שעלינו לעשות בכל שלב, הוא לבדוק האם הקשת שאנו שוקלים לצרף, סוגרת מעגל חיובי, או סוגרת מעגל שלילי שני.

הגדרות:
מסלול חיובי – מסלול שמספר הקשתות השליליות בו הוא זוגי.
מסלול שלילי – מסלול שמספר הקשתות השליליות בו הוא אי-זוגי.

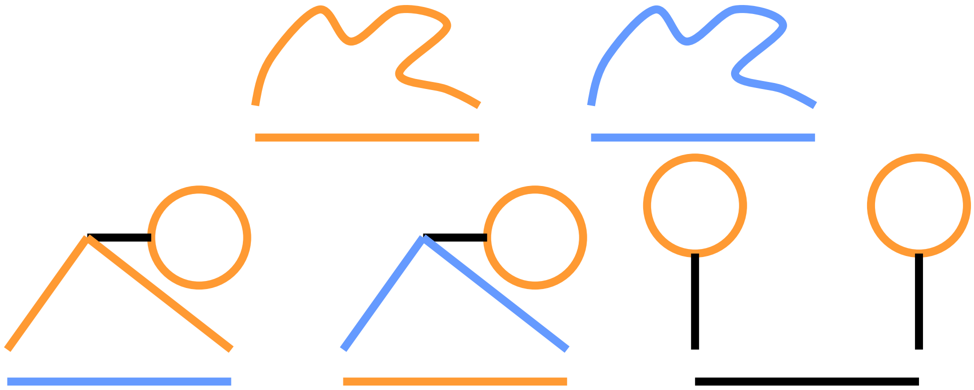
הזוגיות של מסלול נקבעת על-ידי הזוגיות של מספר הקשתות השליליות במסלול. באופן אינטואיטיבי, קשת שלילית הופכת את הזוגיות של המסלול.

מצאנו כי יש בדיוק 5 דרכים שונות לתמוך בקשת (בניגוד לדרך אחת ויחידה שקיימת כאשר כל הקשתות הן חיוביות). בנוסף זה קל להראות, כי המקדמים בקומבינציה הלינארית של וקטורי קשתות לא ממושקלים הם כולם $0,\pm1,\pm2,\pm\frac{1}{2}$.

חמש הדרכים הן הבאות:
1. תמיכה בקשת חיובית באמצעות מסלול חיובי.
2. תמיכה בקשת שלילית באמצעות מסלול שלילי.
3. תמיכה בקשת חיובית באמצעות רכיב קשירות הכולל מעגל שלילי ומסלול שלילי בין נקודות הקצה של הקשת.
4. תמיכה בקשת שלילית באמצעות רכיב קשירות הכולל מעגל שלילי ומסלול חיובי בין נקודות הקצה של הקשת.

5. תמיכה בקשת באמצעות שני רכיבי קשירות הכוללים מעגל שלילי.

נסמן בכתום קשתות / מסלולים / מעגלים שליליים, בכחול קשתות / מסלולים חיוביים ובשחור מסלולים בזוגיות כלשהי. השרטוט הבא מתאר את חמש הדרכים לתמוך בקשת:



יהי $u_k = \sqrt{|a_{ij}|}\langle ij\rangle$ (או $u_k = \sqrt{|a_{ij}|} > ij <$) וקטור ב – *U* שמשקלו *w*. מה נוכל להגיד על המקדמים של הוקטורים ב – *V* בקומבינציה הלינארית שמייצגת את *u*? אם נביט בקומבינציה הלינארית של הוקטורים הלא ממושקלים ב – *V* לוקטור <*ij*>, הרי טענו כי המקדמים האלו כולם $0,\pm1,\pm2,\pm\frac{1}{2}$, ובפרט כולם קטנים בערכם המוחלט מ – 2. אם נתחשב במשקלים, אז את המקדם של הוקטור ה – *k* ב – *V* נצטרך להכפיל ב - $\frac{\sqrt{|a_{ij}|}}{\sqrt{|b_k|}}$ (כאשר b_k הוא משקל הוקטור ה – *k*). הראינו כבר שכיון ש – *V* בסיס מקסימלי, $\sqrt{|a_{ij}|} \le \sqrt{|b_k|}$. מכאן שהמקדמים כולם קטנים או שווים מ – 2 בערכם המוחלט. כלומר מצאנו *W* שמקיים *U*=*VW*, כך שכל הערכים ב – *W* קטנים או שווים בערכם המוחלט מ – 2.

$$\lambda_{\max}(A,M) \le \sigma\left(UU^T, VV^T\right) \le \|W\|_2^2 \le \|W\|_F^2 = \sum_{i,j} {w_{ij}}^2 \le 4mn$$

מצאנו גם דרך להכליל את אלגוריתם הוספת הקשתות של Vaidya ל – preconditioners המבוססים על בסיס מקסימלי. הרעיון מבוסס על הפעלת TreePartition על כל רכיב קשירות בבסיס (לאחר הסרת קשת במעגל אם קיים מעגל). בין כל זוג חלקים שונים יש להוסיף לכל היותר שני וקטורים כדי להשלים לבסיס.