



دانشگاه صنعتی شریف
دانشکده ریاضی
سمینار محاسبات عددی

عنوان:
الگوریتم های چولسکی تغییر یافته

نگارش:
رضا پیشکو
۹۸۱۰۰۳۶۷

استاد درس:
دکتر مهدوی امیری

تیر ۱۴۰۲

چکیده

این گزارش بر گرفته از ([1] Haw-ren Fang, Dianne P.O'Leary) درباره الگوریتم چولسکی تغییر یافته است که هدف آن تبدیل ماتریس متقارن A که احتمالا نامعین است به ماتریس معین $A + E$ است. از آنجا که فاکتور سازی این ماتریس مثبت معین در مسائل بهینه سازی برای محاسبه جهت نیوتن استفاده می شود، مطلوب است که ماتریس تغییر یافته خوش تعریف و نزدیک به A باشد. تعدادی الگوریتم برای این کار پیش از این وجود داشته و این مقاله با ترکیب کردن این الگوریتم ها و تغییر دادن آنها روش های جدیدی معرفی می کند.

۱- مقدمه

الگوریتم های چولسکی تغییر یافته در مسائل بهینه سازی غیر خطی برای محاسبه جهت نیوتن کاربرد بسیاری دارد. با داشتن ماتریس A احتمالا نامعین که تقریبی از هسین تابعی است که می خواهیم کمینه سازی کنیم، هدف پیدا کردن ماتریس $A + E$ است که E به کوچک باشد. در این صورت جهت جست و جو یا Δx از حل سیستم $(A + E)\Delta x = -g(x)$ که $g(x)$ گرادیان تابع است، به دست می آید.

هنگام محاسبه E چهار خاصیت زیر باید برقرار باشد.

۱. اگر A به اندازه کافی مثبت معین باشد، آنگاه $E = 0$.

۲. اگر A مثبت معین نباشد، آنگاه $\|E\|$ خیلی بزرگ تر از $\inf\{\|\Delta A\| : A + \Delta A\}$ نباشد.

۳. ماتریس $A + E$ خوش تعریف باشد.

۴. هزینه الگوریتم فقط به اندازه یک ضریب کوچکی از n^2 بزرگ تر از فاکتور سازی استاندارد چولسکی $(\frac{1}{6}n^3 + O(n^2))$ باشد.

خاصیت ۱ این اطمینان را می دهد که روش نیوتن روی مسائل بهینه سازی محدب همچنان همگرایی سریع خود را حفظ کند. خاصیت ۲ جهت جست و جو را نزدیک به جهت نیوتن نگه می دارد. خاصیت ۳ پایداری محاسباتی را نتیجه می دهد. خاصیت ۴ باعث می شود که محاسبات نسبت به فاکتور سازی یک ماتریس چگال، کم باشد.

دو نوع کلی الگوریتم وجود دارد، که از حالت ساده ای که $A = \text{diag}(d_1, d_2, \dots, d_n)$ قطری باشد، نشأت می گیرند. در این حالت می توان ماتریس $A + E = \text{diag}(\hat{d}_1, \hat{d}_2, \dots, \hat{d}_n)$ را با انتخاب $\hat{d}_k = \max\{|d_k|, \delta\}$ مثبت معین کرد که $\delta > 0$ یک مقدار کوچک از قبل مشخص شده است. به چنین روشی، الگوریتم نوع ۱ می گویند. همچنین، می توان مقادیر ماتریس جدید را به صورت $\hat{d}_k = \max\{d_k, \delta\}$ در نظر گرفت که به این روش، الگوریتم نوع ۲ می گویند.

سه تجزیه پر استفاده از ماتریس A متقارن به صورت زیر وجود دارد.

۱. $PAP^T = LDL^T$ به طوری که D قطری باشد

۲. $PAP^T = LBL^T$ به طوری که B قطری-بلوکی با بلوک های ۱ یا ۲ بعدی باشد

۳. $PAP^T = LTL^T$ به طوری که T ماتریس سه قطری باشد.

که در ادامه دو روش بر اساس LDL^T گفته می شود و دو الگوریتم جدید از روی آنها معرفی می شود. همچنین دو الگوریتم بر اساس LBL^T گفته می شود و دو الگوریتم جدید از روی آنها معرفی می شود.

