دانشگاه صنعتی شاهرود
دانشکده علوم ریاضی
گروه ریاضی

پایان‌نامه کارشناسی ارشد ریاضی

عنوان

بررسی کدهای خطی و غیر خطی و برخی کاربردهای آن(کدهایDPCL)

نگارش
مهناز علیین

استاد راهنما
آقای دکتر صادق رحیمی شعرباف

استاد مشاور
آقای دکتر جعفری‌راد

۱۳۹۰/۲/۱۳

صفحة تصويب نامه توسط هيأت داوران (فرم پيوست ۳ يا ۴ با امضاي اصل هيأت داوران مورد قبول است)

قدردانی

به نام یاربی همتا

من لم یشکر المخلوق لم یشکر الخالق

با سلام و درود به ارواح طیبه شهدا از صدر اسلام تا کنون و تهیت و ثنا بر یگانه منجی عالم بشریت حضرت ولی عصر (عج) روحی فداک و نائب بر حقش، بر خود دانستم تا مراتب تشکر و قدردانی خود را از کلیه اساتید که بنده حقیر را در این چند سال مورد عنایت و راهنمایی قرار دادند، بنمایم و خواستار صحت و توفیقات روزافزون ایشان و خانواده محترمشان از خداوند بی همتا باشم. امید آن دارم که این تلاش جزئی و ناچیز که تقدیم به مهدی فاطمه سلام ا... علیه نموده‌ام مورد رضایت قلبی ایشان قرار گیرد.

تقدیر، تقویم انسان‌های عادیست و تغییر، تدبیر انسان‌های عالی. لحظاتی سرشار از تغییرات زیبا را برای کلیه محققان و اساتید گرامی از خداوند منان خواستارم.

چکیده

استفاده از کدهای LDPC به دلیل ساختار ساده، روش‌های کدگشایی تکراری و برخی دلایل دیگر در سال‌های اخیر بسیار مورد توجه بوده است. در این پایان‌نامه با نگاهی کوتاه به تاریخچه نظریه کدگذاری و بیان خلاصه‌ای از مفاهیم مربوط به کدهای خطی به شرح کدهای LDPC و نحوه کدگذاری آنها می‌پردازیم. سپس با استفاده از تعریف گراف فاکتور، برخی از الگوریتم‌های کدگشایی عبورپیام را معرفی می‌کنیم. علاوه‌براین روش کدگشایی برنامه‌ریزی خطی و برخی از مهمترین بهبودهای انجام شده بروی این الگوریتم کدگشایی را مورد بررسی قرار می‌دهیم و در پایان به مقایسه چند الگوریتم کدگشایی با استفاده از نتایج برخی مقالات اخیر می‌پردازیم.

واژه‌های کلیدی: کدهای $LDPC$ ، گراف فاکتور ، گراف تنر ، الگوریتم عبور پیام ، گدگشایی ، کدگذاری، کدهای خطی ، برنامه‌ریزی خطی ، کدگشای LP .

پیشگفتار

با نگاهی به پیشرفت‌های روزافزون در زمینه ارتباطات ، نظریه اطلاعات و نظریه کدگذاری و با توجه به کاربرد وسیع مفاهیم موجود در ریاضیات محض و کاربردی در این رشته‌ها ، زمینه مناسبی برای انجام تحقیقات کاربردی با استفاده از مفاهیم ریاضی از جمله نظریه ترکیبیات و گراف در این حیطه وجود دارد. به عنوان مثال می‌توان به ارتباط بین طرح‌های ترکیبیاتی با کد و مفاهیمی مانند فاصله در نظریه کدگذاری و همچنین ساخت کدهایی با استفاده از ماتریس هادامار اشاره کرد. علاوه‌براین از برخی تعاریف و قضایای موجود در نظریه گراف می‌توان به نتایجی مشابه با برخی قضایا و نتایج ارئه شده در نظریه کدگذاری رسید. خواننده گرامی می‌تواند برای بررسی این نتایج به مراجع [۱۹]،[۱۶] ، [۱] و مراجعه کند. علاوه‌براین نظریه کدگذاری در بسیاری از علوم مورد استفاده قرار گرفته است که برخی از این کاربردها داخل متن ذکر شده است. یکی از کاربردهای جذاب نظریه کدگذاری در زمینه ژنتیک و کدگشایی DNA است [۲] . آنچه در این پایان‌نامه بیان می‌گردد بررسی کاربرد برخی تکنیک‌های ریاضی مانند استفاده از گراف دو بخشی ، برنامه‌ریزی خطی ، برش گوموری و برخی مفاهیم آماری در نظریه کدگذاری و به طور خاص در کدهای LDPC است [۴]،[۳].

فهرست مطالب

لیست تصاویر

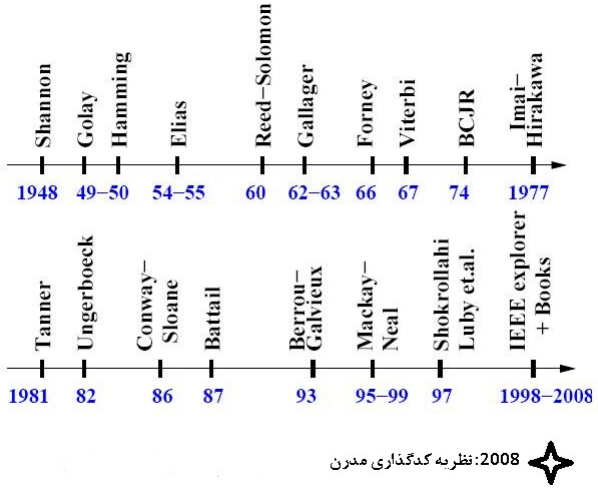
فصل ۱
مقدمه و تاریخچه

مساله اساسی ارتباطات تولید دوباره یک پیام (که در یک نقطه انتخاب شده است) در نقطه‌ای دیگراست. (کلود شانون) [۱]

۱.۱ تاریخچه

شانون در سال ۱۹۴۸ [۲۰] ، قوانینی را براساس ریاضیات بدست آورد که معین می‌کردند ، چگونه اطلاعات می‌توانند به سرعت از طریق یک کانال پارازیت‌دار ارسال شوند. مطالبی که شانون در این مقاله بیان می‌کند و چارچوب ریاضی آن ، مبنای تولید یک رشته کاملا جدید به نام **نظریه اطلاعات** [۲] گردید. تقریبا همزمان با شانون، **ریچارد هامینگ**[۳] (۱۹۵۰) به تحقیق درباره امکان کشف و تصحیح خطا در پیام‌های دریافتی پرداخته است. به این ترتیب با کار این دو (براساس چارچوب مطرح شده توسط شانون) [۴] نظریه کدگذاری یا به طور دقیق‌تر **نظریه کدهای تصحیح کننده خطا** [۵] شروع به کار کرد. می‌توان این گونه برداشت کرد که در حقیقت شانون نظریه کدگذاری را به دو قسمت **نظریه کدگذاری منبع** [۶] و **نظریه کدگذاری کانال** [۷] (کدهای تصحیح کننده خطا) تقسیم کرده است.

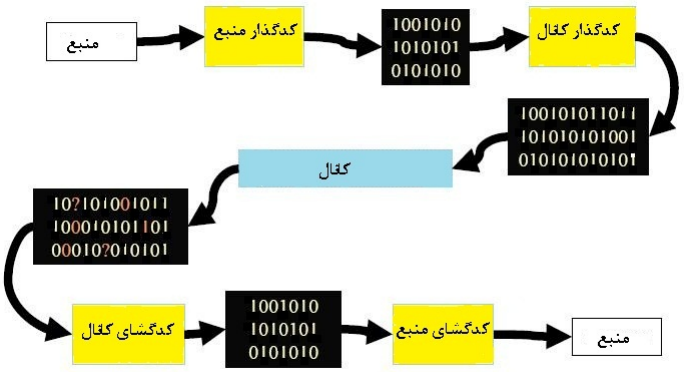
[۱] Claude Shannon
[۲] Information theory
[۳] R.W. Hamming
[۴] Coding theory
[۵] Theory of Error-Correcting Codes
[۶] Source coding theory
[۷] Channel coding theory



شکل ۱.۱: برخی از رویدادهای مهم در کدگذاری

نظریه کدگذاری شانون ادعا و به صورت نظری اثبات می‌کند که برای هر کانال، ماکزیمم نرخی وجود دارد که در آن نرخ می‌توان داده‌ها را به گونه‌ای که احتمال خطا صفر شود، مخابره کرد. این ماکزیمم نرخ به **ظرفیت کانال** [۸] مشهور شده است. علاوه براین شانون اثبات می‌کند که تقریبا هر کد بسیار بزرگ، می‌تواند به این ظرفیت برسد. البته این اثبات، چگونگی ساخت این کدها و نحوه **کدگذاری** [۹] و **کدگشایی** [۱۰] آنها را بیان نمی‌کند. در حقیقت یک کد تصادفی به اندازه دلخواه بزرگ ممکن است با استفاده از اصول فنی به خوبی اجرا شود ولی زمان‌های (پیچیدگی زمانی) کدگذاری و کدگشایی آن ممکن است بسیار بزرگ باشند. به این ترتیب (پس از کارهای شانون) یکی از اهداف اصلی نظریه کدگذاری، ساخت کدهایی شد که با پیچیدگی کدگذاری و کدگشایی قابل کنترل به ظرفیت کانال می‌رسند. از جمله موفقیت‌هایی که در نتیجه این تلاش‌ها حاصل شد، ساخت کدهای **توربو** [۱۱] در سال ۱۹۹۳ [۷] بود با ساخت این کدها، **کدگشایی تکراری** [۱۲] (که باعث اجرای عالی و پیچیدگی پایین گردید) مطرح شد. پس از آن **کدهای LDPC** [۱۳] دوباره مطرح شدند (قبل از آن توسط گالاگر در سال ۱۹۶۲ کشف

[۸] Capacity of channel
[۹] Encoding
[۱۰] Decoding
[۱۱] Turbo code
[۱۲] Iteration decoding
[۱۳] Low-density parity-check codes



شکل ۲.۱: سیستم ارتباطی

شده بود) .

آنچه با کار شانون از سال ۱۹۴۸ آغاز شد را به طور خلاصه می‌توان در شکل ۱.۱ دید. در فاصله زمانی ۱۹۴۸ تا ۱۹۹۳ (قبل از کدهای توربو) کدهایی با ساختار جبری از جمله کدهای خطی (که توسط الیاس [۱۴] کشف شد) مطرح شدند. پس از آن کدهای خوبی مانند **ریدمولر**[۱۵] و **کدهای پیچشی**[۱۶] ارائه شدند.

آنچه در این فصل ارئه می‌شود عبارت است از: در بخش ۲.۱ سیستم ارتباطی شرح داده می‌شود. در بخش ۳.۱ کانال‌های ارتباطی مورد نیاز تعریف می‌شوند. در بخش ۴.۱ فرضیاتی که در این پایان‌نامه مورد استفاده قرار می‌گیرند بیان می‌شوند. در بخش ۵.۱ مقیاس گذاری LLR برای کانال‌های مورد نظر بیان می‌گردد.

۲.۱ سیستم ارتباطی

همانطور که در شکل ۲.۱ می‌بینید ، در یک سیستم ارتباطی یک پیام (از منبع) ابتدا توسط کدگذار منبع، کدگذاری (رمزگذاری) می‌گردد (در شکل ۲.۱ الفبای کدگذاری میدان دودویی در نظر گرفته شده است ولی می‌تواند تغییر کند) . حاصل این عمل، برداری به طول k است. سپس کدگذار کانال به این بردار با توجه به مدل کدگذاری کانال

[۱۴] Elias
[۱۵] Reed-Muller
[۱۶] Convolutional codes

مورد نظر **افزونگی**[۱۷] اضافه می‌کند. که این افزونگی موجب می‌شود که کدگشای کانال بتواند خطای حاصل از پارازیت کانال را در صورت امکان کشف و تصحیح کند. بردار جدید حاصل از مرحله کدگذاری کانال را **کدواژه**[۱۸] می‌نامند که عنصری متعلق به $F_۲^n$ (با فرض الفبای دودویی $GF(۲)$ یا $F_۲$) است. به این ترتیب k را **بعد** (اندازه) کد و n را **طول کد** (طول کدواژه) می‌نامند. کدواژه از طریق کانال ارسال می‌گردد. کانال معمولا خواصی دارد که (این خواص) موجب تغییر سیگنال‌هایی که از کانال عبور می‌کنند، می‌گردند و آنها را تحریف (خراب) می‌کند. یکی از مواردی که موجب تحریف کدواژه ارسالی می‌گردد وجود پارازیت (نوفه) در کانال است. در انتهای کانال، گیرنده واژه‌ای را از کانال دریافت می‌کند که (با توجه به تغییرات رخ داده در پیام ارسالی در اثر پارازیت) ممکن است کدواژه نباشد. کدگشای کانال، سعی خواهد کرد خطاهای احتمالی واژه دریافتی را در صورت امکان کشف و تصحیح کند و به این ترتیب محتمل‌ترین کدواژه (که به طور محتمل‌تر همان کدواژه ارسالی است) را به کدگشای منبع (رمزگشا) تحویل می‌دهد. کدگشای منبع نیز آن‌را رمزگشایی و پیام اصلی را از آن استخراج می‌کند. به مثال زیر توجه کنید.

مثال ۱.۲.۱. [۲۱] عمل کپی کردن فیلم یا اطلاعات روی DVD را در نظر بگیرید. در این عمل، کامپیوتر به عنوان کدگذار منبع و کدگذار کانال عمل می‌کند. یعنی اطلاعات را به داده‌های دودویی تبدیل می‌کند و با توجه به مدل کدگذاری کانال مورد استفاده افزونگی مورد نیاز را به آن اضافه می‌کند. سپس اطلاعات را روی DVD کپی می‌کند. در اینجا DVD، کانال است. DVDممکن است خراشیده یا کثیف گردد، به این ترتیب برخی از اطلاعات روی آن خراب خواهند شد. اجراکننده DVD (DVD player) می‌تواند این داده‌ها را تعمیر کند و آنها را بخواند (یعنی به عنوان کدگشای کانال و کدگشای منبع عمل می‌کند).

♢

به این ترتیب می‌توان فرض کرد:

[۱۷] Redundancy
[۱۸] Codeword

- بردار اطلاعات: $u = \left[\begin{array}{ccc} u_۱ & \dots & u_k \end{array} \right]^T \in F_۲^k$
- کدواژه: $x = \left[\begin{array}{ccc} x_۱ & \dots & x_n \end{array} \right]^T \in \mathcal{C}$ ، که $x \in F_۲^n$ و $\mathcal{C}$ مجموعه همه کدواژه‌های (مجاز) است، به طوری که $|\mathcal{C}| = ۲^k$ (برای کدهای خطی که در فصل بعد بیان خواهد شد) یعنی:

$$\mathcal{C} = \{x : \ \ x = enc(u) \ , \ u \in F_۲^k\}$$

(منظور از $enc(u)$ کدگذاری u است).

- واژه دریافتی $y = \left[\begin{array}{ccc} y_۱ & \dots & y_n \end{array} \right]^T$ ، مجموعه الفبای y به کانال مورد استفاده وابسته است.
- کدواژه برآورد شده $\hat{x} = \left[\begin{array}{ccc} \hat{x}_۱ & \dots & \hat{x}_n \end{array} \right]^T \in F_۲^n$
- بردار اطلاعات برآورد شده $\hat{u} = \left[\begin{array}{ccc} \hat{u}_۱ & \dots & \hat{u}_k \end{array} \right]^T \in F_۲^k$
- نرخ کد: $R = k/n$ (برای کدهای خطی که در فصل بعد بیان خواهد شد).

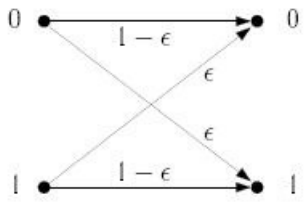
۳.۱ مدل‌های کانال مورد استفاده در این پایان‌نامه

تعریف ۱.۳.۱. یک کانال را **بی‌حافظه**[۱۹] گویند اگر خروجی کانال در هر لحظه فقط به ورودی کانال در همان لحظه وابسته باشد. به عبارت دیگر اگر $y = y_۱, \dots, y_n$ خروجی و $x = x_۱, \dots, x_n$ ورودی باشد آنگاه:

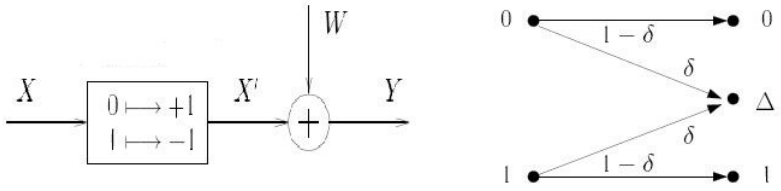
$$p(y|x) = \prod_{i=۱}^{n} p(y_i|x_i) \qquad (۱.۱)$$

در این حالت کانال به طور کامل توسط الفبای ورودی و خروجی شرح داده می‌شود.

[۱۹] Memoryless



شکل ۱.۳: BSC با احتمال همگذری ϵ



شکل ۱.۴: نمودار سمت راست BECبا احتمال پاکشدگی کانال δ و نمودار سمت چپ کانال (BI-AWGNC) نرمال شده است.

تعریف ۱.۳.۲. کانال متقارن دودویی (BSC)[۲۰] ، کانالی ورودی دودویی، خروجی دودویی با پارامتر ϵ است که در آن $p(۰|۱) = p(۱|۰) = \epsilon$ و $p(۱|۱) = p(۰|۰) = ۱ - \epsilon$ است و به ϵ **احتمال همگذری**[۲۱] می گویند.

به نمودار (۳.۱) توجه کنید.

تعریف ۱.۳.۳. کانال پاک شدگی دودویی BEC[۲۲] کانالی با پارامتر δ است که به آن **احتمال پاک‌شدگی**[۲۳] می گویند و الفبای ورودی آن $\{۰, ۱\}$ و الفبای خروجی آن $\{۰, ۱, ?\}$ است و در آن

$p(۱|۱) = p(۰|۰) = ۱ - \delta$ و $p(?|۱) = p(?|۰) = \delta$ است (منظور از ? بیت پاک‌شده است). به نمودار ۴.۱ توجه کنید.

تعریف ۱.۳.۴. در کانال $(BI\text{-}AWGNC)$ [۲۴] ، ورودی x به $x' \in \{+۱, -۱\}$ نگاشته می‌شود سپس پارازیت سفید نرمال به آن اضافه می‌شود. به این ترتیب خروجی $y = x' + w$ است که $w \sim \mathcal{N}(۰, N_۰/۲E_s)$. الفبای

[۲۰] Binary Symmetric Channel
[۲۱] Cross over probability
[۲۲] Binary Erasure Channel
[۲۳] erasure probability
[۲۴] Binary Input Additive White Gaussian Noise Channel

خروجی این کانال $y \in \mathbb{R}$ است. به نمودار ۴.۱ توجه کنید) $N_0/2E_s$ نسبت سیگنال به پارازیت هر نماد کد (SNR) است).

۱.۳.۱ نسبت‌های درستنمایی

یکی از مفاهیمی که در این پایان‌نامه به دفعات مورد استفاده قرار می‌گیرد، **لگاریتم نسبت درستنمایی (LLR)**[۲۵] است که به صورت زیر تعریف می‌شود:

$$l = \lambda_i = L(y_i|x_i) = \ln\left(\frac{p(y_i|x_i = 0)}{p(y_i|x_i = 1)}\right) \qquad (۲.۱)$$

مقدار l ، میزان قطعی بودن اینکه x ، ۰ یا ۱ است را شرح می‌دهد. اگر l مثبت باشد آنگاه

$$p(y_i|x_i = 0) > p(y_i|x_i = 1)$$

و باید $\hat{x} = 0$ باشد. به همین ترتیب اگر l منفی باشد آنگاه

$$p(y_i|x_i = 0) < p(y_i|x_i = 1)$$

و $\hat{x} = 1$. هرچه مقدار $|l|$ بزرگتر باشد آنگاه قابلیت اطمینان نماد، بیشتر خواهد بود.

قانون تصمیم سخت l عبارت است از:

$$\hat{x} = \begin{cases} 0 & l \geq 0 \\ 1 & l < 0 \end{cases}$$

به این ترتیب مقدار l برای کانالهای تعریف شده در این فصل به صورت زیر است:

- BSC :

$$L_{BSC}(y_i|x_i) = \begin{cases} \ln\left(\frac{1-\epsilon}{\epsilon}\right) & y_i = 0 \\ \ln\left(\frac{\epsilon}{1-\epsilon}\right) & y_i = 1 \end{cases}$$

[۲۵] Log likelihood ratio

- BEC :

$$L_{BEC}(y_i|x_i) = \begin{cases} \infty & y_i = ۰ \\ ۰ & y_i =? \\ -\infty & y_i = ۱ \end{cases}$$

- BI-AWGNC :

$$L_{BI-AWGNC}(y_i|x_i) = ۴\frac{E_s}{N_۰}y_i$$

۴.۱ فرضیات

در همه این پایان‌نامه عناصر منبع دارای توزیع یکسان و مستقل هستند. به این ترتیب برای متغیرهای مستقل $\{u_k\}$ داریم:

$$p(u) = \prod_{i=۱}^{k} p(u_k) \qquad (۳.۱)$$

با توجه به اینکه $\forall i = ۱, \ldots, k$ ، $u_k \in F_۲$ توزیع یکسانی دارند، پس:

$$\forall i = ۱, \ldots, k \qquad p(u_k) = ۱/۲ \qquad (۴.۱)$$

به این ترتیب از (۳.۱) و (۴.۱) داریم:

$$p(u) = ۱/۲^k \qquad (۵.۱)$$

همچنین فرض می‌کنیم کانال‌های مورد بررسی در این پایان‌نامه بی‌حافظه و بدون پس‌خورد هستند. یعنی:

$$p(y|x) = \prod_{i=۱}^{n} p(y_i|x_i)$$

و بدون پس‌خورد بودن کانال یعنی:

$$p(y_i|x) = p(y_i|x_i)$$

در سراسر این پایان‌نامه مبنای لگاریتم را در مواردی که به طور صریح مبنا ذکر نشده است e درنظر می‌گیریم (لگاریتم طبیعی) و همچنین منظور از بردار ، ماتریس ستونی است.

۵.۱ مقیاس گذاری LLR

مقیاس گذاری LLR برای یک کانال در هنگام انجام فرآیند کدگشایی می‌تواند نقش مهمی را ایفا کند. تحت کانال BSC با احتمال همگذری ϵ مقیاس گذاری $\lambda_i(LLR)$ توسط ثابت β به صورت زیر است:

$$\beta.\lambda_i = \beta.\ln\left(\frac{p(y_i|x_i=۰)}{p(y_i|x_i=۱)}\right) = \begin{cases} \beta.\ln\left(\frac{۱-\epsilon}{\epsilon}\right) & y_i = ۰ \\ \beta.\ln\left(\frac{\epsilon}{۱-\epsilon}\right) & y_i = ۱ \end{cases}$$

به این ترتیب می‌توان λ_i را توسط β به ± ۱ مقیاس گذاری کرد. بنابراین آگاهی از احتمال همگذری (پارازیت) موجود در BSC هنگام استفاده از هر الگوریتم کدگشایی (با فرض اینکه الگوریتم اجازه استفاده از مقیاس گذاری را بدهد) مهم نخواهد بود. به طور مشابه برای کانالBEC مقیاس گذاری به صورت زیر است:

$$\beta.\lambda_i = \beta.\ln\left(\frac{p(y_i|x_i=۰)}{p(y_i|x_i=۱)}\right) = \begin{cases} \beta.\infty & y_i = ۰ \\ \beta.۰ & y_i =? \\ \beta.-\infty & y_i = ۱ \end{cases}$$

و در کانال BI-AWGNC داریم:

$$\beta.\lambda_i = \beta.(۴E_s/N_۰y) \qquad (۶.۱)$$

بنابراین مقیاس گذاری نسبت پارازیت به سیگنال توسط β ، نشان می‌دهد که هنگام تعیین λ_i برای کدگشایی، آگاهی از پارازیت لازم نیست.

آنچه در این پایان‌نامه خواهیم دید :

- در **فصل دوم** ، برخی تعاریف ابتدایی مورد نیاز در پایان‌نامه مطرح می‌گردد.
- در **فصل سوم** ، به معرفی کدهای LDPC، گراف‌های فاکتور و روش کدگذاری کدهای LDPC می‌پردازیم.

- در **فصل چهارم** ، و فصل ۵، دو روش کدگشایی کدهای LDPC بیان می‌شود. در فصل ۴ روش کدگشایی عبور پیام که بر مبنای گراف فاکتور است مطرح می‌شود.

- در **فصل پنجم** ، روش کدگشایی کدهای LDPC با استفاده از برنامه‌ریزی خطی و دو روش بهبود آن مطرح می‌شود.

- در **فصل ششم** ، مقایسه‌ای بین برخی روش‌های کدگشایی مطرح شده در این پایان‌نامه بیان می‌شود.

فصل ۲

مروری بر کدهای خطی و برخی مفاهیم مورد نیاز

۱.۲ مقدمه فصل

کدهای خطی ، کلاس مهمی از کدهای هستند که به دلیل ساختار جبری و خواصی که دارند بسیار مورد توجه قرار گرفته‌اند. در این فصل برخی تعاریف مورد نیاز در این پایان‌نامه در ارتباط با کدهای خطی بیان خواهد شد.

۲.۲ برخی تعاریف

مجوعه q عنصری $F = \{f_۱, \ldots . f_q\}$ را (که معمولا یک میدان متناهی است) به عنوان **الفبای کد** در نظر می‌گیرند و عناصر آن را **نماد کد** می‌نامند. کد $\mathcal{C}$ به طول n ، تحت الفبای F_q زیرمجموعه‌ای از

$$F^n = F \times F \times \ldots \times F$$

(یعنی F_q^n) است ، به عبارت دیگر مجموعه‌ای از n تایی‌ها مرتب (تحت F_q) است که به هر یک از این n تایی‌های مرتب (عناصر یک کد به طول n) یک **کدواژه** می‌گویند . علاوه بر این باید توجه داشت که به هریک از عناصر F_q^n ، در حالت کلی **واژه** می‌گویند. برای نمایش یک کد $\mathcal{C}$ از (n, M)- کد $\mathcal{C}$ استفاده می‌کنند که در آن n طول کد (طول بلوکی) و M اندازه کد (تعداد کدواژه‌های متعلق به کد $\mathcal{C}$ یعنی : $|\mathcal{C}|$) است. نرخ کد $\mathcal{C}$ برابراست با : $\frac{log_q|\mathcal{C}|}{n}$.

تعریف ۱.۲.۲. اگر همه کدواژه‌های متعلق به یک کد ، طول مساوی داشته باشند آنگاه به آن **کد بلوکی**[۱] می گویند.

تعریف ۲.۲.۲. به ازای هر دو کدواژه x, y متعلق به کد $\mathcal{C}$ ، **فاصله هامینگ**[۲] $d(x, y)$ بین آن دو عبارت است از تعداد مکان‌هایی که آن دو کدواژه مؤلفه‌های متفاوتی دارند. **مینیمم فاصله کد** $\mathcal{C}$ عبارت است از : کوچکترین فاصله از مجموعه فاصله‌های بین هر دو کدواژه مجزا متعلق به کد $\mathcal{C}$. آنچه درباره مینیمم فاصله هامینگ مهم است این است که مینیمم فاصله یک کد ، توانایی تصحیح خطای کد را نشان می‌دهد. **وزن**[۳] یک کدواژه عبارت است از تعداد مؤلفه‌های غیر صفر آن کدواژه.

به این ترتیب یک (n, M)-کد با مینیمم فاصله d را می‌توان به صورت (n, M, d)-کد نمایش داد. توانایی یک کد توسط تعداد کدواژه‌ها و فاصله بین کدواژه‌های آن کد مشخص می‌شود. به این ترتیب دو کد را که این دو کمیت آنها مساوی است ، کدهای هم ارز در نظر می‌گیرند. با استفاده از دوعملگر (روی کدواژه‌های کد $\mathcal{C}$) که در ادامه بیان می‌شود می‌توان کد هم ارز با $\mathcal{C}$ را ساخت. این دو عملگر عبارت‌اند از :

۱. جایگشت دلخواه مؤلفه‌های همه کدواژه‌ها.

۲. جایگشت‌های دلخواه نمادها (متعلق به مجموعه الفبا)که به طور مستقل روی هر مجموعه از مؤلفه‌ها به کار می‌رود.

۳.۲ کدهای بلوکی خطی

کد بلوکی خطی[۴] به طول n تحت F_q ، زیر فضایی از فضای برداری F_q^n است با فرض اینکه بعد (اندازه) کد k باشد ، با $[n, k]$-کد بلوکی خطی ، نمایش داده می‌شود. به این ترتیب یک کد ، کد خطی است اگر در شرایط زیر صدق کند :

[۱] Block code
[۲] Hamming distance
[۳] Weight
[۴] Linear block codes

۱. جمع یا تفاضل هر دو کدواژه ، یک کدواژه است.

۲. بردار صفر ، همواره یک کدواژه متعلق به کد خطی است.

۳. تعداد کدواژه‌های هر $[n,k]$- کد بلوکی خطی برابر است با q^k .

(علت ۳ : زیرا k بردار پایه در $\mathcal{C}$ وجود دارد و تعداد کل ترکیبات خطی ممکن با توجه به اینکه هر ضریب عددی را به q حالت (از میدان F_q) می توان انتخاب کرد برابر با q^k حالت است.) در کد خطی با توجه به اینکه تعداد کدواژه‌ها را می‌توان با استفاده از بعد زیر فضا محاسبه کرد لذا می‌توان از نماد $[n,k,d]$ برای کدهای خطی استفاده کرد.

گزاره ۱.۳.۲. [۵] در یک کد خطی ، مینیمم فاصله مساوی با مینیمم وزن کدواژه‌های غیر صفر آن کد است.

اثبات. اگر $\mathcal{C}$ کد خطی باشد و $x,y \in \mathcal{C}$ آنگاه $x-y \in \mathcal{C}$ (زیرا $\mathcal{C}$ خطی است پس ترکیب خطی هر دو کدواژه باز هم یک کدواژه است) بنابراین داریم:

$$d(x,y) = d(x-y, 0) = wt(x-y)$$

□

برای ساخت کدهای خطی از دو ماتریس مولد و ماتریس بررسی‌زوجیت استفاده می‌کنند ، که در ادامه بیان خواهد شد.

تعریف ۲.۳.۲. فرض کنید $\mathcal{C}$ یک $[n,k]$- کدخطی باشد ، ماتریس $k \times n$ ، G را **ماتریس مولد**[۵] کد خطی $\mathcal{C}$ گویند هرگاه سطرهای آن بردارهای پایه زیر فضای خطی $\mathcal{C}$ باشند.

فرض کنید $\{u : u \in F_q^k\}$ ، مجموعه بردارهای اطلاعات باشند، به این ترتیب کدخطی $\mathcal{C}$ متناظر با این مجموعه به صورت زیر بدست می‌آید:

$$\mathcal{C} = \{u^T G : u \in F_q^k\}$$

[۵] Generator matrix

۴.۲ کدهای خطی هم‌ارز

در مفهوم کلی کدهای هم‌ارز، چون لازم نیست همه خواص کدهای خطی در نظر گرفته شود به همین دلیل ممکن است یک کد غیر خطی با یک کد خطی هم‌ارز باشد. به منظور یافتن کدهای هم‌ارز با یک کد خطی، که خواص کد خطی را داشته باشند باید عملگرهای مجاز را محدود کنیم. بنابراین داریم:

۱. جایگشت دلخواه مولفه‌ها

۲. ضرب یک اسکالر غیر صفر (با توجه به مجموعه الفبای کد مورد نظر) در هر مختصات.

به این ترتیب دو کد خطی هم ارز هستند اگر بتوان یکی از آن دو را با انجام تعداد متناهی از دو عمل بیان شده از دیگری بدست آورد. با انجام اعمال سطری (ستونی) مقدماتی متناظر با دو عملی که در بالا ذکر شد، روی ماتریس مولد G کد خطی، یک ماتریس مولد برای کد هم‌ارز تولید می‌شود. باتوجه به اینکه رتبه G ، k است می‌توان با انجام اعمال سطری مقدماتی مجاز ماتریس G را به فرم $G = [I_k \;\; A]$ تبدیل کرد و یک کد خطی هم‌ارز بدست آورد. که در آن I ماتریس همانی $k \times k$ و A ماتریس $k \times (n-k)$ است و آنرا **فرم استاندارد** G می‌نامند.

مثال ۱.۴.۲. فرض کنید $\mathcal{C}$ یک $[۷, ۴]$- کد خطی باشد که توسط ماتریس مولد G بدست می‌آید

$$G = \begin{bmatrix} ۱ & ۰ & ۰ & ۰ & ۱ & ۱ & ۰ \\ ۰ & ۱ & ۰ & ۰ & ۰ & ۱ & ۱ \\ ۰ & ۰ & ۱ & ۰ & ۱ & ۱ & ۱ \\ ۰ & ۰ & ۰ & ۱ & ۱ & ۰ & ۱ \end{bmatrix}$$

به این ترتیب کدواژه‌های کد $\mathcal{C}$ به صورت زیر است:

$$\begin{array}{cccc} ۰۰۰۰۰۰۰ & ۱۱۰۱۰۰۰ & ۰۱۱۰۱۰۰ & ۰۰۱۱۰۱۰ \\ ۰۰۰۱۱۰۱ & ۱۰۰۰۱۱۰ & ۰۱۰۰۰۱۱ & ۱۰۱۰۰۰۱ \\ ۱۱۱۱۱۱۱ & ۰۰۱۰۱۱۱ & ۱۰۰۱۰۱۱ & ۱۱۰۰۱۰۱ \\ ۱۱۱۰۰۱۰ & ۰۱۱۱۰۰۱ & ۱۰۱۱۱۰۰ & ۰۱۰۱۱۱۰ \end{array} \qquad (۱.۲)$$

چون مینیمم فاصله این کد ۳ است بنابراین یک کد تصحیح ۱-خطا است. این کد را $[۷, ۴, ۳]$- کد هامینگ می‌نامند. (در ادامه شرح داده می‌شوند)

$\diamondsuit$

ماتریس بررسی‌زوجیت

فرض کنید $\mathcal{C}$ یک کد باشد **مکمل متعامد** [۶] کد $\mathcal{C}$ عبارت است از مجموعه همه بردارهایی که به هر بردار متعلق به $\mathcal{C}$ عمود هستند (حاصل ضرب نقطه‌ای آنها صفر است)، این مجموعه یک زیر فضا است [۵]. بنابراین یک کدخطی است که به آن **کددوگان**[۷] می‌گویند و با $\mathcal{C}^{\perp}$ نمایش می‌دهند. اگر $\mathcal{C}$ یک $[n,k]$-کد خطی باشد آنگاه $\mathcal{C}^{\perp}$ یک $[n,n-k]$-کد خطی است. ماتریس مولد $\mathcal{C}^{\perp}$ را **ماتریس بررسی‌زوجیت**[۸] H ، کد $\mathcal{C}$ می‌نامند که یک ماتریس $(n-k)\times n$ است. با استفاده از H می‌توان کدواژه‌های کد $\mathcal{C}$ را بازیابی کرد، زیرا آنها باید بر سطرهای H عمود باشند (بردارهای پایه $\mathcal{C}^{\perp}$) به این ترتیب کد $\mathcal{C}$ را می‌توان به صورت زیر نیز به دست آورد:

$$\mathcal{C}=\{x\in F_q^n:\ Hx=\mathbf{0}\}$$

(منظور از $\mathbf{0}$ ماتریس ستونی صفر از مرتبه $(n-k)\times ۱$ است)

مثال ۲.۴.۲. ماتریس بررسی‌زوجیت $[۷,۴,۳]$-کد هامینگ به صورت زیر است:

$$H=\begin{bmatrix} ۱ & ۰ & ۱ & ۱ & ۱ & ۰ & ۰ \\ ۱ & ۱ & ۱ & ۰ & ۰ & ۱ & ۰ \\ ۰ & ۱ & ۱ & ۱ & ۰ & ۰ & ۱ \end{bmatrix}$$

کدواژه $[۰۰۰۱۱۰۱]^T$ باتوجه به مثال (۱.۴.۲) متعلق به کد مورد نظر است بنابراین

$$\begin{bmatrix} ۱ & ۰ & ۱ & ۱ & ۱ & ۰ & ۰ \\ ۱ & ۱ & ۱ & ۰ & ۰ & ۱ & ۰ \\ ۰ & ۱ & ۱ & ۱ & ۰ & ۰ & ۱ \end{bmatrix}\begin{bmatrix} ۰ \\ ۰ \\ ۰ \\ ۱ \\ ۱ \\ ۰ \\ ۱ \end{bmatrix}=\begin{bmatrix} ۰ \\ ۰ \\ ۰ \end{bmatrix}$$

$\diamondsuit$

قضیه ۳.۴.۲. [۵] فرض کنید H ماتریس بررسی‌زوجیت $[n,k]$ - کد خطی $\mathcal{C}$ باشد. هر مجموعه از $s-۱$ ستون‌های H مستقل خطی اند اگر و فقط اگر مینیمم فاصله هامینگ $\mathcal{C}$ حداقل s باشد.

---

[۶] Orthogonal complemen
[۷] Dual code
[۸] Parity check matrix

از قضیه (۳.۴.۲) می‌توان به این نتیجه رسید که مینیمم فاصله کدخطی $\mathcal{C}$ با ماتریس بررسی‌زوجیت H ، d است اگر و فقط اگر هر $d-1$ ستون H مستقل خطی باشد و مجموعه از d ستون H وابسته خطی باشند. به این ترتیب از این قضیه می‌توان برای یافتن مینیمم فاصله کدخطی استفاده کرد.

قضیه ۴.۴.۲. [۵] اگر $G = [I_k \ A]$ ماتریس مولد (فرم استاندارد)، $[n,k]$-کد خطی $\mathcal{C}$ باشد آنگاه

$$H = [-A^T \ I_{n-k}]$$

ماتریس بررسی‌زوجیت $\mathcal{C}$ است.

۵.۲ کد هامینگ

کد هامینگ از مرتبه r تحت F_q یک $[n,k]$-کد خطی است که $n = (q^r-1)/(q-1)$ و $k = n-r$ ، با ماتریس بررسی‌زوجیت H_r که ماتریسی $r \times n$ به طوری که ستون‌های H_r غیرصفراند و هیچ دو ستون آن مضربی (متعلق به مجموعه الفبای مورد نظر) ازیکدیگر نیستند. توجه کنید که q^r-1 تعداد بردارهای غیر صفر به طول r تحت F_q است و $q-1$ تعداد عناصر غیر صفر است. بنابراین n ماکزیمم تعداد بردارهایی غیر صفر به طول r است که هیچ یک مضارب اسکالر یکدیگر نیستند. به این ترتیب از قضیه ۳.۴.۲ نتیجه می‌گیریم که مینیمم فاصله همه کدهای هامینگ دقیقا ۳ است و بنابراین کدهای تصحیح کننده ۱-خطا هستند.

مثال ۱.۵.۲. [۱۸] کد هامینگ مرتبه $r = ۳$ تحت $F_۲$ که توسط ماتریس بررسی‌زوجیت

$$H_۳ = \begin{bmatrix} ۱ & ۰ & ۰ & ۱ & ۰ & ۱ & ۱ \\ ۰ & ۱ & ۰ & ۱ & ۱ & ۰ & ۱ \\ ۰ & ۰ & ۱ & ۰ & ۱ & ۱ & ۱ \end{bmatrix}$$

تولید می‌شود یک $[۷, ۴]$-کد خطی با مینیمم فاصله ۳ است. با مرتب کردن مجدد ستون‌های $H_۳$ یک کد هامینگ هم ارز تعریف می‌شود.

$\diamondsuit$

۶.۲ کدگشایی

روش‌های کدگشایی که در این بخش تعریف می‌شوند با توجه به نیازهای فصل‌های بعد انتخاب شده‌اند.

کدگشایی ماکزیمم درستنمایی(ML)

فرض کنید x کدواژه ارسالی (از طریق کانال پارازیت‌دار)، y واژه دریافتی باشد و $\hat{x}$ نتیجه حاصل از کدگشایی y باشد. هدف از **کدگشایی ماکزیمم درستنمایی (بلوکی)**[۹] یافتن کدواژه‌ای است که به طور محتمل‌تر کدواژه ارسالی باشد و این کار به صورت زیر انجام می‌گردد:

$$\hat{x} = \underset{x \in \mathcal{C}}{\operatorname{argmax}} \; p(y|x) \qquad (۲.۲)$$

به این ترتیب $\hat{x}$ ماکزیمم احتمال پیشین ($p(y|x)$) را دارا می‌باشد.

با توجه به اینکه ممکن است بیش از یک کد واژه با ماکزیمم احتمال پیشین وجود داشته باشد دو نوع کدگشای ML تعریف می‌شود:

۱. کدگشای ML دقیق (کامل): در این کدگشا اگر بیش از یک کدواژه با احتمال پیشین ماکزیمم وجود داشته باشد کدگشا یکی از آنها را انتخاب می‌کند.

۲. کدگشای ML ناقص: در این کدگشایی اگر بیش از یک کد واژه با ماکزیمم احتمال پیشین وجود داشته باشد آنگاه کدگشا تقاضای ارسال مجدد می‌کند.

کدگشای MAP

راه دیگر کدگشایی واژه دریافتی y ، یافتن **ماکزیمم احتمال پسین (MAP)**[۱۰] و سپس حدس‌زدن کدواژه ارسالی است. یعنی:

$$\hat{x} = \underset{x \in \mathcal{C}}{\operatorname{argmax}} \; p(x|y) \qquad (۳.۲)$$

[۹] Maximum likelihood decoding
[۱۰] Maximum a posteriori probability

به این کدگشا، **کدگشای MAP بلوکی** می‌گویند. اگر احتمال پسین به صورت نمادی انجام گیرد ($p(x_i|y)$)، آنگاه کدگشای حاصل را **کدگشای MAP نمادی** گویند. یعنی:

$$\hat{x}_i = \underset{x_i \in \{0,1\}}{\text{argmax}} \; p(x_i|y) \tag{۴.۲}$$

(با فرض اینکه $\hat{x}_i$ نمادکد برآورد شده باشد)

برای کدگشای MAP بلوکی با استفاد ه از قانون بیز داریم:

$$\begin{aligned} \hat{x} &= \underset{x \in \mathcal{C}}{\text{argmax}} \; \frac{p(x,y)}{p(y)} \\ &= \underset{x \in \mathcal{C}}{\text{argmax}} \; p(y|x)\,\frac{p(x)}{p(y)} \end{aligned} \tag{۵.۲}$$

با توجه به این فرض که کدگذار نگاشتی یک به یک است و همچنین فرض هم‌احتمال بودن (۵.۱)، بنابراین ثابت $= p(x)$. از طرفی با توجه به اینکه ماکزیمم‌سازی فقط تحت x انجام می‌گیرد پس احتمال واژه دریافتی ($p(y)$) را می‌توان به عنوان یک ثابت در نظر گرفت. به این ترتیب داریم:

$$\hat{x} = \underset{x \in \mathcal{C}}{\text{argmax}} \; p(y|x) \tag{۶.۲}$$

که رابطه (۶.۲) همان کدگشای ML بلوکی است. به طور مشابه می‌توان ML نمادی را نیز از MAP نمادی به دست آورد، که حاصل کار به صورت زیر است:

$$\begin{aligned} \hat{x}_i &= \underset{x_i \in \{0,1\}}{\text{argmax}} \; p(x_i|y) \\ &= \underset{x_i \in \{0,1\}}{\text{argmax}} \; p(y|x_i) \end{aligned} \tag{۷.۲}$$

به طور خلاصه این ۴ کدگشا را می‌توان در جدول (۸.۲) دید.

نام کدگشا	فرمول
بلوکی ML	$\hat{x} = \underset{x \in \mathcal{C}}{\text{argmax}} \; p(y\|x)$
نمادی ML	$\hat{x_i} = \underset{x_i \in \{۰,۱\}}{\text{argmax}} \; p(y\|x_i) \quad \forall i = ۱,\ldots,n$
بلوکی MAP	$\hat{x} = \underset{x \in \mathcal{C}}{\text{argmax}} \; p(x\|y)$
نمادی MAP	$\hat{x_i} = \underset{x_i \in \{۰,۱\}}{\text{argmax}} \; p(x_i\|y) \quad \forall i = ۱,\ldots,n$

(۸.۲)

فصل ۳

کدهای LDPC

۱.۳ مقدمه فصل

در دهه ۱۹۹۰ پیشرفت‌هایی در زمینه کدگذاری اتفاق افتاد که موجب طلوع نظریه **کدگذاری مدرن**[۱] شد. در نظریه کدگذاری مدرن ، کدها توسط مدل‌های گرافی تنک (اسپارس) و تصادفی توصیف می‌شوند. علاوه براین کدگذاری وکدگشایی آنها توسط الگوریتم‌های محلی کارا انجام می شوند. ایده کدهای تصادفی اولین بار توسط شانون [۲۰] مطرح شد. آنچه که در این نظریه جدید است، استفاده از مدل‌های گرافی تنک و خاصیت محلی بودن الگوریتم‌ها است.