۲- الگوریتم

۲-۱ الگوریتم های LDL^T تغییر یافته (کارهای پیشین)

در الگوریتم تغییر یافته LDL^T برای ماتریس مثبت معین $\hat{A} = A + E$ که $E = \text{diag}(\delta_1, \delta_2, \dots, \delta_n)$ مقدار $\delta_k \geq 0$ در قدم k ام از تجزیه محاسبه می شود. به این صورت که اگر ماتریس مکمل شور در مرحله k ام را به صورت $A_k = \begin{bmatrix} a_k & c_k^T \\ c_k & \bar{A}_k \end{bmatrix}$ در نظر بگیریم، با شروع از $A_1 = A$ داریم:

$$L(k+1:n, k) = \frac{c_k}{a_k + \delta_k} \quad D(k, k) = a_k + \delta_k \quad A_{k+1} = \bar{A}_k - \frac{c_k c_k^T}{a_k + \delta_k} \quad (1)$$

چالش این الگوریتم ها انتخاب δ به گونه ای است که چهار خاصیت ذکر شده را ارضا کنند. در ادامه ابتدا به الگوریتمی که Gill و Murray معرفی کردند شروع میکنیم که به اختصار به آن GMW۸۱ می گوئیم و سپس به الگوریتم ابتدایی و ثانویه ای که Schnabel و Eskow معرفی کردند می پردازیم که به ترتیب به آنها SE۹۰ و SE۹۹ می گوئیم.

۲-۱-۱-۲ الگوریتم GMW۸۱ [۲]

این الگوریتم مقدار δ_k را به صورت زیر محاسبه می کند

$$a_k + \delta_k = \max \left\{ \delta, |a_k|, \frac{\|c_k\|_\infty^2}{\beta^2} \right\}$$

که $\beta > 0$ و $\delta > 0$ از قبل مشخص شده اند. مقدار δ را در آزمایش ها برابر با ϵ_M قرار داده اند. مقدار β هم در واقع کرانی روی قدر مطلق درایه های غیر قطری ماتریس پایین مثلثی L است. Gill و Murray نشان داده اند که

$$\|E\|_2 \leq \left(\frac{\xi}{\beta} + (n-1)\beta \right)^2 + (\eta + (n-1)\beta^2) + \delta =: f(\beta)$$

که η و ξ به ترتیب مقدار بیشینه قدر مطلق درایه های قطری و درایه های غیر قطری ماتریس A هستند. هزینه محاسبات این الگوریتم نسبت به هزینه تجزیه چولسکی استاندارد به اندازه $O(n^2)$ بیشتر است و خاصیت ۴ ارضا می شود. حالا خاصیت ۲ را در نظر بگیرید. مقدار کمینه عبارت بالا برابر است با

$$\min_{\beta} f(\beta) = 2\xi(\sqrt{n^2 - n} + n - 1) + \eta + \delta \leq 4n\xi + \eta + \delta$$

که با $\beta^2 = \frac{\xi}{\sqrt{n^2 - 1}}$ برای $n > 1$ به این مقدار می رسیم.

برای برقراری خاصیت ۱، نشان داده شده است که اگر $\beta^2 \geq \eta$ و $\lambda_{\min}(A) \geq \delta$ آنگاه $E = 0$.

در نتیجه مقدار β را به صورت زیر مشخص کرده اند

و در نهایت با قرار دادن این مقدار β در رابطه ای که برای کران $\|E\|_2$ به دست آوردند نتیجه می شود $\|E\|_2 = O(n^2)$.

الگوریتم SE۹۰ الهام گرفته از لم مرتبط با دایره Gerschgorin [۵] است. شعاع و دایره Gerschgorin به صورت زیر تعریف می شود

$$R_i(A) = \sum_{j \neq i} |a_{i,j}| \quad C_i(A) = \{z : \|z - a_{i,i}\| \leq R_i(A)\}$$

Gerschgorin نشان داده است که مقادیر ویژه ماتریس A درون اجتماع دایره های $C_i(A)$ قرار دارند. پس با قرار دادن $\delta_k = \max\{0, -a_{kk} + R_k(A)\}$ می توان A را مثبت نیمه معین کرد. حالا با در نظر گرفتن لم زیر می توان δ_k تغییر یافته را به دست آورد.