آنچه در این فصل ارائه خواهد شد، عبارت است از:

در بخش ۲.۳ **گراف های فاکتور** [۲] که در کدگشایی عبورپیام مورد استفاده قرار می‌گیرند، بیان می‌شوند. علاوه براین در این بخش ، گراف خاصی به نام **گراف تنر** [۳] که برای نمایش کدهای خطی بلوکی (از جمله کدهایLDPC) مورد استفاده قرار می‌گیرد ، تعریف شده است. در بخش ۳.۳ **کدهای LDPC** و ساختارهای مختلف آنها شرح داده می‌شود. در بخش ۴.۳ نحوه کدگذاری این نوع کدها بیان شده است.

به‌این‌ترتیب ، قبل از بیان کدهای LDPC نیازمند بیان تعاریفی هستیم .

[۱] Modern coding theory
[۲] Factor graphs
[۳] Tanner graph

۲.۳ گراف فاکتور

گراف فاکتور (FG) ، نوعی نمایش گرافی، برای ساخت **توابع حاشیه‌ای** [۴] مربوط به توابع چند متغیره است. برخی از مسائل که توسط الگوریتم‌ها حل می‌شوند را می‌توان با استفاده از این روش تفسیر کرد. الگوریتم کدگشایی (عبور پیام) که در فصل ۴ بیان می‌شود، نمونه‌ای از این مسائل است.

۱.۲.۳ قانون توزیع پذیری و گراف فاکتور

فرض کنید F (می‌توان فرض کرد که $F = \mathbb{R}$) یک میدان باشد، برای هر $a, b, c \in F$ قانون توزیع‌پذیری به صورت (۱.۳) است.

$$ab + ac = a(b + c) \qquad (۱.۳)$$

با استفاده از این قانون می‌توان پیچیدگی محاسبات را تا حدود زیادی کاهش داد. به عنوان مثال می‌توان $\sum_{i,j} a_i b_j$ را به صورت $(\sum_i a_i)(\sum_j b_j)$ محاسبه کرد که در آن به ازای هر j, i ، $a_i, b_j \in F$ است.

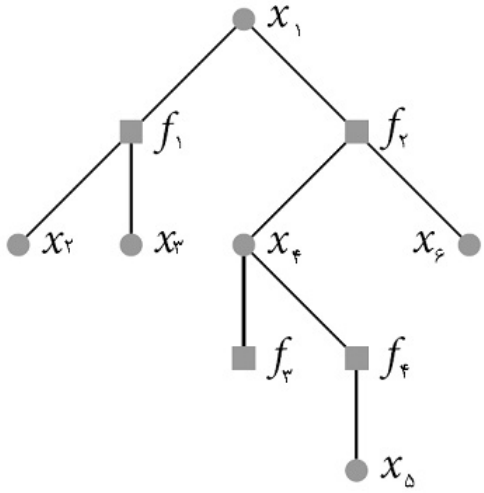
تعریف ۱.۲.۳. گراف فاکتور ، گرافی دوبخشی است که ساختار فاکتورگیری [۵] را نمایش می دهد. راس‌های بخش اول را راسهای متغیر [۶] (متناظر با متغیر x_i) و راس‌های بخش دوم را راس‌های فاکتور [۷] (متناظر با فاکتور (تابع) f_j) می‌نامند. راس متغیر x_i توسط یالی به راس فاکتور f_j متصل می شود اگر وتنها اگر متغیر x_i آرگومان (متغیر مستقل) فاکتور f_j باشد.

مثال ۲.۲.۳. [۱۸] تابع f که به صورت زیر فاکتورگیری شده است را در نظر بگیرید:

$$f(x_۱, x_۲, x_۳, x_۳, x_۴, x_۵, x_۶) = f_۱(x_۱, x_۲, x_۳,)f_۲(x_۱, x_۴, x_۶)f_۳(x_۴)f_۴(x_۴, x_۵) \qquad (۲.۳)$$

(گراف متناظر با آن در شکل (۱.۳) نشان داده شده است). فرض کنید بخواهیم تابع حاشیه‌ای f را نسبت به $x_۱$

[۴] Marginal
[۵] Factorization
[۶] Variable node(vertex)
[۷] Factor node(vertex)



شکل ۱.۳: گراف فاکتور متناظر با مثال ۲.۲.۳ درخت است. از دایره برای نمایش راس متغیر و از مربع برای نمایش راس کنترل استفاده می‌شود .

محاسبه کنیم اگر این تابع حاشیه‌ای را به صورت $f(x_1)$ نمایش دهیم، می‌نویسیم:

$$\begin{aligned} f(x_1) &= \sum_{x_2,x_3,x_4,x_5,x_6} f(x_1,x_2,x_3,x_3,x_4,x_5,x_6) \\ &= \sum_{\sim x_1} f(x_1,x_2,x_3,x_3,x_4,x_5,x_6) \end{aligned} \tag{۳.۳}$$

با فرض اینکه همه متغیرها مقادیرشان را از مجموعه الفبایی به نام χ اخذ کنند.

$\diamondsuit$

قرارداد: [۱۸] منظور از $\sum_{\sim x_i}$ جمع تحت همه متغیرهای به غیر از x_i است. (این قرارداد به منظور مختصرنویسی اتخاذ شده است).

تعریف ۳.۲.۳. فرض کنید $\mathcal{C}$ کد خطی دودویی با ماتریس بررسی‌زوجیت $H_{m\times n}$ باشد (نیاز نیست سطرهای H مستقل‌خطی باشند). **گراف تنر** متناظر با H گرافی دوبخشی است که برای نمایش فاکتورگیری تابع عضویت کد استفاده می‌شود. راس‌های این گراف شامل n راس متغیر، متناظر با مولفه‌های (بیت‌های) کدواژه (یا به عبارت دیگر ستون‌های H) و m راس بررسی متناظر با مجموعه معادلات بررسی‌زوجیت(یا به عبارت دیگر سطرهای H) است. راس بررسی j به راس متغیر i متصل است اگر وفقط اگر $H_{j,i} = 1$ باشد (به بیان دیگر اگر متغیر i در

معادله بررسی‌زوجیت j شرکت داشته باشد. منظور از $H_{j,i}$ عنصر سطر j ام ستون i ام ماتریس H است) .

اولین بار فردی به نام تنر، برای نمایش کدها و تجسم کدگشایی تکراری (از جمله الگوریتم عبور پیام) از گراف‌های دوبخشی استفاده کرد.

مثال ۳.۲.۴. [۱۸] کد دودویی $\mathcal{C}$ با ماتریس بررسی‌زوجیت زیر را در نظر بگیرید:

$$H = \begin{bmatrix} ۱ & ۱ & ۰ & ۱ & ۰ & ۰ & ۰ \\ ۰ & ۰ & ۱ & ۱ & ۰ & ۱ & ۰ \\ ۰ & ۰ & ۰ & ۱ & ۱ & ۰ & ۱ \end{bmatrix}$$

فرض کنید $F_۲$ میدان دودویی با عناصر $\{۰, ۱\}$ و $X = (x_۱, \ldots, x_۷)$ باشد. تابع $f(x_۱, \cdots, x_۷)$ به صورت زیر تعریف می‌شود:

$$f : F_۲^۷ \longmapsto \{۰, ۱\} \subset \mathbb{R}$$

$$f(x_۱, \ldots, x_۷) = \mathbb{I}_{\{x \in C(H)\}} = \begin{cases} ۱ & H \otimes x = \mathbf{0} \\ ۰ & o.w \end{cases}$$

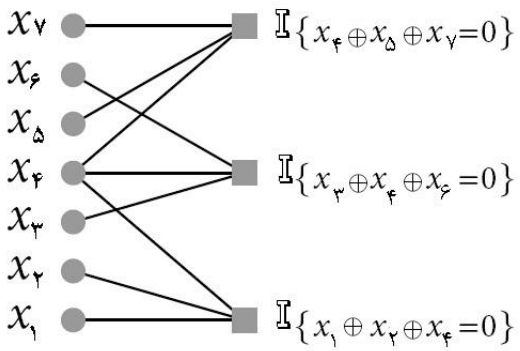
(منظور از $\mathcal{C}(H)$ کد $\mathcal{C}$ است که توسط H تعریف می‌شود و $\mathbf{0}$ ماتریس ستونی صفر است. علاوه بر این $\oplus$ جمع به پیمانه ۲ و $\otimes$ ضرب به پیمانه ۲ است، با توجه به اینکه نتایج ضرب به پیمانه ۲ روی عناصر متعلق به $\{۰, ۱\}$ مشابه ضرب معمولی است در برخی مواقع از نوشتن آن خودداری می‌کنیم.) f را به صورت زیر فاکتورگیری می‌کنیم:

$$f(x_۱, \ldots, x_۷) = \mathbb{I}_{\{x_۱ \oplus x_۲ \oplus x_۴ = ۰\}} \mathbb{I}_{\{x_۳ \oplus x_۴ \oplus x_۶ = ۰\}} \mathbb{I}_{\{x_۴ \oplus x_۵ \oplus x_۷ = ۰\}}$$

که در آن $\mathbb{I}_{\{.\}}$ تابع نشانگر [۸] است. گاهی اوقات تابع f را تابع عضویت کد می‌نامند. گراف فاکتور متناظر با f که همان گراف تنر است، در شکل ۲.۳ نشان داده شده است.

$\diamondsuit$

[۸] Indicator function



شکل ۲.۳: گراف فاکتور تابع عضویت مثال ۴.۲.۳ (گراف تنر) این گراف نیز درخت است .

۲.۲.۳ تعیین فرم بازگشتی توابع حاشیه‌ای

تابع عمومی g که گراف فاکتور متناظر با (فاکتورگیری) آن درخت است را در نظر بگیرید. فرض کنید حاشیه‌ای g نسبت به متغیر z مورد نظر باشد، یعنی:

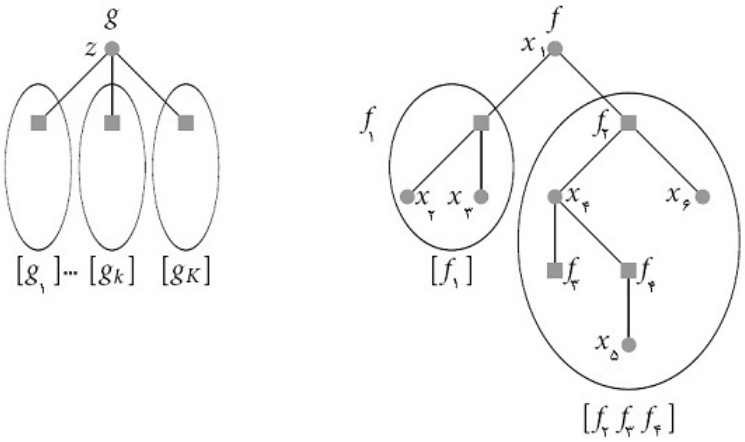
$$g(z) = \sum_{\sim z} g(z, \ldots)$$

با توجه به اینکه گراف فاکتور متناظر با g ، دوبخشی است، تابع g فاکتورگیری کلی به صورت زیر خواهد داشت:

$$g(z, \ldots) = \prod_{k=۱}^{K} g_k(z, \ldots)$$

($k \in \mathbb{Z}$) با این خاصیت که: متغیر z در هر فاکتور g_k ظاهر می‌شود ولی سایر متغیرها فقط در یک فاکتور ظاهر می‌شوند. زیرا اگر متغیری مانند z_i (به غیر از z) در دو فاکتور وجود داشته باشد، آنگاه غیر از مسیری که از طریق z این دو فاکتور را به یکدیگر متصل می‌کند، مسیر دیگری وجود خواهد داشت که توسط متغیر z_i این دو فاکتور را به یکدیگر متصل می کند. در این صورت با درخت بودن گراف فاکتور مورد نظر در تناقض است.

مثال ۵.۲.۳. [۱۸] در مثال ۲.۲.۳ گراف فاکتور متناظر با f ، درخت است بنابراین می‌توان فاکتورگیری آن را به صورت زیر نوشت:



شکل ۳.۳: فاکتور گیری کلی g و مثال ۵.۲.۳

$$f(x_1,\ldots) = [f_1(x_1,x_2,x_3)][f_2(x_1,x_4,x_6)f_3(x_4)f_4(x_4,x_5)]$$

همانطور که می‌بینید $k = 2$ است. این مثال را هم زمان با شرح مطالب مورد نظر ادامه می‌دهیم (شکل ۳.۳ را ببینید).

$\diamondsuit$

حاشیه‌ای g نسبت به z را می‌توان با توجه به قانون توزیع‌پذیری به صورت زیر نوشت:

$$\begin{aligned}\sum_{\sim z} g(z,\ldots) &= \sum_{\sim z} \prod_{k=1}^{K}[g_k(z,\ldots)] \\ &= \prod_{k=1}^{K} [\sum_{\sim z}[g_k(z,\ldots)]\end{aligned} \qquad (۴.۳)$$

(سطر آخر رابطه (۴.۳) حاصلضرب توابع حاشیه‌ای است) به این ترتیب تابع حاشیه‌ای g نسبت به z ($\sum_{\sim z} g(z,\ldots)$) به صورت حاصل ضرب توابع حاشیه‌ای انفرادی نوشته می‌شود.

مثال ۶.۲.۳. [۱۸] (ادامه مثال ۵.۲.۳)

$$f(x_1) = [\sum_{\sim x_1} f_1(x_1, x_2, x_3)][\sum_{\sim x_1} f_2(x_1, x_4, x_6) f_3(x_4) f_4(x_4, x_5)] \ .$$

$\diamondsuit$

استفاده از قانون توزیع‌پذیری به طور کلی موجب کاهش غیر قابل اغماض پیچیدگی می‌شود. می‌توان این تفکر بازگشتی را برای هر $g_k(z, \ldots)$ (که به طور کلی حاصل ضرب تعدادی فاکتور است) ادامه داد. چون گراف فاکتور متناظر با g درخت است، باید فاکتورگیری کلی به صورت زیر داشته باشد:

$$g_k(z, \ldots) = \underbrace{h(z, z_1, \ldots, z_J)}_{\text{هسته}} \underbrace{\prod_{j=1}^{J} [h_j(z_j, \ldots)]}_{\text{فاکتورها}}$$

که در آن z فقط در هسته $h(z, z_1, \ldots, z_J)$ و هر z_j $(j = 1, \ldots, J)$ حداکثر دوبار ظاهر می‌شود، یعنی هر z_j ممکن است، در هسته و حداکثر در یکی از فاکتورهای $h_j(z_j, \ldots)$ ظاهر شود. علاوه‌براین همه متغیرهای دیگر (به غیر از z و z_j ، $j = 1, \ldots, J$) به طور منحصربه‌فردی در یک فاکتور قرار خواهند داشت.

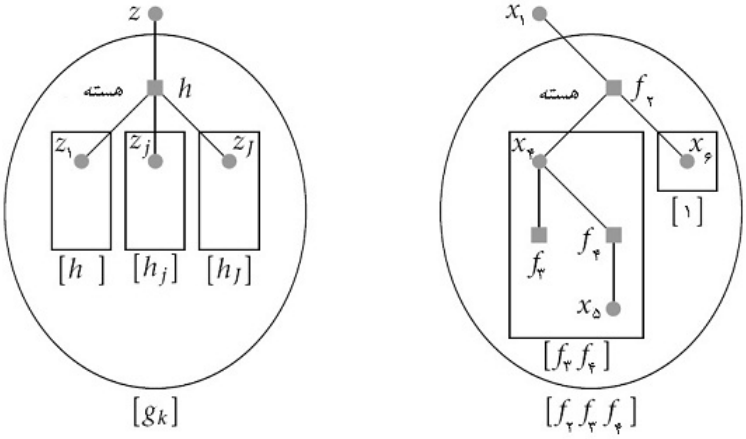
مثال ۳.۲.۷. [۱۸] (ادامه مثال ۳.۲.۵) آنچه در بالا گفته شد را می‌توان برای مثال ۳.۲.۵ به صورت زیر نوشت:

$$f_2(x_1, x_4, x_6) f_3(x_4) f_4(x_4, x_5) = \underbrace{f_2(x_1, x_4, x_6)}_{\text{هسته}} \underbrace{[f_3(x_4) f_4(x_4, x_5)]}_{\text{فاکتور وابسته به } x_4}$$

(x_5 متغیری غیر از x_1, x_4 و x_6 است که فقط در یک فاکتور شرکت دارد).

$\diamondsuit$

فاکتورگیری کلی و مثال ۳.۲.۷ در شکل ۳.۴ نشان داده شده است. با استفاده مجدد از قانون توزیع پذیری



شکل ۴.۳: فاکتورگیری کلی g_k و مثال ۷.۲.۳

می‌توان نوشت:

$$\begin{aligned}\sum_{\sim z} g_k(z,\ldots) &= \sum_{\sim z} h(z,z_1,\ldots,z_J) \prod_{j=1}^{J}[h_j(z_j,\ldots)] \\ &= \sum_{\sim z}^{J} h(z,z_1\ldots,z_J) \underbrace{\prod_{j=1}^{J}[\sum_{\sim z_j} h_j(z_j,\ldots)]}_{\text{ضرب توابع حاشیه‌ای}}\end{aligned} \tag{۵.۳}$$

به این ترتیب هر فاکتور $h_j(z_j,\ldots)$ فرم کلی مشابه با تابع اصلی $g(z,\ldots)$ دارد که می‌تواند به قطعات کوچکتری شکسته شود. این فرآیند تا هنگامی که به برگ‌های درخت برسیم ادامه میابد. بنابراین برای محاسبه تابع حاشیه‌ای مورد نظر باید در جهت عکس یعنی از برگ‌ها آغاز کنیم. در کل راس‌های گراف، توابع حاشیه‌ای را محاسبه می‌کنند که توابعی تحت مجموعه الفبای χ هستند .

در بخش بعدی حاشیه‌سازی ازطریق عبورپیام، شرح داده می‌شود. به طور کلی این روش به این صورت است که پیام‌ها توابع حاشیه‌ای هستند. قوانین پردازش پیام‌ها در راس‌های فاکتور در رابطه (۵.۳) بیان شده است و در راس‌های متغیر، ضرب مؤلفه در مؤلفه [۹] انجام می‌شود. به عبارت دیگر در راس‌های برگ می‌توان دو حالت را در نظر گرفت:

[۹] Pointwise multiplication

اگر راس برگ، راس فاکتور باشد، فرم کلی این راس به صورت

$$\sum_{\sim z} g(z) = g_k(z)$$

است. یعنی پیامی که راس‌های برگ فاکتور ارسال می‌کنند، خود فاکتور (تابع) است. اگر راس برگ متغیر داشته باشیم پیامی که این راس ارسال می‌کند را با مثالی شرح خواهیم داد.

مثال ۳.۸.۲. [۱۸] تابع $f(x_1, x_2)$ را در نظر بگیرید حاشیه‌ای آن نسبت به x_1

$$f(x_1) = \sum_{\sim x_1} f(x_1, x_2)$$

است. در اینجا x_2 راس برگ متغیر است. براساس قانون عبور پیام در رابطه (۵.۳) داریم:

$$f(x_1) = \sum_{\sim x_1} f(x_1, x_2) \cdot \mu(x_1)$$

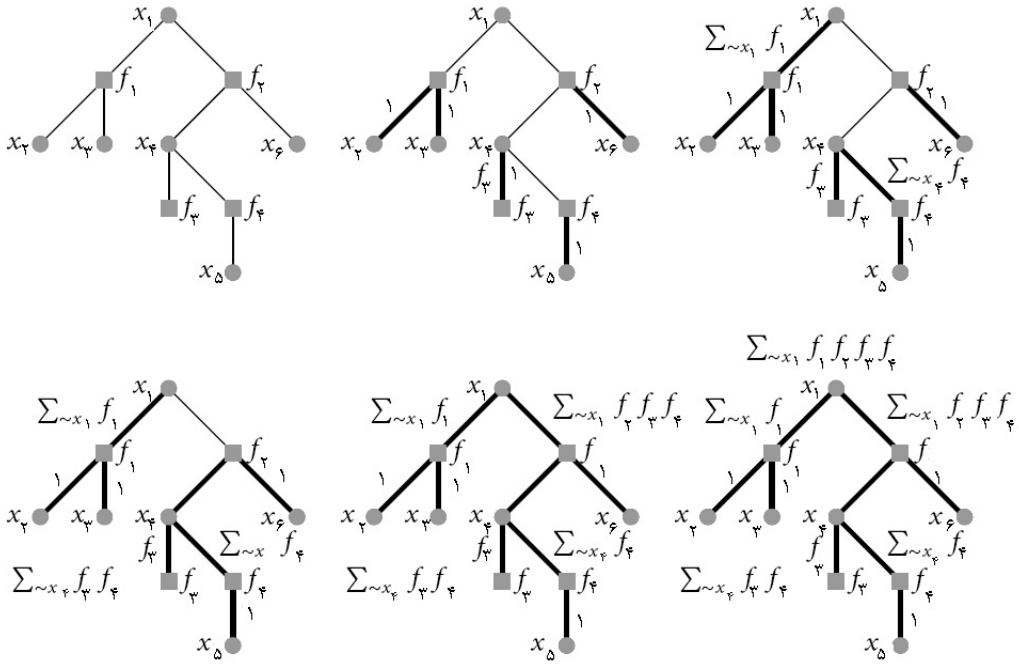
$\mu(x_1)$ پیام‌آغازی است که از راس برگ متغیر برای هسته ارسال شده است (همان طور که می‌بینید در صورتی نتیجه درست حاصل می‌شود که $\mu(x_1) = 1$) .

$\diamondsuit$

۳.۲.۳ حاشیه‌سازی از طریق عبور پیام

همانطور که مشاهده شد در حالتی که گراف فاکتور درخت است، مساله حاشیه‌سازی می‌تواند مطابق با ساختار درخت به اعمال کوچکتری شکسته شود. پیام‌هایی که در طول یال‌های گراف ارسال می‌شوند، توابعی تحت χ یا به طور معادل بردارهایی به طول $|\chi|$ هستند. ابتدا نحوه عمل الگوریتم عبور پیام را برای مثال ۵.۲.۳ شرح می‌دهیم.

مثال ۳.۹.۲. [۱۸] همانطور که در شکل ۵.۳ می‌بینید (سطر اول ، گراف وسط) عبورپیام از راسهای برگ آغاز می‌شود. راس‌های برگ متغیر x_2, x_3, x_5 و x_6 تابع ثابت ۱ (همان‌گونه که در انتهای زیر بخش قبل بیان

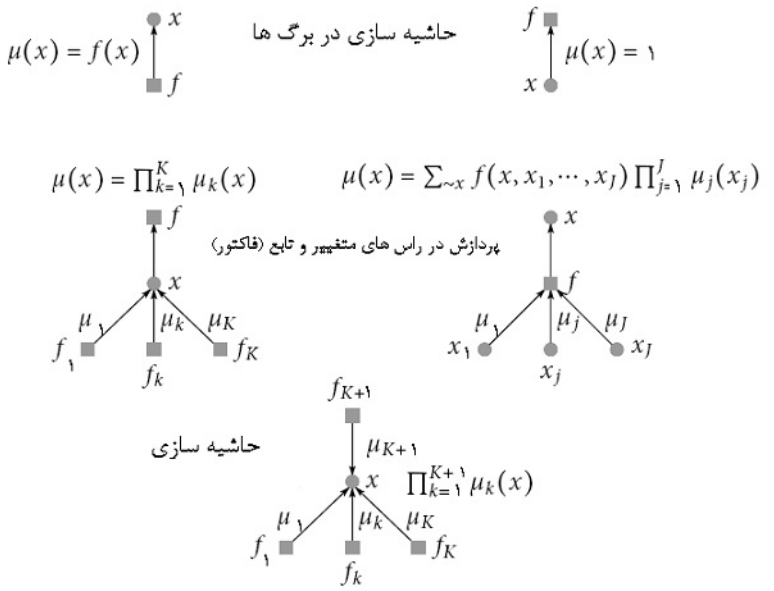


شکل ۵.۳: حاشیه‌سازی تابع f از طریق عبور پیام. عبور پیام از راس‌های برگ آغاز می‌شود. پیام‌هایی که هر راس دریافت می‌کند پس از پردازش به راس والد خود می‌فرستد .

شد) و راس برگ فاکتور f_3 ، تابع f_3 را به راس والد خود ارسال می‌کند. در مرحله بعد راس فاکتور f_1 پیام‌هایی که از دو فرزند خود دریافت کرده است را براساس رابطه (۵.۳) پردازش می‌کند و نتیجه را به راس والد خود ارسال می‌کند، یعنی $\sum_{\sim x_1} f_1(x_1, x_2, x_3)$. به طور مشابه راس فاکتور f_4 پیام $\sum_{\sim x_4} f_4(x_4, x_5)$ را به راس والد خود ارسال می‌کند. حال راس متغیر x_4 حاصل ضرب پیام‌هایی را که از فرزندان خود دریافت کرده است (مطابق رابطه (۴.۳)) یعنی $\sum_{\sim x_1} f_3(x_4) f_4(x_4, x_5)$ را به راس فاکتور f_2 ارسال می‌کند. f_2 نیز پیام $\sum_{\sim x_1} f_2(x_1, x_4, x_6) f_3(x_4) f_4(x_4, x_5)$ را به x_1 ارسال می‌کند. در پایان آنچه در ریشه x_1 محاسبه می‌شود، حاشیه‌ای مورد نظر است، یعنی $\sum_{\sim x_1} f_1(x_1, x_2, x_3) f_2(x_1, x_4, x_6) f_3(x_4) f_4(x_4, x_5)$ است.

$\diamondsuit$

با توجه به مثال قبل، عبور پیام از راس‌های برگ آغاز می‌شود. در شکل ۶.۳ خلاصه قوانین عبور پیام، بیان شده است.



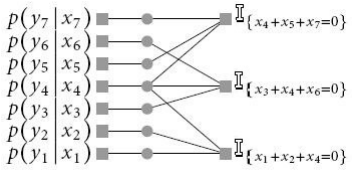
شکل ۶.۳: [۱۸] قوانین عبور پیام

۴.۲.۳ کدگشایی MAPبیتی

با توجه به قانون کدگشای ماکزیمم پسین بیتی MAP (که در بخش (۶.۲) بیان شد) داریم:

$$\begin{aligned}
\widehat{x}_i &= \underset{\substack{x_i \in \{۰,۱\}}}{\text{argmax}}\, p(x_i|y) \\
&\text{(قانون احتمال کل)} \quad = \underset{\substack{x_i \in \{۰,۱\}}}{\text{argmax}} \sum_{\sim x_i} p(x|y) \\
&\text{(قانون بیز)} \quad = \underset{\substack{x_i \in \{۰,۱\}}}{\text{argmax}} \sum_{\sim x_i} p(y|x)p(x) \\
&= \underset{\substack{x_i \in \{۰,۱\}}}{\text{argmax}} \sum_{\sim x_i} \Big(\prod_j p(y_j|x_j)\Big) \mathbb{I}_{\{x \in \mathcal{C}\}}
\end{aligned} \tag{۶.۳}$$

توجه کنید که سطر آخر (۶.۳) با فرض فاقد حافظه بودن کانال و هم احتمال بودن کدواژه‌ها است. فرض کنید تابع عضویت کد $\mathbb{I}_{\{x \in \mathcal{C}\}}$ را می توان به فرم فاکتورگیری نوشت. از سطر آخر (۶.۳) واضح است که مساله



شکل ۷.۳: گراف فاکتور مساله کدگشایی MAP مثال (۱۰.۲.۳)

کدگشایی بیتی معادل با محاسبه تابع حاشیه‌ای یک تابع (که فاکتورگیری روی آن انجام شده است) و سپس انتخاب ماکزیمم مقدار این تابع حاشیه‌ای است.

مثال ۱۰.۲.۳. [۱۸] ماتریس بررسی‌زوجیت مثال (۴.۲.۳) را در نظر بگیرید. در این مورد

$$\begin{array}{c}\text{argmax}\ \ p(x_i|y)\\ x_i \in \{0,1\}\end{array}$$

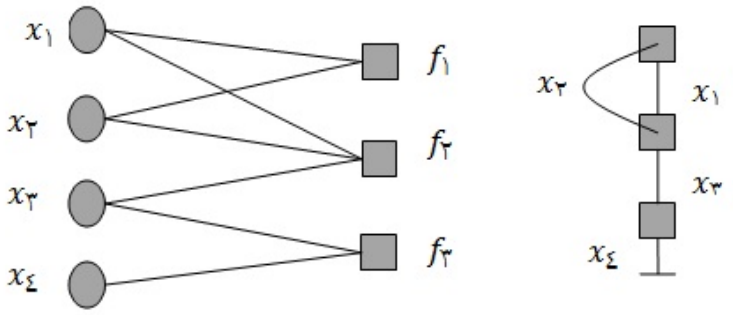
را می‌توان به صورت زیر فاکتور کرد :

$$\begin{array}{c}\text{argmax}\ \ \sum_{\sim x_i}(\prod_{j=1}^{7} p(y_j|x_j)\ )\mathbb{I}_{\{x_1\oplus x_2\oplus x_4\}}\mathbb{I}_{\{x_3\oplus x_4\oplus x_6\}}\mathbb{I}_{\{x_4\oplus x_5\oplus x_7\}}\\ x_i \in \{0,1\}\end{array}$$

گراف فاکتور متناظر با این فاکتورگیری در شکل ۷.۳ نمایش داده شده است. در این گراف اثر کانال نیز نمایش داده شده است.

$\diamondsuit$

پیام $\mu(x)$ در حالت دودویی را می‌توان، برداری به طول ۲ یعنی $(\mu(0),\mu(1))$ در نظر گرفت. این پیام از راس برگ فاکتور ارسال می‌شود. در حقیقت این پیام می‌تواند $(p(y_j|0),p(y_j|1))$ باشد که کانال به راس متغیر j ارسال می‌کند.



شکل ۸.۳: [۱۸]شکل سمت راست گراف فاکتور سبک فورنی ، شکل سمت چپ گراف فاکتور

۵.۲.۳ گراف‌های فاکتور سبک فورنی

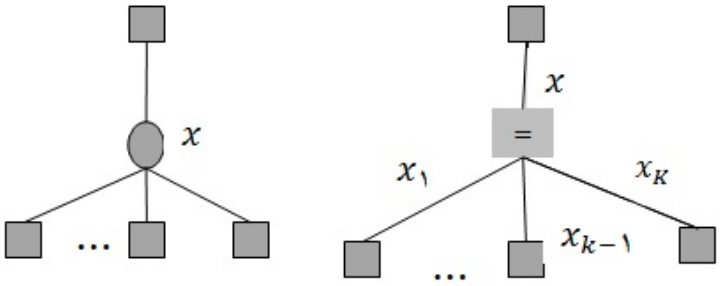
همان‌طور که بیان شد گراف‌های فاکتور (FG) روشی برای نمایش رابطه بین توابع و مولفه‌های محلی (توابع حاشیه‌ای) آنها هستند. گراف‌های **فاکتور سبک فورنی** [۱۰] که گاهی اوقات به آنها گراف‌های نرمال[۱۱] نیز می‌گویند نمایشی دیگر برای بیان این رابطه‌اند.

اگر گراف دوبخشی FG را به گرافی با این قاعده تبدیل کنیم که در آن، یال‌ها (نیم‌یال‌ها)، همان راس‌های متغیر متعلق به گراف فاکتور باشند، گراف حاصل را گراف فاکتور سبک فورنی (FSFG) می‌گویند.

همانطور که در شکل ۸.۳ می‌بینید، شکل سمت چپ گراف FG و شکل سمت راست، گراف FSFG است. راس‌های متغیر گراف FSFG در شکل ۸.۳ از درجه ۱ یا ۲ هستند. در حالتی که درجه راس‌های متغیر بیش از ۲ باشد، یال متناظر با این متغیر را در گراف FSFG به تعداد کافی تکرار می‌کنیم و این یال‌ها را توسط فاکتور تساوی به یکدیگر متصل می‌کنیم (شکل ۹.۳ را ببینید).

در شکل ۹.۳ فرض شده است که درجه راس متغیر x ، $k+1$ است. همان‌طور که می‌بینید در گراف FSFG معادل با گراف FG مورد نظر، در شکل، k متغیر $x_k, \ldots, x_2, x_1$ توسط فاکتور تساوی با متغیر x مساوی شده‌اند،

[۱۰] Fornay - style factor graph
[۱۱] Normal graphs



شکل ۳.۹: [۱۸]شکل سمت راست گراف فاکتور سبک فورنی ، شکل سمت چپ گراف فاکتور

یعنی $x = x_1 = \ldots = x_2 = x_1$.

باید توجه کرد که در گراف FSFG فقط به قانون پردازش راس متغیر، نیاز داریم که مشابه با قانون پردازش راس متغیر در گراف FG است. به عبارت دیگر در گراف FSFG برای راسی از درجه $J+1$ پیام خروجی در امتداد یال x عبارت است از:

$$\sum_{\sim x} f(x, x_1, \ldots, x_J) \prod_{j=1}^{J} \mu_j(x_j) \qquad (۷.۳)$$

به این ترتیب چون گراف‌های FSFG راس‌های کمتری نیاز دارند، ساده‌تر هستند. متغیرهای داخلی را متغیرهای حالت [۱۲] می‌نامند و نیم‌یال‌ها (مانند آخرین یال شکل ۷.۳ سمت راست) را متغیرهای خارجی[۱۳] می‌نامند.

۳.۳ کدهای بررسی‌زوجیت کم‌چگال (LDPC)

کدهای LDPC اولین بار در سال ۱۹۶۲ توسط **گالاگر**[۱۴] [۱۴]،[۱۵] پیشنهاد شد. اما این کدها حدود ۳۰ سال چندان مورد توجه نبودند . دلیل این عدم توجه، نبودن تکنولوژی رشدیافته‌ای بود که قادر باشد به روشی کارا، این کدها را پیاده‌سازی کند. موفقیت‌هایی که در زمینه کدگشایی تکراری کدهای توربو بدست آمد، موجب مطرح شدن دوباره کدهای LDPC در دهه ۱۹۹۰ توسط **مک کی**[۱۵] و**نیل**[۱۶] شد.

[۱۲] State variable
[۱۳] External variable
[۱۴] Gallager
[۱۵] MacKay
[۱۶] Neal

یکی از ویژگی‌های مهم کدهای LDPC این است که اجازه می‌دهند، داده‌ها با نرخی نزدیک به حد شانون ارسال شوند. به عنوان مثال طرحی از کدهای LDPC ارئه شده است که در حدود dB ۰/۰۰۴۵ حد شانون می‌رسد [۸]. امروزه استفاده از کدهای LDPC در تصحیح خطای اطلاعات، برای مواردی از قبیل تلفن همراه، **ارسال ماهواره‌ای تلویزیون دیجیتال** [۱۷] و . . . متداول شده‌اند.

به عنوان مثالی خاص از کاربرد کدهای LDPC می توان به **نسل دوم استاندارد رادیو تلویزیون دیجیتال** [۱۸] (DVB-S2) اشاره کرد [۹] .

کدهای LDPC، کدهایی خطی بلوکی هستند که ماتریس بررسی‌زوجیت آنها (همان طور که از نام آنها می‌توان فهمید) تعداد بسیار کمی عنصر غیر صفر دارد. در حقیقت کدهای LDPC، حداقل یک ماتریس بررسی‌زوجیت تنک H دارند. تنک بودن H ضمانتی است برای اینکه پیچیدگی کدگشایی و همچنین مینیمم فاصله با افزایش n (طول هر کدواژه یا به عبارت دیگر طول بلوکی) به طور خطی نسبت به n افزایش بیابد.

۱.۳.۳ انواع کدهای LDPC

کدهای LDPC در ابتدا توسط ساخت ماتریس بررسی‌زوجیت تنک طراحی می‌شوند، پس از آن، ماتریس مولد کد تعیین می شود. بزرگترین تفاوت بین کدهای LDPC و کدهای بلوکی کلاسیک، نحوه کدگشایی آنها است. روش های کدگشایی مورد استفاده کدهای بلوکی کلاسیک به طور معمول الگوریتم‌هایی مشابه **ML** (ماکزیمم درستنمایی) هستند. بنابراین این الگوریتم‌ها اغلب کوتاه می‌باشند. کدهای LDPC به طور تکراری با استفاده از گراف تنر متناظر با ماتریس بررسی‌زوجیت H کد مورد نظر کدگشایی می‌شوند، به این ترتیب کدگشایی آنها با تمرکز روی خواص H ، طراحی شده‌اند. (در این پایان‌نامه فقط کدهای LDPC دودویی مورد بحث قرار می‌گیرد) .

می توان دو دسته کلی کدهای LDPC باقاعده و بی‌قاعده را به صورت زیر تعریف کرد:

۱. **کدهای LDPC باقاعده** [۱۹] :

[۱۷] Satelling transmission of digital television
[۱۸] Secend generation digital video broadcating
[۱۹] regular LDPC codes

ماتریس بررسی‌زوجیت کدهای LDPC باقاعده، ماتریس **(ω_c, ω_r) - باقاعده** نامیده می‌شود، اگر ω_c عنصر ۱ ، در هر ستون و ω_r عنصر ۱ ، در هر سطر ماتریس بررسی‌زوجیت موردنظر وجود داشته باشد. به این ترتیب در گراف تنر متناظر با ماتریس بررسی‌زوجیت (ω_c, ω_r) - باقاعده H ، درجه هر راس متغیر ω_c و درجه هر راس بررسی ω_r می باشد. تعداد یال‌های گراف تنر (ω_c, ω_r) - کد LDPC با قاعده $n\omega_c$ است (n طول کدواژه‌ها) لذا با افزایش n ، تعداد یال‌های گراف به طور خطی افزایش می‌یابد.

در کدهای LDPC دو نوع نرخ متفاوت وجود دارد:

(الف) **نرخ واقعی** [۲۰] (همان نرخ کد است):

$$R = \frac{k}{n} \qquad (۸.۳)$$

(ب) **نرخ طرح** [۲۱]

$$R_d = ۱ - \frac{m}{n} \qquad (۹.۳)$$

($H \in F_۲^{m \times n}$ و k طول بردار اطلاعات اصلی، قبل از اضافه شدن افزونگی ($(n-k) \leq m$) است).

نرخ طرح در کدهای LDPC باقاعده:

$$R_d = ۱ - \frac{\omega_c}{\omega_r} \qquad (۱۰.۳)$$

[۲۰] true rate
[۲۱] design rate

بنابراین برای کدهایLDPC باقاعده می‌توان نوشت:

$$R_d \leq R \qquad (۱۱.۳)$$

زیرا

$$n\omega_c = m\omega_r \Rightarrow \frac{m}{n} = \frac{\omega_c}{\omega_r}$$

با توجه به اینکه برخی معادلات بررسی‌زوجیت، ممکن است تکراری باشند (سطرهای H در حالت کلی مستقل‌خطی نیستند)، لذا $(n-k) \leq m$ ، از طرفی داریم:

$$\begin{aligned} R &= \frac{k}{n} &&= \frac{n-(n-k)}{n} \\ &= ۱ - \frac{n-k}{n} &&\geq ۱ - \frac{m}{n} \\ & &&\geq ۱ - \frac{\omega_c}{\omega_r} \end{aligned}$$

مثال ۱.۳.۳. [۱۷] ماتریس بررسی‌زوجیت باقاعده زیر را در نظر بگیرید:

$$H = \begin{pmatrix} ۱ & ۱ & ۰ & ۱ & ۰ & ۰ \\ ۰ & ۱ & ۱ & ۰ & ۱ & ۰ \\ ۱ & ۰ & ۰ & ۰ & ۱ & ۱ \\ ۰ & ۰ & ۱ & ۱ & ۰ & ۱ \end{pmatrix}$$

می‌بینیم که رتبه سطری H مساوی با ۳ و ۲ $= \omega_c$ ، ۳ $= \omega_r$ و $R = \frac{۱}{۲}, R_d = \frac{۱}{۳}$ است. ♢

۲. کد LDPC بی قاعده[۲۲]:

در ماتریس بررسی‌زوجیت بی‌قاعده، تعداد ۱ها در هر سطر و همچنین تعداد ۱ها در هر ستون می‌تواند متفاوت باشد. در ادامه به بیان مفاهیم مورد نیاز برای کدهای LDPC بی‌قاعده (در حقیقت مفاهیم کلی) می‌پردازیم.

[۲۲] Irrequlars LDPC codes

تعریف ۲.۳.۳. [۱۸] فرض کنید طول بلوکی کد LDPC، n باشد، اگر تعداد راس‌های متغیر درجه i را با Λ_i نمایش دهیم، آنگاه $\sum_i \Lambda_i = n$ ، به طور مشابه اگر تعداد راس‌های بررسی درجه i را با P_i نمایش دهیم، آنگاه $\sum_i P_i = n(۱ - R_d)$ (R_d نرخ طرح است). علاوه براین چون تعداد یال‌ها در هر دو طرف (طرف راس‌های متغیر و طرف راس‌های بررسی) برابر است، بنابراین:

$$\sum_i i\Lambda_i = \sum_i iP_i$$

به منظور راحتر شدن کار می‌توان این مفاهیم را به صورت زیر نوشت:

$$P(x) = \sum_{i=۱}^{\omega_{r\ max}} P_i x^i \ , \ \Lambda(x) = \sum_{i=۱}^{\omega_{c\ max}} \Lambda_i x^i \qquad (۱۲.۳)$$

که $\Lambda(x)$ و $P(X)$ چندجمله‌ای‌هایی با بسط نامنفی حول صفر هستند و ضرایب صحیح Λ_i و P_i همان تعداد راس‌های درجه i می‌باشند($\omega_{r\ max}$ ماکزیمم تعداد ۱، در سطرهای ماتریس H و $\omega_{c\ max}$ ماکزیمم تعداد ۱، در ستون‌های ماتریس H است). براساس این تعریف داریم:

$$\Lambda(۱) = n \ , \ P(۱) = n(۱ - R_d) \ , \ \Lambda'(۱) = p'(۱) \qquad (۱۳.۳)$$

و بنابراین می‌توان نوشت:

$$R_d(\Lambda, P) = ۱ - \frac{P(۱)}{\Lambda(۱)} \qquad (۱۴.۳)$$

Λ و P را به ترتیب **توزیع‌های درجه** [۲۳] متغیر و بررسی از **دیدگاه راس** می‌نامند.

گاهی اوقات استفاده از توزیع‌های درجه نرمال شده، به جای Λ و P مناسب‌تر است به همین منظور داریم:

$$L(x) = \frac{\Lambda(x)}{\Lambda(۱)} \ , \ R(x) = \frac{P(x)}{p(۱)} \qquad (۱۵.۳)$$

[۲۳] Degree distributions

مثال ۳.۳.۳. [۱۸] برای $[۷, ۴, ۳]$ - کدهامینگ داریم:

$$\Lambda(x) = ۳x + ۳x^۲ + x^۳ \qquad P(x) = ۳x^۴$$
$$L(x) = \frac{۳}{۷}x + \frac{۳}{۷}x^۲ + \frac{۱}{۷}x^۳ \qquad R(x) = x^۴$$

$\diamondsuit$

در حقیقت رفتار کدهای LDPC با استفاده از تفاوت درجه راس‌ها، به طور قابل توجهی بهبود می‌یابد، بنابراین عملکرد کدهای LDPC بی‌قاعده می‌تواند بهتر باشد.

۲.۳.۳ ساخت کدهای LDPC

نسخه اصلی کدهای LDPC که توسط گالاگر ارائه شده است، کد LDPC باقاعده است. سطرهای ماتریس بررسی‌زوجیت گالاگر به ω_c مجموعه تقسیم می‌شود که هر مجموعه شامل $\frac{m}{\omega_c}$ سطر (m تعداد سطرهای H) است. در مجموعه اول از سطرهای H ، ω_r عنصر ۱ موجود در هر سطر به طور متوالی از چپ به راست قرار گرفته‌اند به طوری که ω_r عنصر ۱ موجود در سطر بعدی، در ادامه سطر قبل قرار گیرد. هر یک از مجموعه سطرها دیگر توسط جایگشت‌های تصادفی ستون‌های مجموعه اول حاصل می‌شوند. در نتیجه هر یک از ستون‌های H فقط یک عنصر ۱ در هر یک از ω_c مجموعه دارد.