لم ۱. ماتریس متقارن $A = \begin{bmatrix} a & c^T \\ c & \bar{A} \end{bmatrix}$ را در نظر بگیرید و فرض کنید $\delta \geq \{0, -a + \|c\|_1\}$ که نتیجه دهد $a + \delta \geq \|c\|_1$. ماتریس مکمل شور برابر است با $\hat{A} = \bar{A} - \frac{cc^T}{a+\delta}$. در این صورت $C_i(\hat{A}) \subset C_i + 1(A)$.

در این صورت با قرار دادن $\delta_k = \max\{0, -a_k + \|c\|_1\}$ می توان $A + E$ مثبت معین را ایجاد کرد. با تکرار کردن لم ۱ نتیجه می شود که $\delta_k \leq \max\{0, -a_{kk} + R_k(A)\}$. ولی با این روش ساده ممکن است خاصیت ۱ برقرار نشود. به همین دلیل الگوریتم SE۹۰ شامل دو فاز است. در فاز اول همان قدم های الگوریتم چولسکی استاندارد را انجام می دهد و قرار می دهد $\delta_k = 0$. تا زمانی با فاز اول پیش می رود که مقادیر قطری مکمل شور بعدی به اندازه کافی مثبت باشند. شبه کد این الگوریتم به صورت زیر است.

Algorithm 1 Phase 1 of a 2-Phase Strategy.

{Given a symmetric $A \in R^{n \times n}$ and a small tolerance $\delta > 0$.}
 $A_1 := A, k := 1$
 Pivot on the maximum diagonal element of A_1 .
 {Denote $A_k = \begin{bmatrix} a_k & c_k^T \\ c_k & \bar{A}_k \end{bmatrix}$, then $\text{Diag}(\bar{A}_k) \leq a_k I_{n-k}$ after pivoting.}
if $a_1 \geq \delta$ **then**
 while $\text{Diag}(\bar{A}_k - \frac{c_k c_k^T}{a_k}) \geq \delta I_{n-k}$ and $k < n$ **do**
 $A_{k+1} := \bar{A}_k - \frac{c_k c_k^T}{a_k}$
 $k := k + 1$
 Pivot on maximum diagonal of A_k .
 end while
end if

ولی مشکل این الگوریتم این بود که برای ماتریس هایی که نزدیک به مثبت معین بودند مقدار $\|E\|_2$ بزرگی داشت و برای همین در SE۹۹ الگوریتم دو فاز آرم سازی شده ای ارائه دادند. در این الگوریتم به مقادیر روی قطر فاز ۱ قبل که منفی هستند و به اندازخ کافی کوچک هستند هم اجازه می دهد در فاز ۱ آرام شده بیایند. شبه کد این الگوریتم در زیر آمده است.

Algorithm 2 Relaxed Phase 1 of a 2-Phase Strategy.

{Given a symmetric $A \in R^{n \times n}$, $\delta > 0$ and $0 < \mu \leq 1$.}
 $\eta := \max_{1 \leq i \leq n} |A_{ii}|$
if $\text{Diag}(A) \geq -\mu\eta I_n$ **then**
 $A_1 := A, k := 1$
 Pivot on the maximum diagonal element of A_1 .
 {Denote $A_k = \begin{bmatrix} a_k & c_k^T \\ c_k & \bar{A}_k \end{bmatrix}$, then $\text{Diag}(\bar{A}_k) \leq a_k I_{n-k}$ after pivoting.}
while $a_k \geq \delta$ and $\text{Diag}(A_k) \geq -\mu a_k I_{n-k+1}$ and $\text{Diag}(\bar{A}_k - \frac{c_k c_k^T}{a_k}) \geq -\mu\eta I_{n-k}$ and $k < n$ **do**
 $A_{k+1} := \bar{A}_k - \frac{c_k c_k^T}{a_k}$
 $k := k + 1$
 Pivot on maximum diagonal of A_k .
end while
end if

در نهایت بعد از اتمام فاز ۱، مقادیر δ_k در فاز ۲ به صورت زیر مشخص می‌شوند

$$\delta_k = \max \{ \delta_{k-1}, -a_{k-1} + \max \{ \|c_k\|_1, \tau\eta \} \}$$

که $\tau = \sqrt[3]{\epsilon_M}$. دلیل انتخاب δ_k به این صورت که باعث شود $\delta_k \geq \delta_{k-1}$ به این دلیل است که افزایش δ_k نسبت به δ_{k-1} مقدار $\|E\|_1$ را در این نقطه افزایش نمی‌دهد ولی ممکن است δ_i را برای $k < i \leq n$ کاهش دهد.