مثال ۴.۳.۳. [۱۷] ماتریس بررسی‌زوجیت گالاگر (۳،۴)- منظم به طول بلوکی ۱۲ به صورت زیر است:

$$H = \left(\begin{array}{cccccccccccc}
۱ & ۱ & ۱ & ۱ & ۰ & ۰ & ۰ & ۰ & ۰ & ۰ & ۰ & ۰ \\
۰ & ۰ & ۰ & ۰ & ۱ & ۱ & ۱ & ۱ & ۰ & ۰ & ۰ & ۰ \\
۰ & ۰ & ۰ & ۰ & ۰ & ۰ & ۰ & ۰ & ۱ & ۱ & ۱ & ۱ \\
\hline
۱ & ۰ & ۱ & ۰ & ۰ & ۱ & ۰ & ۰ & ۰ & ۱ & ۰ & ۰ \\
۰ & ۱ & ۰ & ۰ & ۰ & ۰ & ۱ & ۱ & ۰ & ۰ & ۰ & ۱ \\
۰ & ۰ & ۰ & ۱ & ۱ & ۰ & ۰ & ۰ & ۱ & ۰ & ۱ & ۰ \\
\hline
۱ & ۰ & ۰ & ۱ & ۰ & ۰ & ۱ & ۰ & ۰ & ۱ & ۰ & ۰ \\
۰ & ۱ & ۰ & ۰ & ۰ & ۱ & ۰ & ۱ & ۰ & ۰ & ۱ & ۰ \\
۰ & ۰ & ۱ & ۰ & ۱ & ۰ & ۰ & ۰ & ۱ & ۰ & ۰ & ۱
\end{array}\right)$$

$\diamondsuit$

روش متداول دیگر برای ساخت کدهای LDPC، روشی است که توسط مک‌کی و نیل ارائه شده است. در این روش ستون‌های H ازچپ به راست، به صورت یکی یکی به ماتریس اضافه می‌شوند. تعداد ۱ ها در هر ستون با توجه به توزیع درجه متغیر تعیین می‌شود و موقعیت عناصر غیرصفر (۱ ها) در هر سطر به صورت تصادفی و با توجه به این که چه سطرهایی هنوز (با توجه به توزیع درجه بررسی) پرنشده‌اند، انتخاب می‌شوند. اگر سطرهایی وجود داشته باشند که هنوز پرنشده‌اند ، آنگاه ستون‌های بعدی اضافه می‌شوند.

مثال ۳.۳.۵. [۱۷] ماتریس بررسی‌زوجیت مک‌کی- نیل (۳،۴)- منظم به طول بلوکی ۱۲:

$$H = \begin{pmatrix} ۱ & ۰ & ۰ & ۰ & ۰ & ۱ & ۰ & ۱ & ۰ & ۱ & ۰ & ۰ \\ ۱ & ۰ & ۰ & ۱ & ۱ & ۰ & ۰ & ۰ & ۰ & ۰ & ۱ & ۰ \\ ۰ & ۱ & ۰ & ۰ & ۱ & ۰ & ۱ & ۰ & ۱ & ۰ & ۰ & ۰ \\ ۰ & ۰ & ۱ & ۰ & ۰ & ۱ & ۰ & ۰ & ۰ & ۰ & ۱ & ۱ \\ ۰ & ۰ & ۱ & ۰ & ۰ & ۰ & ۱ & ۱ & ۰ & ۰ & ۰ & ۱ \\ ۰ & ۱ & ۰ & ۰ & ۱ & ۰ & ۰ & ۰ & ۱ & ۰ & ۱ & ۰ \\ ۱ & ۰ & ۰ & ۱ & ۰ & ۰ & ۱ & ۰ & ۰ & ۱ & ۰ & ۰ \\ ۰ & ۱ & ۰ & ۰ & ۰ & ۱ & ۰ & ۱ & ۰ & ۱ & ۰ & ۰ \\ ۰ & ۰ & ۱ & ۱ & ۰ & ۰ & ۰ & ۰ & ۱ & ۰ & ۰ & ۱ \end{pmatrix}$$

هنگام اضافه کردن ستون ۱۱ ام می‌بینیم که سطرهای ۲ ، ۴ ، ۵ ، ۶ و ۹ پرنشده‌اند، از بین آنها سطرهای ۲ ، ۴ و ۶ را اضافه می‌کنیم.

$\diamondsuit$

تعریف ۳.۳.۶. [۱۸] (**دسته استاندارد** $LDPC(\Lambda, P)$) برای جفت توزیع درجه (Λ, P) **دسته**[۲۴] گراف‌های دوبخشی به صورت زیر تعریف می‌شود:

هر گراف $LDPC(\Lambda, P)$ ، $\Lambda(۱)$ راس متغیر و $p(۱)$ راس بررسی دارد (به عبارت دیگر Λ_i راس متغیر و P_i راس بررسی از درجه i دارد). می‌توان فرض کرد که هر راس درجه i ، i سوکت دارد که i یال از آن خارج می‌شود، به طوری که در هر طرف گراف (دوبخشی) $\Lambda'(۱) = P'(۱)$ سوکت وجود خواهد داشت. سوکت‌های هر طرف

[۲۴] ensemble

را با استفاده از مجموعه $\{۱, ۲, \cdots, \Lambda'(۱)\} = [\Lambda'(۱)]$ به طور تصادفی و ثابت بر چسب می‌زنیم. فرض کنید σ جایگشتی روی $[\Lambda'(۱)]$ باشد، با اتصال سوکت i ام طرف راس‌های متغیر، توسط یک یال به سوکت $\sigma(i)$ طرف راس‌های بررسی، گراف دوبخشی به σ وابسته می‌شود. بنابراین متناظر با هر σ تعریف شده روی $[\Lambda'(۱)]$ گرافی دوبخشی ساخته می‌شود. به این ترتیب مجموعه‌ای از گراف‌های دوبخشی به وجود می‌آیند، که روی این مجموعه یک توزیع احتمال، با استفاده از قرار دادن توزیع احتمال یکنواخت روی مجموعه جایگشت‌ها، تعریف می‌شود. این مجموعه همان دسته گراف‌های دوبخشی LDPC است. حال با استفاده از ماتریس بررسی‌زوجیت، کدی را به هر عنصر $LDPC(\Lambda, P)$ ، مربوط می‌کنیم. چون امکان رخ دادن یال‌های چندگانه وجود دارد، از طرفی کدگذاری تحت میدان $F_۲$ انجام می‌شود، در ماتریس بررسی‌زوجیت H عنصر سطر i ام ستون j ام غیر صفر است اگر تعداد یال‌هایی که راس بررسی i ام را به راس متغیر j ام متصل می‌کند فرد باشد، در غیر این صورت صفر خواهد بود. به این ترتیب چون به هر گراف، کدی مرتبط می‌شود، کدها را به عنوان عناصر $LDPC(\Lambda, p)$ (چنانچه بتوان این دو را به جای یکدیگر بکار برد) بیان کرد. در حقیقت این گراف‌ها (یا به طور دقیق تر، سوکت‌ها) هستند که برچسب خورده‌اند و توزیع احتمالی یکنواخت دارند. پس کدهای حاصل از آنها برچسب‌دار نیستند، به این ترتیب توزیع احتمال روی مجموعه کدها، لزوما یکنواخت نیست. در نتیجه اگر می‌گوییم که یک کد را به طور یکنواخت تصادفی انتخاب می‌کنیم، منظور واقعی ما انتخاب گرافی تصادفی از دسته گراف‌ها است و کد مربوط به آن را در نظر می‌گیریم.

در بخش ۱.۳.۳ مفاهیم و روابطی برای کدهای LDPC از دیدگاه درجه راس‌ها مطرح شد. به منظور تحلیل مناسب‌تر کدهای LDPC (که در فصل بعد خواهیم دید) همان مفاهیم از دیدگاه یالی مطرح می‌شود.

تعریف ۷.۳.۳. [۱۸] فرض کنید λ و ρ چندجمله‌ای‌هایی با بسط نامنفی حول صفر باشند، تعریف می‌شود:

$$\begin{aligned} \lambda(x) &= \sum_i \lambda_i x^{i-۱} = \frac{\Lambda'(x)}{\Lambda'(۱)} = \frac{L'(x)}{L'(۱)} \\ \rho(x) &= \sum_i \rho_i x^{i-۱} = \frac{P'(x)}{P'(۱)} = \frac{R'(x)}{R'(۱)} \end{aligned} \tag{۱۶.۳}$$

که در آن λ_i (ρ_i) کسری از یال‌ها است که به راس‌های متغیر (راس‌های بررسی) درجه i متصل شده‌اند. در حقیقت λ_i (ρ_i) احتمال انتخاب یکنواخت تصادفی یالی است که به راس متغیر (بررسی) درجه i در گراف مورد نظر متصل شده است. زیرا:

$$\begin{aligned} \Lambda(x) &= \sum_{i}^{\omega_{c\ max}} \Lambda_i x^i \\ \Lambda'(x) &= \sum_{i}^{\omega_{c\ max}} i\Lambda_i x^{i-1} \\ \Rightarrow \forall i \quad \lambda_i &= \frac{i\Lambda_i}{\Lambda'(1)} = \frac{i\Lambda_i}{\sum_{i}^{\omega_{c\ max}} i\Lambda_i} \end{aligned} \qquad (۱۷.۳)$$

که در سطر آخر $i\Lambda_i$ ، تعداد کل یال‌های متصل به راس‌های درجه i است. λ و ρ ، را به ترتیب **توزیع‌های درجه متغیر و بررسی از دیدگاه یالی** می‌نامند. عکس روابط بالا به صورت زیر است:

$$\begin{aligned} \frac{\Lambda(x)}{n} &= L(x) = \frac{\int_0^x \lambda(z)dz}{\int_0^1 \lambda(z)dz} \\ \frac{P(x)}{n(1-r)} &= R(x) = \frac{\int_0^x \rho(z)dz}{\int_0^1 \rho(z)dz} \end{aligned} \qquad (۱۸.۳)$$

متوسط درجه‌های متغیر و بررسی را می‌توان به صورت زیر بیان کرد:

$$\begin{aligned} \omega_{c\ avg} &= L'(1) = \frac{1}{\int_0^1 \lambda(x)dx} \\ \omega_{r\ avg} &= R'(1) = \frac{1}{\int_0^1 \rho(x)dx} \end{aligned} \qquad (۱۹.۳)$$

به این ترتیب نرخ طرح عبارت است از:

$$\begin{aligned} r(\lambda,\rho) &= 1 - \frac{\omega_{c\ avg}}{\omega_{r\ avg}} = 1 - \frac{L'}{R'} \\ &= 1 - \frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx} \end{aligned} \qquad (۲۰.۳)$$

(توجه: $r(\lambda,\rho)$ همان R_d است، که با توجه به نحوه نام گذاری توزیع‌ها متغیر و بررسی، در اینجا به این صورت ذکر شده است)

در مورد نرخ طرح باید به این نکته توجه کنیم که: با فرض اینکه معادلات بررسی‌زوجیت (سطرهای H) مستقل‌خطی باشند آنگاه نرخ طرح مساوی با نرخ کد است زیرا:

برای کد $\mathcal{C}$ با ماتریس بررسی‌زوجیت $H \in F_۲^{m\times n}$ که در آن $m = n - k$ (سطرهای H مستقل‌خطی‌اند) داریم:

$$\begin{aligned} r(\lambda,\rho) &= ۱ - \frac{m}{n} \\ &= ۱ - \frac{n-k}{n} \\ &= \frac{k}{n} \\ &= R \end{aligned} \qquad (۲۱.۳)$$

مثال ۳.۳.۸. [۱۸] جفت (Λ, P) را به صورت زیر در نظر بگیرید:

$$\Lambda(x) = ۶۱۳x^۲ + ۲۰۲x^۳ + ۵۷x^۴ + ۸۴x^۷ + ۴۴x^۸ \quad , \; P(x) = ۵۰۰x^۶$$

با

$$\Lambda(۱) = ۱۰۰۰ \; , \; P(۱) = ۵۰۰ \; , \; \Lambda' = P' = ۳۰۰۰$$

$$\lambda(x) = \frac{۱۲۲۷}{۳۰۰۰}x + \frac{۶۰۶}{۳۰۰۰}x^۲ + \frac{۲۲۸}{۳۰۰۰}x^۳ + \frac{۵۵۸}{۳۰۰۰}x^۶ + \frac{۳۵۲}{۳۰۰۰}x^۷ \quad , \quad \rho(x) = x^۵$$

$\diamondsuit$

با توجه به اینکه (Λ, P) ، (n, L, R) و (n, λ, ρ) اطلاعات مشابهی را در بر می‌گیرند، می توان با توجه به دیدگاه مورد نظر از هریک از آنها استفاده کرد.

قرارداد [۱۸]: اگر بخواهند توزیع‌های درجه را در فرم‌های مختلف که معادل هستند نشان دهند، به صورت زیر می‌نویسند:

$$(\Lambda, P) \triangleq (n, L, R) \triangleq (n, \lambda, \rho)$$

اغلب دسته مورد نظر را به صورت $LDPC(n, \lambda, \rho)$ نشان می‌دهند.

۴.۳ کدگذاری

پیش از این بیان شد که ابتدا ماتریس بررسی‌زوجیت H کد LDPC ساخته می‌شود سپس با استفاده از آن ماتریس مولد کد تعیین می‌شود. برای این کار می توان روش حذفی گوس-جردن را روی H اجرا کرد و آنرا به صورت زیر تبدیل نمود:

$$H = [A \quad I_{n-k}]$$

که در آن A ، ماتریسی $(n-k) \times k$ و I_{n-k} ماتریس همانی $(n-k) \times (n-k)$ است.

به این ترتیب ماتریس مولد عبارت است از:

$$G = \begin{bmatrix} I_k & A^T \end{bmatrix}$$

در ادامه با ذکر مثالی به شرح این روش می‌پردازیم.

مثال ۱.۴.۳. [۱۷] برای کدکردن کد LDPC به طول ۱۰ (طول بلوکی) و نرخ $\frac{۱}{۲}$ ماتریس بررسی‌زوجیت زیر را در نظر بگیرید:

$$H = \begin{bmatrix} ۱ & ۱ & ۰ & ۱ & ۱ & ۰ & ۰ & ۱ & ۰ & ۰ \\ ۰ & ۱ & ۱ & ۰ & ۱ & ۱ & ۱ & ۰ & ۰ & ۰ \\ ۰ & ۰ & ۰ & ۱ & ۰ & ۰ & ۰ & ۱ & ۱ & ۱ \\ ۱ & ۱ & ۰ & ۰ & ۰ & ۱ & ۱ & ۰ & ۱ & ۰ \\ ۰ & ۰ & ۱ & ۰ & ۰ & ۱ & ۰ & ۱ & ۰ & ۱ \end{bmatrix}$$

ابتدا باید H را به شکل سطری پلکانی تبدیل شود، که این کار توسط اعمال سطری مقدماتی تحت میدان $(F_۲)$ انجام می‌شود (یعنی جابجایی دو سطر یا اضافه کردن یک سطر به سطر دیگر به پیمانه ۲).

حاصل جمع به پیمانه ۲ سطرهای ۱ و ۴ را در سطر ۴ قرار می‌دهیم و سطر ۳ و ۵ را جابجا می‌کنیم.

در نهایت جمع به پیمانه ۲ سطر ۴ و ۵ را در سطر ۵ قرار می‌دهیم ماتریس جدید H_r که شکل سطری پلکانی

ماتریس H است به صورت زیر است:

$$H_r = \begin{bmatrix} ۱ & ۱ & ۰ & ۱ & ۱ & ۰ & ۰ & ۱ & ۰ & ۰ \\ ۰ & ۱ & ۱ & ۰ & ۱ & ۱ & ۱ & ۰ & ۰ & ۰ \\ ۰ & ۰ & ۱ & ۰ & ۰ & ۱ & ۰ & ۱ & ۰ & ۱ \\ ۰ & ۰ & ۰ & ۱ & ۱ & ۱ & ۱ & ۱ & ۱ & ۰ \\ ۰ & ۰ & ۰ & ۰ & ۱ & ۱ & ۱ & ۰ & ۰ & ۱ \end{bmatrix}$$

سپس ماتریس حاصل را به ماتریس سطری پلکانی تقلیل یافته تبدیل می‌کنیم:

$$H_{rr} = \begin{bmatrix} ۱ & ۰ & ۰ & ۰ & ۰ & ۰ & ۱ & ۱ & ۱ & ۰ \\ ۰ & ۱ & ۰ & ۰ & ۰ & ۱ & ۰ & ۱ & ۰ & ۰ \\ ۰ & ۰ & ۱ & ۰ & ۰ & ۱ & ۰ & ۱ & ۰ & ۱ \\ ۰ & ۰ & ۰ & ۱ & ۰ & ۰ & ۰ & ۱ & ۱ & ۱ \\ ۰ & ۰ & ۰ & ۰ & ۱ & ۱ & ۱ & ۰ & ۰ & ۱ \end{bmatrix}$$

و در نهایت با استفاده از جابجایی‌های ستونی ماتریس بررسی‌زوجیت H_{rr} را به فرم استاندارد تبدیل می‌کنیم:

$$H_{std} = \begin{bmatrix} ۰ & ۱ & ۱ & ۱ & ۰ & ۱ & ۰ & ۰ & ۰ & ۰ \\ ۱ & ۰ & ۱ & ۰ & ۰ & ۰ & ۱ & ۰ & ۰ & ۰ \\ ۱ & ۰ & ۱ & ۰ & ۱ & ۰ & ۰ & ۱ & ۰ & ۰ \\ ۰ & ۰ & ۱ & ۱ & ۱ & ۰ & ۰ & ۰ & ۱ & ۰ \\ ۱ & ۱ & ۰ & ۰ & ۱ & ۰ & ۰ & ۰ & ۰ & ۱ \end{bmatrix}$$

(اگر به ستون‌های ماتریس H_{rr} اندیس‌های $\begin{bmatrix} ۱۰ & ۹ & ۸ & ۷ & ۶ & ۵ & ۴ & ۳ & ۲ & ۱ \end{bmatrix}$ را نسبت دهیم) شماره ستون‌ها در ماتریس H_{std} به صورت زیر است:

$$\begin{bmatrix} ۶ & ۷ & ۸ & ۹ & ۱۰ & ۱ & ۲ & ۳ & ۴ & ۵ \end{bmatrix}$$

باید توجه کرد که در مرحله آخر با جابجایی ستون‌های ماتریس H_{rr} ، کدواژه‌های حاصل از ماتریس H_{std} ، نسخه جابجاشده کدواژه‌های حاصل از H می‌باشند. بنابراین قبل از اینکه کدواژه حاصل از H_{std} ارسال شود باید (با توجه به شماره ستون‌های جابجا شده) جابجایی معکوس را روی کدواژه انجام گردد.

اگر کانال فاقد حافظه باشد، چون ترتیب بیت‌های کدواژه مهم نیست ، می‌توان جابجایی مورد نظر را روی H اجرا کرد با این کار ماتریس بررسی‌زوجیت جدید عبارت است از:

$$H' = \begin{bmatrix} ۱ & ۱ & ۰ & ۱ & ۱ & ۰ & ۰ & ۱ & ۰ & ۰ \\ ۰ & ۱ & ۱ & ۰ & ۱ & ۱ & ۱ & ۰ & ۰ & ۰ \\ ۰ & ۰ & ۰ & ۱ & ۰ & ۰ & ۰ & ۱ & ۱ & ۱ \\ ۱ & ۱ & ۰ & ۰ & ۰ & ۱ & ۱ & ۰ & ۱ & ۰ \\ ۰ & ۰ & ۱ & ۰ & ۰ & ۱ & ۰ & ۱ & ۰ & ۱ \end{bmatrix}$$

که همان خواص H را دارد با این تفاوت که بیت‌های کدواژه‌های آن براساس H_{std} مرتب شده‌اند. به این ترتیب ماتریس مولد، برای کدی که ماتریس‌های بررسی‌زوجیت آن H_{std} و H' عبارت است از:

$$G = \begin{bmatrix} ۱ & ۰ & ۰ & ۰ & ۰ & ۰ & ۱ & ۱ & ۰ & ۱ \\ ۰ & ۱ & ۰ & ۰ & ۰ & ۱ & ۰ & ۰ & ۰ & ۱ \\ ۰ & ۰ & ۱ & ۰ & ۰ & ۱ & ۱ & ۱ & ۱ & ۰ \\ ۰ & ۰ & ۰ & ۱ & ۰ & ۱ & ۰ & ۰ & ۱ & ۰ \\ ۰ & ۰ & ۰ & ۰ & ۱ & ۰ & ۰ & ۱ & ۱ & ۱ \end{bmatrix}$$

$\diamondsuit$

اجرای روش حذفی گوس-جردن به همراه جابجایی ستونی بروی H ، برای رسیدن به حالت مطلوب به $o(n^۳)$ عملیات نیاز دارد. از طرفی در کدگذاری، فقط یکبار نیازمند انجام این عملیات نیاز هستیم. به این ترتیب می‌توان این عملیات را به عنوان پیش‌پردازش دید. عملیات مورد نظر برای هر کدگذاری $o(n^۲)$ است، زیرا برای هر کدواژه c متعلق به $\mathcal{C}$ کدگذاری آن به صورت

$$c^T = u^T \otimes G$$

است که در آن u بردار اطلاعات و G ماتریس مولد است.

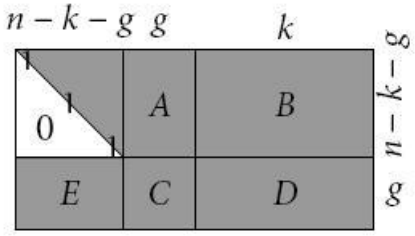
۱.۴.۳ کدگذاری پیچیدگی زمانی‌خطی ، برای کدهای LDPC

فرض کنید، یک کد LDPC به طول n و بعد k توسط ماتریس بررسی‌زوجیت $(n-k) \times n$ ، H تولید شده است (یعنی یک کد شامل مجموعه بردارهای x به طول n به طوری که):

$$H \otimes x = \mathbf{0}$$

فرض کنید H رتبه کامل سطری دارد. اگر بتوان H را به صورت زیر تبدیل کرد (در صورت لزوم با جابجایی ستون‌های H) می‌توان کدگذاری را انجام داد:

$$H = [H_p \;\; H_S]$$



شکل ۳.۱۰: [۱۸] فرم بالا مثلثی تقریبی H

از این پس ماتریس H را هنگامی که به فرم

$$H_t = [H_p \;\; H_S]$$

تبدیل شود H_t می‌نامیم. بردار x به دو قسمت سیستماتیک $s \in F_2^k$ ، (منظور از سیستماتیک، ناحیه اطلاعات اصلی به طول k در کدواژه است که در روش استاندارد معمولا این قسمت در ابتدای همه کدواژه‌ها قرار دارد) و بخش بررسی $p \in F_2^{n-k}$ ، یعنی $x = [p \;\; s]^T$ تقسیم می‌شود. به جای یافتن ماتریس مولد متناظر با ماتریس بررسی‌زوجیت، کد LDPC می‌تواند به طور مستقیم با استفاده از ماتریس بررسی‌زوجیت کدگذاری شود. برای این منظور می‌توان از تبدیل ماتریس بررسی‌زوجیت به شکل ماتریسی که زیر ماتریسی بالا مثلثی دارد و جایگذاری برعکس استفاده کرد. در ابتدا ماتریس بررسی‌زوجیت، فقط با استفاده از جابجایی‌های سطری و ستونی به فرم زیر تبدیل می‌شود:

$$H_t = \begin{bmatrix} T & A & B \\ E & C & D \end{bmatrix} \qquad (۲۲.۳)$$

با توجه به $H_t = [H_p \;\; H_S]$ ، پس $H_p = \begin{bmatrix} T & A \\ E & C \end{bmatrix}$. (شکل ۳.۱۰ را ببینید) لازم است که T مربعی و بالا مثلثی باشد. به این ترتیب می‌گوییم H به فرم بالا مثلثی تقریبی تبدیل می‌شود و با H_t نمایش می‌دهیم. نکته مهم این است که چون این تبدیل فقط با استفاده از جابجایی‌های سطری و ستونی انجام شده است، ماتریس حاصل هنوز هم تنک است. بنابراین اگر بتوان T را بگونه‌ای ساخت که از اندازه $n-k$ باشد، آنگاه کدگذاری با پیچیدگی $o(n)$ امکانپذیر خواهد بود.

به طور کلی (همانطور که در ادامه نشان داده خواهد شد) اگر بتوان T را از اندازه g $n-k-g$ ساخت، آنگاه می‌توان کدگذاری را با پیچیدگی $o(n+g^{۲})$ به انجام رساند، به این ترتیب هدف ساختن چنین تجزیه‌ای است که اندازه T تا جای ممکن به $n-k$ نزدیک شود.

تعداد سطرهای زیر ماتریس $\begin{bmatrix} E & C & D \end{bmatrix}$ را **شکاف** [۲۵] گویند و با g نمایش می‌دهند.

اندازه زیر ماتریس‌های H_t به صورت زیر است:

$$\begin{array}{lcllcl} T & : & (n-k-g)\times(n-k-g) & A & : & (n-k-g)\times g \\ B & : & (n-k-g)\times k & E & : & g\times(n-k-g) \\ C & : & g\times g & D & : & g\times k \end{array}$$

فرض کنید ماتریس بررسی‌زوجیت H به فرم (۲۲.۳) تبدیل شده است. کدگذاری در حقیقت حل دستگاه معادلات $H_t\otimes[p\ \ s]^T=\mathbf{0}$ برای p است (یعنی p را بیابیم، s ها معین هستند). با ضرب ماتریس

$$\begin{bmatrix} I & \mathbf{0} \\ -ET^{-۱} & I \end{bmatrix} \qquad (۲۳.۳)$$

از چپ در H_t داریم:

$$\begin{bmatrix} T & A & B \\ \mathbf{0} & C-ET^{-۱}A & D-ET^{-۱}B \end{bmatrix} \qquad (۲۴.۳)$$

به این ترتیب با اجرای روش حذفی گوس برای حذف E فقط $C-ET^{-۱}A$ و $D-ET^{-۱}B$ تحت تاثیر قرار می‌گیرند و بقیه ماتریس بررسی‌زوجیت، تنک باقی می‌ماند.

$x=[p\ \ s]^T$ را به صورت $x=[p_۱\ \ p_۲,s]^T$ می‌نویسیم که در آن $p_۱$ و $p_۲$ به ترتیب زیرماتریس‌های ستونی $(n-k-g)\times ۱$ و $g\times ۱$ هستند. با توجه به (۲۴.۳) معادله $H_t\otimes x=\mathbf{0}$ را به دو معادله زیر تبدیل می‌کنیم:

$$Tp_۱+Ap_۲+Bs=\mathbf{0} \qquad (۲۵.۳)$$

$$(C-ET^{-۱}A)p_۲+(D-ET^{-۱}B)s=\mathbf{0} \qquad (۲۶.۳)$$

[۲۵] Gap

عملیات	پیچیدگی
Bs	$o(n)$
$T^{-1}[Bs]$	$o(n)$
$-E[T^{-1}Bs]$	$o(n)$
Ds	$o(n)$
$Ds - E[T^{-1}Bs]$	$o(n)$
$-\phi^{-1}[Ds - ET^{-1}Bs]$	$o(n + g^2)$

(۲۷.۳)

تعریف می‌کنیم :

$$\phi = C - ET^{-1}A$$

فرض وارون‌پذیر بودن ϕ معادل فرض وارون‌پذیر بودن H_p است (اگر ϕ وارون‌پذیر باشد). از (۲۶.۳) و فرض وارون‌پذیری ϕ داریم:

$$p_2^T = -\phi^{-1}(D - ET^{-1}B)s \qquad (۲۸.۳)$$

بنابراین اگر ماتریس $-\phi^{-1}(D - ET^{-1}B)$ که از اندازه $g \times k$ است، محاسبه شود، آنگاه پیچیدگی محاسبه p_2، $o(g \times k)$ است. این پیچیدگی همانطور که در جدول (۲۷.۳) نشان داده شده است، می تواند بیشتر کاهش یابد. با توجه به جدول (۲۷.۳)، برای کاهش پیچیدگی در گام اول $(D - ET^{-1}B)s$ محاسبه شده است. سپس $-\phi^{-1}$ که ماتریسی $(g \times g)$ است، محاسبه و از چپ در $(D - ET^{-1}B)s$ ضرب شده است. با توجه به جدول (۲۷.۳) پیچیدگی محاسبه گام اول $o(n)$ است، زیرا ابتدا Bs محاسبه شده است که با توجه به تنک بودن B پیچیدگی آن $o(n)$ است. سپس نتیجه در T^{-1} ضرب شده است. چون $T^{-1}[Bs] = Y$ معادل با دستگاه $[Bs] = TY$ است. بنابراین پیچیدگی آن براساس جایگزینی برعکس (یادآوری می‌کنیم که T بالا مثلثی و تنک است) $o(n)$ است. در

نهایت پیچیدگی محاسبه $p_۲$ ، $o(n+g^۲)$ است.

(۲۹.۳)

عملیات	پیچیدگی
$AP_۲$	$o(n)$
Bs	$o(n)$
$[AP_۲]+[Bs]$	$o(n)$
$-T^{-۱}[AP_۲+Bs]$	$o(n)$

از رابطه (۲۵.۳) می توان نوشت $Tp_۱ = -(Ap_۲ + Bs)$. می‌توان $P_۱$ را همانطور که جدول (۲۹.۳) نشان می‌دهد با پیچیدگی $o(n)$ محاسبه کرد.

به این ترتیب می‌توان خلاصه فرآیند کدگذاری (که شامل دو مرحله پیش‌پردازش و کدگذاری واقعی است) را به صورت زیر بیان کرد:

در مرحله پیش‌پردازش، ابتدا جابجایی‌های سطری و ستونی برای تبدیل ماتریس بررسی زوجیت به فرم بالا مثلثی تقریبی با g تا حد ممکن کوچک انجام می‌شود. سپس از چپ ماتریس (۲۳.۳) در ماتریس H_t ضرب می‌شود تا ϕ بدست آید (می‌توان ضرب رابطه (۲۳.۳) از چپ در ماتریس H_t را به طور کارایی توسط اجرای الگوریتم حذفی گوس انجام داد) به منظور بررسی اینکه آیا ϕ منفرد هست یا خیر، تا جایی که قطر برای $n-k$ سطر ادامه دارد حذف گوسی ادامه میابد. با این کار سه حالت ممکن است رخ دهد، اول، ممکن است حذف گوسی بدون نیاز به جابجایی‌های ستونی با موفقیت انجام شود. در این حالت ϕ نامنفرد (دترمینان مخالف صفر) است. دوم، حذف گوسی موفقیت آمیز است ولی نیاز به برخی جابجایی‌های ستونی است. در این حالت با به کار بردن جابجایی‌های ستونی مورد نیاز ماتریس جدید و هم ارز ϕ نامنفرد است. سوم، اگر هیچ جابجایی ستونی وجود نداشته باشد به طوری که حذفی گوسی با موفقیت انجام شود، آنگاه ماتریس اصلی، رتبه کامل ندارد. باید توجه داشت که مرحله پیش‌پردازش فقط یکبار انجام می‌گیرد.

خلاصه روش کدگذاری بیان شده در ادامه بیان می‌شود:

- **پیش پردازش**

ورودی : ماتریس بررسی‌زوجیت نامنفرد H .

خروجی : ماتریس بررسی‌زوجیت هم‌ارز H ، به فرم $H_t = \begin{bmatrix} T & A & B \\ E & C & D \end{bmatrix}$ ، به طوری که

$C - ET^{-1}A$ نامنفرد باشد.

۱. [مثلثی کردن] اجرای جابجایی‌های سطری و ستونی برای تبدیل ماتریس بررسی‌زوجیت به فرم بالا مثلثی تقریبی

$$H_t = \begin{bmatrix} T & A & B \\ E & C & D \end{bmatrix}$$

که در آن شکاف g تا اندازه ممکن کوچک است.

۲. [بررسی رتبه] استفاده از حذفی گوس برای اجرای

$$\begin{bmatrix} I & \mathbf{0} \\ -ET^{-1} & I \end{bmatrix} \begin{bmatrix} T & A & B \\ E & C & D \end{bmatrix} = \begin{bmatrix} T & A & B \\ \mathbf{0} & C - ET^{-1}A & D - ET^{-1}B \end{bmatrix}$$

بررسی نامنفرد بودن $\phi = C - ET^{-1}A$ (با اجرای جابجایی‌های ستونی بیشتر، درصورت نیاز به این خاصیت). (منفرد بودن H در اینجا حذف می‌شود)

- **کدگذاری**

ورودی : ماتریس بررسی‌زوجیت به فرم $H_t = \begin{bmatrix} T & A & B \\ E & C & D \end{bmatrix}$ به طوری که $C - ET^{-1}A$ نامنفرد باشد و بردار $s \in F^k$.

خروجی : بردار $x = [p_1 \;\; p_2 \;\; s]^T$ که $p_1 \in F^{n-k-g}$ ، $p_2 \in F^g$ به طوری که $H_t \otimes x = \mathbf{0}$.

۱. تعیین p_2 همانطور که در جدول (۲۷.۳) نشان داده شده است.

۲. تعیین p_1 همانطور که در جدول (۲۹.۳) نشان داده شده است.

مثال ۳.۴.۲. [۱۸] ماتریس بررسی‌زوجیت مربوط به (۳،۶)-کد با قاعده به طول ۱۲ که به صورت زیر است را در نظر بگیرید:

$$H = \begin{array}{c} \\ ۱ \\ ۲ \\ ۳ \\ ۴ \\ ۵ \\ ۶ \end{array} \begin{array}{c} \begin{array}{cccccccccccc} ۱ & ۲ & ۳ & ۴ & ۵ & ۶ & ۷ & ۸ & ۹ & ۱۰ & ۱۱ & ۱۲ \end{array} \\ \left[\begin{array}{cccccccccccc} ۰ & ۰ & ۱ & ۱ & ۰ & ۱ & ۰ & ۱ & ۰ & ۱ & ۱ & ۰ \\ ۱ & ۰ & ۰ & ۱ & ۰ & ۱ & ۱ & ۰ & ۱ & ۰ & ۱ & ۰ \\ ۰ & ۱ & ۱ & ۰ & ۰ & ۱ & ۱ & ۱ & ۰ & ۰ & ۰ & ۱ \\ ۰ & ۱ & ۱ & ۰ & ۱ & ۰ & ۱ & ۰ & ۱ & ۱ & ۰ & ۰ \\ ۱ & ۰ & ۰ & ۱ & ۱ & ۰ & ۰ & ۱ & ۱ & ۰ & ۰ & ۱ \\ ۱ & ۱ & ۰ & ۰ & ۱ & ۰ & ۰ & ۰ & ۰ & ۱ & ۱ & ۱ \end{array}\right] \end{array}$$

با انجام جابجایی‌های سطری و ستونی داریم:

$$H_t = \begin{array}{c} \\ ۲ \\ ۱ \\ ۴ \\ ۳ \\ ۵ \\ ۶ \end{array} \begin{array}{c} \begin{array}{cccccccccccc} ۱ & ۴ & ۱۰ & ۳ & ۵ & ۶ & ۷ & ۸ & ۹ & ۲ & ۱۱ & ۱۲ \end{array} \\ \left[\begin{array}{cccccc|cccccc} ۱ & ۱ & ۰ & ۰ & ۰ & ۱ & ۱ & ۰ & ۱ & ۰ & ۱ & ۰ \\ ۰ & ۱ & ۱ & ۱ & ۰ & ۱ & ۰ & ۱ & ۰ & ۰ & ۱ & ۰ \\ ۰ & ۰ & ۱ & ۱ & ۱ & ۰ & ۱ & ۰ & ۱ & ۱ & ۰ & ۰ \\ ۰ & ۰ & ۰ & ۱ & ۰ & ۱ & ۱ & ۱ & ۰ & ۱ & ۰ & ۱ \\ \hline ۱ & ۱ & ۰ & ۰ & ۱ & ۰ & ۰ & ۱ & ۱ & ۰ & ۰ & ۱ \\ ۱ & ۰ & ۱ & ۰ & ۱ & ۰ & ۰ & ۰ & ۰ & ۱ & ۱ & ۱ \end{array}\right] \end{array}$$

شکاف $g = ۲$ است. الگوریتم گوسی را برای تبدیل E به صفر اجرا می‌کنیم، داریم:

$$\begin{array}{c} \\ ۲ \\ ۱ \\ ۴ \\ ۳ \\ ۵ + ۲ \\ ۶ + ۲ + ۱ + ۳ \end{array} \begin{array}{c} \begin{array}{cccccccccccc} ۱ & ۴ & ۱۰ & ۳ & ۵ & ۶ & ۷ & ۸ & ۹ & ۲ & ۱۱ & ۱۲ \end{array} \\ \left[\begin{array}{cccccc|cccccc} ۱ & ۱ & ۰ & ۰ & ۰ & ۱ & ۱ & ۰ & ۱ & ۰ & ۱ & ۰ \\ ۰ & ۱ & ۱ & ۱ & ۰ & ۱ & ۰ & ۱ & ۰ & ۰ & ۱ & ۰ \\ ۰ & ۰ & ۱ & ۱ & ۱ & ۰ & ۱ & ۰ & ۱ & ۱ & ۰ & ۰ \\ ۰ & ۰ & ۰ & ۱ & ۰ & ۱ & ۱ & ۱ & ۰ & ۱ & ۰ & ۱ \\ \hline ۰ & ۰ & ۰ & ۰ & ۱ & ۱ & ۱ & ۱ & ۰ & ۰ & ۱ & ۱ \\ ۰ & ۰ & ۰ & ۰ & ۱ & ۱ & ۰ & ۰ & ۱ & ۰ & ۱ & ۰ \end{array}\right] \end{array}$$

چون $\phi = \begin{bmatrix} ۱ & ۱ \\ ۱ & ۱ \end{bmatrix}$ منفرد است، برای ساختن ϕ وارون‌پذیر ستون‌های ۶ و ۷ را جابجا می‌کنیم، داریم:

$\phi = \begin{bmatrix} ۱ & ۱ \\ ۱ & ۰ \end{bmatrix}$. ماتریس بالا مثلثی تقریبی حاصل برای کدگذاری عبارت است از:

$$H_t = \begin{array}{c} \\ ۲ \\ ۱ \\ ۴ \\ ۳ \\ ۵ \\ ۶ \end{array} \begin{array}{c} \begin{array}{cccccccccccc} ۱ & ۴ & ۱۰ & ۳ & ۵ & ۷ & ۶ & ۸ & ۹ & ۲ & ۱۱ & ۱۲ \end{array} \\ \left[\begin{array}{cccc|cc|cccccc} ۱ & ۱ & ۰ & ۰ & ۰ & ۱ & ۱ & ۰ & ۱ & ۰ & ۱ & ۰ \\ ۰ & ۱ & ۱ & ۱ & ۰ & ۰ & ۱ & ۱ & ۰ & ۰ & ۱ & ۰ \\ ۰ & ۰ & ۱ & ۱ & ۱ & ۱ & ۰ & ۰ & ۱ & ۱ & ۰ & ۰ \\ ۰ & ۰ & ۰ & ۱ & ۰ & ۱ & ۱ & ۱ & ۰ & ۱ & ۰ & ۱ \\ \hline ۱ & ۱ & ۰ & ۰ & ۱ & ۰ & ۰ & ۱ & ۱ & ۰ & ۰ & ۱ \\ ۱ & ۰ & ۱ & ۰ & ۱ & ۰ & ۰ & ۰ & ۰ & ۱ & ۱ & ۱ \end{array}\right] \end{array}$$

فرض کنید $s = [۱\ ۰\ ۰\ ۰\ ۰\ ۰]^T$ ، $P_۲$ براساس جدول (۲۷.۳) محاسبه می‌شود. داریم:

$$Bs = [۱\ ۱\ ۰\ ۱]^T \quad , \qquad T^{-۱}[Bs] = [۰\ ۱\ ۱\ ۱]^T \quad , \qquad E[T^{-۱}Bs] = [۱\ ۱]^T$$

$$Ds = [۰\ ۰]^T \quad , \qquad [Ds] + [ET^{-۱}Bs] = [۱\ ۱]^T \quad ,$$

$$\phi^{-۱}[Ds + ET^{-۱}Bs] = [۱\ ۰]^T = P_۲$$

و به طور مشابه براساس جدول (۲۹.۳) داریم:

$$AP_۲ = [۰\ ۰\ ۱\ ۰]^T \qquad [AP_۲] + [Bs] = [۱\ ۱\ ۱\ ۱]^T$$

$$T^{-۱}[AP_۲ + Bs] = [۱\ ۰\ ۰\ ۱]^T = P_۱$$

بنابراین کدواژه، عبارت است از:

$$x = [P_۱\ \ P_۲\ \ s]^T = [۱\ ۰\ ۰\ ۱\ ۱\ ۰\ ۱\ ۰\ ۰\ ۰\ ۰\ ۰]^T$$

می توان بررسی کرد که در $H_t \otimes x = \mathbf{0}$ صدق می‌کند.

$\diamondsuit$

فصل ۴

کدگشایی کدهای LDPC با استفاده از کدگشای عبورپیام

۱.۴ مقدمه فصل

در فصل قبل کدهای LDPC و نحوه کدگذاری آنها مورد بررسی قرار گرفت. در این فصل و فصل بعد دو روش کدگشایی الگوریتم عبور پیام و کدگشایی برنامه‌ریزی خطی برای این دسته از کدها بیان می‌گردد. این روش‌های کدگشایی مبتنی بر تکرار هستند به همین دلیل آنها را **روش‌های کدگشایی تکراری** می‌نامند. دسته‌ای از الگوریتم‌های کدگشایی که برای کدهای LDPC (و به طور کلی کدهای خطی) مورد استفاده قرار می‌گیرد به طور مجموع **الگوریتم‌های عبور پیام**[۱] نامیده می‌شوند. چون عملیاتی که در این الگوریتم‌ها انجام می‌گیرد با استفاده از عبورپیام در امتداد یال‌های گراف تنر (و به طور کلی گراف فاکتور) شرح داده می‌شود این الگوریتم‌ها را عبور پیام می‌نامند. الگوریتم عبورپیام برای کدهای LDPC اولین بار توسط گالاگر در [۱۵] (درسال ۱۹۶۲) مطرح شد. تفاوت الگوریتم‌های عبور پیام در نوع پیام‌ها یا در عملیاتی است که در راس‌ها انجام می‌گیرد. از جمله الگوریتم‌های عبورپیام می‌توان به **کدگشایی معکوس کردن وضعیت بیت**[۲] که در آن پیام‌ها دودویی هستند، اشاره کرد. یکی دیگر از الگوریتم‌های عبورپیام، **کدگشایی گسترش سنت (عقیده)** [۳] (BP) است که در این روش پیام‌ها احتمالات هستند (که این احتمالات در حقیقت سطحی از عقاید را در مورد مقدار بیت‌های کدواژه مطرح

[۱] Message passing algorithms
[۲] Bit-flipping decoding
[۳] Belief propagation decoding

می‌کنند). البته به جای احتمالات می‌توان از لگاریتم نسبت‌های درستنمایی استفاده کرد که در این حالت کدگشایی گسترش عقیده را اغلب **کدگشایی جمع-ضرب**[۴] می‌نامند. علت این نامگذاری این است که در راس‌های بررسی و متغیر از عملگرهای ضرب وجمع برای پیام‌ها (که لگاریتم نسبت‌های درستنمایی هستند) استفاده می‌شود. قبل از بیان الگوریتم‌های عبور پیام، به بیان مطالبی می‌پردازیم که ممکن است در ساده‌سازی مفاهیمی که در ادامه بیان می‌شود، کمک کند. البته برای مطالعه بیشتر می‌توان به [۱۸] مراجعه کرد.

می‌دانیم قانون توزیع‌پذیری، یکی از قوانینی است که در هر میدان F وجود دارد. همچنین می‌دانیم نیم‌حلقه جابجایی $\mathbb{K}$ با دو عملگر $+$ و $\centerdot$ در اوصول موضوعه زیر صدق می‌کند:

۱. این دو عملگر خاصیت تعویض‌پذیری و شرکت‌پذیری دارند.

۲. عنصر خنثی عمل $+$ ، ۰ ، و عنصر خنثی عمل $\centerdot$ ، ۱ است.

۳. به ازای هر $x, y, z \in \mathbb{K}$ قانون توزی‌پذیری به صورت زیر برقرار است:

$$(x + y) \centerdot z = (x \centerdot z) + (y \centerdot z)$$

همانطور که می‌دانیم حلقه جابجایی در مقایسه با میدان F نیاز به وجود وارون نسبت به جمع و ضرب ندارد. در جدول (۱.۴) لیستی از حلقه‌های جابجایی که در کدگشایی مورد استفاده قرار می‌گیرند بیان می‌شود.

(۱.۴)

$\mathbb{K}$	$(\centerdot, ۱)$	$(+, ۰)$	
F	$(\centerdot, ۱)$	$(+, ۰)$	
$F[x, y, \ldots]$	$(\centerdot, ۱)$	$(+, ۰)$	
$\mathbb{R} \geq ۰$	$(\centerdot, ۱)$	$(+, ۰)$	ضرب-جمع
$\mathbb{R} > ۰ \cup \{\infty\}$	$(\centerdot, ۱)$	(min, ∞)	$min -$ ضرب
$\mathbb{R}$	$(\centerdot, ۱)$	$(max, ۰)$	$max -$ ضرب
$\mathbb{R} \cup \{\infty\}$	$(+, ۰)$	(min, ∞)	$min -$ جمع
$\mathbb{R} \cup \{-\infty\}$	$(+, ۰)$	$(max, -\infty)$	$max -$ جمع
$\{۰, ۱\}$	$(AND, ۱)$	$(OR, ۰)$	بولی

[۴] Sum-product decoding

یکی از کدگشایی‌های تکراری بسیار مهم، حلقه جمع-ضرب (با دو عملگر، جمع و ضرب معمولی) است. مثال مهم دیگر حلقه max-جمع است که برای اجرای کدگشایی بلوکی مناسب است. به منظور مشاهده عملکرد این حلقه در قانون توزیع‌پذیری، جمع را با max و ضرب را با جمع جابجا می‌کنیم داریم :

$$max\{x+y, x+z\} = x + max\{y, z\}$$

به طور کلی تر:

$$\max_{i,j}\{x_i + y_j\} = \max_{i}\{x_i\} + \max_{j}\{y_j\}$$

به این ترتیب آنچه در این فصل بیان می‌شود عبارت است از:

در بخش ۲.۴ عبور پیام تحت کانال پاک‌شدگی دودویی بیان می‌شود. در بخش ۳.۴ کدگشایی معکوس کردن وضعیت بیت مورد بررسی قرار می‌گیرد. در بخش ۴.۴ کدگشای جمع-ضرب بیان می‌شود و در بخش ۵.۴ تحلیل کدگشای عبور پیام مطرح می‌شود.