در فاز ۱ آرام‌سازی شده چون مقادیر منفی کوچک هم وارد می‌شوند، تغییراتی در مقدار δ_k برای فاز ۲ صورت می‌گیرد که در نهایت نتیجه می‌شود

$$\delta_k = \max \{ \delta_{k-1}, -a_{k-1} + \max \{ \|c_k\|_1, \bar{\tau}\eta \} \}$$

که $\bar{\tau} = \sqrt[3]{\epsilon_M^2}$. در هر دو حالت الگوریتم دو فاز و همچنین الگوریتم دو فاز آرام‌سازی شده نتیجه می‌شود $\|E\|_2 = O(n)$.

۲-۲ الگوریتم های LDL^T تغییر یافته (کار های جدید)

۲-۲-۱ الگوریتم GMW-I

الگوریتم GMW۸۱ از الگوریتم های نوع یک است که در آن $\|E\|_2 = O(n^2)$ در حالی که پیشتر دیدیم در الگوریتم های SE۹۹ SE۹۰، $\|E\|_2 = O(n)$ در این روش الگوریتم دو فاز را با الگوریتم GMW۸۱ ترکیب می‌کنیم. در واقع الگوریتم GMW-I از فاز یک آرام‌سازی شده SE۹۹ استفاده می‌کند و در فاز ۲ از الگوریتم GMW۸۱ استفاده می‌کند. و در اینصورت کرانی که GMW۸۱ برای نرم E داشت به کران زیر تغییر می‌کند.

$$\|E\|_2 \leq \left(\frac{\hat{\xi}}{\beta} + (n - K - 1)\beta \right)^2 + 2(\hat{\eta} + (n - K - 1)\beta^2) + \delta$$

که K تعداد قدم های الگوریتم فاز یک است و $\hat{\xi}$ و $\hat{\eta}$ به ترتیب مقدار بیشینه مقادیر روی قطر و مقادیر غیر قطری A_{k+1} هستند. در این الگوریتم مقدار β به صورت زیر تعیین می‌شود چرا که نیازی به شرط $\beta^2 \geq \hat{\eta}$ نداریم.

$$\beta^2 = \max \left\{ \frac{\hat{\xi}}{\sqrt{(n-K)^2 - 1}}, \epsilon_M \right\}$$

که در نهایت با این پارامتر ها نتیجه می شود $\|E\|_2 = O(n)$.

۲-۲-۲ GMW-II

این الگوریتم نسخه نوع دو از الگوریتم GMW^{۸۱} است که از استراتژی غیر نزولی در آن استفاده شده است و δ_k به صورت زیر مشخص شده است.

$$a_k + \delta_k = \max \left\{ \delta, a_k + \delta_{k-1}, \frac{\|c_k\|_\infty^2}{\beta^2} \right\}$$

که $\beta, \delta > 0$ مقادیری هستند که از قبل مشخص شده اند و همچنین $\delta_0 = 0$ در این الگوریتم همچنان مقادیر غیر قطری L با مقدار β باند خورده اند. همچنین، کران نرم E در این الگوریتم به صورت زیر تغییر پیدا می کند.

$$\|E\|_2 \leq \left(\frac{\xi}{\beta} + (n-1)\beta \right)^2 + (\eta + (n-1)\beta^2) + \delta =: f(\beta)$$

که η, ξ به ترتیب مقدار بیشینه قدر مطلق مقادیر قطری و غیر قطری A هستند. مینیمم عبارت بالا به صورت زیر است:

$$\min_{\beta} f(\beta) = 2\xi(\sqrt{n^2 - n} + n - 1) + \eta + \delta \leq 4n\xi + \eta + \delta$$

مینیمم به ازای $\beta^2 = \frac{\xi}{\sqrt{n^2 - n}}$ بدست می آید و در نتیجه مقدار β به صورت زیر تعیین می شود.

$$\beta^2 = \max \left\{ \eta, \frac{\xi}{\sqrt{n^2 - n}}, \epsilon_M \right\}$$

که با این پارامتر ها داریم $\|E\|_2 = O(n^2)$. حال می توان الگوریتم دو فازی را روی GMW-II هم اجرا کرد و نرم E را به $O(n)$ کاهش داد.