باید توجه کرد که در الگوریتم‌های عبور پیام یک تکرار الگوریتم از ارسال پیام‌ها از راس‌های بررسی آغاز و با بازگشت پیام‌ها به راس‌های بررسی پایان می‌یابد.

۲.۴ عبور پیام تحت کانال پاک‌شدگی دودویی

وقتی بیتی (مؤلفه‌ای از کدواژه دودویی) توسط کانال BEC ارسال می‌شود، آنچه دریافت می‌گردد یا خود بیت است یا با احتمال δ بیت به طور کامل پاک شده است. به ترتیب چون بیت‌های معلوم دریافتی همیشه درست هستند بنابراین وظیفه کدگشا تعیین مقادیر بیت‌های مجهول است. به این ترتیب اگر معادله بررسی‌زوجیت فقط شامل یک بیت پاک شده باشد این بیت پاک شده توسط معادله بررسی‌زوجیت (راس بررسی) مورد نظر می‌تواند تعیین شود. برای شرح بیشتر به مثال بعد توجه کنید.

مثال ۱.۲.۴. [۱۷] کد دودویی را که در معادلات بررسی‌زوجیت زیر صدق می‌کند را در نظر بگیرید:

$$\begin{aligned} x_۱ \oplus x_۲ \oplus x_۴ &= ۰ \\ x_۲ \oplus x_۳ \oplus x_۵ &= ۰ \\ x_۱ \oplus x_۲ \oplus x_۳ \oplus x_۶ &= ۰ \end{aligned}$$

اگر کدواژه $[c_۱\ c_۲\ c_۳\ c_۴\ c_۵\ c_۶]^T$ ارسال شده باشد، با جایگذاری در معادله بررسی‌زوجیت اول

$$c_۱ \oplus c_۲ \oplus c_۴ = ۰$$

اگر $c_۱ = ۰$ ، $c_۲ = ۱$ و $c_۴ =?$ باشد، آنگاه:

$$\begin{aligned} c_۴ &= x_۱ \oplus x_۲ \\ &= ۰ \oplus ۱ \\ &= ۱ \end{aligned}$$

$\diamondsuit$

فرض کنید پیام‌های آغازی (پیامهای راس‌های برگ فاکتور) ۰ یا ? یا ۱ باشد، با توجه به قوانین عبور پیام استاندارد که در فصل ۳، شکل ۵.۳ و مطالب بیان شده در بخش ۴.۲.۳ پیام آغازی می‌تواند

$$(\mu_j(۰), \mu_j(۱)) = (p(y_j|۰), p(y_j|۱)\ )$$

باشد به این ترتیب برای کانالBEC، پیام‌های آغازی مساوی با $(\delta, ۰\ ,۱-\delta)$ یا (δ, δ) یا $(۰\ ,۱-\delta)$ هستند که به ترتیب متناظر با احتمال دریافت ۰ یا ? یا ۱ می‌باشند. با توجه به اینکه نرمال‌سازی پیام تاثیری در کدگشایی ندارد (همانطور که گفتیم پیام‌ها در الگوریتم‌های عبور پیام مختلف می‌توانند متفاوت باشند) بنابراین می‌توانیم این احتمالات را به ترتیب با پیام‌های $(۱\ ,۰)$ ، $(۱\ ,۱)$ و $(۰\ ,۱)$ جایگزین کنیم. با توجه به این مطالب و شکل ۵.۳ قوانین پردازش پیام در راس‌های متغیر و بررسی برای BEC به صورت زیر هستند: در هر راس متغیر پیام خروجی، پاک‌شدگی (?) است اگر همه پیام‌های ورودی پاک‌شدگی باشند. در غیر این صورت چون کانال هرگز خطاهایی از نوع تبدیل ۰ به ۱ یا برعکس ایجاد نمی‌کند، همه پیام‌های غیر پاک‌شدگی باید موافق باهم، ۰ یا ۱ باشند. در این حالت پیام خروجی برابر با این مقدار مشترک است. در راس بررسی ، پیام خروجی پاک‌شدگی است اگر در بین

پیام‌های ورودی به آن راس پاک‌شدگی وجود داشته باشد. در غیر این صورت اگر همه پیام‌ها ۰ یا ۱ باشند آنگاه پیام خروجی جمع به پیمانه ۲ پیام‌های ورودی است. ادعای اول را در نظر بگیرید: اگر همه پیام‌های ورودی به یک راس متغیر به فرم $\{(۱,۱),(۰,۱)\}$ باشند آنگاه پیام خروجی (مساوی با ضرب مؤلفه‌ای پیام‌ها ورودی متناظر با قانون کلی عبورپیام) عنصری از این مجموعه است. علاوه براین، پیام خروجی مساوی با $(۱,۱)$ است اگر همه پیام‌های ورودی به فرم $(۱,۱)$ باشند. این مطلب درباره مجموعه $\{(۱,۱),(۰,۱)\}$ نیز درست است. (توجه کنید که چون این کانال جز پاک کردن بیت خطای دیگری ایجاد نمی‌کند، بنابراین جز این دو حالت، حالت دیگری ایجاد نمی‌شود) حال برای راس بررسی، کافی است یک راس بررسی درجه ۳ با دو پیام ورودی را درنظر بگیریم چون راس بررسی با درجه بیشتر از ۳ را می‌توان به عنوان اتصالات سری چندین راس بررسی مدل کرد به طوری که هر یک از آنها دو ورودی و یک خروجی دارند (به عنوان مثال $(x_۱ + x_۲ + x_۳ = (x_۱ + x_۲) + x_۳)$ ، $(\mu_۱(۰),\mu_۱(۱))$ و $(\mu_۲(۰),\mu_۲(۱))$ دو پیام ورودی باشند، براساس قانون عبور پیام استاندار، پیام خروجی عبارت است از:

$$\begin{aligned}(\mu(۰),\mu(۱)) &= \Big(\sum_{x_۱,x_۲}\mathbb{I}_{\{x_۱\oplus x_۲=۰\}}\mu_۱(x_۱)\mu_۲(x_۲), \sum_{x_۱,x_۲}\mathbb{I}_{\{x_۱\oplus x_۲=۱\}}\mu_۱(x_۱)\mu_۲(x_۲)\Big)\\ &= \big(\mu_۱(۰)\mu_۲(۰)\oplus\mu_۱(۱)\mu_۲(۱), \mu_۱(۰)\mu_۲(۱)\oplus\mu_۱(۱)\mu_۲(۰)\big)\end{aligned}$$

اگر $(\mu_۲(۰),\mu_۲(۱)) = (۱,۱)$ آنگاه $(\mu(۰),\mu(۱)) = (۱,۱)$. این نشان می‌دهد که اگر هر یک از دو ورودی پاک‌شدگی باشد آنگاه خروجی پاک‌شدگی است. علاوه‌براین اگر هردو پیام معین (غیر پاک‌شدگی) باشند آنگاه با یک بررسی ساده می‌توان دید که قانون عبور پیام، جمع به پیمانه ۲ پیام‌ها است.

قبل از بیان الگوریتم عبور پیام تحت کانال BEC برخی نمادهای مورد نیاز در این الگوریتم را بیان می‌کنیم. فرض کنید پیامی که راس متغیر j ارسال می‌کند M_j و پیامی که راس بررسی i به راس متغیر j ارسال می‌کند $E_{i,j}$ باشد. B_i مجموعه اندیس متغیرهای متعلق به معادله بررسی‌زوجیت i و A_j مجموعه معادلات بررسی‌زوجیتی (راس‌های بررسی) که متغیر j ام را مورد بررسی قرار می‌دهند در نظر بگیرید. مثالی که در ادامه بیان می‌شود این نمادها را مورد استفاده قرار می‌دهد.

مثال ۴.۲.۲. [۱۷] کد LDPC با ماتریس بررسی زوجیتی باقاعده با $\omega_r = ۳$ و $\omega_c = ۲$ ($rank(H) = ۳$) را در نظر بگیرید :

$$H = \begin{bmatrix} ۱ & ۱ & ۰ & ۱ & ۰ & ۰ \\ ۰ & ۱ & ۱ & ۰ & ۱ & ۰ \\ ۱ & ۰ & ۰ & ۰ & ۱ & ۱ \\ ۰ & ۰ & ۱ & ۱ & ۰ & ۱ \end{bmatrix}$$

$$B_۱ = \{۱, ۲, ۴\} \ , \ B_۲ = \{۲, ۳, ۵\} \ , \ B_۳ = \{۱, ۵, ۶\} \ , \ B_۴ = \{۳, ۴, ۶\} \ ,$$

$$A_۱ = \{۱, ۳\} \ , \ A_۲ = \{۱, ۲\} \ , \ A_۳ = \{۲, ۴\} \ , \ A_۴ = \{۱, ۴\} \ , \ A_۵ = \{۲, ۳\} \ , \ A_۶ = \{۳, ۴\} \ ,$$

$\diamondsuit$

الگوریتم عبورپیام تحت BEC ([۱۷]) :

ورودی : پیام دریافتی $y = [y_۱ \ \dots \ y_n]^T$ و I_{max} (ماکزیمم تکرار الگوریتم) .

خروجی : $M = [M_۱ \ \dots \ M_n]^T$.

گام صفر (آغاز) : $\forall j = ۱, \dots, n$ قرار دهید $M_j = y_j$ و $I = ۰$.

گام یک (به روزسازی پیام‌ها در راس‌های بررسی) : $\forall i = ۱, \dots, m$ و $\forall j \in B_i$ ، اگر به جز M_j همه پیام‌هایی که به راس بررسی i رسیده‌اند معلوم (غیر پاک‌شدگی) باشند آنگاه :

$$E_{i,j} = \sum_{j' \in B_i, j' \neq j} M_j$$

(جمع بالا به پیمانه ۲ است) ، در غیر این صورت

$$E_{i,j} = ?$$

.

گام دوم (به روزسازی پیام‌ها در راس‌های متغیر) : $\forall j = ۱, \ldots, n$ اگر پاک‌شدگی $M_j = \exists i \in A_j$ و

به طوری که $E_{i,j} \neq ?$ آنگاه :

$$M_j = E_{i,j}$$

مرحله بررسی : اگر همه M_j ها معلوم هستند یا $I = I_{max}$ آنگاه پایان. در غیراین صورت $I = I + ۱$ و به گام ۱ بروید.

مثال ۳.۲.۴. [۱۷] کد مثال (۲.۲.۴) را در نظر بگیرید. فرض کنید $c = [۰\ ۰\ ۱\ ۰\ ۱\ ۱]^T$ تحت کانال BEC ارسال شده است و بردار $y = [۰\ ۰\ ۱\ ?\ ?\ ?]^T$ دریافت شود. با استفاده از الگوریتم عبور پیام تحت کانال BEC بیت‌های پاک‌شده را بازیابی می‌کنیم :

گام صفر : $M = [۰\ ۰\ ۱\ ?\ ?\ ?]^T$

گام یک : محاسبه پیام‌های راس‌های بررسی :

راس بررسی ۱ به راس‌های متغیر ۱ ، ۲ و ۴ متصل است بنابراین پیام‌های ورودی ۱ ، ۰ و ? هستند. داریم :

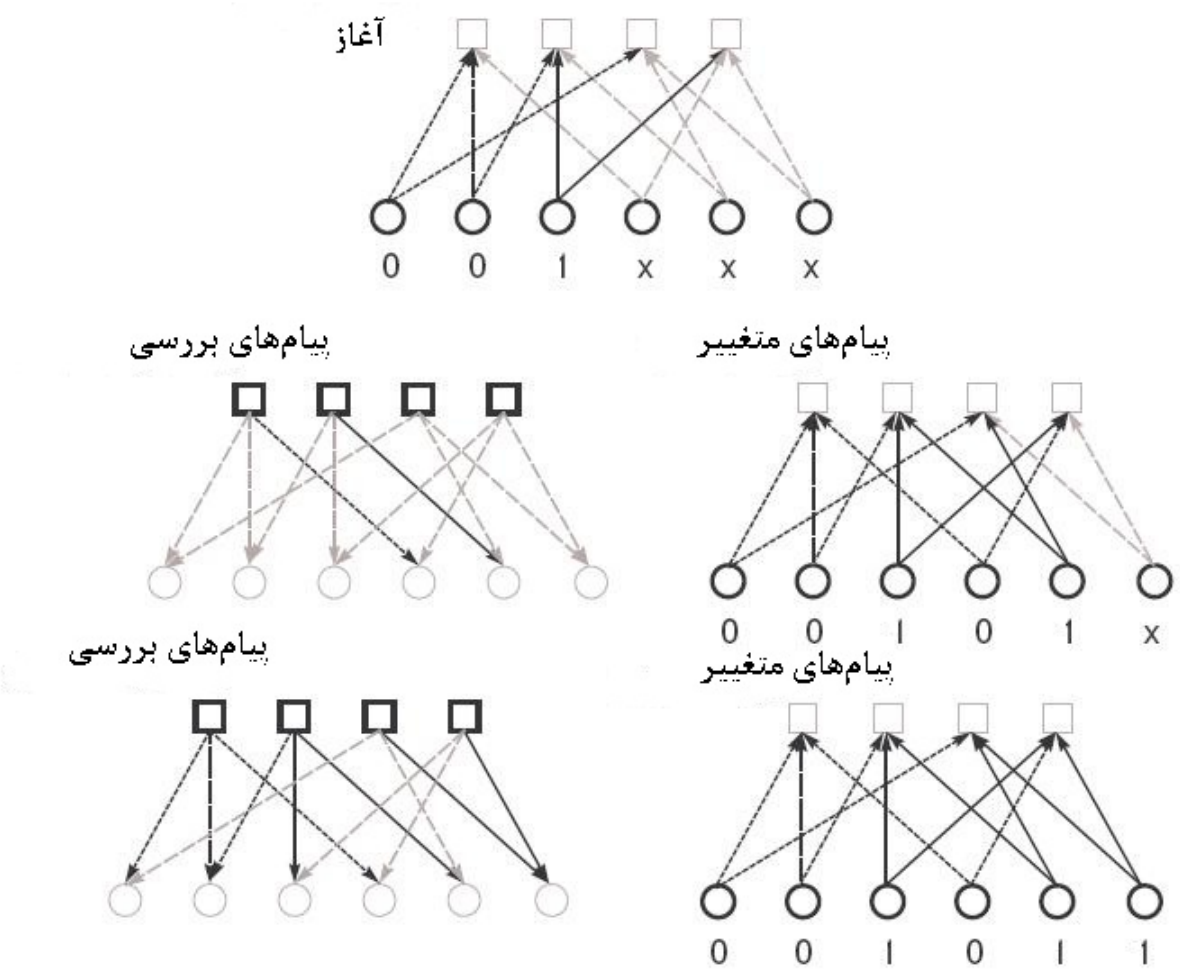
$$\begin{aligned} E_{۱,۴} &= M_۱ \oplus M_۲ \\ &= ۰ \oplus ۰ \\ &= ۰ \end{aligned}$$

برای سایر راس‌ها :

$$\begin{aligned} E_{۲,۵} &= M_۲ \oplus M_۳ \\ &= ۰ \oplus ۱ \\ &= ۱ \end{aligned}$$

راس بررسی ۳ به راس‌های متغیر ۱ ، ۵ و ۶ متصل است. به این ترتیب پیام‌های ورودی ۰ ، ? و ? هستند. چون به این راس بررسی دو ? رسیده است بنابراین آنچه به راس‌های متغیر متصل به خود ارسال می‌کند ? است. به طور مشابه ، راس بررسی۴ ام به راس‌های متغیر ۳ ، ۴ و ۶ متصل است و چون دو پیام ? را دریافت می‌کند لذا پیامی را که به همه راس‌های متغیر متصل به خود ارسال می‌کند ? خواهد بود.

گام دوم : هر راس متغیر که مقدار پاک‌شدگی دارد ، می‌تواند با استفاده از پیام‌های ورودی خود در صورت امکان مقدار کنونی خود را بهبود ببخشد. مقدار راس ۴ ، ? است و پیام‌های ورودی آن $E_{۱,۴} = ۰$ و $E_{۴,۴} = ?$



شکل ۱.۴: کدگشای عبور پیام بردار دریافتی $y = [0\ 0\ 1\ ?\ ?\ ?]^T$. هریک از شکل‌ها بیانگر تصمیمی است که الگوریتم براساس پیام‌های گام قبلی اتخاذ می‌کند. برای نمایش عبور پیام ۰ از یک یال ، آن یال با نقطه چین پر رنگ ، عبور پیام ۱ با خط و عبور پیام ? با نقطه چین کم رنگ نشان داده شده است .

هستند. به این ترتیب راس مقدار خود را به ۰ تغییر می‌دهد. مقدار راس متغیر ۵ نیز ? است و پیام‌های ورودی $E_{۲,۵} = ۱$ و $E_{۳,۵} =?$ می‌باشند لذا مقدار خود را به ۱ تغییر می‌دهد. مقدار راس متغیر ۶ نیز ? است. ولی پیام‌های ورودی $E_{۳,۶} =?$ و $E_{۴,۶} =?$ هستند. بنابراین راس نمی‌تواند مقدار خود را تغییر دهد. در انتهای گام دوم داریم :

$$M = [۰\ ۰\ ۱\ ۰\ ۱\ ?]^T$$

با توجه به اینکه هنوز مقدار راس ۶ ، ? است ، الگوریتم ادامه می‌یابد.

تکرار گام یک :

$$\begin{array}{lll} E_{۳,۶} = & M_۱ \oplus M_۵ & E_{۴,۶} = \quad M_۳ \oplus M_۴ \\ = & ۱ \oplus ۰ & = \quad ۰ \oplus ۱ \\ = & ۱ & = \quad ۱ \end{array}$$

راس متغیر ۶ پیام‌های $E_{۳,۶} = ۱$ و $E_{۴,۶} = ۱$ را دریافت می‌کند و به این ترتیب مقدار خود را به ۱ تغییر می‌دهد. توجه داشته باشید که چون بیت‌های دریافتی از کانال همیشه صحیح هستند (اگر پاک شده نباشند) بنابراین پیام‌های دریافتی از راس‌های بررسی همیشه موافق خواهند بود. در انتهای تکرار گام دوم داریم :

$$M = [۰\ ۰\ ۱\ ۰\ ۱\ ۱]^T$$

به این ترتیب چون بیت پاک‌شده‌ای وجود ندارد ، الگوریتم متوقف می‌شود و M را به عنوان کدواژه کدگشایی شده بر می‌گرداند. شکل ۱.۴ فرآیند انجام شده را نشان می‌دهد.

$\diamondsuit$

۳.۴ کدگشای معکوس کردن وضعیت بیت

الگوریتم کدگشای معکوس کردن وضعیت بیت ، الگوریتم عبور پیام با تصمیم سخت برای کدهای LDPC است. تصمیم سخت دودویی درباره هر بیت دریافتی توسط آشکارساز اتخاذ می‌شود و به کدگشا ارسال می‌گردد. در الگوریتم کدگشای معکوس کردن وضعیت بیت ، پیام‌هایی که در امتداد یال‌های گراف تنر عبور می‌کند ، دودویی

هستند. در این حالت هر راس متغیر پیام دریافتی را که صفر یا یک است به راس‌های بررسی مجاور به خود ارسال می‌کند . هر راس بررسی پیامی را براساس اطلاعات در دسترس خود ، به راس‌های متغیر مجاور به خود ارسال می‌کند ، علاوه براین هر راس بررسی ، درستی معادله بررسی‌زوجیت مربوط به خود را با توجه به پیام‌های دریافتی تعیین می‌کند (جمع به پیمانه دو پیام‌های ورودی به راس بررسی صفر هست یا خیر ؟) . اگر اکثریت پیام‌های دریافتی توسط یک راس متغیر با مقدار قبلی همان راس متفاوت باشد ، راس متغیر مقدار کنونی خود را تغییر می‌دهد. این فرآیند تا هنگامی که همه معادلات بررسی‌زوجیت برقرار باشند یا الگوریتم به ماکزیمم تعداد تکرار تعیین شده در الگوریتم نرسیده باشد ، ادامه می‌یابد. با توجه به اینکه برقرار بودن همه معادلات بررسی‌زوجیت بیانگر متعلق بودن کدواژه به کد مورد نظر است این شرط در همه الگوریتم‌های عبورپیام از جمله کدگشای معکوس کردن وضعیت بیت وجود دارد. انجام این شرط بررسی در الگوریتم‌ها دو مزیت دارد :

۱. با یافتن جواب ، از تکرار اضافی الگوریتم جلوگیری می‌شود.

۲. در صورت شکست خوردن الگوریتم این شکت همیشه نمایان می‌گردد (توجه کنید که قوانین کلی عبور پیام در فصل ۳ شکل ۵.۳ بیان شده است) .

الگوریتم کدگشای معکوس کردن وضعیت بیت براساس این اصل است که یک بیت (مؤلفه) کدواژه ، اگر در تعداد زیادی از معادلات برسی‌زوجیت ناصحیح قرار داشته باشد به طور محتمل‌تر خودش ناصحیح است.

الگوریتم کدگشایی معکوس کردن وضعیت بیت ([۱۷]) :

ورودی : بردار $y = [y_1, \ldots, y_n]^T$ حاصل از اتخاذ تصمیم سخت روی واژه دریافتی

و حداکثر تعداد تکرار الگوریتم I_{max} .

خروجی : $M = [M_1, \ldots, M_n]^T$.

گام صفر (آغاز) : $\forall j = 1, \ldots n$ قرار دهید:

$$I = 0 \ , \ M_j = y_j$$

گام یک (مرحله به‌روزسازی پیام‌ها در راس‌های بررسی) :

$\forall j = ۱,\ldots,n$ و $\forall i = ۱,\ldots,m$ محاسبه کنید :

$$E_{i,j} = \sum_{j' \in B_i, j' \neq j} M_j$$

(جمع بالا به پیمانه ۲ انجام می‌گیرد) .

گام ۲(مرحله به‌روزسازی پیام در راس‌های متغیر) :

اگر پیام‌های $E_{i,j}$ با y_j موافق نباشند آنگاه :

$$M_j = (y_j + ۱)$$

مرحله بررسی (برقرار بودن معادلات بررسی‌زوجیت):

$\forall i = ۱,\ldots,m$ محاسبه کنید :

$$L_i = \sum_{j' \in B_i} M_{j'}$$

(جمع به پیمانه ۲ انجام می‌گیرد) اگر به ازای هر $i = ۱,\ldots,m$ ، $L_i = ۰$ یا $I = I_{max}$ آنگاه پایان . در غیر این صورت قرار دهید :

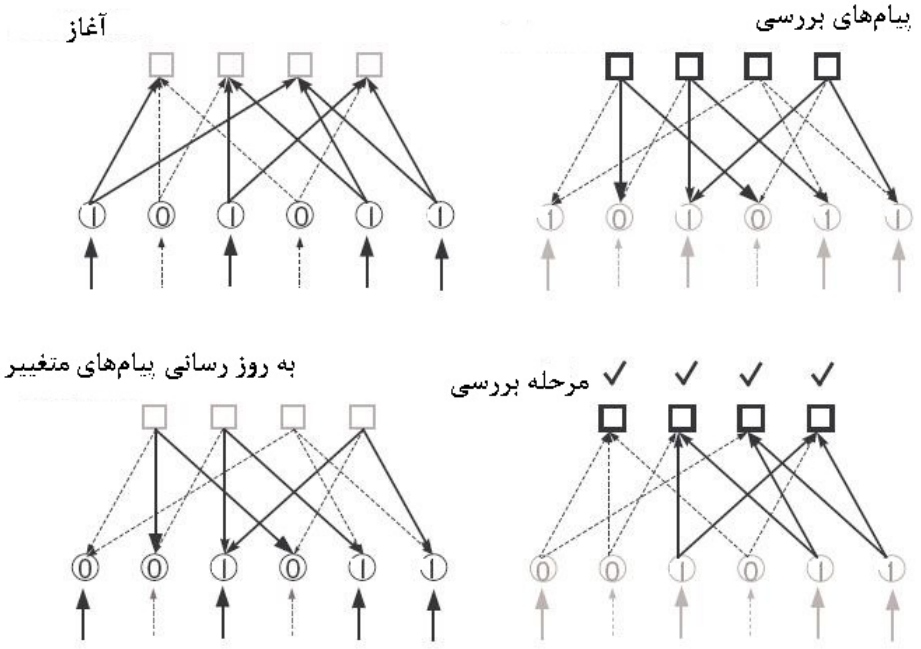
$$I = I + ۱$$

و به گام ۱ بروید.

مثال ۱.۳.۴. [۱۷] کد LDPC مثال (۲.۲.۴) را در نظر بگیرید. فرض کنید کدواژه $x = [۰\ ۰\ ۱\ ۰\ ۱\ ۱]^T$ تحت کانال BSC با احتمال همگذری $p = ۰/۲$ ارسال و $y = [۱\ ۰\ ۱\ ۰\ ۱\ ۱]^T$ پیام دریافتی باشد. مراحل الگوریتم کدگشایی معکوس کردن وضعیت بیت به صورت زیر انجام می‌گیرد:

گام صفر : (آغاز) $M_j = y_j$ پس

$$M = [۱\ ۰\ ۱\ ۰\ ۱\ ۱]^T$$



شکل ۲.۴: کدگشای معکوس کردن وضعیت بیت برای بردار دریافتی $y = [۱\ ۰\ ۱\ ۰\ ۱\ ۱]^T$. هریک از شکل‌ها بیانگر تصمیمی است که الگوریتم براساس پیام‌های گام قبلی اتخاذ می‌کند. علامت $\times$ بیانگر این است که معادله بررسی‌زوجیت برقرار نیست و علامت $\surd$ بیانگر این است که معادله بررسی‌زوجیت برقرار است. برای نمایش عبورپیام صفر از یک یال ، آن یال را با نقطه چین و عبورپیام یک را با خط نشان می‌دهیم.

گام یک : محاسبه پیام‌های خروجی از راس‌های بررسی :

$$B_۱ = \{۱, ۲, ۴\} \qquad E_{۱,۱} = M_۲ \oplus M_۴ = ۰ \oplus ۰ = ۰ \qquad E_{۱,۲} = M_۱ \oplus M_۴ = ۱ \oplus ۰ = ۱ \qquad E_{۱,۴} = M_۱ \oplus M_۲ = ۱ \oplus ۰ = ۱$$

$$B_۲ = \{۲, ۳, ۵\} \qquad E_{۲,۲} = M_۳ \oplus M_۵ = ۱ \oplus ۱ = ۰ \qquad E_{۲,۳} = M_۲ \oplus M_۵ = ۰ \oplus ۱ = ۱ \qquad E_{۲,۵} = M_۲ \oplus M_۳ = ۰ \oplus ۱ = ۱$$

برای سایر راس‌های بررسی‌زوجیت داریم :

$$E_{۳,۱} = ۰\ , \quad E_{۳,۵} = ۰\ , \quad E_{۳,۶} = ۰$$
$$E_{۴,۳} = ۰\ , \quad E_{۴,۴} = ۰\ , \quad E_{۴,۶} = ۰$$

گام دوم : به‌روزسازی پیام‌ها در راس‌های متغیر :

$A_۱ = \{۱, ۳\}$ هر دو پیام ورودی به راس متغیر ۱ ، صفر هستند. بنابراین اکثر پیامهایی که به راس متغیر ۱

وارد می‌شوند ، از مقدار کنونی متفاوت هستند لذا راس متغیر ۱ مقدار خودش را تغییر (مقدار بیت را معکوس می‌کند) می‌دهد . $A_۲ = \{۱, ۲\}$ هر دو پیام ورودی به راس متغیر ۲ با مقدار کنونی راس یکسان هستند. بنابراین راس متغیر ۲ مقدار خود را تغییر نمی‌دهد. برای سایر راس‌های متغیر نیز تعداد کافی پیام‌های ورودی متفاوت با مقدار کنونی آن راس وجود ندارند به همین دلیل مقدار خود را تغییر نمی‌دهند. پیام‌های جدید که از راس‌های متغیر به راس‌های بررسی وارد می‌شوند ، عبارت‌اند از :

$$M = [۰\ ۰\ ۱\ ۰\ ۱\ ۱]^T$$

حال الگوریتم به بررسی درستی این جواب در معادلات بررسی‌زوجیت می‌پردازد. به این ترتیب :

$$\begin{aligned} L_۱ &: \quad M_۱ \oplus M_۲ \oplus M_۴ = \ ۰ \oplus ۰ \oplus ۰ = ۰ \\ L_۲ &: \quad M_۲ \oplus M_۳ \oplus M_۵ = \ ۰ \oplus ۱ \oplus ۱ = ۰ \end{aligned}$$

به طور مشابه $L_۳ = ۰$ و $L_۴ = ۰$ لذا بردار M در همه معادلات بررسی‌زوجیت صدق می‌کند ، بنابراین الگوریتم متوقف می‌شود و $M = [۰\ ۰\ ۱\ ۰\ ۱\ ۱]^T$ کدواژه کدگشایی شده است. در شکل ۲.۴ مراحل انجام شده، نشان داده شده است.

$\diamondsuit$

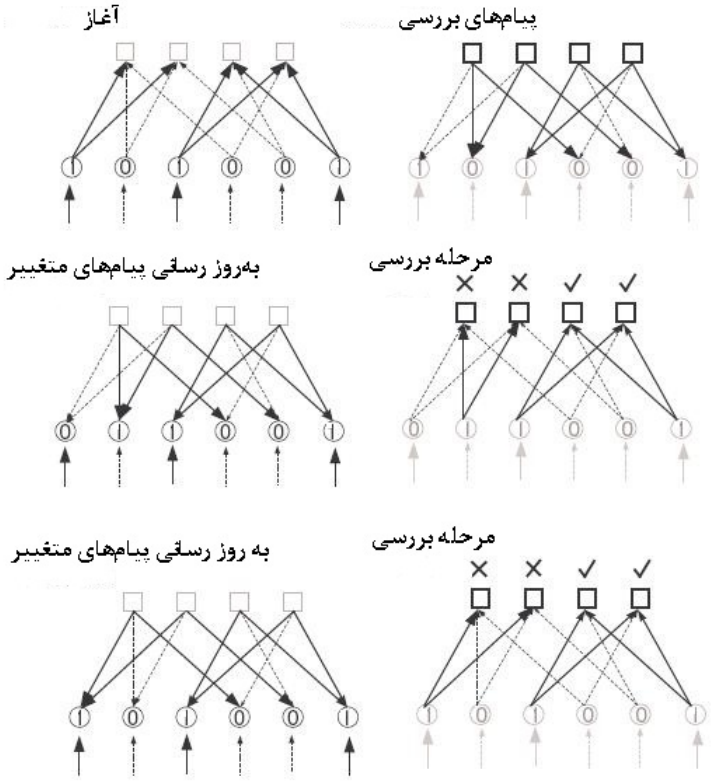
مثال ۰.۲.۳.۴ [۱۷]

کد متناظر با گراف تنر شکل ۳.۴ را در نظر بگیرید فرض کنید کدواژه $x = [۰\ ۰\ ۱\ ۰\ ۰\ ۱]^T$ تحت کانال پارازیت‌دار سفید گوسی جمعی ورودی دودویی با (BPSK) [۵] (که یک روش مدلاسیون است و با توجه به تخصصی بودن آن نیازی به شرح بیشتر ندارد) ارسال شده و پیام دریافتی

$$[-۱/۱ \quad +۱/۵ \quad -۰/۵ \quad +۱ \quad +۱/۸ \quad -۲]$$

باشد. با اتخاذ تصمیم سخت آشکارساز روی هر بیت (مؤلفه) ، کدواژه $y = [۱\ ۰\ ۱\ ۰\ ۰\ ۱]^T$ حاصل می‌شود. همانطور که می‌بینید بیت اول واژه دریافتی ناصحیح است. مراحل الگوریتم کدگشایی معکوس کردن وضعیت بیت

[۵] signaling and with binary phase shift keying



شکل ۳.۴: کدگشای معکوس کردن وضعیت بیت برای بردار دریافتی $y = [۱\ ۰\ ۱\ ۰\ ۰\ ۱]$. هریک از شکل‌ها بیانگر تصمیمی است که الگوریتم براساس پیام‌های گام قبلی اتخاذ می‌کند. علامت $\times$ بیانگر این است که معادله بررسی‌زوجیت برقرار نیست و علامت $\surd$ بیانگر این است که معادله بررسی‌زوجیت برقرار است. برای نمایش عبور پیام صفر از یک یال ، آن یال را با نقطه چین و عبور پیام یک را با خط نشان می‌دهیم.

در شکل (۳.۴) نمایش داده شده است. مقدار آغازی راس‌های متغیر به ترتیب ۱ ، ۰ ، ۱ ، ۰ ، ۰ و ۱ است. پیام‌ها به راس‌های بررسی ارسال می‌گردند. در گام اول مشخص می‌گردد که معادلات بررسی‌زوجیت ۱ و۳ برقرار نیستند بنابراین الگوریتم ادامه می‌یابد. در گام دوم بیت‌های ۱ و ۲ اکثر پیام‌هایی که دریافت می‌کنند از مقدار کنونی آنها متفاوت است به همین دلیل مقدار آنها تغییر می کنند. با تکرار گام اول خواهیم دید که معادلات بررسی‌زوجیت ۱ و ۲ برقرار نیستند. در تکرارهای بعدی مقدار دو بیت ۱ و ۲ تغییر می‌کنند به طوری که همیشه یکی از آنها ناصحیح است و الگوریتم شکست می‌خورد. به این ترتیب چون هر دو بیت اول کدواژه در دو معادله بررسی‌زوجیت مشابه قرار دارند. بنابراین هنگامی که هر دو معادله برقرار نباشند ، تعیین بیت خطا غیر ممکن است.

$\diamondsuit$

۴.۴ کدگشای جمع - ضرب

الگوریتم جمع-ضرب ، یک الگوریتم با تصمیم نرم است. این الگوریتم مشابه الگوریتم کدگشایی معکوس کردن وضعیت بیت است که در بخش قبل مورد بررسی قرار گرفت. با این تفاوت که در الگوریتم جمع - ضرب پیام‌ها احتمالات هستند ، در حالی که کدگشای معکوس کردن وضعیت بیت ، کار را با یک تصمیم سخت روی بیت‌های دریافتی (به عنوان ورودی) آغاز می کند. الگوریتم جمع - ضرب ، احتمال هر بیت دریافتی را به عنوان ورودی می‌پذیرد. در حقیقت ورودی احتمالات بیتی هستند که آنها را **احتمالات پیشین**[۶] برای بیت‌های دریافتی می‌نامند. زیرا این احتمالات قبل از شروع کار کدگشا LDPC مشخص می‌شوند. احتمالات بیتی که توسط کدگشا برگردانده می‌شوند را **احتمالات پسین** [۷] می‌نامند. در کدگشای جمع-ضرب این احتمالات به صورت لگاریتم نسبت‌های درستنمایی بیان می‌شوند. برای متغیر دودویی x به راحتی می‌توان $p(x=۱)$ را با فرض معلوم بودن $p(x=۰)$ یافت ($p(x=۱)=۱-p(x=۰)$) به این ترتیب فقط نیاز به ذخیره یکی از این دو احتمال برای x هستیم. لگاریتم نسبت‌های درستنمایی که برای نمایش میزان درستی متغیر دودویی مورد استفاده قرار می‌گیرد، مقدار منحصر به فرد زیر می‌باشد :

$$L(x)=\ln\left(\frac{p(x=۰)}{p(x=۱)}\right) \qquad (۲.۴)$$

اگر $p(x=۰)>p(x=۱)$ آنگاه $L(x)$ مثبت است و برعکس اگر $p(x=۱)>p(x=۰)$ باشد، $L(x)$ منفی است. به این ترتیب علامت $L(x)$ یک تصمیم سخت را روی x مهیا می‌کند و اندازه $|L(x)|$ قابلیت اعتماد (اعتبار) این تصمیم است. برای تبدیل لگاریتم درستنمایی به احتمالات داریم:

$$p(x=۱)=\frac{p(x=۱)/p(x=۰)}{۱+(p(x=۱)/p(x=۰))}=\frac{e^{-L(x)}}{۱+e^{-L(x)}} \qquad (۳.۴)$$

[۶] A priori probabilities
[۷] A posteriori probability

و

$$p(x=۰)=\frac{p(x=۰)/p(x=۱)}{۱+(p(x=۱)/p(x=۰))}=\frac{e^{L(x)}}{۱+e^{L(x)}} \qquad (۴.۴)$$

مزیت استفاده از نمایش لگاریتمی احتمالات این است که هنگامی که احتمالات باید در هم ضرب شوند، لگاریتم نسبت‌های درستنمایی فقط با هم جمع می‌شوند ، که این عمل موجب کاهش پیچیدگی اجرا می‌گردد. هدف کدگشای جمع-ضرب، محاسبه ماکزیمم احتمالات پسین (MAP) برای هر بیت کدواژه است . یعنی $P_i=p(c_i=۱|N)$ (احتمال اینکه مولفه i ام کدواژه ۱ باشد به شرط اینکه بدانیم رخداد N در همه محدودیت‌های بررسی‌زوجیت صدق می‌کند(. اطلاعات اضافی درباره بیت i ام ، که از محدودیت بررسی‌زوجیت دریافت شده است را **اطلاعات بیرونی** [۸] می‌نامند. الگوریتم جمع-ضرب به طور تکراری مقدار تقریبی MAP را برای هر بیت محاسبه می‌کند. اگر چه احتمالات پسین که توسط کدگشا جمع-ضرب برگردانده می‌شوند با فرض اینکه گراف تنر **دور آزاد** [۹] (به عبارت دیگر فاقد دور) باشد فقط احتمالات MAP دقیق هستند. به طور خلاصه اطلاعات بیرونی حاصل از محدودیت بررسی‌زوجیت برای یک بیت در تکرار اول مستقل از احتمالات پیشین برای همان بیت است ، البته اطلاعات بیرونی به احتمالات پیشین سایر بیت‌های کدواژه (بقیه راس‌های متغیر گراف تنر) وابسته است. اطلاعات بیرونی حاصل برای بیت i ام در بقیه تکرارها نیز از احتمال پیشین اصلی همان بیت مستقل است تا هنگامی که احتمال پیشین از طریق یک دور متعلق به گراف تنر به بیت i ام برگردانده شود. همبستگی اطلاعات بیرونی به احتمال پیشین اصلی بیت موجب می‌شود که نتایج احتمالات پسین دقیق نباشد. در کدگشا جمع-ضرب اطلاعات بیرونی از راس بررسی i به راس متغیر j ، $E_{i,j}$ عبارت است از LLR (لگاریتم نسبت درستنمایی) احتمال اینکه بیت j ام موجب شود راس بررسی i برقرار باشد. اگر بیت j ام ۱ باشد احتمال اینکه معادله بررسی‌زوجیت برقرار باشد ، برابراست با :

$$P_{i,j}^{ext}=\frac{۱}{۲}-\frac{۱}{۲}\prod_{j'\in B_i,j'\neq j}\left(۱-۲P_{j'}^{int}\right) \qquad (۵.۴)$$

[۸] Extrinsic information
[۹] Cycle free

در (۵.۴) $P_{i,j'}^{int}$ احتمال این است که بیت j' ، ۱ باشد. اگر بیت j ام صفر باشد احتمال اینکه معادله بررسی‌زوجیت i ام برقرار باشد $1 - P_{i,j}^{ext}$ است. به این ترتیب داریم :

$$E_{i,j} = LLR(P_{i,j}^{ext}) = ln\left(\frac{1 - P_{i,j}^{ext}}{P_{i,j}^{ext}}\right) \qquad (6.4)$$

با جایگذاری (۵.۴) در (۶.۴) داریم :

$$E_{i,j} = ln\left(\frac{\frac{1}{2} + \frac{1}{2}\prod_{j' \in B_i, j' \neq j}(1 - 2P_{j'}^{int})}{\frac{1}{2} - \frac{1}{2}\prod_{j' \in B_i, j' \neq j}(1 - 2P_{j'}^{int})}\right) \qquad (7.4)$$

رابطه زیر را در نظر بگیرید :

$$tanh\left(\frac{1}{2}ln(\frac{1-P}{P})\right) = 1 - 2P \qquad (8.4)$$

با استفاده از (۸.۴) داریم:

$$E_{i,j} = ln\left(\frac{1 + \prod_{j' \in B_i, j' \neq j} tanh(M_{i,j'}/2)}{1 - \prod_{j' \in B_i, j' \neq j} tanh(M_{i,j'/2})}\right) \qquad (9.4)$$

که :

$$M_{i,j'} = LLR(P_{i,j'}^{int}) = ln\left(\frac{1 - P_{i,j'}^{int}}{P_{i,j'}^{int}}\right) \qquad (10.4)$$

با توجه به رابطه زیر :

$$2tanh^{-1}(p) = ln(\frac{1+p}{1-p})$$

می‌توان (۹.۴) را به صورت زیر نوشت :

$$E_{i,j} = 2tanh^{-1}(\prod_{j' \in B_i, j' \neq j} tanh(M_{i,j'/2})\ ) \qquad (11.4)$$

هر راس متغیر j به LLR پیشین r_i و LLR های راس‌های بررسی‌زوجیت مجاور با خودش (راس متغیر j) دسترسی دارد. به این ترتیب LLR کل بیت j ام عبارت است از جمع این LLR ها ، یعنی :

$$L_j = LLR(P_j^{int}) = r_j + \sum_{i \in A_j} E_{i,j} \qquad (۱۲.۴)$$

باید توجه کرد که پیام‌های $M_{i,j}$ ارسال شده از راس‌های متغیر به راس بررسی ، مقدار LLR همه راس‌های متغیر نیستند. به بیان دقیق‌تر به منظور جلوگیری از بازگشت اطلاعاتی که توسط یک راس بررسی ارسال شده است به خود آن راس بررسی ، پیام راس متغیر j به راس بررسی i عبارت است از (۱۲.۴) بدون مؤلفه $E_{j,i}$ ای که از راس بررسی i ام دریافت شده است. یعنی :

$$M_{i,j} = \sum_{i' \in A_j, i' \neq i} E_{i',j} + r_j \qquad (۱۳.۴)$$

الگوریتم کدگشای جمع- ضرب ([۱۷]) :

ورودی : لگاریتم نسبت‌های درستنمایی برای احتمالات پیشین $r_j = ln\frac{(c_t = ۰)}{(c_t = ۱)}$ (پیام‌های ورودی) و ماتریس بررسی‌زوجیت H و ماکزیمم تکرار الگوریتم I_{max} .

خروجی : برآورد احتمالات پسین بیتی ، بیت‌های دریافتی به صورت لگاریتم نسبت‌های درستنمایی .

گام صفر(آغاز) : $\forall j = ۱, \ldots, n \ , \ i = ۱, \ldots, m$

$$M_{i,j} = r_j$$

تا هنگام رسیدن به پایان گام‌های زیر را انجام دهید.

گام یک(مرحله به‌روزسازی پیام‌ها در راس‌های بررسی) :

$\forall i = ۱, \ldots, m$ و $j \in B_i$ محاسبه کنید :

$$E_{i,j} = ln \left(\frac{۱ + \prod_{j' \in B_i, j' \neq j} tanh(M_{i,j'}/۲)}{۱ - \prod_{j' \in B_i, j' \neq j} tanh(M_{i,j'/۲})} \right)$$

مرحله بررسی(برقرار بودن معادلات بررسی‌زوجیت) :

$\forall j = ۱, \ldots, n$ محاسبه کنید :

$$L_j = \sum_{i \in A_j} E_{i,j} + r_j \ , \ z_j = \begin{cases} ۱ & L_j \leq ۰ \\ ۰ & L_j > ۱ \end{cases}$$

اگر $I = I_{max}$ یا $H \otimes z = \mathbf{0}$ آنگاه پایان . در غیر این صورت :

گام دوم (مرحله به‌روزسازی پیام‌های راس‌های متغیر) :

$\forall j = ۱, \ldots, n$ و $i \in A_j$ محاسبه کنید :

$$M_{i,j} = \sum_{i' \in A_j, i' \neq i} E_{i,j} + r_j$$

.