۲-۳ الگوریتم های LBL^T تغییر یافته (کار های پیشین)

در این قسمت ابتدا به بیان یک قضیه می پردازیم.

قضیه ۱. فرض کنید یک ماتریس متقارن $M \in C^{n \times n}$ و یک ماتریس وارون ناپذیر $S \in C^{n \times n}$ داریم. در این صورت یک مقدار $\theta_k > 0$ وجود دارد به طوری که :

$$\lambda_k(SMS^T) = \theta_k \lambda_k(M)$$

که در آن $\lambda_1(SS^T) \leq \theta_k \leq \lambda_n(SS^T)$.

هر ماتریس متقارن A یک تجزیه LBL^T دارد که B قطری بلوکی با مرتبه بلوک های ۱ یا ۲ است. در الگوریتم های تغییر یافته LBL^T برخلاف الگوریتم های قبلی ابتدا تجزیه LBL^T را محاسبه کرده و $\hat{B} = B + \Delta B$ را طوری بدست می آوریم که مثبت معین باشد و در اینصورت باعث می شود $P(A + E)P^T = L\hat{B}L^T$ هم مثبت معین باشد. Sorensen و Moré یک الگوریتم تغییر یافته LBL^T ارائه دادند که به آن MSV۹ می گوئیم.

اگر هر بلوک 1×1 در B را d بنامیم، آنگاه آنرا به صورت $\hat{d} = \max\{\delta, |d|\}$ تغییر می دهیم. در MSV۹ و اگر هر بلوک 2×2 در B را D بنامیم، و تجزیه طیفی آنرا به صورت $D = U \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} U^T$ $\hat{\lambda}_i = \max\{\delta, |\lambda_i|\}$ تغییر می دهیم که $\hat{D} = U \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} U^T$

Higham و Cheng یک الگوریتم تغییر یافته LBL^T دیگر ارائه دادند با این تفاوت که در آن $\hat{d} = \max\{\delta, d\}$ و $\hat{\lambda}_i = \max\{\delta, \lambda_i\}$ به این الگوریتم هم به اختصار CH۹۸ می گوئیم.

وجه تمایز این دو الگوریتم در این است که الگوریتم MSV۹ یک الگوریتم نوع یک و الگوریتم CH۹۸ یک الگوریتم نوع دو است.

برای هر دو الگوریتم MSV۹ و CH۹۸ اگر $\lambda_{\min}(B) \geq \delta$ آنگاه $E = 0$. با توجه به قضیه ۱ می دانیم که اگر A مثبت معین باشد آنگاه $\lambda_{\min}(B) \geq \frac{\lambda_{\min}(A)}{\lambda_{\max}(LL^T)}$ در نتیجه اگر داشته باشیم:

$$\lambda_{\min}(A) \geq \delta \|LL^T\|_2$$

آنگاه حتما داریم $E = 0$. در نتیجه خاصیت ۱ برقرار است. همچنین دو روابط زیر برای درستی خاصیت ۲ برای MSV۹ و CH۹۸ وجود دارند.

$$\|E\|_2 \leq -\delta \|LL^T\|_2 \quad \text{برای الگوریتم MSV۹}$$

$$\|E\|_2 \leq \delta \|LL^T\|_2 - \lambda_{\min}(A) \kappa_2(LL^T) \quad \text{برای الگوریتم CH۹۸}$$

برای درستی خاصیت ۳ روابط زیر وجود دارند:

$$\kappa_2(A + E) \leq \kappa_2(LL^T) \max\left(1, \frac{\lambda_{\max}(A)}{\lambda_{\min}(LL^T)\delta}\right) \quad \text{برای الگوریتم CH۹۸}$$

$$\kappa_2(A + E) \leq \kappa_2(LL^T)^2 \kappa_2(A) \quad \text{برای الگوریتم MSV۹}$$

سه الگوریتم Bunch-Parlett (BP) و Bunch-Kaufman (BK) و Bunch-Parlett سریع (FBP) برای تجزیه LDL^T وجود دارند. هزینه محاسباتی هر کدام از آنها LDL^T در جدول زیر آمده است که می توان مشاهده کرد خاصیت ۴ را ارضا نمی کنند.

Symmetric matrix Case	General	
	Worst	Best
BP	$O(n^3)$	
FBP	$O(n^3)$	$O(n^2)$
BBK	$O(n^3)$	$O(n^2)$

۳- روشی جدید با تجزیه $LBL^T - LTL^T$

در این روش تجزیه را به صورت $PAP^T = LTL^T$ نشان می دهیم که T یک ماتریس متقارن سه قطری است و L یک ماتریس پایین مثلثی واحد است به طوری که $|L_{ij}| \leq 1$ و درایه های غیر قطری ستون اول آن صفر هستند. این الگوریتم از این موضوع استفاده می کند که A مثبت معین است اگر و تنها اگر T مثبت معین باشد. در واقع با ساختن ماتریس مثبت معین $\hat{T} = T + \Delta T$ باعث می شود که ماتریس $\hat{A} = P(A + E)P^T = L\hat{T}L^T$ هم مثبت معین باشد.

این روش از آنجا الهام می گیرد که ساختار سه قطری برای یک ماتریس متقارن در تجزیه LDL^T یا LTL^T تغییر نمی کند. در نتیجه برای مثبت معین کردن ماتریس T می توان هر کدام از الگوریتم های تغییر یافته LBL^T و LDL^T که قبلا مطرح کردیم را روی T اعمال کنیم و این تجزیه تغییر یافته خاصیت یک را برقرار می کند زیرا الگوریتم هایی که قبلا بحث کرده بودیم این خاصیت را ارضا می کردند.

بنابراین با توجه به صحبت های گفته شده، الگوریتم های LBL^T و LDL^T تغییر یافته روی یک ماتریس متقارن سه قطری اعمال می شوند که یعنی این روش کارا است.

همانطور که مشاهده کردیم الگوریتم های $MSV9$ و $CH98$ در خاصیت چهار دچار مشکل بودند. ولی با اعمال $MSV9$ یا $CH98$ روی ماتریس سه قطری متقارن T می توانیم هزینه را به $O(n^2)$ کاهش دهیم که این الگوریتم ها را LTL^T - $MS79$ و LTL^T - $CH98$ می نامیم.

الگوریتم ذکر شده را می توان به این صورت فرمول بندی کرد. ماتریس متقارن A را در نظر بگیرید و تجزیه LTL^T آنرا به صورت $PAP^T = LTL^T$ در نظر بگیرید. همچنین، فرض کنید تجزیه LBL^T ماتریس T به صورت $\tilde{P}T\tilde{P}^T = \tilde{L}\tilde{B}\tilde{L}^T$ باشد. خواهیم داشت:

$$PAP^T = L(\tilde{P}^T \tilde{L} \tilde{B} \tilde{L}^T \tilde{P})L^T$$

با اضافه کردن $\Delta \tilde{B}$ به \tilde{B} برای مثبت کردن آن، خواهیم داشت:

$$P(A + E)P^T = L(\tilde{P}^T \tilde{L}(\tilde{B} + \Delta \tilde{B})\tilde{L}^T \tilde{P})L^T$$

در این الگوریتم درستی خاصیت ها را می توان در روابط زیر دید:

$$\lambda_{\min}(A) \geq \delta \|\tilde{L}\tilde{L}^T\| \cdot \lambda_{\min}(LL^T) \Rightarrow E = 0 \quad (2)$$

$$\|E\|_2 \leq -2\lambda_{\min}(A) \cdot \kappa_2(LL^T) \cdot \kappa_2(\tilde{L}\tilde{L}^T) \quad LTL^T\text{-MS}79 \quad (3)$$

$$\|E\|_2 \leq \delta \|LL^T\| \|\tilde{L}\tilde{L}^T\| - \lambda_{\min}(A) \cdot \kappa_2(LL^T) \cdot \kappa_2(\tilde{L}\tilde{L}^T) \quad LTL^T\text{-CH}98 \quad (4)$$

$$\kappa_2(A + E) \leq \kappa_2(LL^T)^2 \kappa_2(\tilde{L}\tilde{L}^T)^2 \kappa_2(A) \quad LTL^T\text{-MS}79 \quad (5)$$

$$\kappa_2(A + E) \leq \kappa_2(LL^T) \kappa_2(\tilde{L}\tilde{L}^T) \max\{1, \frac{\lambda_{\max}(A)}{\lambda_{\min}(LL^T) \lambda_{\min}(\tilde{L}\tilde{L}^T) \delta}\} \quad LTL^T\text{-CH}98 \quad (6)$$

هزینه محاسباتی این الگوریتم با الگوریتم های LDL^T تغییر یافته در جدول زیر مقایسه شده است.