مثال ۱.۴.۴. [۱۷] مثال ۱.۳.۴ را در نظر بگیرید. فرض کنید کدواژه

$$c = [۰ \ ۰ \ ۱ \ ۰ \ ۱ \ ۱]^T$$

تحت کانال BSC با احتمال همگذری $p = ۰/۲$ ارسال شده است و $y = [۱ \ ۰ \ ۱ \ ۰ \ ۱ \ ۱]^T$ واژه دریافتی باشد. چون کانال متقارن دودویی است ، احتمال اینکه بیت ۰ ارسال و ۱ دریافت گردد (یعنی احتمال همگذری) p است .به این ترتیب احتمالات پیشین برای BSC به صورت زیر است :

$$r_j = \begin{cases} ln \frac{p}{۱ - p} & y_i = ۱ \\ ln \frac{۱ - p}{p} & y_i = ۰ \end{cases}$$

در این کانال داریم :

$$ln\frac{p}{۱-p} = ln\frac{۰/۲}{۰/۸} = -۱/۳۸۶۳$$

$$ln\frac{۱-p}{p} = ln\frac{۰/۸}{۰/۲} = ۱/۳۸۶۳$$

و بنابراین لگاریتم نسبت‌های درستنمایی عبارتند از :

$$r = [-۱/۳۸۶۳ \;\; ۱/۳۸۶۳ \;\; -۱/۳۸۶۳ \;\; ۱/۳۸۶۳ \;\; -۱/۳۸۶۳ \;\; -۱/۳۸۶۳]^T$$

ماکزیمم تعداد تکرار الگوریتم کدگشایی را ۳ قرار می‌دهیم. در آغاز :

$$M_{i,j} = r_j$$

بیت اول (به طور متناظر راس متغیر اول) در معادله‌های بررسی‌زوجیت ۱ و ۳ قرار دارد (به راس‌های بررسی ۱ و ۳ متصل است) بنابراین :

$$M_{۱,۱} = r_۱ = -۱/۳۸۶۳ \quad M_{۳,۱} = r_۱ = -۱/۳۸۶۳$$

و برای سایر بیت‌ها داریم :

$$\begin{array}{lll} j=۲ & M_{۱,۲} = r_۲ = ۱/۳۸۶۳ & M_{۲,۲} = r_۲ = ۱/۳۸۶۳ \\ j=۳ & M_{۲,۳} = r_۳ = -۱/۳۸۶۳ & M_{۴,۳} = r_۳ = -۱/۳۸۶۳ \\ j=۴ & M_{۱,۴} = r_۴ = ۱/۳۸۶۳ & M_{۴,۴} = r_۴ = ۱/۳۸۶۳ \\ j=۵ & M_{۲,۵} = r_۵ = -۱/۳۸۶۳ & M_{۳,۵} = r_۵ = -۱/۳۸۶۳ \\ j=۶ & M_{۳,۶} = r_۶ = -۱/۳۸۶۳ & M_{۴,۶} = r_۶ = -۱/۳۸۶۳ \end{array}$$

برای گام ۱ ، احتمالات خارجی محاسبه می‌شوند. معادله بررسی‌زوجیت اول شامل بیت‌های ۱ و ۲ و ۴ است و بنابراین احتمالات خارجی از راس بررسی ۱ به بیت ۱ به احتمالات بیت‌های ۲ و ۴ وابسته است :

$$\begin{aligned} E_{۱,۱} &= ln\left(\frac{۱+tanh(M_{۱,۲}/۲)tanh(M_{۱,۴}/۲)}{۱-tanh(M_{۱,۲}/۲)tanh(M_{۱,۴}/۲)}\right) \\ &= ln\left(\frac{۱+tanh(۱/۳۸۶۳/۲)tanh(۱/۳۸۶۳/۲)}{۱-tanh(۱/۳۸۶۳/۲)tanh(۱/۳۸۶۳/۲)}\right) \\ &= ln\left(\frac{۱+۰/۶\times۰/۶}{۱-۰/۶\times۰/۶}\right) \\ &= ۰/۷۵۳۸ \end{aligned}$$

به این ترتیب سایر احتمالات خارجی را پس از محاسبه ، داخل ماتریس زیر قرار داده شده اند (در این ماتریس (.) بیایگر این است که LLR برای سطر i و ستون j وجود ندارد و در سایرنقاط عنصر سطر i و ستون j بیانگر $E_{i,j}$ است) :

$$E = \begin{bmatrix} ۰/۷۵۳۸ & -۰/۷۵۳۸ & . & -۰/۷۵۳۸ & . & . \\ . & ۰/۷۵۳۸ & -۰/۷۵۳۸ & . & -۰/۷۵۳۸ & . \\ ۰/۷۵۳۸ & . & . & . & ۰/۷۵۳۸ & ۰/۷۵۳۸ \\ . & . & -۰/۷۵۳۸ & ۰/۷۵۳۸ & . & -۰/۷۵۳۸ \end{bmatrix}$$

حال برای هر بیت (راس متغیر) احتمالات بیرونی و درونی را می‌یابیم. به عنوان مثال بیت اول LLR های بیرونی‌اش را از راس‌های بررسی ۱ و ۳ و LLR درونی‌اش را از کانال دریافت می‌کند ، به این ترتیب LLR کل برای بیت ۱ (با توجه به قوانین عبورپیام که در شکل ۵.۳ بیان شد) :

$$L_۱ = r_۱ + E_{۱,۱} + E_{۳,۱} = -۱/۳۸۶۳ + ۰/۷۵۳۸ + ۰/۷۵۳۸ = ۰/۱۲۱۳$$

براساس علامت LLR کل ، تصمیم سخت روی بیت ۱ نشان می‌دهد که این بیت ۰ است. به همین روش برای بیت‌های ۲ تا ۶ داریم :

$$\begin{aligned} L_۲ &= r_۲ + E_{۱,۲} + E_{۲,۲} = ۱/۳۸۶۳ \\ L_۳ &= r_۳ + E_{۲,۳} + E_{۴,۳} = -۲/۸۹۳۸ \\ L_۴ &= r_۴ + E_{۱,۴} + E_{۴,۴} = ۱/۳۸۶۳ \\ L_۵ &= r_۵ + E_{۲,۵} + E_{۳,۵} = -۱/۳۸۶۳ \\ L_۶ &= r_۶ + E_{۳,۶} + E_{۴,۶} = -۱/۳۸۶۳ \end{aligned}$$

با اجرای تصمیم سخت روی بیت‌ها با توجه به علامت LLR ها داریم :

$$z = [۰\ ۰\ ۱\ ۰\ ۱\ ۱]^T$$

حال به بررسی درستی z می‌پردازیم :

$$s = H \otimes z = \begin{bmatrix} ۱ & ۱ & ۰ & ۱ & ۰ & ۰ \\ ۰ & ۱ & ۱ & ۰ & ۱ & ۰ \\ ۱ & ۰ & ۰ & ۰ & ۱ & ۱ \\ ۰ & ۰ & ۱ & ۱ & ۰ & ۱ \end{bmatrix} [۰\ ۰\ ۱\ ۰\ ۱\ ۱]^T = [۰\ ۰\ ۰\ ۰]^T$$

با توجه به اینکه s صفر است ، بنابراین z کدواژه معتبر (متعلق به کدمورد نظر) است و کدگشا متوقف می‌شود و z به عنوان واژه کدگشایی شده برگردانده می‌شود.

◇

می‌توان پیچیدگی الگوریتم جمع-ضرب را کاهش داد. برای این منظور می توان معادله (۱۱.۴) تعویض کرد.

برای یافتن معادله جایگزین (۱۱.۴) داریم:(در ادامه برای تسهیل کار عبارت جایگزینی

$$\prod_{j'} = \prod_{j' \in B_i, j' \neq j}$$

را در نظر می‌گیریم). علاوه‌براین می‌توان $M_{i,j'}$ را به صورت زیر نوشت :

$$M_{i,j'} = \alpha_{i,j'} \beta_{i,j'}$$

که

$$\alpha_{i,j'} = sign(M_{i,j'}) \ , \ \beta_{i,j'} = |M_{i,j'}|$$

با استفاده از این مفهوم می‌توان نوشت :

$$\prod_{j'} tanh(M_{i,j'}/۲) = \prod_{j'} \alpha_{i,j'} \prod_{j'} tanh(\beta_{i,j'}/۲)$$

بنابراین معادله (۱۱.۴) به صورت زیر تبدیل خواهد شد :

$$\begin{aligned} E_{j,i} = \ & ۲tanh^{-۱}(\prod_{j'} \alpha_{i,j'} \prod_{j'} tanh(\beta_{i,j'}/۲) \ ) \\ = \ & (\prod_{j'} \alpha_{i,j'}) ۲tanh^{-۱}(\ \prod_{j'} tanh(\beta_{i,j'}/۲) \ ) \end{aligned} \qquad (۱۴.۴)$$

با مرتب کردن معادله (۱۴.۴) و تبدیل ضرب به جمع می‌توان آن‌را به صورت زیر نوشت :

$$\begin{aligned} E_{j,i} = \ & (\prod_{j'} \alpha_{i,j'}) ۲tanh^{-۱} ln^{-۱} ln \ (\ \prod_{j'} tanh(\beta_{i,j'}/۲) \ ) \\ = \ & (\prod_{j'} \alpha_{i,j'}) ۲tanh^{-۱} ln^{-۱}(\ \sum_{j'} ln \ tanh(\beta_{i,j'}/۲) \ ) \end{aligned} \qquad (۱۵.۴)$$

حال با تعریف تابع

$$\phi(x) = -ln \ tanh(x/۲) = ln \frac{e^x + ۱}{e^x - ۱}$$

و با توجه به اینکه :

$$\phi(\phi(x)) = ln\frac{e^{\phi(x)}+۱}{e^{\phi(x)}-۱} = x$$

(یعنی $\phi = \phi^{-۱}$) در نهایت معادله (۱۵.۴) را می‌توان به صورت زیر نوشت :

$$E_{i,j} = (\prod_{j'} \alpha_{i,j'})\phi\left(\sum_{j'} \phi(B_{i,j'})\right) \qquad (۱۶.۴)$$

اگر روی هر $M_{i,j'}$ یک تصمیم سخت اتخاذ شود ، ضرب علامت‌ها را می‌توان با استفاده از جمع به پیمانه ۲ این تصمیم‌ها محاسبه کرد و تابع ϕ را نیز می‌توان با استفاده از یک جدول (جستجو) بدست آورد.

۵.۴ تحلیل کدگشای عبور پیام (MPA)

قبل از اینکه به تحلیل کدگشای عبور پیام بپردازیم ، یک تعریف و یک حکم را بیان می‌کنیم.

حکم :[۱۸] فرض کنید $\mathcal{T}$ گراف تنر متناظر با کدخطی دودویی $\mathcal{C}$ باشد که تحت کانال BEC با احتمال پاک‌شدگی δ ارسال شده است و پس از دریافت پیام ، کدگشای عبورپیام روی $\mathcal{T}$ اجرا گردد. اگر $p(\mathcal{T},\delta,l,x)$ احتمال (بلوکی یا بیتی) پاک‌شدگی بعد از تکرار l ام الگوریتم باشد با فرض اینکه $x \in \mathcal{C}$ ارسال شده باشد آنگاه

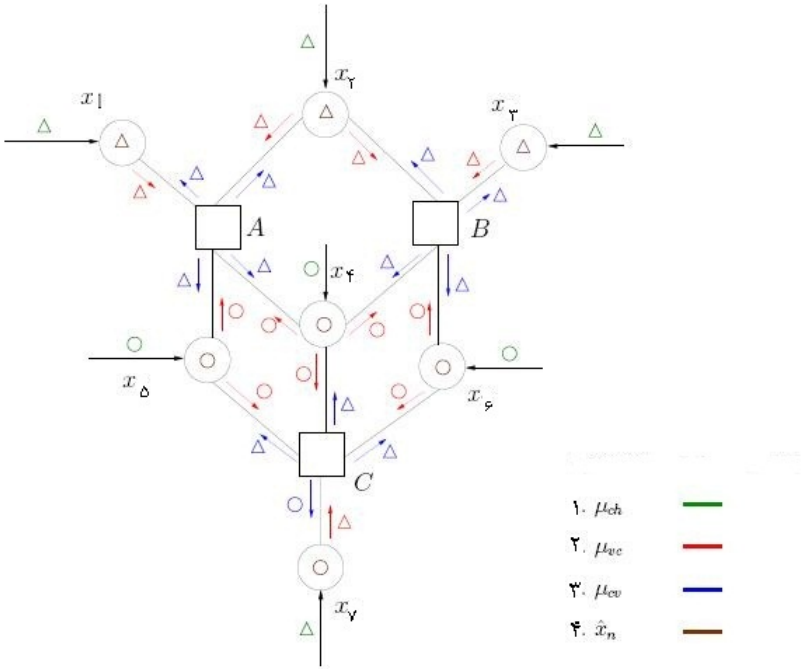
$$p(\mathcal{T},\delta,l,x) = \frac{۱}{|\mathcal{C}|}\sum_{x'\in\mathcal{C}} p(\mathcal{T},\delta,l,x') = p(\mathcal{T},\delta,l)$$

یعنی $p(\mathcal{T},\delta,l,x)$ مستقل از کدواژه ارسالی است.

به این ترتیب برای تحلیل کارایی و اجرای الگوریتم عبورپیام می‌توان فرض کرد که کدواژه ارسالی ، کدواژه صفر است که در هر کد خطی وجود دارد.

تعریف ۱.۵.۴. مجموعه‌ای از نمادهای کد که توسط MPA قابل حل نیستند ، **مجموعه توقف** [۱۰] S نامیده می‌شوند، به عبارت دیگر برای کانال BEC ، اگر راس‌های بررسی (مجاور با S یعنی $\mathcal{N}(S)$) حداقل دو ورودی پاک‌شدگی از راس‌های متغیر متعلق به S (که S زیرمجموعه‌ای از مجموعه اندیس‌های راس‌های متغیر $\mathcal{V}$ است

[۱۰] Stoping set



شکل ۴.۴: مجموعه $S = \{۱, ۲, ۳\}$ نشان‌دهنده مجموعه توقف $[۷, ۴, ۳]$ - کدهمینگ .

) داشته باشند آنگاه پیام خروجی $E_{i,j}$ همیشه پاک‌شدگی خواهد بود. به بیان دقیق‌تر فرض کنید $\varepsilon \subseteq \mathcal{V}$ مجموعه پاک‌شدگی‌های کانال باشد ، بنابراین اگر $S \subseteq \varepsilon$ آنگاه مجموعه S نمادهای کد غیرقابل حل را تشکیل خواهد داد و این نمادها (نمادهای متعلق به S) تحت BEC امکان کدگشایی ندارند .

مثال ۲.۵.۴. [۲۵] کد $\mathcal{C}$ که $[۷, ۴, ۳]$-کد همینگ است را در نظر بگیرید. فرض کنید $x \in \mathcal{C}$ تحت کانال BEC ارسال و واژه دریافتی y توسط MPA کدگشایی شود. یکی از مجموعه‌های توقف $S = \{۱, ۲, ۳\}$ است که در شکل ۴.۴ نشان داده شده است. همان‌طور که می‌بینید $\mathcal{N}(S) = \{A, B\}$ است که هریک بنابر تعریف مجموعه توقف حداقل با دو یال به S متصل شده اند. اگر واژه دریافتی $[? \ ۰ \ ۰ \ ۰ \ ? \ ? \ ?]$ باشد آنگاه مجموعه پاک‌شدگی $\varepsilon = \{۱, ۲, ۳, ۷\}$ است ، بنابراین $S \subseteq \varepsilon$. در شکل ۴.۴ می‌توان دید که S هرگز تحت این شرایط کدگشایی نمی‌شود و فقط $x_۷$ توسط MPA کدگشایی می‌گردد.

$\diamondsuit$

هدف از این بخش تحلیل اجرای کدگشایی عبور پیام و درک این مطلب است که چگونه انتخاب کد LDPC بر این اجرا اثر می‌گذارد. درحقیقت برای گراف تنر معین $\mathcal{T}$ می‌خواهیم بدانیم برای چه سطحی از پارازیت کانال ، کدگشای عبورپیام قادر است خطاها را تصحیح کند (یا برای چه سطحی نمی‌تواند) . متاسفانه این مساله در حالت کلی هنوز حل نشده باقی‌مانده است. ولی آنچه ممکن است تعیین شود ، چگونگی رفتار یک دسته از گراف‌های تنر تحت کانال فاقد حافظه ، با فرض دورآزاد بودن گراف تنر است. برای انجام این کار تکامل تدریجی تابع چگالی احتمال (خطا) در طی اجرای الگوریتم عبور پیام دنبال می‌شود ، که این فرآیند را **تکامل چگالی** [۱۱] می‌نامند. تکامل چگالی می‌تواند برای یافتن ماکزیمم سطح پارازیت کانال که (به طور محتمل‌تر) توسط دسته خاصی از گراف‌های تنر با استفاده از الگوریتم عبور پیام تصحیح می‌شود ، استفاده گردد . این ماکزیمم سطح پارازیت را **آستانه (مرز)** [۱۲] این دسته خاص می‌نامند. به این ترتیب با شناسایی دسته‌ای که بهترین آستانه را دارد می‌توان به طراحی کدی متعلق به این دسته پرداخت (که در این پایان‌نامه به دنبال بررسی آن نخواهیم بود) .

۱.۵.۴ تکامل چگالی روی BEC

با در نظر گرفتن الگوریتم عبور پیام تحت BEC ، فرض می‌کنیم که گراف تنر مورد نظر درخت باشد. در ادامه به بیان تکامل چگالی تحت این کانال برای کدهای LDPC باقاعده و بی قاعده می‌پردازیم.

کدهای LDPC با قاعده

دسته گراف‌های تنر $\mathcal{T}(\omega_c, \omega_r)$ که شامل همه گراف‌های تنر LDPC با قاعده با راس‌ها متغیری از درجه ω_c و راس‌های بررسی از درجه ω_r است را در نظر بگیرید. می‌خواهیم بدانیم کدگشای عبورپیام روی کانال پاک‌شدگی دودویی با استفاده از کدهای این دسته چگونه رفتار می‌کند. فرض کنید q_l احتمال این باشد که در تکرار l ام الگوریتم، یک پیام از **راس بررسی به راس متغیر** ? (پاک‌شدگی) باشد و p_l احتمال این باشد که در تکرار l ام الگوریتم ، یک پیام از **راس متغیر به بررسی** ? (پاک‌شدگی) باشد (یعنی p_l احتمال این است که بیت

[۱۱] Density - evolution
[۱۲] Thershold

کدواژه در تکرار l ام الگوریتم هنوز پاک‌شدگی است). پیام بررسی به متغیر ، روی یک یال $?$ است اگر یک پیام ورودی یا بیشتر، از طریق $(\omega_r - ۱)$ یال دیگر به راس بررسی $?$ باشد. برای محاسبه احتمال q_l فرض می‌کنیم که همه پیام‌های ورودی مستقل از یکدیگر هستند. این فرض از دو فرض زیر حاصل می‌شود:

۱. فاقد حافظه بودن کانال و در نتیجه مستقل بودن احتمالات بیتی اصلی .

۲. هیچ دوری به طول $۲l$ یا کمتر در گراف تنر وجود ندارد. زیرا وجود دور باعث می‌شود که پیام‌ها همبسته شوند.

با این فرض احتمال اینکه هیچ یک از $(\omega_r - ۱)$ پیام ورودی دیگر به راس بررسی پاک‌شدگی $(?)$ نباشد برابر است با ضرب احتمالات $(۱ - p_l)$. بنابراین احتمال اینکه یک یا بیشتر از پیام‌های ورودی دیگر پاک‌شدگی باشند عبارت است از :

$$q_l = ۱ - (۱ - p_l)^{(\omega_r - ۱)} \qquad (۱۷.۴)$$

در تکرار l ام پیام متغیر به بررسی $?$ است اگر پیام اصلی ارسال شده از کانال به متغیر $?$ باشد (که با احتمال δ این حالت رخ می‌دهد) و همه پیام‌های ورودی به متغیر در تکرار $(l - ۱)$ ام پاک‌شدگی باشد (که هر یک با احتمال $q_{l-۱}$ رخ می‌دهد) ، به این ترتیب با فرض مستقل بودن همه پیام‌های ورودی این احتمال برابر است با :

$$p_l = \delta(q_{l-۱})^{(\omega_c - ۱)} \qquad (۱۸.۴)$$

با جایگذاری $q_{l-۱}$ از (۱۷.۴) داریم :

$$p_l = \delta(۱ - (۱ - p_{l-۱})^{(\omega_r - ۱)})^{(\omega_c - ۱)} \qquad (۱۹.۴)$$

اگر $p_۰$ احتمال پاک‌شدگی یک بیت کدواژه توسط کانال باشد، داریم :

$$p_۰ = \delta$$

به این ترتیب برای دسته (ω_c, ω_r)- باقاعده :

$$p_۰ = \delta \ , \ p_l = \delta(۱ - (۱ - p_{l-۱})^{(\omega_r - ۱)})^{(\omega_c - ۱)} \qquad (۲۰.۴)$$

همان‌طور که می‌بینید احتمال پاک‌شدگی در کدگشایی عبورپیام به عنوان تابعی از تعداد تکرار l برای کدهای LDPC (ω_c, ω_r)- باقاعده است. به این ترتیب با محاسبه رابطه بازگشتی (۲۰.۴) می‌توان تعیین کرد که برای چه احتمال‌های پاک‌شدگی ، کدگشای عبورپیام به طور محتمل‌تر پاک‌شدگی را تصحیح می‌کند.

مثال ۳.۵.۴. [۱۷] یک کد از دسته $(۳, ۶)$- باقاعده تحت کانال پاک‌شدگی دودویی با احتمال پاک‌شدگی $\delta = ۰/۳$ ارسال شده است و الگوریتم کدگشا عبور پیام برای کدگشایی مورد استفاده قرار می‌گیرد. احتمال اینکه یک بیت کدواژه بعد از l تکرار کدگشایی عبورپیام (اگر گراف تنر دور آزاد باشد) پاک‌شدگی باقی بماند، با توجه به (۲۰.۴) رابطه بازگشتی زیر بدست می‌آید :

$$p_۰ = ۰/۳ \ , \ p_l = (۱ - (۱ - p_{l-۱})^۵)^۲$$

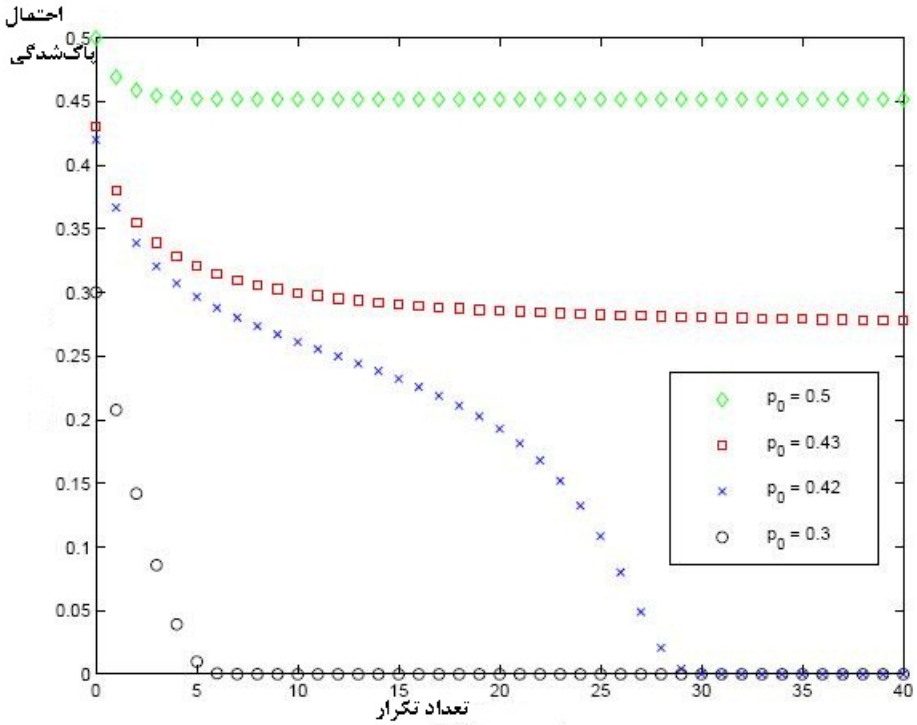
این رابطه بازگشتی را برای $l = ۷$ محاسبه می‌کنیم ، داریم :

$$\begin{array}{lll} p_۰ = ۰/۳۰۰۰ & p_۱ = ۰/۲۰۷۶ & p_۲ = ۰/۱۴۱۹ \\ p_۳ = ۰/۰۸۵۸ & p_۴ = ۰/۰۳۹۲ & p_۵ = ۰/۰۰۹۸ \\ p_۶ = ۰/۰۰۰۷ & p_۷ = ۰/۰۰۰۰ & \end{array}$$

بنابراین در یک کدواژه از کد LDPC $(۳, ۶)$- باقاعده ۴ - دور آزاد که تحت کانال BEC با احتمال پاک‌شدگی $\delta = ۰/۳$ ارسال شده است بعد از $l = ۷$ تکرار الگوریتم عبورپیام احتمال پاک‌شدگی ، صفر می‌شود.

$\diamondsuit$

مثال ۴.۵.۴. [۱۷] در شکل ۵.۴ برای محدوده‌ای از احتمال‌های پاک‌شدگی کانال ، رابطه بازگشتی (۲۰.۴) برای یک کد متعلق به دسته $(۳،۶)$- باقاعده محاسبه شده است . همانطور که می‌بینیم هنگامی که $\delta \geq ۰/۴۳$ احتمال اینکه پاک‌شدگی (بعد از تعدادی تکرار) باقی بماند به صفر کاهش نمی‌یابد حتی اگر l خیلی بزرگ باشد.



شکل ۵.۴: احتمالات پاک‌شدگی محاسبه شده در مثال (۳.۵.۴)

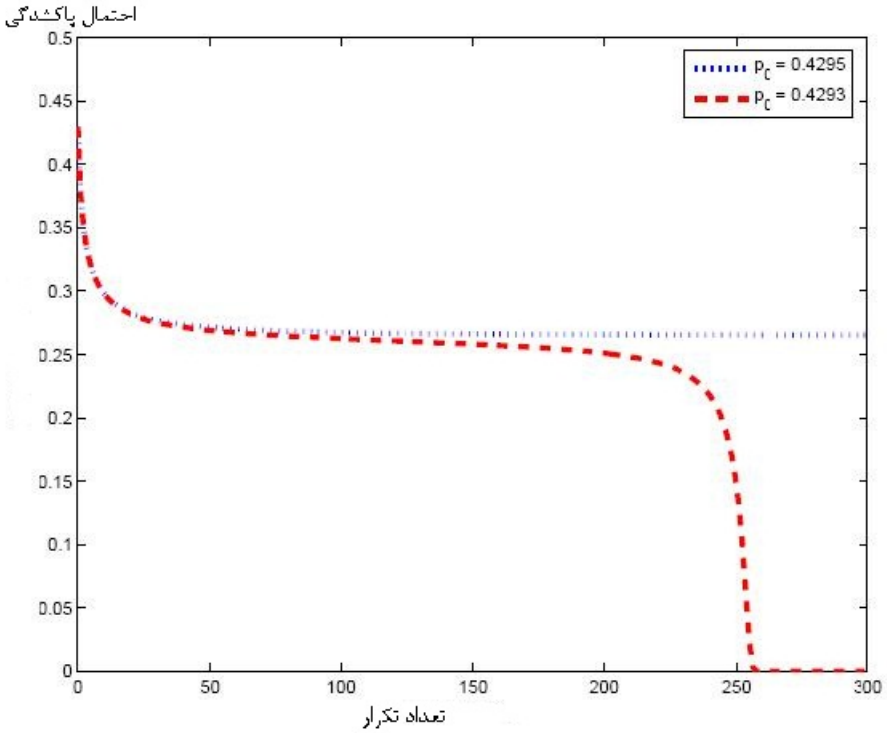
در حالی که برای $\delta \leq$ ۰/۴۲ وقتی $l \to \infty$ احتمال پاک‌شدگی به صفر میل می‌کند. مقدار δ ای که بین این دو مقدار به دست می‌آید را ، آستانه (مرز) دسته $(۳,۶)$ - باقاعده می‌نامند. با محاسبه مجدد رابطه بازگشتی (۲۰.۴) همانطور که در شکل ۶.۴ نشان داده شده است می‌توان دید که آستانه دسته (۳.۶) - باقاعده تحت کانالBEC بین ۰/۴۲۹۳ و ۰/۴۲۹۴ است.

$\diamondsuit$

کدهای LDPC بی قاعده

همانطور که قبلا نیز بیان شد برای بررسی تکامل چگالی کدهای LDPC بی‌قاعده از توزیع‌های درجه برحسب یال‌های گراف تنر استفاده می‌شود. براساس تعریف (۷.۳.۳) داریم :

$$\sum_i \rho_i = ۱ \quad , \quad \sum_i \lambda_i = ۱$$



شکل ۶.۴: احتمالات پاک‌شدگی محاسبه شده در مثال (۴.۵.۴)

یادآوری می‌کنیم که :

$$\begin{aligned}\lambda(x) &= \lambda_۲x + \lambda_۳x^۲ + \ldots + \lambda_i x^{i-۱} + \ldots \\ \rho(x) &= \rho_۲x + \rho_۳x^۲ + \ldots + \rho_i x^{i-۱} + \ldots\end{aligned}$$

از (۱۷.۴) می‌دانیم که در تکرار l ام کدگشای عبورپیام احتمال اینکه پیام بررسی به متغیر ؟ باشد (بافرض مستقل بودن پیاهای ورودی) عبارت است از :

$$q_l = ۱ - (۱ - p_l)^{(\omega_r - ۱)}$$

(برای یک یال متصل به راس بررسی از درجه ω_r) . همانطور که در تعریف (۷.۳.۳) نیز بیان شد ، برای یک گراف تنر بی‌قاعده احتمال اینکه یک یال به راس بررسی درجه ω_r متصل شده باشد ρ_{ω_r} است. بنابراین با محاسبه میانگین تحت همه یال‌های متعلق به گراف تنر بی‌قاعده، متوسط احتمال اینکه پیام بررسی به متغیر خطا (

پاک‌شدگی) باشد :

$$q_l = \sum_i \rho_i(۱ - (۱ - p_l)^{(i-۱)}) = ۱ - \sum_i \rho_i(۱ - p_l)^{(i-۱)} \qquad (۲۱.۴)$$

با استفاده از تعریف $\rho(x)$ داریم :

$$q_l = ۱ - \rho\,(۱ - p_l)$$

از (۱۹.۴) می‌دانیم که : احتمال اینکه در تکرار l ام الگوریتم یک پیام متغیر به بررسی ? باشد (با فرض مستقل بودن پیام‌های ورودی) عبارت است از :

$$p_l = \delta(۱ - (۱ - p_{l-۱})^{(\omega_r-۱)})^{(\omega_c-۱)}$$

(برای یک یال که به راس متغیر درجه ω_c متصل است) . برای گراف تنر بی‌قاعده احتمال اینکه یک یال به راس متغیر درجه ω_c متصل باشد λ_{ω_c} است. به این ترتیب با محاسبه میانگین تحت همه یالهای گراف تنر احتمال متوسط اینکه یک پیام متغیر به بررسی پاک‌شدگی باشد :

$$p_l = \delta \sum_i \lambda_i (q_{l-۱})^{(i-۱)} \qquad (۲۲.۴)$$

در نهایت با جایگذاری $q_{l-۱}$ داریم :

$$p_l = \delta\lambda\,(۱ - \rho\,(۱ - p_{l-۱})\,)$$

با فرض $p_۰ = \delta$ ، در نتیجه برای کدهای LDPC بی‌قاعده داریم :

$$p_۰ = \delta\ ,\ p_l = p_۰\lambda\,(۱ - \rho\,(۱ - p_{l-۱})\,) \qquad (۲۳.۴)$$

۲.۵.۴ آستانه

هدف از تکامل چگالی، تعیین این مطلب است که برای چه احتمال پاک‌شدگی کانال (δ) کدگشای عبورپیام به طور محتمل‌تر همه بیت‌های پاک‌شدگی را تصحیح می‌کند. برای آزمایش اثر δ روی p_l تابع زیر تعریف می‌شود :

$$f(p,\delta) = \delta\lambda(۱ - \rho(۱ - p)\ )$$

احتمال پاک‌شدگی در تکرار l عبارت است از :

$$p_l = f(p_{l-۱}, \delta)$$

که p و δ احتمالات هستند بنابراین $۰ \leq \delta, p \leq ۱$ در اینجا تابع $f(p,\delta)$ برای $\delta > ۰$ در p اکیدا صعودی است. بنابراین اگر $p_l > p_{l-۱}$ آنگاه

$$p_{l+۱}(\delta) = f(p_l, \delta) \geqslant f(p_{l+۱}, \delta) = p_l$$

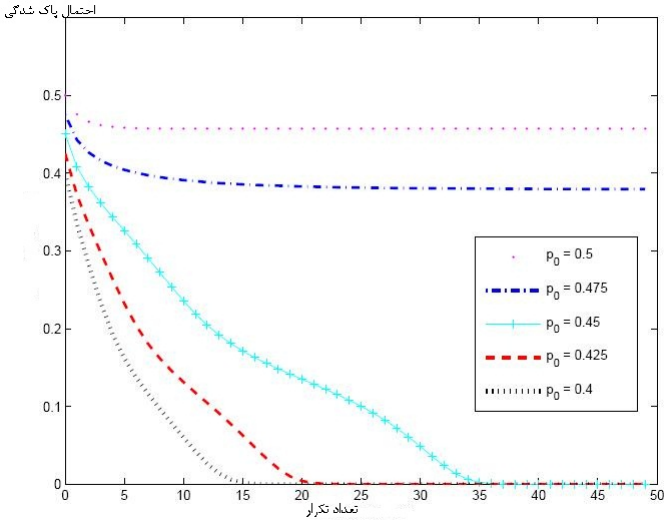
برای $\delta \in [۰, ۱]$ ، $p_l(\delta)$ در $p = ۰$ کران پایین دارد :

$$f(۰, \delta) = \delta\lambda(۱ - \rho(۱)) = \delta\lambda(۱ - ۱) = ۰$$

و در $p = ۱$ کران بالا دارد :

$$f(۱, \delta) = \delta\lambda(۱ - \rho(۱ - ۱)) = \delta\lambda(۱ - ۰) = \delta$$

چون $f(p,\delta)$ در p تابع اکیدا صعودی است و $۰ \leqslant f(p,\delta) \leqslant \delta$ بنابراین به ازای $p, \delta \in [۰, ۱]$ هنگامی که $l \to \infty$ ، p_l به عنصری متعلق به $p_\infty \in [۰, \delta]$ میل می‌کند. به این ترتیب برای جفت توزیع (λ, ρ) ، وجود دارد $\delta^* \in [۰, ۱]$ به طوری که هنگامی که تعداد تکرارها به سمت بی‌نهایت میل می‌کند $p_l(\delta^*) \to ۰$ علاوه‌براین به ازای $\delta < \delta^*$ نیز هنگامی که تعداد تکرارها به سمت بی‌نهایت میل می‌کند $p_l(\delta) \to ۰$ به این ترتیب δ^* را آستانه



شکل ۷.۴: احتمالات پاک‌شدگی محاسبه شده در مثال (۵.۵.۴)

می‌گویند. در حقیقت :

$$\delta^*(\lambda,\rho) = sup\left\{\delta \in [۰,۱] : \begin{array}{c} p_l(\delta) \to \\ l \to \infty \end{array} ۰ \right\}$$

مثال ۵.۵.۴. [۱۷] دسته LDPC بی‌قاعده با توزیع‌های درجه زیر را در نظر بگیرید :

$$\lambda(x) = ۰/۱x + ۰/۴x^{۲} + ۰/۵x^{۱۹}$$

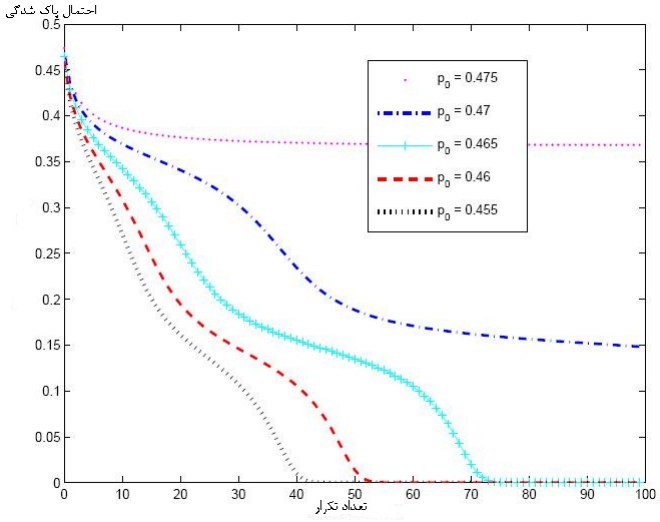
و

$$\rho(x) = ۰/۵x^{۷} + ۰/۵x^{۸}$$

نرخ این کد :

$$r(\lambda,\rho) = ۱ - \frac{\int_۰^۱ \rho(x)d(x)}{\int_۰^۱ \lambda(x)d(x)} = ۱ - \frac{\sum_i \rho_i/i}{\sum_i \lambda_i/i} \approx ۰/۵$$

برای یافتن آستانه، رابطه بازگشتی (۲۳.۴) را برای محدوده‌ای از احتمالات متفاوت پاک‌شدگی کانال محاسبه می‌شود. همانطورکه در شکل ۷.۴ می‌بینید برای $\delta \geq ۰/۵$ احتمال اینکه پاک‌شدگی باقی مانده‌باشد به صفر کاهش



شکل ۸.۴: احتمالات پاک‌شدگی محاسبه شده در مثال (۵.۵.۴) (بررسی دوباره)

نمی‌یابد (حتی هنگامی که l خیلی بزرگ باشد ، در حالی که برای $\delta \leq ۰/۴$ احتمال خطا به صفر می‌رسد. به این ترتیب برای نزدیک شدن به آستانه، رابطه بازگشتی (۲۳.۴) با احتمالات پاک‌شدگی کانال بین این دو مقدار محاسبه می‌شود. شکل ۷.۴ نشان می‌دهد که برای $\delta \geq ۰/۴۷۵$ احتمال اینکه پاک‌شدگی باقی بماند حتی اگر l خیلی بزرگ باشد به صفر کاهش نمی‌یابد ، در حالی که برای مقادیر $\delta \leq ۰/۴۵$ احتمال خطا به صفر میل می‌کند. به این ترتیب دوباره رابطه بازگشتی مورد نظر برای مقداری بین این دو مقدار جدید محاسبه می‌شود. همانطور که در شکل ۸.۴ می‌بینید برای $\delta \geq ۰/۴۷۵$ احتمال اینکه پاک‌شدگی باقی بماند حتی اگر l خیلی بزرگ باشد به صفر کاهش نمی‌یابد ، در حالی که برای مقادیر

$$\delta \leq ۰/۴۶۵$$

احتمال خطا به صفر میل می‌کند. بنابراین نتیجه می‌گیریم که آستانه برای این دسته ، احتمال پاک‌شدگی بین $۰/۴۶۵$ و $۰/۴۷۵$ است.

$\diamondsuit$

۳.۵.۴ پایداری

با توجه به رابطه (۲۰.۴) می‌توان نتیجه گرفت که در چندجمله‌ای‌هایی با درجه خیلی بالا تعداد تکرار افزایش می‌یابد. برای درک رفتار چندجمله‌ای وقتی p_l کوچک است می‌توان آن‌را توسط بسط تیلور حول صفر تقریب کرد :

$$p_l = f(p_{l-۱}, \delta) \approx f'(p, \delta)\, p_{l-۱} \qquad (۲۴.۴)$$

برای

$$f(x) = g(h(x))$$

داریم :

$$\frac{df}{dx} = \frac{dg}{dh} \times \frac{dh}{dx}$$

بنابراین :

$$f(p, \delta) = \delta\lambda\,(h(p)\,)$$

که

$$h(p) = ۱ - \rho\,(۱ - p)$$

با مشتقگیری نسبت به p داریم :

$$\frac{df(p, \delta)}{dp} = \delta \frac{d\,\lambda}{d\,h} \times \frac{d\,h}{d\,p}$$

با محاسبه مقدار مشتق در $p = ۰$ داریم :

$$h(p = ۰) = ۱ - \rho(۱) = ۰$$

$$\frac{d\,\lambda}{d\,h}|_{p=۰} = \frac{d\,\lambda}{d\,h}|_{h=۰} = (\lambda_۲ + ۲\lambda_۳ h + \ldots + (i-۱)\lambda_i h^{(i-۲)} + \ldots)|_{h=۰} = \lambda_۲$$

و

$$\frac{d\,h}{d\,p}|_{p=۰} = \frac{d(۱-\rho\,(۱-p)}{d\,p}|_{(۱-p)=۱} = \rho'(۱)$$

با جایگذاری در (۲۴.۴) داریم :

$$p_l \approx \delta\lambda_۲\rho'(۱)p_{l-۱} \qquad (۲۵.۴)$$

هنگامی‌که $p_l \to ۰$. هنگامی‌که $l \to \infty$ برای اینکه p_l به صفر میل کند باید : $p_l < p_{l-۱}$ (باتوجه به اینکه p_l احتمال است پس باید $p_l < p_{l-۱} \leq ۱$) و بنابراین از (۲۵.۴) داریم :

$$\delta\lambda_۲\rho' < ۱ \qquad (۲۶.۴)$$

بنابراین برای جفت توزیع درجه (λ, ρ) تحت کانال پاک‌شدگی با احتمال پاک‌شدگی δ برای اینکه احتمال پاک‌شدگی به صفر میل کند $\lambda_۲$ باید کران بالایی به صورت (۲۷.۴) داشته باشد :

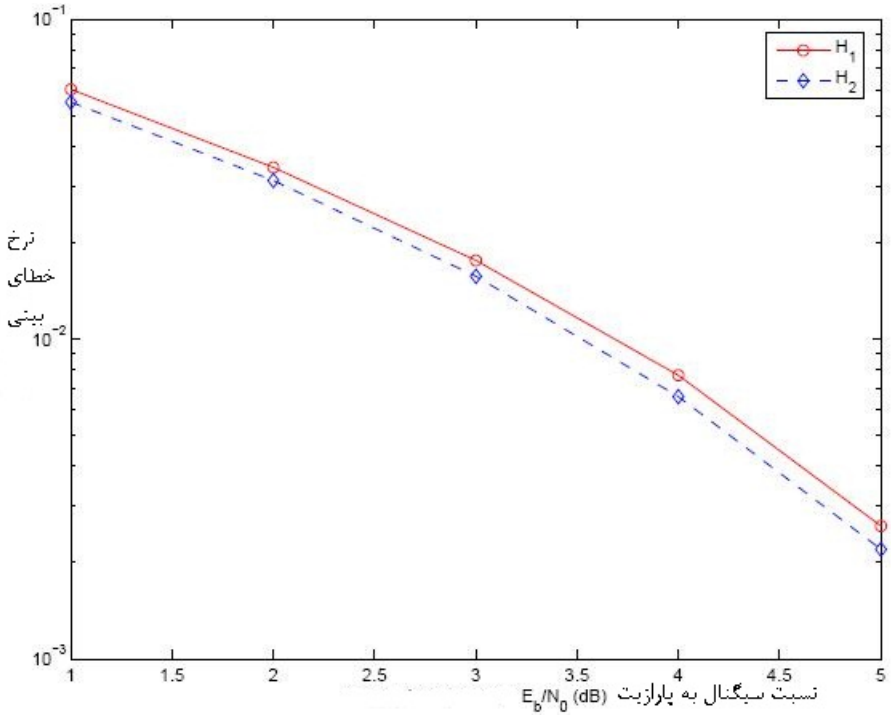
$$\lambda_۲ < \frac{۱}{\delta\rho'(۱)} \qquad (۲۷.۴)$$

نامعادله (۲۷.۴) **محدودیت پایداری**[۱۳] نامیده می‌شود.

۴.۵.۴ کمر

همانطور که بیان شد وجود دور در گراف تنر موجب همبستگی احتمال‌های حاشیه‌ای (پیام‌ها)می‌گردد. به این ترتیب وجود دور در گراف تنر روی همگرایی کدگشا (به جواب) اثر می‌گذارد و کوچکترین دور که **کمر**[۱۴] نامیده می‌شود ، بیشترین اثر را دارد. یک نکته بسیار مهم در مورد خواص کد LDPC این است که اغلب خواص یک کد با انتخاب ماتریس بررسی‌زوجیت متفاوت ، تغییر می‌کند. به مثال بعد توجه کنید.

[۱۳] Stability constraint
[۱۴] Girth



شکل ۹.۴: نرخ خطای بیتی اجرای کدگشای جمع - ضرب تحت کانال AWGN با استفاده از ماتریس بررسی‌زوجیت مثال (۶.۵.۴)

مثال ۶.۵.۴. [۱۷] دو ماتریس بررسی‌زوجیت

$$H_۱ = \begin{bmatrix} ۱ & ۱ & ۰ & ۱ & ۰ & ۰ \\ ۰ & ۱ & ۱ & ۰ & ۱ & ۰ \\ ۱ & ۰ & ۰ & ۰ & ۱ & ۱ \\ ۱ & ۰ & ۱ & ۱ & ۱ & ۰ \end{bmatrix}$$

$$H_۲ = \begin{bmatrix} ۱ & ۱ & ۰ & ۱ & ۰ & ۰ \\ ۰ & ۱ & ۱ & ۰ & ۱ & ۰ \\ ۱ & ۰ & ۰ & ۰ & ۱ & ۱ \\ ۰ & ۰ & ۱ & ۱ & ۰ & ۱ \end{bmatrix}$$

هر دو یک کد را شرح می‌دهند اما گراف‌های تنر آنها کمرهای متفاوتی دارد. شکل ۹.۴ اجرای کدگشای جمع-ضرب را برای هر یک از این دو ماتریس بررسی‌زوجیت نشان می‌دهد. حذف تک دور به طول ۴ (۴-دور) از گراف تنر حاصل از ماتریس بررسی‌زوجیت (در گراف تنر متناظر با $H_۲$ دور به طول ۴ که در گراف تنر متناظر با $H_۱$ وجود دارد، حذف می‌شود) اجرای کدگشای جمع-ضرب را به طور قابل ملاحظه‌ای کاهش می‌دهد.

$\diamondsuit$

هرچند فاقد دور بودن گراف تنر موجب عملکرد بهینه برخی از الگوریتم‌های کدگشایی می‌شود ولی با توجه به لمی که در ادامه مطرح می‌شود می‌توان دید که این کدها (که گراف تنر متناظر با آن فاقد دور است) به اندازه کافی توانا نیستند.

لم ۷.۵.۴. [۱۸] فرض کنید $\mathcal{C}$ یک کد خطی دودویی با نرخ r است که گراف تنر متناظر با آن جنگل باشد. آنگاه $\mathcal{C}$ شامل حداقل n ($\frac{۲r-۱}{۲}$) کدواژه به وزن ۲ است.

با توجه به اینکه عملکرد تصحیح خطای یک کد به مینیمم فاصله همینگ وابسته است به این ترتیب وجود کدواژه‌های به وزن ۲ به خصوص اگر تعداد آنها زیاد باشد تاثیر نامطلوبی در عملکرد کد خواهد گذاشت.

۵.۵.۴ کدواژه‌های کاذب

همانطور که پیش از این بیان شد، یک دلیل پایین بودن پیچیدگی الگوریتم عبور پیام ، عملکرد محلی الگوریتم روی گراف تنر متناظر با ماتریس H است (یعنی هر راس در الگوریتم کدگشایی فقط روی پیام‌هایی که از آن راس می‌بیند عمل می‌کند نه روی دیگر پیام‌های متعلق به گراف) . از طرفی این عملکرد همچنین موجب یک ضعف اساسی در الگوریتم می‌گردد : چون الگوریتم به طور محلی عمل می‌کند ، هر راس تمام ساختار گراف را نمی‌داند. برای شرح بیشتر به ادامه مطالب توجه کنید، یک گراف جایگزین (برای گراف تنر) که (به طور محلی) LLR هایی مشابه با LLR های گراف تنر تولید می‌کند را گراف **پوشش** [۱۵] می‌نامند. مجموعه کدواژه‌های معتبر متعلق به همه گراف‌های پوشش یک گراف تنر را **کدواژه‌های کاذب** [۱۶] می‌نامند. چون کدگشای عبورپیام به طور محلی نمی‌تواند بین گراف تنر واقعی و گراف‌های پوشش فرق بگذارد ، کدواژه متعلق به هر یک از گراف‌های پوشش را مشابه کدواژه متعلق به گراف تنر می‌داند. بنابراین وقتی می‌گوییم کدگشا شکست می‌خورد در حقیقت به یکی از کدواژه‌های کاذب همگرا است.

[۱۵] Cover
[۱۶] Pseudo-codewords

فصل ۵

کدگشایی کدهای LDPC با استفاده از الگوریتم کدگشایی LP

۱.۵ مقدمه فصل

فلدمن و همکارانش [۱] از برنامه‌ریزی خطی (LP) یا به بیان دقیق‌تر از **برنامه ریزی خطی کاهشی** [۲] برای ساختن یک الگوریتم کدگشایی استفاده کردند. آنها با ارائه مقالاتی این روش کدگشایی را برای کدهای خطی ، همچنین کدهای توربو و کدهای LDPC شرح داده‌اند و به تحلیل کدگشایی LP پرداخته‌اند. از آن پس کارهای زیادی در این زمینه انجام شده است.

استفاده از کدگشایی LP برای کدهای LDPC در سال‌های اخیر به دلیل عملکردی نزدیک به عملکرد الگوریتم‌های عبورپیام و همچنین توانایی تحلیل طول بلوک- متناهی افزایش یافته است. اساس مطالبی که در این فصل بیان می شود بر مبنای کارهای فلدمن و همکارانش است.

آنچه در این فصل خواهیم دید:

در بخش ۲.۵ پیش نیازهایی از برنامه‌ریزی خطی از جمله تعاریف و قضایای مورد نیاز مطرح می‌شود. در بخش ۳.۵ برنامه‌ریزی خطی کاهشی برای کدگشایی فرمول‌بندی می‌شود. در ۴.۵ تفسیر هندسی مورد نیاز بیان می‌شود. در ۵.۵ یک روش جایگزینی برای تعریف محدودیت‌های LP مطرح می‌شود. در ۶.۵ دو روش بهبود برای عملکرد

[۱] Feldman et.al.
[۲] Linear programming relaxation

کدگشایی LP مطرح می‌شود.

۲.۵ پیش نیاز ها

در این بخش به یادآوری برخی مفاهیم مورد نیاز در برنامه‌ریزی‌خطی می‌پردازیم. می‌دانیم در هر برنامه‌ریزی‌خطی مانند برنامه‌ریزی خطی زیر، هر بردار x که در همه محدودیت‌های برنامه‌ریزی‌خطی صدق کند، جواب شدنی نامیده می‌شود.

$$\begin{aligned} \min. \quad & C^T x \\ s.t. \quad & A x = b \\ & x \geq ۰ \end{aligned}$$

مجموعه همه جواب‌های شدنی را مجموعه شدنی می‌نامند (منظور از A در برنامه‌ریزی خطی استاندارد بالا ماتریسی $m \times n$ است) . جواب شدنی x^* که تابع هدف را نیز مینیمم می‌کند، جواب شدنی بهینه یا به طور ساده جواب بهینه می‌نامند. با توجه به اینکه یک نیم‌فضا مجموعه‌ای از نقاط به صورت $\{x : px \geq k\}$ است که در آن p یک بردار ناصفر در فضای اقلیدسی n بعدی (E^n) و k یک اسکالر است، اشتراک تعداد متناهی نیم‌فضا (متعلق به $\mathcal{R}^n$) را **چندوجهی** [۳] و یک چندوجهی کراندار را **چندسقفی** [۴] می‌نامند. مجموعه X را در فضای F^n محدب می‌گویند اگر به ازای هر دو نقطه مفروض $x_۱$ و $x_۲$ در X و به ازای هر $\lambda \in [۰ \ , \ ۱]$ داشته باشیم $\lambda x_۱ + \lambda x_۲ \in X$ (یادآوری می‌کنیم که هر نقطه به شکل $\lambda x_۱ + \lambda x_۲$ را ترکیب محدب $x_۱$ و $x_۲$ می‌گویند و اگر $\lambda \in (۰, ۱)$ باشد آنگاه $\lambda x_۱ + \lambda x_۲$ را ترکیب محدب اکید $x_۱, x_۲$ می‌نامند) . کوچکترین مجموعه محدب شامل مجموعه دلخواه A را پوسته محدب A گویند.

تعریف ۱.۲.۵. نقطه x متعلق به مجموعه محدب X را نقطه راسی گویند ، اگر نتوان x را به صورت ترکیب محدب اکید دو نقطه متمایز متعلق به X نوشت.

[۳] Polyhedron
[۴] Polytope

قضیه ۲.۲.۵. [۶] فرض کنید $\mathcal{M}$ یک چندوجهی ناتهی باشد. اگر بردار ضرایب محدودیت‌هایی که $\mathcal{M}$ را تعریف می‌کنند گویا باشند ، آنگاه هر نقطه راسی $\mathcal{M}$ یک بردار گویا است .

قضیه ۳.۲.۵. [۶] اگر در یک برنامه‌ریزی خطی جواب بهینه وجود داشته باشد، آن‌گاه یک نقطه راسی بهینه وجود خواهد داشت.

۳.۵ فرمولبندی برنامه‌ریزی‌خطی کاهشی برای کدگشایی

با توجه به ماهیت کدواژه‌های دودویی که مؤلفه‌های آنها متعلق به $\{۰, ۱\}$ است، منطقی به نظر می‌رسد که برنامه‌ریزی‌خطی مورد نیاز، برنامه‌ریزی‌خطی عدد صحیح (ILP) باشد. اما مساله ILP حاصل متاسفانه در کل NP-hard است [۱۱] . یک راه حل طبیعی برای برطرف کردن این مشکل، یافتن جوابی تقریبی برای یک ILP است. لذا با حذف محدودیت‌های عدد صحیح ، ابتدا مساله LP حاصل حل می‌گردد ، سپس جواب حاصل به آنچه مورد نظر است تبدیل می‌شود. یک روش حل مساله ILP استفاده از **برنامه‌ریزی‌خطی کاهشی** [۵] است.

۱.۳.۵ کدگشایی ماکزیمم درستنمایی برای LP

کدخطی دودویی $\mathcal{C} \in F_۲^n$ با ماتریس بررسی‌زوجیت $H \epsilon F_۲^{m \times n}$ را در نظر بگیرید.فرض کنید کدواژه $x = \begin{bmatrix} x_۱ & \cdots & x_n \end{bmatrix}^T$ تحت کانال فاقدحافظه و پارازیت‌دار ارسال شده است و $y = \begin{bmatrix} y_۱ & \cdots & y_n \end{bmatrix}^T$ بردار دریافتی و $\lambda = \begin{bmatrix} \lambda_۱ & \cdots & \lambda_n \end{bmatrix}^T$ بردار لگاریتم نسبت درستنمایی LLR باشد. (یادآوری : لگاریتم نسبت درستنمایی LLR $\lambda_i = \ln\left(\frac{P(y_i|x_i=۰)}{P(y_i|x_i=۱)}\right)$) حال باید y کدگشایی شود. یک راه کدگشایی انتخاب کدواژه‌ای است که ماکزیمم احتمال درستنمایی (محتمل‌ترین کدواژه) را دارد. به همین دلیل به آن، **کدواژه ماکزیمم درستنمایی** [۶] (ML) می‌گویند. در این بخش تابع هدف برنامه‌ریزی خطی مورد نظر براساس کدگشایی

[۵] LP relaxation
[۶] Maximmum likelihood codeworde

ماکزیمم درستنمایی بدست می آید.

$$\hat{x} = \underset{x \in \mathcal{C}}{\text{argmax}} \quad P(y|x) \tag{۱.۵}$$

چون کانال بی‌حافظه است، متغیرها مستقل‌اند لذا:

$$\begin{aligned} \hat{x} &= \underset{x \in \mathcal{C}}{\text{argmax}} \quad \prod_{i=۱}^{n} P(y_i|x_i) \\ &= \underset{x \in \mathcal{C}}{\text{argmax}} \quad \ln \prod_{i=۱}^{n} P(y_i|x_i) \\ &= \underset{x \in \mathcal{C}}{\text{argmin}} \quad (-\ln \prod_{i=۱}^{n} P(y_i|x_i)) \\ &= \underset{x \in \mathcal{C}}{\text{argmin}} \quad (-\sum_{i=۱}^{n} \ln P(y_i|x_i)) \end{aligned} \tag{۲.۵}$$

عبارت $\sum_{i=۱}^{n} \ln P(y_i|۰)$ مستقل از x است ، چون مینیمم‌سازی تحت x انجام می‌شود لذا این عبارت را می‌توان به عنوان ثابت به سطر آخر (۲.۵) اضافه کرد:

$$\begin{aligned} \hat{x} &= \underset{x \in \mathcal{C}}{\text{argmin}} \ (\sum_{i=۱}^{n} \ln P(y_i|۰) - \sum_{i=۱}^{n} \ln P(y_i|x_i)) \\ &= \underset{x \in \mathcal{C}}{\text{argmin}} \ \sum_{i=۱}^{n} \ln \frac{P(y_i|۰)}{P(y_i|x_i)} \end{aligned} \tag{۳.۵}$$

عبارت $\sum_{i=۱}^{n} \ln \frac{P(y_i|۰)}{P(y_i|x_i)}$ وقتی $x_i = ۰$ ، مساوی با ۰ (صفر) و هنگامی که $x_i = ۱$ به صورت زیر است:

$$\sum_{i=۱}^{n} \ln \frac{P(y_i|۰)}{P(y_i|x_i)} = \sum_{i=۱}^{n} \lambda_i$$

بنابراین (۳.۵) را می توان به صورت زیر نوشت :

$$\begin{aligned} \hat{x} &= \underset{x \in \mathcal{C}}{\operatorname{argmin}} \sum_{i=1}^{n} x_i \cdot (\ln \frac{P(y_i|0)}{P(y_i|x_i)}) \\ &= \underset{x \in \mathcal{C}}{\operatorname{argmin}} \sum_{i=1}^{n} \lambda_i x_i \\ &= \underset{x \in \mathcal{C}}{\operatorname{argmin}} \lambda^T x \end{aligned} \qquad (۴.۵)$$

معادله (۴.۵) تابع هدف مورد نظر و $\hat{x}$ ، کدواژه ماکزیمم درستنمایی است. به این ترتیب مساله بهینه‌سازی مورد نظر به صورت زیر خواهد بود:

$$\begin{aligned} \min. \quad & \lambda^T x \\ s.t. \quad & x \in \mathcal{C} \end{aligned} \qquad (۵.۵)$$

(به این ترتیب تابع هدف مساله ، $\lambda^T x$ است که می توان λ را برای حالتی که کانال BSC با احتمال همگذری p است به این صورت تفسیر کرد: اگر بیت دریافتی $y_i = 1$ باشد پس $\lambda_i = \ln\left(\frac{p}{1-p}\right)$ ، در این حالت λ_i هزینه کدگشایی $\hat{y}_i = 1$ است).

می‌دانیم اگر $x \in \mathcal{C}$ آنگاه $H \otimes x = \mathbf{0}$ بنابراین از (۵.۵) داریم:

$$\begin{aligned} \min. \quad & \lambda^T x \\ s.t. \quad & H \otimes x = \mathbf{0} \\ & x_j \in \{0, 1\} \end{aligned} \qquad (۶.۵)$$

محدودیت‌ها و متغیرهای مساله بهینه‌سازی (۶.۵) عدد صحیح‌اند. با توجه به مطالب گفته شده به منظور استفاده از برنامه‌ریزی‌خطی کاهشی ، محدودیت $x_j \in \{0, 1\}$ با $x_j \in [0, 1]$ تعویض می‌شود . علاوه‌براین باید معادلات بررسی‌زوجیت که تحت F_2 انجام می‌شوند به صورت معادلات خطی تحت $\mathbb{R}$ بازنویسی شوند. مراحل انجام این کار، ابتدا در قالب مثال زیر بیان خواهد شد.

مثال ۱.۳.۵. [۲۵] ماتریس بررسی‌زوجیت زیر را در نظر بگیرید:

$$H = \begin{bmatrix} ۱ & ۰ & ۰ & ۱ & ۱ & ۰ \\ ۰ & ۱ & ۰ & ۱ & ۰ & ۱ \\ ۰ & ۰ & ۱ & ۰ & ۱ & ۱ \end{bmatrix}$$

اگر $x = \begin{bmatrix} x_۱ & \dots & x_۶ \end{bmatrix}^T$ ، معادلات بررسی‌زوجیت متناظر با H به صورت زیر خواهد بود:

$$H \otimes x = \left\{ \begin{array}{lcl} r(۱) & : & x_۱ \oplus x_۴ \oplus x_۵ = ۰ \\ r(۲) & : & x_۲ \oplus x_۴ \oplus x_۶ = ۰ \\ r(۳) & : & x_۳ \oplus x_۵ \oplus x_۶ = ۰ \end{array} \right\} \qquad (۷.۵)$$

که منظور از $r(i)$ ، $(i = ۱, ۲, ۳)$ معادله متناظر با سطر i ام ماتریس H است. حال برای اینکه معادلات بررسی‌زوجیت تحت $\mathbb{R}$ بیان شود، نیازمند فرمول‌بندی و متغیرهای جدیدی هستیم. ابتدا این فرمول‌بندی جدید ، برای معادله متناظر با سطر اول یعنی $r(۱)$ بیان می‌شود:

- $\mathbb{S}_۱$ را مجموعه اندیس‌های نمادهای کد که در معادله بررسی‌زوجیت $r(۱)$ استفاده شده‌اند در نظر بگیرید، بنابراین

$$\mathbb{S}_۱ = \{۱, ۴, ۵\}$$

- $x_{\mathbb{S}_۱}$ **کدواژه محلی** [۷] نامیده می شود که متناظر با نمادهای کد مورد استفاده در معادله بررسی‌زوجیت $r(۱)$ است یعنی:

$$x_{\mathbb{S}_۱} = \begin{bmatrix} x_۱ & x_۴ & x_۵ \end{bmatrix}^T$$

ماتریسی که $x_{\mathbb{S}_۱}$ را از x استخراج می‌کند $B_۱$ می‌نامیم. بنابراین داریم :

$$B_۱ = \begin{bmatrix} ۱ & ۰ & ۰ & ۰ & ۰ & ۰ \\ ۰ & ۰ & ۰ & ۱ & ۰ & ۰ \\ ۰ & ۰ & ۰ & ۰ & ۱ & ۰ \end{bmatrix}$$

پس :

$$x_{\mathbb{S}_۱} = B_۱ x \qquad (۸.۵)$$

توجه کنید که در معادله (۸.۵) جمع و ضرب تحت $\mathbb{R}$ انجام می‌شود.

[۷] Local codeword

- حال ماتریس $A_۱$ به صورت زیر تعریف می‌شود:

ستون‌های $A_۱$ کدواژه‌های محلی هستند که در معادله بررسی‌زوجیت $r(۱)$ صدق‌می‌کنند.این کدواژه‌ها بردارهایی به طول d_c با وزن زوج هستند. یکی از ستون‌های A بردار صفر خواهد بود. بنابراین برای مثال مورد نظر $d_c = ۳$:

$$x_{\mathbb{S}_۱} \in \left\{ \begin{bmatrix} ۰ & ۰ & ۰ \end{bmatrix}^T, \begin{bmatrix} ۰ & ۱ & ۱ \end{bmatrix}^T, \begin{bmatrix} ۱ & ۰ & ۱ \end{bmatrix}^T, \begin{bmatrix} ۱ & ۰ & ۱ \end{bmatrix}^T \right\}$$

بنابراین

$$A_۱ = \begin{bmatrix} ۰ & ۰ & ۱ & ۱ \\ ۰ & ۱ & ۰ & ۱ \\ ۰ & ۱ & ۱ & ۰ \end{bmatrix}$$

- بردار **نشانگر**[۸] $\omega_۱$ که پیکربندی درستی را برای $r(۱)$ انتخاب می‌کند به صورت زیر تعریف می‌شود:

$$\omega_۱ = \begin{bmatrix} \omega_{۱,\emptyset} \\ \omega_{۱,\{۴,۵\}} \\ \omega_{۱,\{۱,۵\}} \\ \omega_{۱,\{۱,۴\}} \end{bmatrix} \in F_۲^۴ \quad \mathbf{1}^T\omega_۱ = ۱ \qquad (۹.۵)$$

باید توجه کنید که ۴ در $F_۲^۴$ متناظر با تعداد کدواژه‌های محلی متفاوتی است که در $r(۱)$ صدق می‌کنند و در حقیقت همان تعداد ستون‌های ماتریس $A_۱$ است. توجه کنید که منظور از $\mathbf{1}^T$ برداری با عناصر ۱ است.

بنابراین با استفاده از (۹،۵) داریم:

$$\omega_۱ \in \left\{ \begin{bmatrix} ۱ \\ ۰ \\ ۰ \\ ۰ \end{bmatrix} \begin{bmatrix} ۰ \\ ۱ \\ ۰ \\ ۰ \end{bmatrix} \begin{bmatrix} ۰ \\ ۰ \\ ۱ \\ ۰ \end{bmatrix} \begin{bmatrix} ۰ \\ ۰ \\ ۰ \\ ۱ \end{bmatrix} \right\} \qquad (۱۰.۵)$$

حال می توان دید که:

$$x_{\mathbb{S}_۱} = A_۱\omega_۱ \qquad (۱۱.۵)$$

[۸] Indicator

توجه کنید که در (۱۱.۵) ضرب و جمع تحت $\mathbb{R}$ انجام می‌شود. حال اگر برای مثال مورد نظر $\omega_۱ = \begin{bmatrix} ۰ & ۱ & ۰ & ۰ \end{bmatrix}^T$ در نظر گرفته شود آنگاه :

$$\omega_{۱,\emptyset} = \omega_{۱,\{۱,۵\}} = \omega_{۱,\{۱,۴\}} = ۰,\ \omega_{۱,\{۴,۵\}} = ۱$$

به طوری که $x_{\mathbb{S}_۱} = \begin{bmatrix} ۱ & ۱ & ۰ \end{bmatrix}$ و $\omega_{۱,\{.\}}$ یک متغیر نشانگر (یا به عبارت دیگر متغیر کمکی) است به این ترتیب $\omega_۱$ پیکربندی مناسبی را با انتخاب کدواژه محلی از ستون‌های $A_۱$ ایجاد می‌کند. با استفاده از روابط (۱۰.۵) و (۱۱.۵) می‌توان نوشت:

$$B_۱x = A_۱\omega_۱ \ ,\ \omega_۱ \in F_۲^۴ \ ,\ \mathbf{1}^T\omega_۱ = ۱ \qquad (۱۲.۵)$$

به این ترتیب (۱۲.۵) معادل با معادله بررسی‌زوجیت $r(۱)$ است. برای سایر معادلات بررسی‌زوجیت نیز می‌توان معادله خطی جدیدی به صورت گفته شده بدست آورد ، حاصل این کار به صورت زیر است:

$$r(۱): \left\{ \begin{array}{c} x_۱ \oplus x_۴ \oplus x_۵ = ۰ \quad \mathbb{S}_۱ = \{۱,۴,۵\} \\ \omega_۱ = \begin{bmatrix} \omega_{۱,\emptyset} \\ \omega_{۱,\{۴,۵\}} \\ \omega_{۱,\{۱,۵\}} \\ \omega_{۱,\{۱,۴\}} \end{bmatrix} \\ B_۱ = \begin{bmatrix} ۱ & ۰ & ۰ & ۰ & ۰ & ۰ \\ ۰ & ۰ & ۰ & ۱ & ۰ & ۰ \\ ۰ & ۰ & ۰ & ۰ & ۱ & ۰ \end{bmatrix} \quad A_۱ = \begin{bmatrix} ۰ & ۰ & ۱ & ۱ \\ ۰ & ۱ & ۰ & ۱ \\ ۰ & ۱ & ۱ & ۰ \end{bmatrix} \end{array} \right\}$$

$$r(۲): \left\{ \begin{array}{c} x_۲ \oplus x_۴ \oplus x_۶ = ۰ \quad \mathbb{S}_۲ = \{۲,۴,۶\} \\ \omega_۲ = \begin{bmatrix} \omega_{۲,\emptyset} \\ \omega_{۲,\{۴,۶\}} \\ \omega_{۲,\{۲,۶\}} \\ \omega_{۲,\{۲,۴\}} \end{bmatrix} \\ B_۲ = \begin{bmatrix} ۰ & ۱ & ۰ & ۰ & ۰ & ۰ \\ ۰ & ۰ & ۰ & ۱ & ۰ & ۰ \\ ۰ & ۰ & ۰ & ۰ & ۰ & ۱ \end{bmatrix} \quad A_۲ = \begin{bmatrix} ۰ & ۰ & ۱ & ۱ \\ ۰ & ۱ & ۰ & ۱ \\ ۰ & ۱ & ۱ & ۰ \end{bmatrix} \end{array} \right\}$$

$$r(\mathbf{۳}):\left\{\begin{array}{c} x_۳\oplus x_۵\oplus x_۶=۰ \quad \mathbb{S}_۳=\{\mathbf{۳},\mathbf{۵},\mathbf{۶}\} \\ \omega_۳=\begin{bmatrix}\omega_{۳,\emptyset}\\ \omega_{۳,\{۵,۶\}}\\ \omega_{۳,\{۳,۶\}}\\ \omega_{۳,\{۳,۵\}}\end{bmatrix} \\ B_۳=\begin{bmatrix}۰&۰&۱&۰&۰&۰\\ ۰&۰&۰&۰&۱&۰\\ ۰&۰&۰&۰&۰&۱\end{bmatrix} \quad A_۳=\begin{bmatrix}۰&۰&۱&۱\\ ۰&۱&۰&۱\\ ۰&۱&۱&۰\end{bmatrix} \end{array}\right\}$$

$\diamondsuit$

همانطور که مشاهده شد اگر تعداد ۱ها در سطرهای ماتریس H یکسان باشد (در مثال قبل این تعداد برابر با $d_c=۳$) ماتریس‌های A_i برای معادلات بررسی‌زوجیت متفاوت، یکسان $(i=۱,\ldots,m)$ خواهد بود.

به این ترتیب فرمول‌بندی کلی به صورت زیر خواهد بود:

برای هر سطر i $(i=۱,\ldots,m)$ ماتریس بررسی‌زوجیت H ، فرض کنید $d_c(i)$ تعداد ۱ها در آن سطر باشد ، $\mathbb{S}_i$ مجموعه اندیس‌های نمادهای کد شرکت‌کننده در معادله بررسی‌زوجیت i و B_i ماتریسی $d_c(i)\times n$ باشد که کدواژه محلی $x_{\mathbb{S}_i}$ را از کدواژه x استخراج می‌کند و A_i ماتریسی $d_c(i)\times ۲^{d_c(i)-۱}$ است که ستون‌های آن توسط کدواژه‌های محلی که در معادله بررسی‌زوجیت i ام صدق می‌کنند ، تشکیل می‌شود.

معادله بررسی‌زوجیت i ام را می‌توان به صورت زیر نوشت :

$$B_ix=A_i\omega_i \quad , \quad \omega_i\in\{۰,۱\}^{۲^{d_c(i)-۱}} \quad , \quad \mathbf{1}^T\omega_i=۱ \qquad (۱۳.۵)$$

توجه کنید که در (۱۳.۵) جمع و ضرب تحت $\mathbb{R}$ انجام می‌گیرد. با جایگذاری (۱۳.۵) در مساله بهینه‌سازی مورد

نظر داریم:

$$\begin{aligned} \min. \quad & \lambda^T x \\ s.t. \quad & B_i x = A_i \omega_i \qquad i = ۱, \dots, m \\ & \omega_i \in \{۰, ۱\}^{۲^{d_c(i)-۱}} \qquad ۱^T \omega_i = ۱ \\ & X_j \in \{۰, ۱\} \qquad j = ۱, \dots, n \end{aligned} \tag{۱۴.۵}$$

باید توجه کرد که x_j مؤلفه‌های بردار x هستند. به این ترتیب همه معادلات خطی تحت $\mathbb{R}$ هستند، اما هنوز مساله بهینه‌سازی، برنامه‌ریزی خطی عدد صحیح (ILP) است. زیرا x_j و ω_i هر دو عدد صحیح اند. بنابراین $\{۰, ۱\}$ را به $[۰, ۱]$ تبدیل می‌کنند.

به این‌ترتیب (۱۴.۵) به صورت زیر بازنویسی می‌شود:

$$\begin{aligned} \min. \quad & \lambda^T x \\ s.t. \quad & B_i x = A_i \omega_i \qquad i = ۱, \dots, m \\ & \omega_i \in [۰, ۱]^{۲^{d_c(i)-۱}} \qquad ۱^T \omega_i = ۱ \\ & X_j \in [۰, ۱] \qquad j = ۱, \dots, n \end{aligned} \tag{۱۵.۵}$$

۲.۳.۵ بررسی جواب‌های کدگشای LP

در صورتی مساله کدگشای LP موفقیت‌آمیز است که، کدواژه کدگشایی شده دقیقا کدواژه ماکزیمم درستنمایی باشد. جواب‌های مساله LP همیشه کدواژه‌های دودویی (عدد صحیح) نیستند، بلکه ممکن است جواب حاصل غیرصحیح باشند. اگر جواب غیرصحیح باشد، کدگشای LP شکست خورده است. جواب‌های غیرصحیح کدگشای LP، **جواب‌های کسری** [۹] نامیده می‌شود (در بخش تفسیر هندسی بیشتر مورد بحث قرار خواهد گرفت) .

با این حال روش‌هایی مانند اضافه کردن معادله بررسی‌زوجیت افزونگی که در بخش (۷.۵) مطرح خواهد شد، ممکن است بتوان جواب‌های کسری را به جوابهای عددصحیح تبدیل کرد.

[۹] Fractional solution

نکته‌ای که باید به آن توجه کرد این است که، در حالتی که جواب کدگشای LP عددصحیح است، نمی‌توان گفت که جواب حاصل حتما کدواژه ارسالی واقعی است بلکه می‌توان گفت، که کدواژه ML است. به این ترتیب اگر کدگشای برنامه‌ریزی خطی به جواب عددصحیح برسد این جواب حتما کدواژه ML است. این خاصیت کدگشای LP را **تصدیق ML** [۱۰] می‌نامند [۱۲] .

این خاصیت در مقایسه کدگشا LP با الگوریتم‌های کدگشا عبور پیام مزیت بسیار مهمی است. زیرا جواب حاصل از الگوریتم عبورپیام ممکن است یک بردار دودویی باشد که کدواژه نیست. در صورتی که اگر جواب کدگشا LP عدد صحیح باشد (بردار دودویی) حتما کدواژه ML است. در حقیقت در جمله قبل دو نکته وجود دارد، نکته اول: اگر جواب کدگشا LP عددصحیح باشد، این جواب حتما کدواژه است و نکته دوم: اگر جواب کدگشا LP کدواژه باشد، این جواب حتما کدواژه ML است.این دو نکته در قالب دو قضیه توسط فلدمن در [۱۳] مطرح شده است که در بخش تفسیر هندسی بیان خواهد شد.

۳.۳.۵ مقیاس گذاری λ

چون در فرمول‌بندی LP مورد نظر تابع هدف ، به صورت

$$\hat{x} = \text{argmin}\lambda^T x$$

بهینه‌سازی است. لذا با ضرب عدد ثابت β در $\lambda^T x$ جواب تغییری نخواهد کرد. یعنی:

$$\hat{x} = \text{argmin}\lambda^T x = \text{argmin}(\beta\lambda^T)x \qquad (۱۶.۵)$$

بنابراین آگاهی از پارازیت مهم نخواهد بود.

[۱۰] ML Certificate

۴.۵ تفسیر هندسی

قبل از اینکه تفسیر هندسی را بیان کنیم، باید برخی مفاهیمی را که توسط فلدمن در [۱۱] ، [۱۲] بیان‌شده است، را شرح دهیم.

فرض کنید به جای اینکه کدواژه‌ها را متعلق به $F_۲^n$ در نظر بگیریم آنها را به عنوان عناصر ابرمکعب واحد که متعلق به $\mathbb{R}^n$ هستند، در نظر بگیریم. به این ترتیب می‌توان مساله بهینه‌سازی مورد نظر را به صورت زیر نوشت:

$$\begin{aligned} \min. \quad & \lambda^T x \\ s.t. \quad & x \in poly(\mathcal{C}) \end{aligned}$$

که در رابطه بالا منظور از $poly(\mathcal{C})$ چندسقفی حاصل از ترکیبات محدب کدواژه‌های متعلق به کد $\mathcal{C}$ است (یادآوری: λ^T بردار لگاریتم نسبت درستنمایی است). یعنی:

$$poly(\mathcal{C}) = \left\{ \sum_{\nu \in \mathcal{C}} \gamma_\nu \nu \qquad : \gamma_\nu \geq ۰ \ , \ \sum_{\nu \in \mathcal{C}} \gamma_\nu = ۱ \right\}$$

فلدمن این چندسقفی را **چندسقفی کدواژه** [۱۱] می‌نامند (که داخل ابر مکعب $[۰, ۱]$ قرار دارد). هر نقطه متعلق به $poly(\mathcal{C})$ ، به صورت $x = (x_۱, \ldots, x_n)$ است که:

$$\forall i = ۱, \ldots, n \quad x_i = \sum_{\nu \in \mathcal{C}} \gamma_\nu \nu_i$$

قبل از اینکه به ادامه بحث بپردازیم به نکته زیر توجه کنید:

با توجه به اینکه جواب مساله بهینه‌سازی (با توجه به مطالب گفته شده در (۲.۵)) حتما یک نقطه راسی است، دو مساله بهینه‌سازی زیر دقیقا جوابهای بهینه مشابهی را تولید می‌کنند (زیرا نقاط راسی آنها یکسان است).

$$\begin{aligned} \operatorname{argmin}_{x \in \mathcal{C}} \lambda^T x = \quad & \operatorname{argmin}_{x \in poly(\mathcal{C})} \lambda^T x \end{aligned} \qquad (۱۷.۵)$$

[۱۱] Codeworde polytope

همانطور که فلدمن در [۱۰] بیان می‌کند، برای هر کد دلخواه، این پوسته محدب بسیار پیچیده است. به همین دلیل فلدمن و همکارانش چندسقفی کاهشی $\mathcal{P}$ را برای کدگشایی تعریف کرده‌اند.

در LP کاهشی که توسط فلدمن و همکارانش در [۱۲] مطرح شده است، هر راس بررسی متعلق به گراف تنر، **کد محلی** [۱۲] و اشتراک همه کدهای محلی به عنوان **کد سراسری** [۱۳] تعریف شده است. علاوه‌براین هر راس بررسی متعلق به گراف تنر در واژگان LP به عنوان **چندسقفی کدواژه محلی** [۱۴] ($\mathcal{P}_i$) و اشتراک همه چندسقفی‌های کدواژه‌های محلی را به عنوان **چندسقفی سراسری** [۱۵] ($\mathcal{P}$) تعریف می‌شود.

فرض کنید $\mathcal{I} = \{۱, \ldots, n\}$ مجموعه اندیس‌های راس‌های بررسی و $\mathcal{J} = \{۱, \ldots, m\}$ مجموعه اندیس‌های راس‌های متغیر باشد. هر یال $(i, j) \in \mathcal{I} \times \mathcal{J}$ متعلق به گراف تنر $\mathcal{T}$ (متناظر با کد $\mathcal{C}$) است اگر و تنها اگر عنصر $H_{i,j}$ متعلق به ماتریس بررسی زوجیت H برابر ۱ باشد. $\forall i \in \mathcal{I}$ ، $\mathcal{N}(i)$ مجموعه اندیس‌های همه راس‌های متغیر مجاور با راس بررسی i و به طور مشابه $\forall j \in \mathcal{J}$ ، $\mathcal{N}(j)$ مجموعه اندیس‌های همه راس‌های بررسی مجاور با راس متغیر j در نظر بگیرید. (توجه کنید که $\mathcal{N}(i)$ معادل $\mathbb{S}_i$ است) $S \subseteq \mathcal{N}(i)$ که شامل تعداد زوجی از اندیس‌های راس‌های متغیر است **مجموعه کدواژه محلی** (متناظر با کدواژه محلی $x_{\mathbb{S}_i}$) تعریف می‌شود به طوری که در کدواژه محلی y متناظر با آن به ازای هر j متعلق به S ، $y_j = ۱$ ، از طرفی $\forall j \in \mathcal{N}(i)$ که $j \notin S$ داریم: $y_j = ۰$ و سایر y_j ها به طور دلخواه انتخاب می‌شوند ($y_j \in \{۰, ۱\}$). فلدمن برای هر S متعلق به مجموعه $\{|S|$زوج است : $E_i = \{S \subseteq \mathcal{N}(i)$ متغیر کمکی $\omega_{i,S}$ که **نشانگر** کدواژه محلی وابسته به S است را تعریف می‌کند.

اگر $\omega_{i,S} = ۱$ آنگاه این تساوی نشان می‌دهد که از بین همه متغیرهایی که در کد محلی متناظر با راس بررسی i وجود دارند (یعنی $x_{\mathbb{S}_i}$) متغیرهای متناظر با عناصر مجموعه S باید مقداری مساوی با ۱ داشته باشند. علاوه‌براین متغیر کمکی $\omega_{i,\emptyset}$ ، بعنوان نشانگر حالتی که به ازای هر راس بررسی همه متغیرها شرکت‌کننده در این راس بررسی،

[۱۲] Local code
[۱۳] Global code
[۱۴] Local codeword polytope
[۱۵] Global polytope

مقدار صفر را اخذ می‌کنند، تعریف می‌شود. متغیرهای کمکی $\omega_{i,S}$ (همان‌طور که در بخش‌های قبل نیز بیان شد) باید در شرایط زیر صدق کنند:

$$\forall S \in E_i \ , \ ۰ \leq \omega_{i,S} \leq ۱ \qquad (۱۸.۵)$$

و علاوه‌براین

$$\sum_{S \in E_i} \omega_{i,S} = ۱ \qquad (۱۹.۵)$$

علت این شرایط این است که: به ازای هر راس بررسی i ، فقط یکی از S ها می‌تواند جواب اصلی باشد (زیرا $\omega_{i,S}$ به عنوان نشانگر قسمتی از کدواژه اصلی است که توسط S در معادله بررسی‌زوجیت i ام صدق می‌کند) .

فلدمن از متغیرهای $(x_۱, \ldots, x_n)$ برای نمایش مؤلفه‌های کدواژه‌ها استفاده می‌کند به این ترتیب:

$$\forall j \in \mathcal{J} \ , \ ۰ \leq x_j \leq ۱ \qquad (۲۰.۵)$$

بنابراین نشانگر x_j هنگامی که راس متغیر j که با راس بررسی i مجاور است باید متعلق به چندسقفی کدواژه محلی متناظر با راس بررسی i باشد، که این شرط منجر به محدودیت زیر می‌شود:

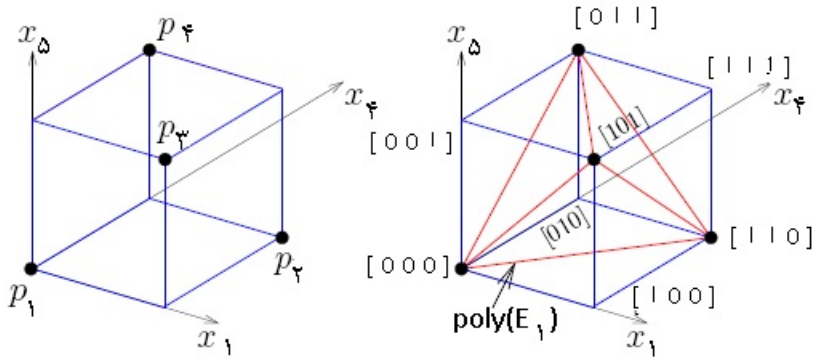
$$\forall j \in \mathcal{N}(i) \ , \ x_j = \sum_{S \in E_i \ , \ S \ni j} \omega_{i,S} \qquad (۲۱.۵)$$

براین اساس چندسقفی کدواژه محلی $\mathcal{P}_j$ (به ازای هر راس بررسی i) عبارت است از همه نقاط (x, ω) که که در ۴ شرط اخیر (یعنی (۱۸.۵) تا (۲۱.۵)) صدق می‌کنند و به این ترتیب $\mathcal{P} = \bigcap_J \mathcal{P}_j$ بیانگر چندسقفی کدواژه است. بنابراین مساله بهینه‌سازی مورد نظر را می‌توان به صورت زیر بیان کرد:

$$\begin{aligned} \min. \quad & \sum_{i=۱}^{n} \lambda_i x_i \\ s.t. \quad & (x, \omega) \in \mathcal{P} \end{aligned}$$

باید توجه شود که اگر $V(\mathcal{P})$ مجموعه نقاط راسی چندسقفی $\mathcal{P}$ باشد ، داریم :

$$\mathcal{C} \subseteq V(\mathcal{P}) \subseteq \mathcal{P}$$



شکل ۱.۵: شکل سمت چپ مکعب واحد و شکل سمت راست پوسته محدب (چندسقفی محلی) $r(1)$

(یا به عبارت دیگر $poly(\mathcal{C}) \subseteq \mathcal{P}$) . ویژگی که چندسقفی $\mathcal{P}$ دارد این است که:

$$\mathcal{C} = \mathcal{P} \cap \{0,1\}^n \qquad (۲۲.۵)$$

یعنی نقاط راسی صحیح چندسقفی $\mathcal{P}$ دقیقا همان کدواژه‌های $\mathcal{C}$ است و سایر نقاط راسی غیرصحیح گویا، **جواب‌های کسری** هستند . چندسقفی که ویژگی گفته شده در (۲۲.۵) را دارد ، در [۱۲] به عنوان **چندسقفی محض** [۱۶] تعریف می‌شود . در زیر بخش ۱.۴.۵ شرح بیشتری در مورد چندسقفی $\mathcal{P}$ بیان خواهد شد .

مثال ۱.۴.۵. [۲۵] در مثال (۱.۳.۵) رابطه (۱۱.۵) را می‌توان به صورت زیر نوشت :

$$\begin{bmatrix} x_1 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \omega_{1,\emptyset} \\ \omega_{1,\{1,4\}} \\ \omega_{1,\{1,5\}} \\ \omega_{1,\{4,5\}} \end{bmatrix} \ , \ \omega_{1,S} \geq 0 \ , \ \mathbf{1}^T\omega = 1 \qquad (۲۳.۵)$$

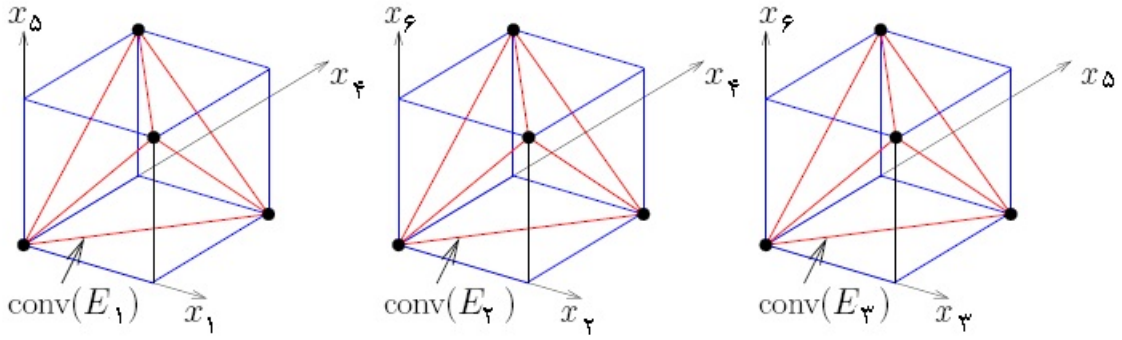
می‌دانیم در حالت کلی پوسته محدب مجموعه $\mathcal{L}$ به صورت زیر تعریف می‌شود:

$$Conv(\mathcal{L}) = \left\{ \sum_{l\in\mathcal{L}} \gamma_l l : \gamma_l \geq 0 \ , \ \sum_{l\in\mathcal{L}} \gamma_l = 1 \right\} \qquad (۲۴.۵)$$

با مقایسه محدودیت‌های رابطه (۲۳.۵) و رابطه (۲۴.۵) می‌بینیم که ω معادل γ است. ماتریس A_1 را به صورت زیر می‌نویسیم:

$$A_1 = [p_1 \quad p_2 \quad p_3 \quad p_4]$$

[۱۶] Proper polytope



شکل ۲.۵: چند سقفی کدواژه‌های محلی

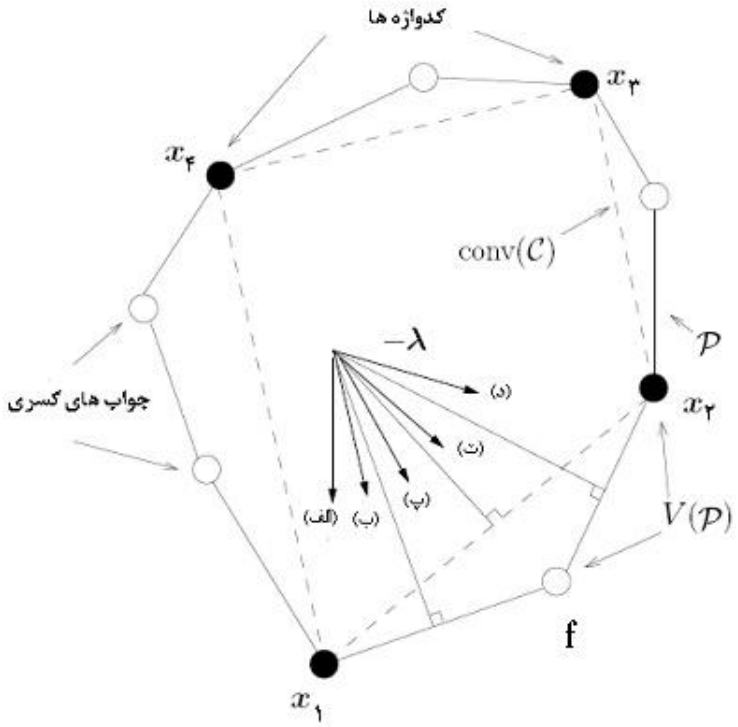
که $E_1 = \{p_i : i = 1, \ldots, 2^{(d_c-1)}\}$ ستون‌های A_1 باشند. شکل ۱.۵ چندسقفی کدواژه محلی حاصل از معادله بررسی‌زوجیت $r(1)$ را نشان می دهد . و شکل ۲.۵ سایر چندسقفی کدواژه‌های محلی مثال (۱.۳.۵) را نشان می دهد. به این ترتیب چندسقفی سراسری به صورت زیر است:

$$\mathcal{P} = \mathcal{P}_1 \cap \mathcal{P}_2 \cap \ldots \cap \mathcal{P}_m$$

$\diamondsuit$

۱.۴.۵ شرح چندسقفی $\mathcal{P}$

در شکل ۳.۵ چندسقفی (کلی) $\mathcal{P}$ در فضای دو بعدی نشان داده شده است. در این شکل چندسقفی حاصل از کدواژه‌های متعلق به کد $\mathcal{C}$ (که همان پوسته محدب کران‌دار یعنی $conv(\mathcal{C})$ است) با نقطه چین و $\mathcal{P}$ توسط خطوط پررنگ نمایش داده شده است. دایره‌های سیاه رنگ، کدواژه‌ها و دایره‌های سفید رنگ راس‌های کسری (که کدواژه نیستند) را نشان می‌دهد. همانطور که می‌بینید $conv(\mathcal{C}) \subseteq \mathcal{P}$. باید توجه کنیم که فقط تابع هدف LP است که به پارازیت وابسته است. تابع هدف LP را می‌توان به عنوان پیکان داخل چندسقفی در نظر گرفت، جواب بهینه، نقطه راسی است که در دورترین نقطه از پیکان قرار دارد . اگر کانال فاقد حافظه باشد، آنگاه کدواژه دریافتی کدواژه ML خواهد بود. بنابراین کدواژه‌ای با کمترین هزینه است. با مقایسه کدگشا LP کاهشی



شکل ۳.۵: [۲۵]رابطه بین چندسقفی $\mathcal{P}$ ، $conv(\mathcal{C})$ ، کدواژه‌ها، جواب های کسری و بردارλ

و کدگشایی ML دقیق می‌توان دید که در کدگشا ML دقیق مجموعه محدودیت‌ها، همان چندسقفی $poly(\mathcal{C})$ (پوسته محدب $conv(\mathcal{C})$) است. در حالی که کدگشای LP از چندسقفی بزرگتر $\mathcal{P}$ استفاده می‌کند. شکل ۳.۵ را در نظر بگیرید، فرض کنید کدواژه $x_۱$ ارسال شده باشد. ۴ پیکان نمایش داده شده در شکل ۳.۵ بیانگر ۴ تابع هدف LP تحت پارازیت‌های متفاوت هستند . (البته پیکان (الف) بیانگر حالتی است که کانال فاقد پارازیت است و در این حالت هردو کدگشا موفق هستند) همانطور که در (ب) می‌بینید، اگر پارازیت بسیار کم باشد آنگاه هر دو کدگشای LP و ML موفق خواهند بود زیرا هردو کدگشا، کدواژه ارسالی $x_۱$ را به عنوان نقطه بهینه در نظر می‌گیرند. در حالت (پ) پارازیت بیشتری نسبت به (ب) در کانال وجود دارد، با توجه به این شرایط کدگشای ML موفق خواهد بود ولی کدگشای LP شکست می‌خورد چون راس کسری f جواب بهینه LP کاهشی است. در (ت) با افزایش پارازیت نسبت به (پ) ML شکست می‌خورد زیرا در این حالت $x_۲$ جواب بهینه خواهد شد، جواب بهینه LP هنوز راس کسری f است. بنابراین کدگشایLP خطا را کشف کرده است. در نهایت در حالت

(د) که پارازیت بسیار زیاد است ، جواب بهینه هر دو کدگشای LP و ML، $x_۲$ است و هر دو روش علاوه بر شکست در کدگشایی قادر به کشف خطا نیستند. توجه کنید که در هر دو حالت آخر، وقتی ML شکست می‌خورد، شکست کدگشای LP به دلیل نقض خود کد (در تضاد با کدگشا LP) است (در بخش تحلیل کدگشای LP شرح کامل‌تری بیان می‌گردد) .