Symmetric matrix Case	General		Tridiagonal	
	Worst	Best	Worst	Best
BP	$O(n^3)$		$O(n^2)$	
FBP	$O(n^3)$	$O(n^2)$	$O(n^2)$	$O(n)$
BBK	$O(n^3)$	$O(n^2)$	$O(n^2)$	$O(n)$

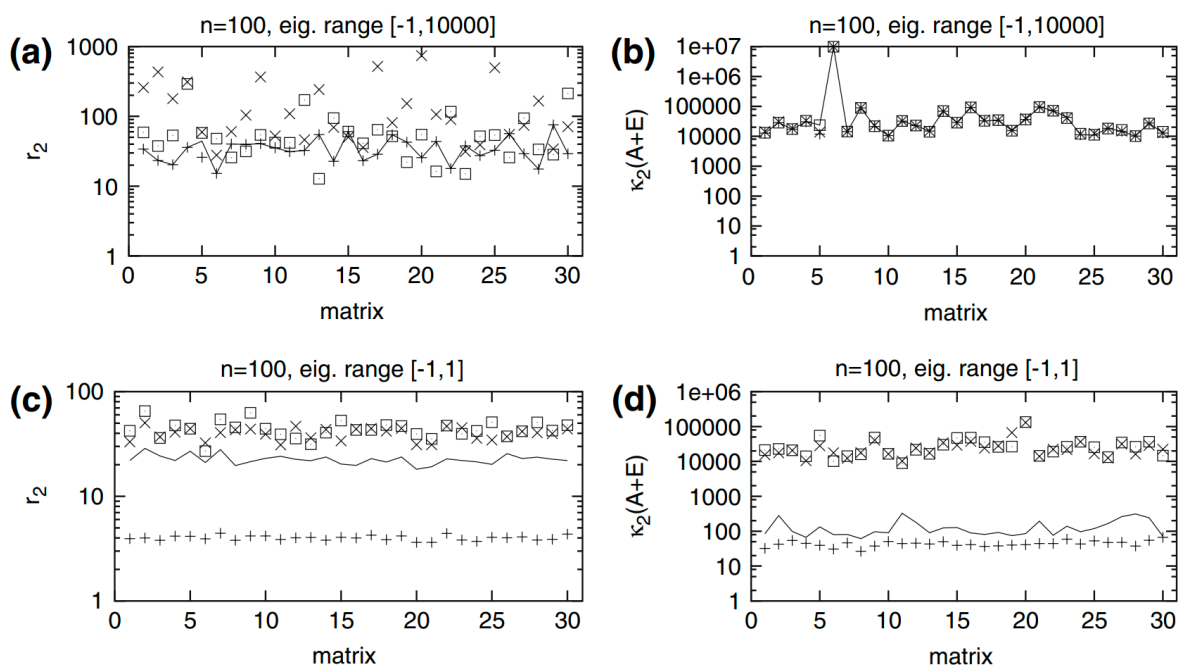
۴- نتایج عددی

در آزمایش های عددی مقاله سه دسته آزمایش انجام شد. ماتریس های تصادفی، ماتریس کارایی سنجی (benchmark matrix) از مقاله [۳] و ۳۳ ماتریس برگرفته از مقاله [۴]. در ادامه ما آزمایش های مربوط به ماتریس های تصادفی را بررسی می کنیم.