به دو گزاره زیر توجه کنید:

گزاره ۲.۴.۵. [۱۱] به ازای هر نقطه $(x,\omega) \in \mathcal{P}$ ، $(x_۱,\ldots,x_n)$ نشان دهنده یک کدواژه است. علاوه براین به ازای هر کدواژه $(y_۱,\ldots,y_n)$ ، وجود دارد ω^y به طوری که (x,ω^y) ، یک نقطه با مقدار صحیح متعلق به $\mathcal{P}$ است، به طوری که به ازای هر $j \in \mathcal{J}$ ، $x_j = y_j$.

اثبات. فرض کنید (x,ω) نقطه‌ای متعلق به $\mathcal{P}$ باشد به طوری که به ازای هر j ، $x_j \in \{۰,۱\}$ و همه $\omega_{i,S} \in \{۰,۱\}$. حال فرض کنید $(x_۱,\ldots,x_n)$ کدواژه نباشد و فرض کنید i اندیس راس بررسی باشد که $\forall j \in \mathcal{J}$ رابطه $y_j = x_j$ برقرار نیست. با توجه به $\sum_{S\in E_i} \omega_{i,S} = ۱$ و این حقیقت که ω عدد صحیح است، به ازای هر $S \in E_i$ ، $\omega_{i,S} = ۱$ و برای سایر $S' \in E_i$ ، $\omega_{i,S'} = ۰$. براساس $\forall j \in \mathcal{N}(i)$ داریم: $x_j = \sum_{S\in E_i, S\in E_i} \omega_{i,S}$

به ازای هر $j \in S$ ، $x_j = ۱$ و به ازای هر $j \in \mathcal{N}(i)$ و $j \notin S$ ، آنگاه $x_j = ۰$. چون $|S|$ زوج است، در محدودیت بررسی‌زوجیت i حتما $y = x$ است که این یک تناقض است.

برای اثبات بخش دوم گزاره، فرض کنید $(y_۱,\ldots,y_n)$ کدواژه باشد و قرار دهید $x_j = y_j$. به ازای هر $i \in \mathcal{I}$ ، S مجموعه‌ای از راس‌های j متعلق به $\mathcal{N}(i)$ باشد که $x_j = y_j = ۱$. چون $(y_۱,\ldots,y_n)$ کدواژه است پس y در محدودیت بررسی زوجیت i صدق می کند. بنابراین $|S|$ زوج است و با معرفی متغیر $\omega_{i,S}$ ، قرار دهید $\omega^y_{i,S} = ۱$ و به ازای سایر $S' \in E_i$، $\omega^y_{i,S'} = ۰$. همه محدودیت‌ها برقرار هستند و همه متغیرها صحیح هستند.

□

گزاره ۳.۴.۵. [۱۱] کدگشایLP خاصیت تصدیق ML دارد (یعنی):

اگر خروجی الگوریتم یک کدواژه باشد، الگوریتم ، کدواژه ML بودن آن کدواژه را تضمین می کند .

اثبات. اگر خروجی الگوریتم کدواژه y باشد آنگاه (y, ω^y) هزینه‌ای کمتر یا مساوی با هزینه نقاط متعلق به $\mathcal{P}$ دارد. برای کدواژه $y' \neq y$ ، براساس گزاره ۲.۴.۵ داریم : $(y', \omega^{y'})$ نقطه‌ای متعلق به $\mathcal{P}$ است . بنابراین y هزینه‌ای کمتر یا مساوی با y' دارد.

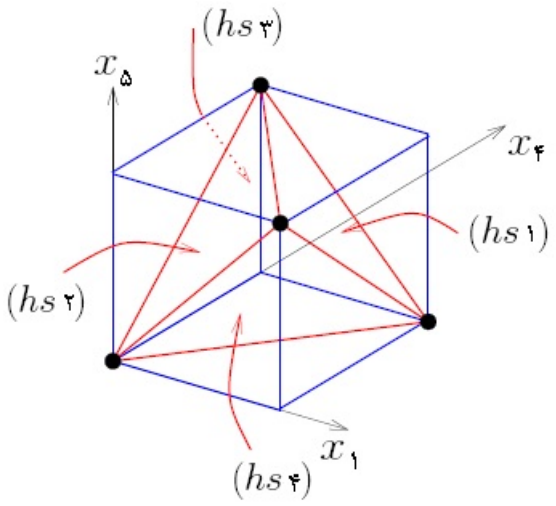
□

۵.۵ فرمول‌بندی جایگزینی

در این بخش یک روش جایگزینی برای ساخت محدودیت‌های LP با استفاده از [۱۱] و [۲۵]بیان می‌شود. علت بیان این روش کاهش پیچیدگی LP با استفاده از حذف متغیرهای کمکی $\omega_{i,S}$ ($i = ۱, \ldots, m$) است. با توجه به اینکه به ازای هر معادله بررسی‌زوجیت تعداد متغیرهای کمکی متفاوت $\omega_{i,S}$ به طور نمایی ($۲^{d_c - ۱}$) افزایش می‌یابد. بنابراین حذف آنها می‌تواند پیچیدگی مساله را کاهش دهد. با توجه به اینکه هدف مینیمم کردن $\lambda^T x$ است، حذف متغیرهای کمکی $\omega_{i,S}$ تاثیری بر $\lambda^T x$ ندارد. این روش فرمول‌بندی ابتدا با ارائه مثالی شرح داده می‌شود، سپس حالت کلی بیان می‌شود.

مثال ۱.۵.۵. [۲۵] در مثال (۱.۳.۵) معادله بررسی‌زوجیت $r(۱)$ را در نظر بگیرید که در رابطه (۲۵.۵) محدودیت $\mathbf{1}^T \omega_{۱} = ۱$ به ماتریس حاصل از این معادله بررسی‌زوجیت اضافه شده است.

$$\begin{bmatrix} ۱ \\ x_۱ \\ x_۴ \\ x_۵ \end{bmatrix} = \begin{bmatrix} ۱ & ۱ & ۱ & ۱ \\ ۰ & ۱ & ۱ & ۰ \\ ۰ & ۱ & ۰ & ۱ \\ ۰ & ۰ & ۱ & ۱ \end{bmatrix} \begin{bmatrix} \omega_{۱,\emptyset} \\ \omega_{۱,\{۱,۴\}} \\ \omega_{۱,\{۱,۵\}} \\ \omega_{۱,\{۴,۵\}} \end{bmatrix} , \ \omega_{۱,S} \geq ۰ \qquad (۲۵.۵)$$



شکل ۴.۵: ۴ نیم‌فضا

ماتریس افزوده (۲۶.۵)، که از (۲۵.۵) حاصل شده است، به شکل سطری پلکانی تقلیل یافته تبدیل می‌شود:

$$\begin{bmatrix} ۱ & ۱ & ۱ & ۱ & ۱ \\ ۰ & ۱ & ۱ & ۰ & x_۱ \\ ۰ & ۱ & ۰ & ۱ & x_۴ \\ ۰ & ۰ & ۱ & ۱ & x_۵ \end{bmatrix} \sim \begin{bmatrix} ۱ & ۰ & ۰ & ۰ & \frac{۲ - x_۱ - x_۴ - x_۵}{۲} \\ ۰ & ۱ & ۰ & ۰ & \frac{x_۱ + x_۴ - x_۵}{۲} \\ ۰ & ۰ & ۱ & ۰ & \frac{x_۱ - x_۴ + x_۵}{۲} \\ ۰ & ۰ & ۰ & ۱ & \frac{-x_۱ + x_۴ + x_۵}{۲} \end{bmatrix} \qquad (۲۶.۵)$$

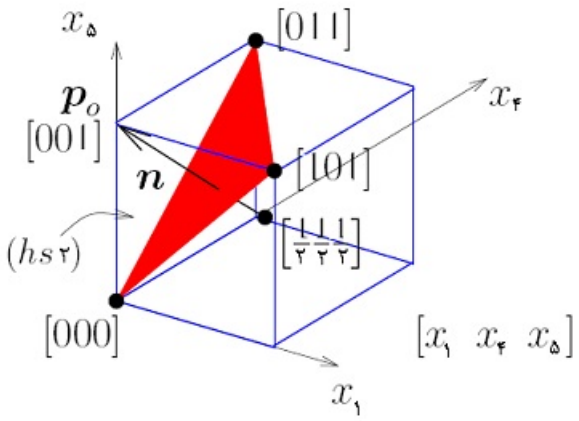
حال اگر محدودیت $\omega_{۱,S} \geq ۰$ را در (-۱) ضرب کنیم و آنرا به (۲۶.۵) اضافه کنیم داریم :

$$\begin{array}{rcll} -۲ + x_۱ + x_۴ + x_۵ & \leq & ۰ & (hs۱) \\ -x_۱ - x_۴ + x_۵ & \leq & ۰ & (hs۲) \\ -x_۱ + x_۴ - x_۵ & \leq & ۰ & (hs۳) \\ x_۱ - x_۴ - x_۵ & \leq & ۰ & (hs۳) \end{array} \qquad (۲۷.۵)$$

توجه کنید که فضای جواب (۲۵.۵) و (۲۷.۵) یکسان است (همه محدودیت‌های (۲۵.۵) در (۲۷.۵) مورد استفاده قرار گرفته‌اند). به این ترتیب محدودیت‌های معادله بررسی‌زوجیت به روشی دیگر بدون استفاده از متغیر کمکی $\omega_{۱,S}$ بیان شد. ۴ نیم‌فضای حاصل از $(hs۱)$-$(hs۴)$ را می‌توان به صورت شکل ۴.۵ نمایش داد.

$\diamondsuit$

قبل از اینکه حالت کلی را بیان کنیم، به شرح مختصری درباره مثال قبل و نیم فضای $(hs۲)$ می‌پردازیم و با استفاده از این شرح حالت کلی را بیان می‌کنیم. شکل ۵.۵ را در نظر بگیرید. با توجه به $(hs۲)$ بردار نرمال صفحه



شکل ۵.۵: نیم‌فضای حاصل از $(hs۲)$

$(hs۲)$ $n = p_۰ - \frac{۱}{۲}\mathbf{1}$ است (که منظور از $\mathbf{1}$ بردار واحد است)، یعنی برداری که نقطه مرکز مکعب واحد را به نقطه‌ای به وزن فرد وصل می‌کند. بنابراین معادله این سطح به طوری که همه x_{S_j} در آن صدق می‌کنند) به صورت زیر است.

$$n^T \left(x_{S_j} - x_{۰,S_j}\right) = ۰ \qquad (۲۸.۵)$$

که n بردار نرمال و $x_{۰,S_j}$ نقطه‌ای متعلق به صفحه مورد نظر است. بنابراین برای مثال مورد نظر در شکل ۵.۵ معادله صفحه $(hs۲)$ به صورت زیر است:

$$\begin{aligned} ۲n^T(x_{S_j} - x_{۰,S_j}) = \quad & ۲(p_۰ - \frac{۱}{۲}\mathbf{1})^T(x_{S_j} - x_{۰,S_j}) = ۰ \\ = \quad & ۲\left(\begin{bmatrix} ۰ \\ ۰ \\ ۱ \end{bmatrix} - \begin{bmatrix} \frac{۱}{۲} \\ \frac{۱}{۲} \\ \frac{۱}{۲} \end{bmatrix}\right)^T \left(\begin{bmatrix} x_۱ \\ x_۴ \\ x_۵ \end{bmatrix} - \begin{bmatrix} ۰ \\ ۰ \\ ۰ \end{bmatrix}\right) = ۰ \\ = \quad & \begin{bmatrix} ۰ \\ ۰ \\ ۱ \end{bmatrix}^T \begin{bmatrix} x_۱ \\ x_۴ \\ x_۵ \end{bmatrix} = ۰ \\ = \quad & -x_۱ - x_۴ + x_۵ = ۰ \end{aligned} \qquad (۲۹.۵)$$

همانطور که می‌بینیم این همان معادله نیم‌فضا $(hs۲)$ در رابطه (۲۷.۵) است. با استفاده از این مثال خاص می‌توان حالت کلی فرمول‌بندی نیم‌فضا را بیان کرد:

$$\begin{aligned} n^T\left(x_{S_j} - x_{۰,S_j}\right) &\leq ۰ \\ ۲n^T x_{S_j} &\leq ۲n^T x_{۰,S_j} \end{aligned} \tag{۳۰.۵}$$

توجه داشته باشید که نوک بردار n یک راس به وزن فرد است به این ترتیب:

$$۲n^T = \overbrace{\underbrace{[+۱\ +۱\ \ldots\ +۱}_{V\ \ s.t.\ \ |V| \text{ فرد است}}\ \underbrace{-۱\ -۱\ \ldots\ -۱]}_{\mathcal{N}(j)-v}}^{\mathcal{N}(j)} \tag{۳۱.۵}$$

که در (۳۱.۵) ، $\mathcal{N}(j)$ مجموعه‌ای از اندیس‌های راس‌های متغیر است که با راس بررسی j مجاور هستند. البته ممکن است مکان -۱ و $+۱$ در $۲n^T$ متفاوت باشد ، در اینجا به منظور راحتی بیشتر به این صورت بیان شده است. V زیرمجموعه‌ای از $\mathcal{N}(j)$ است که در محل اندیس‌ها متعلق به V در بردار $۲n^T$، $+۱$ و در سایر مکان‌های بردار $۲n^T$ ، -۱ قرار می‌گیرد. نقطه $x_{۰,S_j}$ را می‌توان راسی با وزن زوج متعلق به صفحه انتخاب کرد. از طرفی وزن هامینگ یا $\mathcal{W}_H(x_{۰,S_j} \oplus p_۰) = ۱$. این مطلب را می‌توان با استفاده از شکل ۵.۵ به این صورت تفسیر کرد که: با حرکت از یک راس مکعب با وزن زوج در امتداد یک یال متعلق به مکعب به راسی با وزن فرد می‌رسیم. به این ترتیب چون $x_{۰,S_j}$ زوج و فاصله آن از $p_۰$ برابر ۱ است، پس: $\mathbf{1}^T x_{۰,S_j} = |V| \pm ۱$ بنابراین داریم:

$$x_{۰,S_j} = \overbrace{\underbrace{[۱\ ۱\ \ldots\ ۱}_{|V|+۱ \text{ زوج}}\ ۰\ ۰\ \ldots\ ۰]}^{\mathcal{N}(j)} \tag{۳۲.۵}$$

مکان ۱ها و ۰ ها در $x_{۰,S_j}$ می‌تواند تغییر کند، و به منظور راحتی به صورت بالا نوشته شده است. بنابراین:

$$۲n^T x_{۰,S_j} = |V| - ۱ \tag{۳۳.۵}$$

از رابطه (۳۰.۵) داریم:

$$\begin{aligned} ۲n^T x_{S_j} &\leq ۲n^T x_{۰,S_j} \\ \sum_{j\in V} x_j - \sum_{j\in\mathcal{N}(i)-V} x_j &\leq |V| - ۱ \end{aligned} \qquad (۳۴.۵)$$

با توجه به اینکه باید معادله مربوط به همه نیم‌فضاها را بیابیم، پس باید نیم‌فضاهای مربوط به هر راس با وزن فرد متعلق به مکعب واحد را بیابیم که در حقیقت هر راس وزن فرد معادل با انتخاب زیرمجموعه V (با وزن فرد) از $\mathcal{N}(j)$ است. چون تمام جواب‌ها باید داخل متعلق به مکعب واحد باشند، بنابراین محدودیت بررسی‌زوجیت کلی در فرمولبندی جایگزینی به صورت زیر است:

$$\sum_{j\in V} x_j - \sum_{j\in\mathcal{N}(i)-V} x_j \leq |V| - ۱ \quad , \quad \forall V \in \mathcal{N}(i) \quad , \quad |V| \text{فرد} \quad , \quad ۰ \leq x_j \leq ۱ \qquad (۳۵.۵)$$

به منظور بررسی رابطه (۳۵.۵) به مثال زیر توجه کنید.

مثال۵.۵.۲. [۲۵] معادلات بررسی‌زوجیت کد هامینگ (۷،۴،۳) به صورت (۳۶.۵) است. معادله بررسی‌زوجیت A را در نظر بگیرید:

$$H \otimes x = \begin{cases} chk(A) & : \quad x_۱ \oplus x_۲ \oplus x_۴ \oplus_۵ = ۰ \\ chk(B) & : \quad x_۲ \oplus x_۳ \oplus x_۴ \oplus_۶ = ۰ \\ chk(B) & : \quad x_۴ \oplus x_۵ \oplus x_۶ \oplus_۷ = ۰ \end{cases} \qquad (۳۶.۵)$$

همان طور که مشاهده می شود، داریم:

$$\mathcal{N}(A) = \{۱, ۲, ۴, ۵\}$$

رابطه (۳۵.۵) را برای معادله بررسی‌زوجیت A برای راحتی بیشتر به صورت زیر می‌نویسیم:

$$\begin{bmatrix} ۱ & -۱ & -۱ & -۱ \\ -۱ & ۱ & -۱ & -۱ \\ -۱ & -۱ & ۱ & -۱ \\ -۱ & -۱ & -۱ & ۱ \\ ۱ & ۱ & ۱ & -۱ \\ ۱ & ۱ & -۱ & ۱ \\ ۱ & -۱ & ۱ & ۱ \\ -۱ & ۱ & ۱ & ۱ \end{bmatrix} \begin{bmatrix} x_۱ \\ x_۲ \\ x_۴ \\ x_۵ \end{bmatrix} \leq \begin{bmatrix} ۰ \\ ۰ \\ ۰ \\ ۰ \\ ۲ \\ ۲ \\ ۲ \\ ۲ \end{bmatrix} \qquad \begin{array}{l} V = \{۱\} \\ V = \{۲\} \\ V = \{۴\} \\ V = \{۵\} \\ V = \{۱, ۲, ۴\} \\ V = \{۱, ۲, ۵\} \\ V = \{۱, ۴, ۵\} \\ V = \{۲, ۴, ۵\} \end{array} \qquad (۳۷.۵)$$

$\diamondsuit$

۱.۵.۵ نکاتی درباره تفسیر هندس روش جایگزینی

تعداد متغیرها و محدودیت‌های متعلق به چند سقفی $\mathcal{P}$ ، نمایی برحسب درجه راس‌های بررسی است. با فرض اینکه δ_r^+ ماکزیمم درجه راس‌های بررسی متعلق به گراف تنر باشد، مساله LP کاهشی مورد نظر $(n + m2^{\delta_r^+})$ متغیر و محدودیت دارد. برای کدهای LDPC این پیچیدگی خطی برحسب n است، زیرا δ_r^+ ثابت است (برای مطالعه بیشتر و اثبات به [۱۰] ، [۱۱] مراجعه کنید). به این ترتیب برای حل کارا $\mathcal{P}$ نیازمند استفاده از چندسقفی کوچکتری هستیم که جواب‌های مشابهی تولید کند. برای این منظور به تعاریف زیر توجه کنید .

تعریف ۳.۵.۵. [۱۱] فرض کنید P چندسقفی تعریف شده تحت متغیرهای $\{f_j\}_{j\in\mathcal{J}}$ که $0 \leq f_j \leq 1$ و متغیرهای کمکی x باشد ، **چندسقفی تصویر** [۱۷] به صورت زیر تعریف می شود :

$$\bar{P} = \{f : \exists x \;\; s.t. \;\; (f, x) \in P\}$$

می گوییم چندسقفی P با چندسقفی Q هم ارز است ، اگر

$$\bar{P} = \bar{Q}$$

باشد .

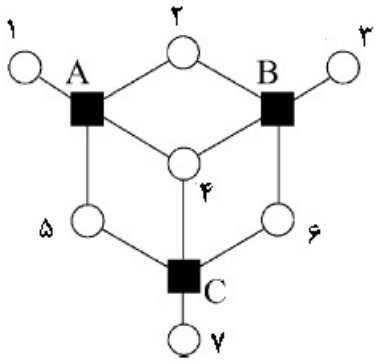
بهینه سازی تحت P و Q نتایج یکسانی را تولید خواهد کرد. به قضیه زیر توجه کنید.

قضیه ۴.۵.۵. [۱۱] فرض کنید به ازای راس بررسی i ، Ω_i مجموعه نقاط x باشد که در محدودیت (۳۵.۵) صدق می‌کند و $\Omega = \cap_j \Omega_i$ ، آنگاه Ω و چندسقفی سراسری $\mathcal{P}$ هم ارز هستند. به عبارت دیگر چندسقفی

$$\bar{\mathcal{P}} = \{f : \exists\, \omega \; s.t. \; (f, \omega) \in \mathcal{P}\}$$

دقیقا مجموعه همه نقاطی است که در (۳۵.۵) صدق می‌کند .

[۱۷] projection polytop



شکل ۶.۵: گراف فاکتور (۷.۴.۳) کد هامینگ . راس‌هایی که با دایره نمایش داده می‌شوند ، راس متغیر و راس‌هایی که با مربع نمایش داده می‌شوند، راس‌های بررسی هستند.

□

۶.۵ بهبود عملکرد کدگشای (LP)

قبل از اینکه برخی روش‌های بهبود عملکرد کدگشای LP بیان شود، به بررسی استفاده از کدگشای LP در دو حالت الف) کدهایی که گراف تنر متناظر با آنها دور آزاد هستند. ب) کدهایی که گراف تنر آنها دارای دور هستند، می‌پردازیم. برای یک گراف تنر فاقد دور (دور آزاد)، می‌توان نشان داد که هر جواب بهینه کدگشای LPعدد صحیح است. در حالی که اگر گراف فاکتور دارای دور باشد، جواب بهینه مساله کدگشای LP ممکن است عدد صحیح نباشد. به مثال زیر توجه کنید.

مثال ۱.۶.۵. [۱۳] کد هامینگ شکل ۶.۵ را در نظر بگیرید.فرض کنید بردار هزینه λ به صورت زیر تعریف شود.برای راس متغیر ۱ ، $\lambda_۱ = -\frac{۷}{۴}$ و برای سایر راس‌های متغیر $\{۲,\dots,۷\}$ ، $\lambda_i = +۱$. می‌توان دید که تحت این تابع هزینه ، همه کدواژه‌ها هزینه نامنفی دارند، زیرا برای اینکه کدواژه‌ای، هزینه‌اش منفی شود باید $y_۱ = ۱$ از طرفی با توجه به شکل ۶.۵ حداقل ۲ تا ازمؤلفه‌های کدواژه باید ۱ باشد ($y_i = ۱$) به این ترتیب به ازای هر کدواژه حداقل هزینه کل $+\frac{۱}{۴}$ است. حال جواب کسری زیر را در نظر بگیرید: $x = (۱, ۱/۲, ۰, ۱/۲, ۰, ۰, ۱/۲)$ برای راس بررسی A قرار می‌هیم: $\omega_{A,\{۱,۲\}} = \omega_{A,\{۱,۴\}} = ۱/۲$ و برای B ، $\omega_{B,\{۲,۴\}} = \omega_{B,\{\emptyset\}}$ و در C ،

$\omega_{C,\{۴,۷\}} = \omega_{C,\emptyset}$ ، می‌توان دید که (x, ω) در همه محدودیت‌ها صدق می‌کند و هزینه این جواب $-۱/۴$ است که از هزینه هر کدواژه کمتر است.

$\diamondsuit$

دوروش بهبود عملکرد کدگشای LP

در این زیر بخش، دو روش بهبود عملکرد کدگشای LP مطرح می‌گردد. در روش اول نشان داده می‌شود ، که با استفاده از محدودیت‌های زوجیت افزونگی ممکن است بتوان جواب بهینه کسری کدگشای LP را (که موجب شکست کدگشای LP می‌شود) بعد از چند تکرار به کدواژه ML تغییر داد. روش دوم نیز بدنبال یافتن محدویت‌هایی برای برش جواب کسری LP است ولی برای رسیدن به این هدف در روش دوم، ابتدا مساله برنامه‌ریزی‌خطی جایگزینی ارائه می‌شود و سپس با دسته‌بندی کردن جواب‌های کسری مساله برنامه‌ریزی‌خطی ارائه شده، الگوریتم بهبود مطرح می‌گردد.

۱.۶.۵ بهبود عملکرد کدگشای LP با استفاده از محدودیت‌های افزونگی

اساس کار در این بخش به این صورت است که، کدگشای LP کارش را با محدودیت‌هایی که توسط معادلات بررسی‌زوجیت ساخته می‌شوند، آغاز می‌کند. سپس در صورت نیاز با ترکیب محدودیت‌های بررسی‌زوجیت مورد نیاز ، محدودیت‌های بررسی‌زوجیت افزونگی را تا هنگامی که کدواژه ML را بیابد (در صورت امکان) به مساله اضافه می‌کند. استفاده از این روش در مقایسه با حالتی که همه محدودیت‌های بررسی‌زوجیت افزونگی، از ابتدا به مساله اضافه می‌شود، ممکن است پیچیدگی کمتری ایجاد کند. قبل از ادامه بحث لازم است نکاتی بیان شود.

۱. فرض کنید x بردار ارسالی تحت کانال فاقدحافظه و λ بردار هزینه آن باشد و $H_{A,.}$ و $H_{B,.}$ سطرهایی متعلق به ماتریس بررسی‌زوجیت H باشد. داریم:

$$H_{A,.} \otimes x = ۰ \wedge \quad H_{B,.} \otimes x = ۰ \qquad (۳۸.۵)$$

$$H_{A,.} \otimes x = ۰ \wedge \quad H_{B,.} \otimes x = ۰ \wedge \quad (H_{A,.} \oplus H_{B,.}) \otimes x = ۰ \qquad (۳۹.۵)$$

(یعنی می‌توان از دو معادله (۳۸.۵) به رابطه‌ای جدید رسید)

معادله جدید $(H_{A,.} \oplus H_{B,.}) \otimes x = ۰$ محدودیت افزونگی است. که از جمع به پیمانه ۲ دو سطر ماتریس بررسی‌زوجیت حاصل شده است.

۲. مفهوم بعدی که در این بخش مورد استفاده قرار می‌گیرد، برش است. یک برش را می‌توان به عنوان نیم‌فضایی در نظر گرفت که جواب کسری را از دامنه جستجوی چندسقفی $\mathcal{P}$ حذف می‌کند. در حقیقت اگر جواب کدگشای LP، کسری باشد ، یک برش می‌تواند به منظور حذف این جواب از چندسقفی مورد استفاده در کدگشای LP بکار رود. به این ترتیب چندسقفی جدیدی تولید می‌شود و کدگشای LP می‌تواند دوباره اجرا شود. این فرآیند تا هنگامی که کدواژه ML پیدا شود یا تعداد تکرار کدگشا به کران بالای مورد نظر برسد (ممکن است در الگوریتم برای تعداد تکرار الگوریتم، کرانی درنظر گرفته شود) ادامه می‌یابد.

پیش از این در فرمولبندی جایگزین محدودیت زیر تعریف شد:

$$\sum_{j \in v} x_j - \sum_{j \in \mathcal{N}(i)-V} x_j \leq |v| - ۱ \quad \forall v \subseteq \mathcal{N}(i) \quad , \quad |v| \text{فرد} \quad ۰ \leq x_j \leq ۱ \qquad (۴۰.۵)$$

که در آن $\mathcal{N}(i)$ اندیس راس‌های متغیر مجاور با راس بررسی i و v زیرمجموعه‌ای از $\mathcal{N}(i)$ است. اگر x_k^* جواب تکرار k ام، در معادله زیر صدق کند:

$$\sum_{j \in v} x_{k,j}^* - \sum_{j \in \mathcal{N}(i)-V} x_{k,j}^* > |v| - ۱ \qquad (۴۱.۵)$$

آنگاه معادله زیر یک برش در x_k^* است:

$$\sum_{j \in v} x_j - \sum_{j \in \mathcal{N}(i)-v} x_j \leq |v| - ۱$$

هنگامی که یک محدودیت بررسی‌زوجیت افزونگی به عنوان یک برش مطرح می‌شود به آن **برش بررسی‌زوجیت افزونگی** [۱۸] (و در برخی از مقالات به آن **مجموعه ممنوع** [۱۹] (FS)) می‌گویند. در زیر بخش بعدی الگوریتمی که از این برش‌های بررسی‌زوجیت افزونگی استفاده می‌کند، ارائه می‌شود.

الگوریتم برش‌های بررسی‌زوجیت اضافی

مراحل این الگوریم به صورت زیر است:

۱. اگر جواب کسری $\hat{x}$ خروجی کدگشای LP باشد. (با فرض اینکه بردار هزینه λ باشد) آنگاه همه راس‌های متغیر متعلق به گراف فاکتور، که متناظر با نمادهای کد با مقدار عدد صحیح هستند را هرس (حذف) کنید. گراف فاکتور جدید را F بنامید.

۲. یک دور در گراف فاکتور F بیابید.

۳. جمع به پیمانه ۲ سطرهایی از ماتریس بررسی‌زوجیت که با راس‌های بررسی متعلق به دور، متناظر هستند را به مساله اضافه کنید.

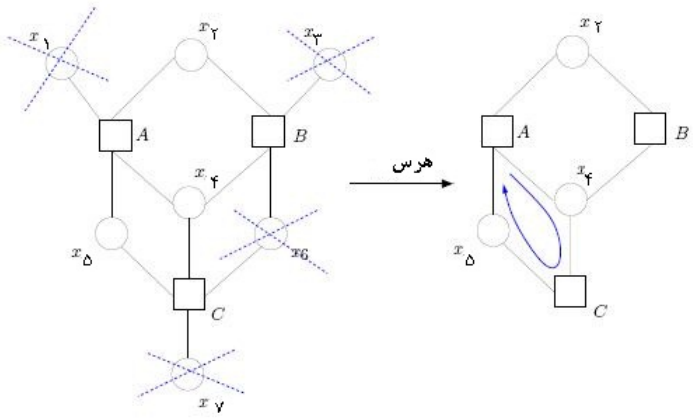
۴. اگر این محدودیت بررسی جدید یک برش را ایجاد می‌کند، کدگشای LP را ادامه دهید و به مرحله ۱ بروید در غیر این صورت به مرحله ۲ بروید.

این ۴ مرحله از قضیه زیر حاصل شده‌اند.

قضیه ۵.۶.۲. [۲۲] محدودیت بررسی‌زوجیت افزونگی می‌تواند یک برش را مطرح کند فقط اگر گراف فاکتور F شامل دور باشد.

مثال ۵.۶.۳. [۲۵] کد (۷،۴،۳) هامینگ را با بردار هزینه $\lambda = \begin{bmatrix} -۷/۴ & ۱ & ۱ & ۱ & ۱ & ۱ & ۱ \end{bmatrix}^T$ در نظر بگیرید. $x = \begin{bmatrix} ۰ & ۰ & ۰ & ۰ & ۰ & ۰ & ۰ \end{bmatrix}$ فرض کنید کانال مورد نظر AWGNC دودویی باشد. خروجی

[۱۸] Redundant parity check cut
[۱۹] Forbidden Set



شکل ۷.۵: گراف فاکتور جدید

کدگشای LP در ابتدا (پس از حل مساله LP)جواب کسری

$$x^*_{\circ} = \begin{bmatrix} ۱ & ۱/۳ & ۰ & ۱/۳ & ۱/۳ & ۰ & ۰ \end{bmatrix}^T$$

است. متناظر با مرحله اول الگوریتم بالا، راس‌های متغیری که جواب صحیح دارند عبارتند از $x_۱$ ، $x_۳$ ، $x_۶$ و $x_۷$ که باید از گراف فاکتور متناظر با این کد هامینگ حذف شود. شکل ۷.۵ را ببینید. با توجه به دور یافت شده، حاصل جمع دو معادله A و C محاسبه می‌شود. یعنی:

$$chk(A) \oplus chk(C) = x_۱ \oplus x_۲ \oplus x_۶ \oplus x_۷ \qquad (۴۲.۵)$$

بررسی خواهیم کرد که آیا این محدودیت بررسی‌زوجیت افزونگی، یک برش تولید می‌کند؟ یعنی باید به بررسی نقض محدودیت (۴۰.۵) بپردازیم. برای این منظور قرار می‌دهیم،

$$x = \begin{bmatrix} x_۱ & x_۲ & x_۳ & x_۴ & x_۵ & x_۶ & x_۷ \end{bmatrix}^T = \begin{bmatrix} ۱ & ۱/۳ & ۰ & ۱/۳ & ۱/۳ & ۰ & ۰ \end{bmatrix}^T$$

بنابراین با توجه به محدودیت (۴۰.۵) داریم:

$$\begin{bmatrix} ۱ & -۱ & ۰ & ۰ & ۰ & -۱ & -۱ \\ -۱ & ۱ & ۰ & ۰ & ۰ & -۱ & -۱ \\ -۱ & -۱ & ۰ & ۰ & ۰ & ۱ & -۱ \\ -۱ & -۱ & ۰ & ۰ & ۰ & -۱ & ۱ \\ ۱ & ۱ & ۰ & ۰ & ۰ & ۱ & -۱ \\ ۱ & ۱ & ۰ & ۰ & ۰ & -۱ & ۱ \\ ۱ & -۱ & ۰ & ۰ & ۰ & ۱ & ۱ \\ -۱ & ۱ & ۰ & ۰ & ۰ & ۱ & ۱ \end{bmatrix} \begin{bmatrix} x_۱ \\ x_۲ \\ x_۳ \\ x_۴ \\ x_۵ \\ x_۶ \\ x_۷ \end{bmatrix} \leq \begin{bmatrix} ۰ \\ ۰ \\ ۰ \\ ۰ \\ ۲ \\ ۲ \\ ۲ \end{bmatrix} \qquad (۴۳.۵)$$

با قرار دادن x در (۴۳.۵) همانطور که می‌بینید سطر اول برقرار نیست: $۱(۱)+۱/۳(-۱) = ۲/۳ \nleq ۰$ بنابراین این محدودیت بعنوان یک برش (برش ۱ در شکل ۸.۵) مطرح می‌شود. حال این محدودیت را به محدودیت‌های مساله اضافه می‌کنیم و مساله را دوباره حل می‌کنیم. جواب حاصل از حل دوباره مساله نیز جواب کسری (اما متفاوت) است، $x_۱^* = [\ ۰ \quad ۱/۳ \quad ۰ \quad ۱/۳ \quad ۰ \quad ۲/۳ \quad ۱\ ]^T$. با توجه به اینکه جواب تغییر کرده است محدودیت اضافه شده واقعا یک برش است، چون جواب هنوز کسری است، بنابراین دوباره به جستجوی یک برش جدید می‌پردازیم. بعد از ۴ بار تکرار که در شکل ۸.۵ و جدول زیر این مراحل نشان داده شده است، خروجی کدگشای LP در نهایت کدواژه ML، $x_۴^* = [\ ۰ \quad ۰ \quad ۰ \quad ۰ \quad ۰ \quad ۰ \quad ۰\ ]^T$ است. به این ترتیب کدگشای LP با استفاده از الگوریتم بهبود موفق می‌شود.

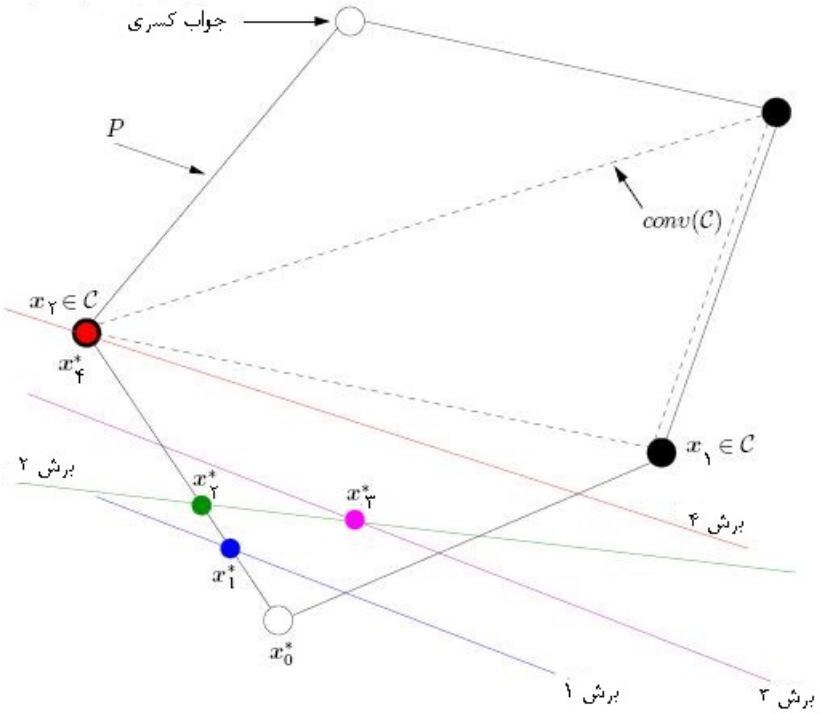
تکرار	برش بررسی زوجیت افزونگی	معادله بررسی زوجیت حاصل از دور	جواب LP x_i^*
۰			$[\ ۱ \quad ۱/۳ \quad ۰ \quad ۱/۳ \quad ۱/۳ \quad ۰ \quad ۰\ ]^T$
۱	$x_۱ - x_۲ - x_۶ - x_۷ \leq ۰$	$chk(A) \oplus chk(C)$	$[\ ۱ \quad ۲/۳ \quad ۰ \quad ۱/۳ \quad ۰ \quad ۱/۳ \quad ۰\ ]^T$
۲	$x_۲ - x_۳ - x_۵ - x_۷ \leq ۰$	$chk(B) \oplus chk(C)$	$[\ ۱ \quad ۱/۲ \quad ۰ \quad ۱/۲ \quad ۰ \quad ۰ \quad ۱/۲\ ]^T$
۳	$x_۱ - x_۳ - x_۵ - x_۶ \leq ۰$	$chk(A) \oplus chk(B)$	$[\ ۱ \quad ۱/۲ \quad ۰ \quad ۰ \quad ۱/۲ \quad ۱/۲ \quad ۰\ ]^T$
۴	$x_۱ - x_۳ - x_۴ - x_۷ \leq ۰$	$chk(A) \oplus chk(B) \oplus chk(C)$	$[\ ۰ \quad ۰ \quad ۰ \quad ۰ \quad ۰ \quad ۰ \quad ۰\ ]^T$

$\diamondsuit$

روش بهبود دوم (الگوریتم جداسازی جدید)

روش دوم نیازمند بیان مقدماتی است که در ادامه مطرح می‌شود.

تعریف ۴.۶.۵. [۲۳] چندوجهی گویا و کراندار $P \in \mathbb{R}^n$ و بردار گویا $x^* \in \mathbb{R}^n$ را در نظر بگیرید، می‌توان نتیجه گرفت که $x^* \in P$ یا (اگر نباشد) یک بردار گویا $(\prod, \prod_۰) \in \mathbb{R}^n \times \mathbb{R}$ وجود دارد به طوری که به ازای



شکل ۸.۵: شرح مثال ۳.۶.۵

هر $x \in P$ ، $\prod^T x \leq \prod_۰$ و $\prod^T x \leq \prod_۰ x^*$. در حالت دوم ($x^* \notin P$) $(\prod_۰, \prod_۰)$ را **برش قابل‌پذیرش** [۲۰] می‌گویند.

گزاره ۵.۶.۵. [۲۳] (این گزاره بیان دیگری از قضیه (۴.۵.۵) است) نامساوی‌های

$$\sum_{j \in v} x_j - \sum_{j \in \mathcal{N}(i)-v} x_j \leq |v| \qquad \forall v \in \sum_i \tag{۴۴.۵}$$

که $\sum_i = \{v \subseteq \mathcal{N}_i : \ |v|$فرد$\}$ حاصل از سطر $r(i)$ ($i = \{۱, \ldots, m\}$) ماتریس بررسی‌زوجیت H و نامساوی‌های $۰ \leq x \leq ۱$ ، به طور کامل پوسته محدب $conv(\mathcal{C}_i)$ چندسقفی کدواژه محلی $\mathcal{C}_i$ را شرح می‌دهند.

اثبات. در قضیه ۴ مرجع [۱۱] بیان شده است.

□

[۲۰] Valid cut

در الگوریتم‌های جداسازی (مانند الگوریتمی که در روش بهبود اول بیان شد) خانواده Λ از برش‌های قابل‌پذیرش محاسبه می‌شوند (تا هنگامی که برش دیگری وجود نداشته باشد). الگوریتم جداسازی جدیدی که در این بخش بیان خواهد شد، براساس فرمولبندی برنامه‌ریزی‌خطی IP جایگزینی است که در [۲۳]بیان شده است. این برنامه‌ریزی‌خطی جایگزینی به صورت زیر است:

$$\begin{array}{lll} \min & \lambda^T x & (IPD) \\ s.t. & Hx - ۲z = ۰ & \\ & x \in \{۰, ۱^n\} & \\ & z \geq ۰, \text{صحیح} & \end{array} \qquad (۴۵.۵)$$

در (۴۵.۵)، متغیر کمکی $z \in \mathbb{Z}^m$ موجب می‌شود که محدودیت دودویی $Hx = ۰$ که تحت $F_۲$ ($GF(۲)$) است به محدودیتی تحت میدان $\mathbb{R}$ تبدیل شود. علاوه‌براین، این عمل موجب می‌شود که تعداد محدودیت‌ها همان تعداد سطرهای ماتریس بررسی‌زوجیت باشد. در این روش ابتدا LP کاهشی حاصل از IPD که در (۴۶.۵) بیان شده است حل می‌شود.

$$\begin{array}{lll} \min & \lambda^T x & (RIPD) \\ s.t. & Hx - ۲z = ۰ & \\ & \prod^T x \leq \prod_۰ \quad (\prod, \prod_۰) \in \Lambda & \\ & ۰ \leq x \leq ۱ & \\ & z \geq ۰ & \end{array} \qquad (۴۶.۵)$$

باید به این نکته توجه کرد که در مرحله آغازی هیچ برشی از نوع $\prod^T x \leq \prod_۰$ وجود ندارد. به عبارت دیگر $\Lambda = \emptyset$. اگر RIPD جواب صحیح $(x^*, z^*) \in \mathbb{Z}^{n+m}$ داشته باشد، آنگاه x^* ، کدواژه ML است. در غیر این صورت برش‌هایی از نوع $\prod^T x \leq \prod_۰$ به منظور حذف جواب غیرصحیح گویا یافت شده (در جریان حل مساله یا تکرار) ساخته می‌شوند. این برش‌ها به عنوان محدودیت‌های جدید به مساله اضافه می‌شوند و RIPD حاصل دوباره حل می‌گردد. هنگامی که جواب RILP غیرصحیح گویا است ، x^* یا z^* (یا هر دو) غیر صحیح گویا است. اگر $x^* \in \mathbb{Z}^n$ و $z^* \in \mathbb{R}^m - \mathbb{Z}^m$ باشد آنگاه برش‌های گوموری مورد استفاده قرار می‌گیرند که این روش یک تکنیک ساخت برش است که در برنامه‌ریزی عدد صحیح مورد استفاده قرار می‌گیرد. در این حالت برش‌های گوموری را می‌توان همان محدودیت‌های (۴۰.۵) در نظر گرفت. به قضیه زیر توجه کنید.