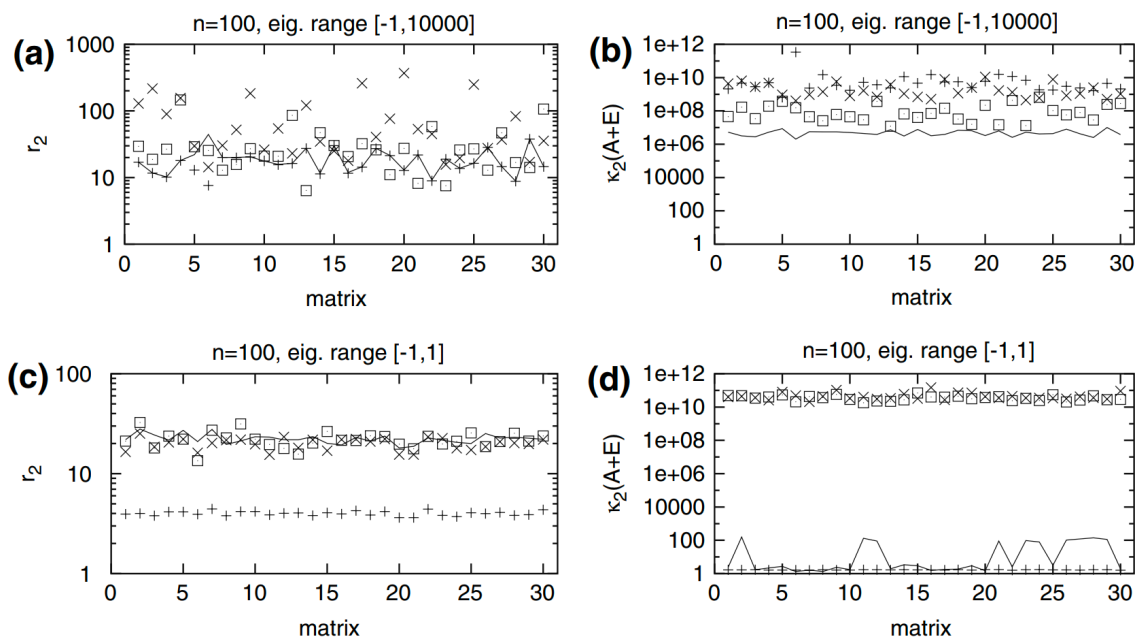
۴-۱- ماتریس های تصادفی

ماتریس هایی ۱۰۰ بعدی با مقادیر ویژه در بازه $[-1, 10000]$ و بازه $[-1, 1]$ به صورت تصادفی تولید و الگوریتم های تجزیه روی آنها اعمال شدند. عملکرد سه الگوریتم نوع ۱ با هم مقایسه شدند. این الگوریتم ها $GMW\text{-I}$, $MS79$, $LTL^T\text{-MS}79$ هستند. همچنین چهار الگوریتم نوع دو $GMW\text{-II}$, $SE99$, $CH98$, $LTL^T\text{-CH}98$ با هم مقایسه شدند.

در شکل زیر نتایج آزمایش روی الگوریتم های نوع ۱ را می توان مشاهده کرد:



در ادامه نتایج آزمایش روی الگوریتم های نوع ۲ آورده شده:



ماتریس های اختلال E با مقدار $r_2 = \frac{\|E\|_2}{|\lambda_{\min}(A)|}$ مقایسه شده اند.

- الگوریتم های نوع یک: در حالتی که مقدار ویژه ها در بازه $[-1, 1]$ قرار دارند، $MS79$ ، LTL^T مقادیر $\|E\|_2$ نزدیک به هم تولید کردند. اما در حالتی که مقدار ویژه ها بین $[-1, 10000]$ بودند، در برخی مثال ها LTL^T - $MS79$ مقادیر $\|E\|_2$ کمتری نسبت به $MSV9$ تولید کرد.
- الگوریتم های نوع دو: در حالت مقادیر ویژه $[-1, 10000]$ الگوریتم های $GMW-II$ ، $SE99$ مقادیر $\|E\|_2$ کمتری داشتند. الگوریتم LTL^T - $CH98$ نیز از $CH98$ عملکرد بهتری داشت. در حالت $[-1, 1]$ الگوریتم $SE99$ بهترین عملکرد را داشت.

- [1] D. P. O. Haw-ren Fang, "Modified cholesky algorithms: a catalog with new approaches," *Math. Program.*, 2007.
- [2] W. M. Philip E. GILL, "Newton-type methods for unconstrained and linearly constrained optimization," *Math. Program.*, 1974.
- [3] E. Schnabel, "A new modified cholesky factorization," *SIAM*, 1990.
- [4] E. Schnabel, "A revised modified cholesky factorization algorithm," *SIAM*, 1999.
- [5] W. Gill, Murray, "Practical optimization," *SIAM*, 1981.