قضیه ۵.۶.۶. [۲۳] فرض کنید $(x^*, z^*) \in \mathbb{Z}^n \times \mathbb{R}^m$ جواب بهینه RIPD باشد به طوری که به ازای $j \in \mathcal{J}$، $z_j^* \in \mathbb{R} - \mathbb{Z}$. آنگاه برش گوموری که توسط (x^*, z^*) نقض می‌شود نامساوی (۴۷.۵) یعنی:

$$\sum_{j \in v} x_j - \sum_{j \in \mathcal{N}(i) - v} x_j \leq |v| - ۱ \qquad (۴۷.۵)$$

است. که $v := \{i \in \mathcal{N}(i) : \ x_j^* = ۱\}$.

فرض کنید جواب بهینه RIPD، (x^*, z^*) باشد به طوری که $x_j^* \in \{۰, ۱\}$ و به ازای حداقل یک $j \in \mathcal{J}$، $z_j^* \in \mathbb{R} - \mathbb{Z}$ ، برش گوموری مورد نظر را می‌توان با استفاده از الگوریتم CGA1 بدست آورد.

الگوریتم ساخت برش [۲۳] (CGA1)

ورودی : (x^*, z^*) به طوری که x^* صحیح باشد و z^* غیرصحیح باشد.

خروجی : برش‌های گوموری.

۱. به ازای $i = ۱$ تا m مراحل زیر را انجام دهید:

- اگر $k_i = ۲z_i^*$ فرد است آنگاه:
  - قرار دهید: $v := \{j \in \mathcal{N}(i) : x_j^* = ۱\}$
  - محدودیت (۴۷.۵) را بسازید.

پیچیدگی این الگوریتم $o(m\delta^{max})$ است (با فرض اینکه δ^{max} بزرگترین درجه راس بررسی باشد). زیرا برای یافتن محدودیت بررسی‌زوجیت نقض شده، حداکثر m دفعه جستجو انجام می‌گیرد و $o(\delta^{max})$ پیچیدگی ساخت محدودیت (۴۷.۵) است. محدودیت‌های بررسی‌زوجیت نقض شده با استفاده از متغیر نشانگر z تعیین می‌شود. به این ترتیب که، فرض کنید محدودیت نقض شده، محدودیت i ام باشد (یعنی با توجه به اینکه اگر $z_i \in \mathbb{Z}$ باشد آنگاه $۲z_i$ حتما زوج است، پس در صورتی ممکن است $۲z_i$ فرد باشد، که $z_i \in \mathbb{R} - \mathbb{Z}$ باشد). محدودیت (۴۷.۵) را می‌توان

به این صورت ساخت که: ضریب x_j برای $\{j \in \mathcal{N}(i) : \ x_j^* = ۱\}$ ، $+۱$ و برای $\{j \in \mathcal{N}(i) : \ x_j^* = ۰\}$ ، -۱ قرار گیرد به طوری که $|v| = k_i$.

حالت دیگری که باید در نظر گرفته شود، حالتی است که $\forall j \in \mathcal{J}$ ، $۰ < x_j^* < ۱$. اگرچه در این روش نیز می‌توان از برش گوموری استفاده کرد ولی به الگوریتم جدیدی نیاز داریم. این الگوریتم از گزاره‌های (۷.۶.۵) و (۸.۶.۵) حاصل می‌شود.

گزاره ۷.۶.۵. [۲۳] فرض کنید x^* جواب بهینه غیر صحیح RIPD و $x^* \in conv(\mathcal{C}(i))$ آنگاه حداقل دو اندیس $j,k \in N(i)$ وجود دارد به طوری که $۰ < x_j < ۱$ و $۰ < x_k < ۱$. به عبارت دیگر راس بررسی i نمی‌تواند فقط با یک راس متغیر با مقدار غیر صحیح مجاور باشد.

کد خطی $\mathcal{C}$ را می‌توان با برخی جایگزینی‌ها و تغییرات با استفاده از ماتریس بررسی‌زوجیت هم ارز H که با $\widehat{H}$ نمایش می‌دهیم، ساخت. چنین ماتریس بررسی‌زوجیتی را می‌توان با استفاده از اعمال سطری مقدماتی بر روی H ساخت. گزاره (۵.۶.۵) برای $\widehat{H}$ برقرار است. همچنین گزاره (۷.۶.۵) به ازای هر راس بررسی‌زوجیت $i \in \{۱,\dots,m\}$ متعلق به گراف تنر متناظر با $\widehat{H}$ برقرار است. نکته‌ای باید به آن توجه کرد این است که سطرهای $\widehat{H}$ (که در حقیقت **کدواژه‌های دوگان** [۲۱] هستند) ممکن است به عنوان محدودیت‌های بررسی‌زوجیت افزونگی تفسیر شوند. بنابراین نامساوی‌های (۴۴.۵) که از سطرهای ماتریس بررسی‌زوجیت جایگزین ($\widehat{H}$) حاصل می‌شوند، را **نامساوی‌های RPC** [۲۳] می‌نامند. و برش‌های حاصل از آنها را **برش‌های نامساوی‌های RPC** می‌نامند. (یک مساله حل نشده که در[۲۳] مطرح شده است، یافتن کدواژه‌های دوگان RPC ای است که محدودیت‌های حاصل از آنها برش RPC باشد) قبل از اینکه الگوریتم (CA) را که از دو ایده برمبنای اندازه مجموعه $A_i = \{j \in J : \ \widehat{H}_{i,j} = ۱, \ ۰ < x_j^* < ۱\}$ (به عبارت دیگر A_i مجموعه اندیس‌های مؤلفه‌های x^* با مقدار غیرصحیح گویا (متعلق به $\widehat{H}_{i,.}$) در جواب بهینه (x^*, z^*) است) ساخته می‌شود را بیان کنیم، حالتی را که $|A_i| = ۱$ و $|A_i| > ۱$ را شرح می‌دهیم.

[۲۱] Dual codeword

- حالتی را در نظر بگیرید که $|A_i| = ۱$. جواب بهینه غیر صحیح x^* مساله RIPD داده شده است. در الگوریتم CGA2 در جستجوی راس بررسی‌زوجیتی هستیم که فقط با یک راس متغیر با مقدار غیر صحیح مجاور است. اگر چنین راسی بررسی‌زوجیتی پیدا شود با توجه به گزاره (۷.۶.۵) می‌گوییم که x^* نمی‌تواند متعلق به پوسته محدب حاصل از این راس بررسی‌زوجیت (چندسقفی محلی) باشد. علاوه‌براین با توجه به گزاره (۵.۶.۵) نامساوی RPC وجود دارد که برشی برای x^* است. این مشاهدات در قضیه زیر بیان شده اند.

قضیه ۸.۶.۵. [۲۳] فرض کنید $(x^*, z^*) \in \mathbb{R}^n \times \mathbb{R}^m$ جواب بهینه مساله RIPD باشد به طوری که x^* غیرصحیح باشد و $\widehat{H}$ ماتریس بررسی‌زوجیت کد $\mathcal{C}$ باشد. اگر $i \in \mathcal{I}$ وجود داشته باشد به طوری که دقیقا برای یک $j \in \widehat{N}_i$ ، x^*_j صحیح نباشد. قرار دهید $k_i = |\{h \in \widehat{N}_i | x^*_h = ۱\}|$. آنگاه اگر k_i فرد باشد،

$$\sum_{h \in \widehat{N}_i:\ x^*_h = ۱} x_h - x_j - \sum_{h \in \widehat{N}_i:\ x^*_h = ۰} x_h \leq k_i - ۱ \qquad (۴۸.۵)$$

یا اگر k_i زوج باشد، آنگاه

$$\sum_{h \in \widehat{N}_i:\ x^*_h = ۱} x_h + x_j - \sum_{h \in \widehat{N}_i:\ x^*_h = ۰} x_h \leq k_i \qquad (۴۹.۵)$$

نامساوی قابل‌پذیرش است که توسط x^* نقض شده است.

به این ترتیب اگر $|A_i| = ۱$ ، (۴۸.۵) یا (۴۹.۵) جواب بهینه را حذف می‌کنند.

- در حالت بعدی، ممکن است به ازای هر سطر $i \in \mathcal{I}$ از $\widehat{H}$ حداقل دو $j \in \widehat{N}_i$ وجود داشته باشد به طوری که x^*_j غیرصحیح باشد ($|A_i| > ۱$) . این حالت در گزاره‌های (۹.۶.۵) و (۱۰.۶.۵) مشخص می‌گردد.

گزاره ۹.۶.۵. [۲۳] فرض کنید به ازای هر سطر $i \in \mathcal{I}$ از $\widehat{H}$ حداقل دو $j \in \widehat{N}_i$ وجود داشته باشد به طوری که x^*_j غیرصحیح باشد ($|A_i| > ۱$)، اگر k_i (که در گزاره قبل تعریف شد) فرد باشد و

آنگاه $\sum_{j\in A_i} x_j^* < ۱$

$$\sum_{h\in \widehat{N}_i:\ x_h^*=۱} x_h - \sum_{j\in A_i} x_j - \sum_{h\in \widehat{N}_i:\ x_h^*=۰} x_h \leq k_i - ۱ \qquad (۵۰.۵)$$

یک برش قابل‌پذیرش است که x^* را حذف می‌کند.

فرض کنید x^* غیر صحیح باشد و j^{max} ، اندیس ماکزیمم مؤلفه ناصحیح متعلق به A_i باشد. یعنی:

$$x_{j^{max}}^* = max\{x_j^* :\ j \in A_i\}$$

گزاره ۱۰.۶.۵. [۲۳] فرض کنید x^* غیر صحیح باشد و j^{max} ، اندیس ماکزیمم مؤلفه ناصحیح متعلق به A_i باشد اگر k_i زوج باشد و $x_j^* > \sum_{j\in A_i-\{j^{max}\}}$ ، آنگاه

$$\sum_{h\in \widehat{N}_i:\ x_h^*=۱} x_h + x_{j^{max}} - \sum_{j\in A_i-\{j^{max}\}} x_j - \sum_{h\in \widehat{N}_i:\ x_h^*=۰} x_h \leq k_i \qquad (۵۱.۵)$$

یک برش قابل‌پذیرش باشد، که x^* را حذف می‌کند.

جستجوی همه برش‌های RPC در حالت کلی رام‌نشدنی است. بنابراین الگوریتم ساخت $\widehat{H}$ که در ادامه بیان می‌شود، به این جستجو کمک خواهد کرد. در این الگوریتم ماتریس H توسط اعمال سطری مقدماتی (تحت $GF(۲)$) به ماتریس هم ارز $\widehat{H}$ تبدیل می‌گردد. هدف (حالت مطلوب) ارائه کد $\mathcal{C}$ با ماتریس بررسی‌زوجیت جایگزینی $\widehat{H}$ است، به طوری که در سطر $\widehat{H}_{i,.}$ دقیقا یک $j \in \widehat{N}_i$ وجود داشته باشد که x_j^* غیرصحیح باشد. الگوریتم ساخت $\widehat{H}$ سعی خواهد کرد که ستون j ماتریس H که x_j^* غیرصحیح گویا است را به بردار واحد تبدیل کند. (حداکثر m ستون از H تبدیل می‌شود)

الگوریتم ساخت $\widehat{H}$ (CA) [۲۳]

ورودی : $(x^*, z*)$ به طور که x^* غیرصحیح باشد.

خروجی : $\widehat{H}$.

۱. قرار دهید $l = ۱$.

۲. برای $j = ۱$ تا n مراحل ۳ تا ۶ را انجام دهید:

۳. اگر $x_j^* \in (۰, ۱)$ آنگاه برو به ۴، در غیر این صورت برو به ۵.

۴. اگر $l \leq m$ آنگاه اعمال سطری مقدماتی را انجام دهید تا هنگامی که $H_{l,j} = ۱$ و به ازای هر $i \in \mathcal{I} - l$ ، $H_{i,j} = ۰$ و برو به ۵ .

۵. قرار دهید $l = l + ۱$

۶. اگر $l > m$ آنگاه پایان .

۷. پایان الگوریتم.

$\widehat{H}$ را می‌توان با پیچیدگی $o(m^۲n)$ محاسبه کرد.

در ادامه به بیان الگوریتم ساخت برش‌های RPC می‌پردازیم.

الگوریتم ساخت برش‌های RPC (CGA2)

[۲۳] **ورودی:** $(x^*, z^*) \in R^{n+m}$ به طوری که x^* غیرصحیح گویا باشد و $\widehat{H}$.

خروجی: نامساوی FS جدید یا خطا.

۱. برای $i = ۱$ تا m گام‌های ۲ تا ۵ را انجام دهید.

۲. مقادیر k_i ، A_i و j^{max} را بیابید.

۳. اگر $|A_i| = ۱$ آنگاه اگر k_i فرد است محدودیت (۴۸.۵) و اگر زوج است محدودیت (۴۹.۵) را محاسبه کنید، پایان .

۴. در غیر این صورت اگر $|A_i| \geq ۲$ ، k_i فرد است و $\sum_{j \in A_i} x_j^* < ۱$ آنگاه محدودیت (۵۰.۵) را محاسبه کنید، پایان .

۵. در غیر این صورت اگر $|A_i| \geq ۲$ ، k_i زوج باشد و $x_j^* > \sum_{j \in A_i - \{j^{max}\}} x_{j^{max}}^*$ آنگاه محدودیت (۵۱.۵) را محاسبه کنید، پایان .

۶. اگر هیچ برشی پیدا نشده است، از الگوریتم خارج شوید، خطا رخ داده است.

پیچیدگی این الگوریتم $o(mn)$ است زیرا در بدترین حالت ورودی $\hat{H}$ فقط یکبار ملاقات می‌شود.

با توجه به مطالب بیان شده حال می‌توان الگوریتم جداسازی را مطرح کرد. در تکرار اول x^* را می‌توان با اتخاذ تصمیم سخت یافت، یعنی اگر $\lambda_j < ۰$ به ازای هر j، x_j برابر با یک قرار داده می‌شود. در یک تکرار مساله $RIPD$ اگر (x^*, z^*) صحیح باشد آنگاه جواب بهینه IPD یافت شده است. اگر x^* صحیح و z^* غیرصحیح گویا باشد الگوریتم $CAG۱$ اجرا می‌شود. با اجرای این الگوریتم محدویت FS ساخته می‌شود. اگرچه اضافه کردن یک محدودیت FS برای برش جواب غیرصحیح کافی است، ولی در این الگوریتم به ازای هر i که z_i غیرصحیح است، محدودیت FS مربوط به آن i به مساله اضافه می‌گردد زیرا ممکن است در تکرارهای بعدی الگوریتم مورد استفاده قرار گیرد. اگر x^* غیرصحیح باشد ابتدا الگوریتم ساخت $\hat{H}$ باید اجرا گردد سپس الگوریتم CGA2 اجرا می‌گردد. اگر یک برش RPC در این الگوریتم پیدا نشد، الگوریتم خطا را برمی‌گرداند. و اگر بیش از یک برش RPC توسط الگوریتم ساخته شود همه این برش‌های جدید را به فرمول RIPD (با توجه به اینکه ممکن است در تکرارهای بعدی مورد استفاده قرار گیرد) اضافه می‌گردد. الگوریتم جداسازی جدید هنگامی متوقف می‌شود که کدواژه ML حاصل شود یا الگوریتم خطا را بازگرداند (به این معنا که برش جدیدی وجود ندارد) .

الگوریتم جداسازی جدید

[۲۳] **ورودی :** بردار هزینه λ و ماتریس H

خروجی : جواب بهینه x^*.

۱. مساله RIPD را حل کنید.

۲. اگر (x^*, z^*) صحیح است آنگاه پایان الگوریتم. در غیر این صورت:

- اگر x^* صحیح است آنگاه الگوریتم (CGA1) را اجرا کنید و محدودیت حاصل از الگوریتم (CGA1) را به مساله RIPD اضافه کنید و به گام ۱ بروید.
- در غیر این صورت الگوریتم ساخت $\hat{H}$ را اجرا کنید و سپس الگوریتم (CGA2) را اجرا کنید. اگر یک برش پیدا شد آنگاه محدودیت مربوط به آن برش را به مساله RIPD اضافه کنید و به گام ۱ بروید. در غیر این صورت پایان الگوریتم .

۷.۵ تحلیل کدگشای LP

هنگامی که از کدگشای LP استفاده می‌شود در دو حالت ممکن است خطا رخ دهد:

۱. هنگامی که جواب LP صحیح نباشد. در این حالت خروجی کدگشای LP خطا است.

۲. جواب بهینه کدگشای LP کامل باشد که در نتیجه کدواژه ML است ولی این کدواژه ML همان کدواژه ارسالی نباشد. در این حالت خود کد شکست خورده است. بنابراین حتی کدگشای ML دقیق ممکن است دچار خطا شود.

با فرض اینکه بدانیم x کدواژه ارسالی است. از مفهوم $P(err|x)$ برای نمایش احتمال خطای کدگشای LP استفاده می‌شود. براساس گزاره (۲.۴.۵) متناظر با کدواژه ارسالی y ، جواب شدنی (y, ω^y) وجود دارد با استفاده از این مطلب می‌توان شرایطی را که تحت آن کدگشای LP موفق خواهد بود را بیان کرد.

قضیه ۱.۷.۵. [۱۱] فرض کنید کدواژه y ارسال شده باشد. اگر همه جواب‌های شدنی کدگشای LP (به جز (y, ω^y)) هزینه‌ای بیشتر از هزینه (y, ω^y) داشته باشند، آنگاه کدگشای LP موفق خواهد شد. در غیر این صورت این کدگشا شکست خواهد خورد.

فلدمن در حالتی که مساله، جواب بهینه چندگانه دارد (با توجه به اینکه ممکن است کدگشا موفق باشد یا خیر) را به عنوان شکست در نظر می‌گیرد. با توجه به قضیه (۱.۷.۵) می‌توان احتمال شکست خوردن کدگشای LP را با استفاده از

$$WER(y) = P(err|y) = P(\exists (f,\omega) \in \mathcal{P}, f \neq y \ : \ \sum_j \lambda_j f_j \leq \sum_j \lambda_j y_j \ ) \qquad (۵۲.۵)$$

(نرخ خطای واژه) مورد بررسی قرار داد.

۱.۷.۵ فرض صفر

همانطور که در بخش (۵.۴) نیز بیان شد، هنگام تحلیل کدهای خطی یک فرض رایج این است که کدواژه ارسالی (تحت کانال) را کدواژه صفر (یعنی برداری به طول n با عناصر صفر) در نظر می‌گیرند. با توجه به قضیه‌ای که در ادامه بیان می‌شود می‌توان دید که این فرض را می‌توان برای کدگشای LP نیز بکار برد.

قضیه ۲.۷.۵. [۱۱] احتمال شکست کدگشای LP مستقل از کدواژه ارسالی است.

در اثبات این قضیه که در ضمیمه مرجع [۱۳] بیان شده است از مفهوم چند سقفی متقارن (و این نکته که چندسقفی $\mathcal{P}$ نیز متقارن است) استفاده شده است به منظور آشنایی با این مفهوم در ادامه به بیان خلاصه‌ای از مفهوم چندسقفی متقارن می‌پردازیم.

به ازای هر نقطه $f \in [۰ \ , \ ۱]^n$ **نقطه نسبی**[۲۲] f نسبت به کدواژه y که با $f^{[y]} \in [۰ \ , \ ۱]^n$ نمایش داده می‌شود عبارت است از:

$$\forall j = ۱, ۲, \ldots, n \ : \ f_j^{[y]} = |f_j - y_j|$$

به این ترتیب داریم: $(f^{[y]})^{[y]} = f$

تعریف ۳.۷.۵. [۱۰] چندسقفی محض $\mathcal{P}$ برای کد خطی دودویی $\mathcal{C}$ ، چندسقفی $\mathcal{C}$-متقارن است اگر به ازای هر نقطه $f \in \mathcal{P}$ و کدواژه $y \in \mathcal{C}$ ، نقطه نسبی $f^{[y]}$ نیز متعلق به $\mathcal{P}$ باشد.

[۲۲] Relative point

قضیه ۴.۷.۵. [۱۰] اگر چندسقفی $\mathcal{P}$ یک چندسقفی محض برای کد $\mathcal{C}$ باشد و $\mathcal{P}$ ، $\mathcal{C}$-متقارن باشد، آنگاه $\mathcal{C}$ باید خطی باشد.

اثبات. برای اثبات کافی است نشان دهیم که کدواژه صفر متعلق به کد $\mathcal{C}$ است و $\forall y, y' \in \mathcal{C} \ , \ y - y' \in \mathcal{C}$. اثبات کامل را می‌توان در [۱۰] مشاهده کرد.

□

قضیه ۵.۷.۵. [۱۰] برا ی هر کدخطی دودویی $\mathcal{C}$ ، چندسقفی تصویر $\bar{\mathcal{P}}$ ، چندسقفی $\mathcal{C}$-متقارن است.

با استفاده از قضیه (۲.۷.۵) می‌توان نتیجه (۶.۷.۵) را بیان کرد.

نتیجه ۶.۷.۵. [۱۱] فرض کنید کدواژه صفر $\mathbf{0}$ ، کدواژه ارسالی باشد. کدگشای LP شکست می‌خورد اگر نقطه‌ای در $\mathcal{P}$ وجود داشته باشد (غیر از $(۰^n, \omega^n)$ که منظور از ۰^n همان کدواژه صفر $\mathbf{0}$ و ω^n برداری از متغیرهای کمکی به طول n است) که هزینه آن نقطه کوچکتر یا مساوی با هزینه کدواژه صفر باشد.

۲.۷.۵ کدواژه‌های کاذب

پیش از این با مفهوم کدواژه‌های کاذب در کدگشایی عبور پیام آشنا شدیم . حال به بیان کدواژه کاذب با توجه به کدگشای LP می‌پردازیم. همانطور که پیش از این بیان شد، کدواژه‌های کدخطی دودویی $\mathcal{C}$ راس‌های چندسقفی محض $\mathcal{P}$ هستند یعنی:

$$\mathcal{C} \subseteq V(\mathcal{P}) \subseteq \mathcal{P} \subseteq \{۰, ۱\}^n$$

نکته‌ای که باید به آن توجه کرد این است که در چندسقفی محض $\mathcal{P}$ علاوه‌بر راس‌های صحیح (کدواژه‌ها)، راس‌های کسری (غیر صحیح) نیز وجود دارند. در کدگشای LP، راس‌ها نقش کدواژه‌های کاذب را ایفا می‌کنند و در حقیقت کدواژه‌ها زیر مجموعه‌ای از کدواژه‌های کاذب هستند. به این ترتیب، همانطور که مجموعه کدواژه‌ها توسط کد تولید می‌شوند، مجموعه کدواژه‌های کاذب توسط الگوریتم کدگشایی زیر-بهینه مورد استفاده، حاصل

می‌شود (همانطور که قبلا بیان شد اگر گراف فاکتور فاقد دور (دور آزاد) باشد جواب کدگشای LP بهینه خواهد بود در غیر این صورت زیر بهینه خواهد بود). کدواژه‌های کاذب وابسته به یک کدگشای زیر-بهینه در تحلیل WER کدگشا مفید هستند.

قضیه ۵.۷.۷. [۱۰] فرض کنید کدواژه صفر، کدواژه ارسالی باشد. کدگشای LP شکست خواهد خورد اگر و فقط اگر کدواژه کاذبی وجود داشته باشد که مقدار تابع هدف به ازای آن (کدواژه کاذب غیر صفر) کوچکتر یا مساوی با صفر باشد.

۳.۷.۵ فاصله کسری

همانطور که پیش از این نیز بیان شد در یک چندسقفی محض $\mathcal{P}$ ، تناظر یک به یکی بین راس‌های صحیح متعلق به $\mathcal{P}$ و کدواژها وجود دارد یعنی: $\mathcal{C} = \mathcal{P} \cap \{۰, ۱\}^n$ فلدمن براساس چندسقفی محض $\mathcal{P}$ ، مفهومی به نام **فاصله کسری**[۲۳] را به عنوان جایگزینی برای فاصله هامینگ (کلاسیک) مطرح می‌کند. در حقیقت او فاصله $l_۱$ بین دو نقطه متعلق به فضای $[۰ \ , \ ۱]^n$ را معادل فاصله بین دو نقطه متعلق به فضای گسسته $\{۰, ۱\}^n$ می‌داند. به این ترتیب با فرض معین بودن چندسقفی محض $\mathcal{P}$ ، فاصله کد را به عنوان مینیمم فاصله $l_۱$ بین دو راس صحیح متعلق به چندسقفی $\mathcal{P}$ معرفی می‌کند. یعنی :

$$d = \begin{array}{l} min \sum_{j=۱}^{n} |y_j - y'_j| \\ y, y' \in V(\mathcal{P} \cap \{۰, ۱\}^n) \\ y \neq y' \end{array} \qquad (۵۳.۵)$$

با توجه به اینکه چندسقفی محض $\mathcal{P}$ شامل راس‌های کسری نیز هست ، فلدمن فاصله کسری d_{frac} چندسقفی $\mathcal{P}$ را به عنوان مینیمم فاصله $l_۱$ بین یک راس صحیح (کدواژه) و هر راس دیگر چندسقفی $\mathcal{P}$ (کدواژه‌های کاذب

[۲۳] Fractional distance

) تعریف می‌کند. به بیان دیگر :

$$d_{frac} = \begin{array}{l} min \sum_{j=1}^{n} |y_j - f_j| \\ y \in \mathcal{C} \\ f \in V(\mathcal{P}) \\ f \neq y \end{array} \quad (۵۴.۵)$$

باید توجه داشت که فاصله کسری همیشه کران بالایی برای فاصله کلاسیک کد است. زیرا هر کدواژه یک راس چندسقفی $\mathcal{P}$ است. از مفهوم فاصله کسری می‌توان برای تحلیل کارایی کدگشای LP استفاده کرد. به قضیه زیر توجه کنید:

قضیه ۸.۷.۵. [۱۳] فرض کنید l یک کد دودویی و $\mathcal{P}$ چندسقفی محض مربوط به کدگشای LP کاهشی کد $\mathcal{C}$ باشد. اگر فاصله کسری $\mathcal{P}$ ، d_{frac} باشد آنگاه کدگشای LP با استفاده از $\mathcal{P}$ موفق خواهد شد اگر حداکثر $\lceil d_{frac}/۲ \rceil - ۱$ بیت توسط کانال متقارن دودویی (BSC) دچار خطا باشد (منظور از $\lceil . \rceil$ جز صحیح است)

فصل ۶

نتایج عددی

۱.۶ مقدمه فصل

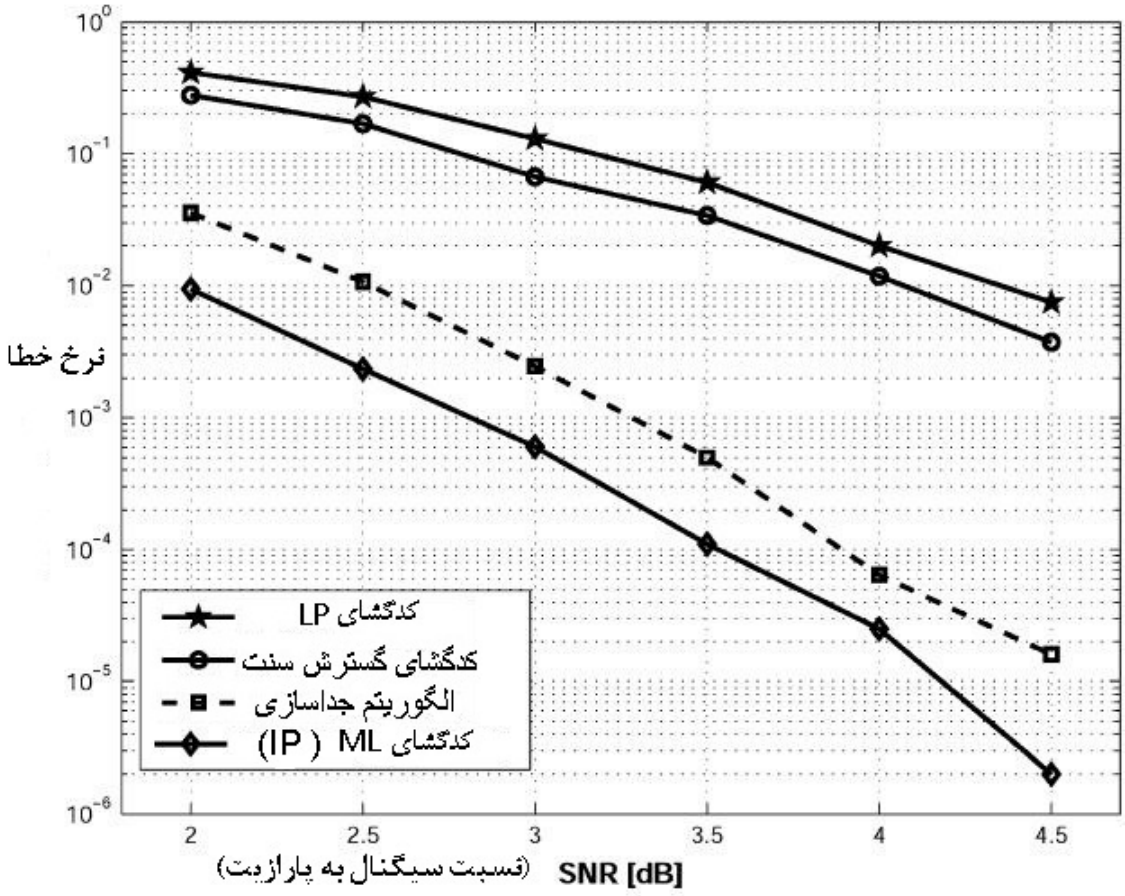
در این فصل به بررسی نتایج عددی حاصل از مقایسه برخی الگوریتم‌های کدگشایی که در این پایان‌نامه ذکر شده‌است، می‌پردازیم.

آنچه در این فصل خواهیم دید عبارت است از :

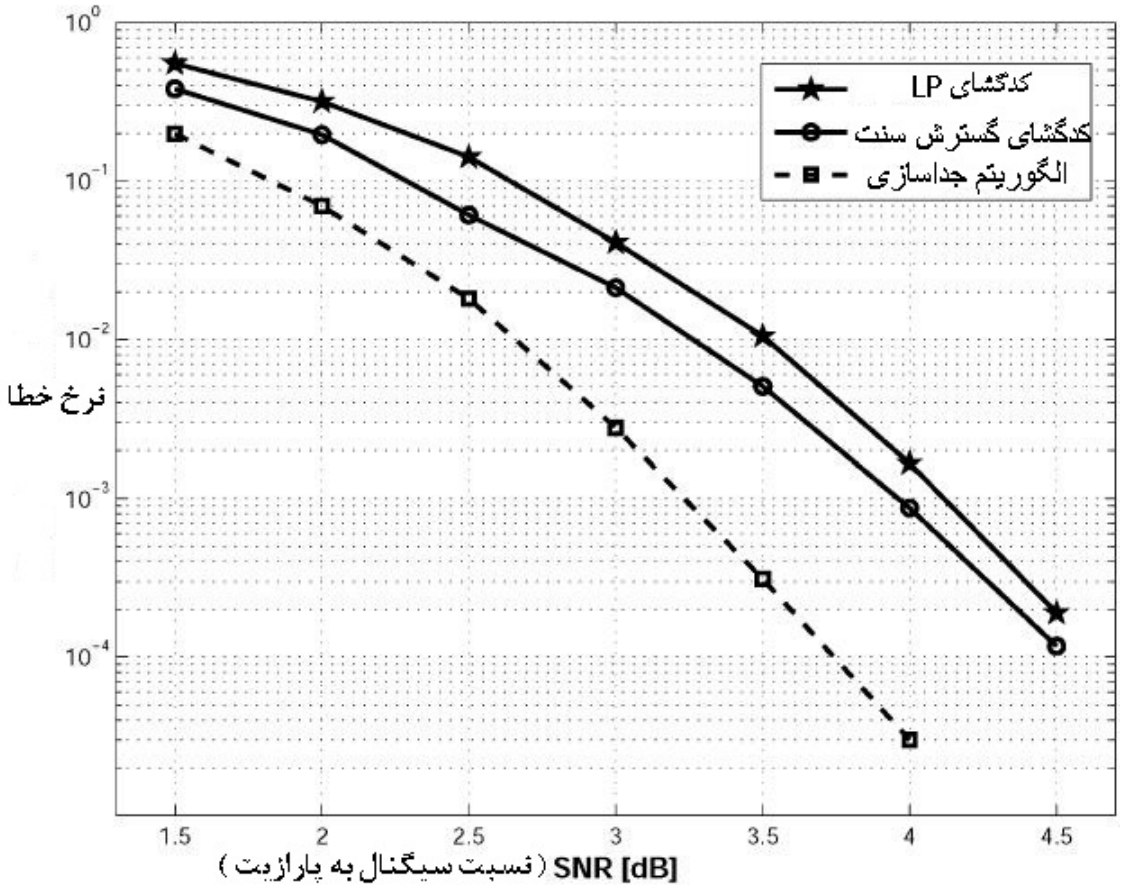
در بخش ۲.۶ گزیده‌ای از مقایسه‌های انجام شده بین برخی از روش‌هایی کدگشایی که در این پایان‌نامه ذکر شده است، بیان می‌گردد.

۲.۶ بررسی نتایج

به منظور مقایسه عملکرد برخی الگوریتم‌های کدگشایی که در این پایان‌نامه ذکر شده‌اند (با استفاده از [۲۳]) از (۳،۴)-کد LDPC با قاعده با طولهای‌بلوکی ۱۰۰ و ۲۰۰ و همچنین (۱۵۵،۶۴)-کد LDPC ساخته شده توسط تنر[۲۴] استفاده شده است. همانطور که در شکل‌های ۱.۶ ، ۲.۶ و ۳.۶ مشاهده می‌شود محور افقی نسبت سیگنال به پارازیت و محور عمودی نرخ خطا است. در شکل ۱.۶ و ۳.۶ یعنی (۳،۴)-کد LDPC به طول ۱۰۰ و (۱۵۵،۶۴)-کدLDPC تنر تفاوت عملکرد الگوریتم جداسازی جدید و اجرای کدگشای ML به میزان ۰.۵dB می‌رسد. در (۳،۴)-کدLDPC به طول ۲۰۰، [۱۸] قادر به محاسبه منحنی کدگشای ML در یک زمان قابل قبول

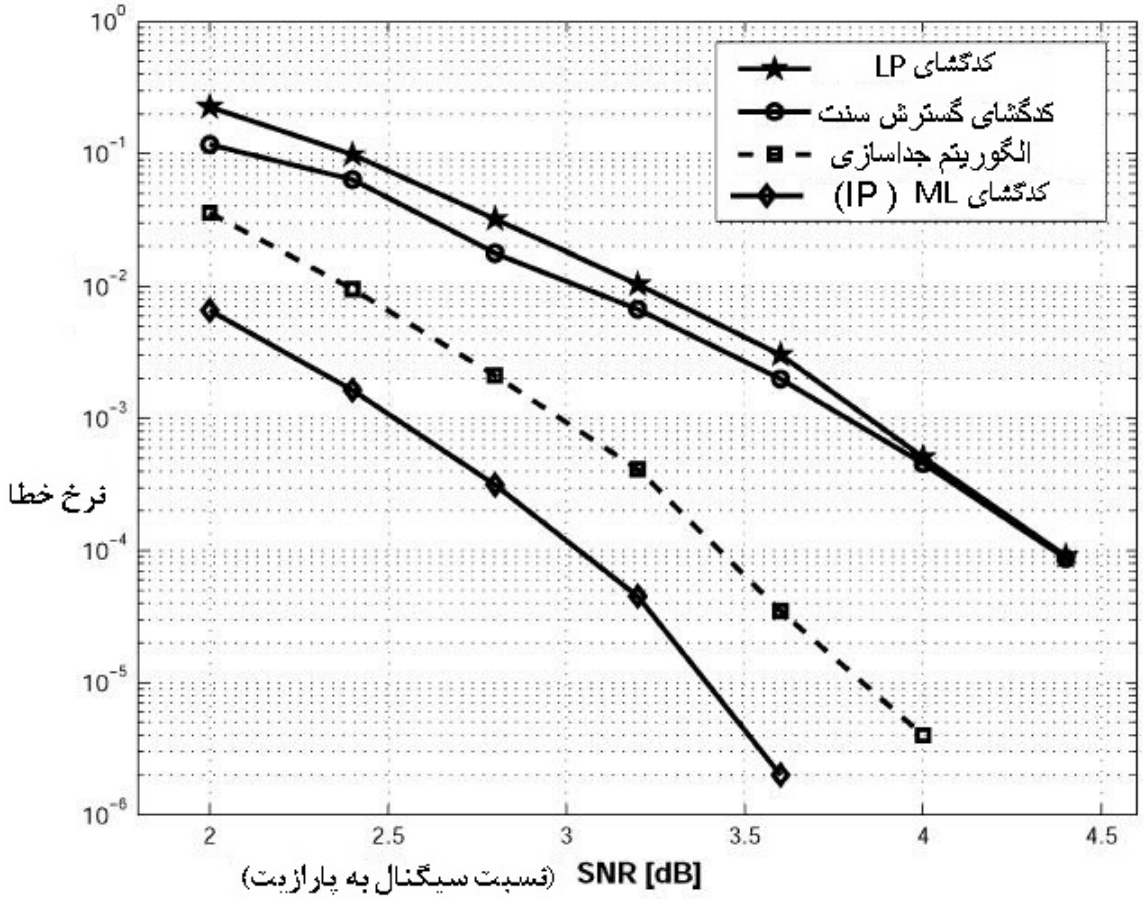


شکل ۱.۶: [۲۳]اجرای برخی الگوریتم‌های کدگشایی برای (۳،۴) - کد LDPC با طول بلوکی ۱۰۰



شکل ۲.۶: [۲۳] اجرای برخی الگوریتم‌های کدگشایی برای (۳،۴)-کد LDPC . با طول بلوکی ۲۰۰

نبوده است. در هر کد ذکر شده عملکرد کدگشای LP نزدیک به عملکرد کدگشای BP است (یاد آوری: BP همان الگوریتم گسترش سنت است). علاوه بر این در هر سه کد نامبرده عملکرد الگوریتم جداسازی جدید در تصحیح خطا از کدگشای BPبرتر است. ولی همانطور که در جدول (۱.۶) می‌بینید این بهبود عملکرد با افزایش پیچیدگی همراه است زیرا هرچه تعداد بیشتری برش به مساله وارد شود مساله LP بزگتری باید حل گردد.



شکل ۳.۶: [۲۳] اجرای برخی الگوریتم‌های کدگشایی برای (۱۵۵،۶۴)- کد LDPC تنر.

کد		کدگشایLP		الگوریتم جداسازی
	برش‌ها	تکرارها	برش‌ها	تکرارها
$(۴۰,۲۰)LDPC$	۱۴/۸	۴/۲	۲۵/۹	۴/۴
$(۸۰,۴۰)LDPC$	۳۳/۸	۵/۷	۱۰۳/۴	۷/۶
$(۱۶۰,۸۰)LDPC$	۷۴/۹	۷/۲	۳۹۷/۰	۱۶/۳
$(۲۰۰,۱۰۰)LDPC$	۹۸/۱	۷/۷	۶۱۲/۱	۲۱/۴
$(۴۰۰,۲۰۰)LDPC$	۲۱۰/۴	۹/۱	۳۰۲۳/۰	۷۲/۷

(۱.۶)

در جدول ۱.۶[۲۳] مقایسه‌ای بین متوسط تعداد تکرارها (تعداد اجرای الگوریتم LP) و متوسط تعداد برش در دو الگوریتم کدگشایی LP استاندارد [۲۲] و الگوریتم جداسازی جدید مطرح شده است. در جدول ۱.۶ SNR برابر با ۱.۵ انتخاب شده است زیرا یافتن کدواژه ML برای مقادیر کمتر SNR دشوارتر است و همچنین ضرایب تابع هدف قابل اطمینان‌تری را تولید می‌کند. علاقمندان به ادامه این مبحث می‌توانند این دسته از کدها را با دید فازی، به ویژه روش‌های کدگشایی بر مبنای دیدگاه فازی ادامه دهند.

پایان

کتاب‌نامه

[۱] مهناز علیین ، صادق رحیمی شعرباف. بررسی مسئله اساسی نظریه کد با استفاده از مفاهیم گراف. چهارمین کنفرانس ریاضی کاربردی ایران، ۱۳۸۸. خ

[۲] مهناز علیین، صادق رحیمی شعرباف. کاربرد نظریه کدگذاری در کد ژنتیکی. اولین همایش ملی الکترونیکی نقش ریاضی در توسعه علوم، ۱۳۸۹. خ

[۳] مهناز علیین، صادق رحیمی شعرباف. کاربرد برنامه‌ریزي‌خطي در کدگشايي کدهاي خطي. چهارمین کنفرانس بین‌المللی انجمن ایرانی تحقیق در عملیات، ۱۳۹۰. خ

[۴] مهناز علیین. کاربرد برنامه ریزی خطی در کدگشایی. اولین همایش ملی الکترونیکی نقش ریاضی در توسعه علوم، ۱۳۸۹.

[5] *Coding Theory A First Course*. Cambridge University Press, 2003.

[6] Nathan Axvig. *Applycation of linear programming to coding thaory*. PhD thesis, University of Nebraska - Lincoln, 2010. 13, 15, 16

[7] Claude Berrou, Alain Glavieux, and Punya Thitimajshima. Near shannon limit error-correcting coding and decoding. international conference on (1993) Volume: 2, Issue: 1, Publisher: IEEE, Pages: 1064-1070, 1993. 92

[8] Sae-young Chung, G.David Forney, J. Richardson Thomas, and Rüdiger Urbanke. On the design of low-density parity-check codes within 0.0045 db of the shannon limit. *IEEE Communications Letters*, 5:58–60, 2001. 2

[9] Mustafa Eroz, Feng-Wen Sun, and Lin-Nan Lee. Dvb-s2 low density parity check codes with near shannon limit performance. In *International Journal on Satellite Communications and Networking*, pages 269–279, 2004. 34

[10] Feldman. *Decoding Erorr-Corection Codes via Linear Programming*. PhD thesis, Massachusetts Institute of Techenology, 2003. 34

[11] J. Feldman, M. J. Wainwright, and D. R. Karger. Using linear programming to Decode Binary linear codes. *Information Theory, IEEE Transactions on*, 51(3):954–972, 2005. 102, 113, 129, 130, 131

[12] Jon Feldman, Martin Wainwright, and David R. Karger. Using linear programming to decode linear codes. In *John Hopkins University*, pages 12–14, 2003. 92, 101, 107, 108, 113, 120, 128, 129, 130

[13] Jon Feldman, Martin J. Wainwright, and David R. Kargerý. Lp decoding. In *In Proc. 41st Annual Allerton Conference on Communication, Control, and Computing*, 2003. 100, 101, 102, 104

[14] R.G Gallager. Low-density parity-check codes. *Information Theory, IEEE Transactions on*, 8(1):21–28, 1962. 114, 132

[15] Robert G. Gallager. *Low-Density Parity-Check Codes*. PhD thesis, Cambridge, Mass, 1963. 33

[16] Michael Huber. Coding Theory and Algebraic Combinatorics. November 2008. 33, 53

[17] Sarah J. Johnson. Introducing low-density parity-check codes.

[18] Tom Richardson and Rüdiger Urbanke. *Modern coding theory*. Cambridge University Press, Cambridge, 2008. 36, 38, 39, 43, 56, 58, 59, 62, 63, 65, 70, 71, 79, 84, 88

[19] S.El Rouayheb and C. N. Georghiades. Graph-theoretic methods in coding theory.invited paper, book in honor of david middleton. *Springer*. , 16, 21, 22, 23, 24, 25, 26, 28, 30, 31, 32, 33, 37, 38, 39, 40, 42, 46, 51, 54, 75, 89

[20] C. E. Shannon. A mathematical theory of communication. *Bell System Tech. J.*, 27:379–423, 623–656, 1948.

[21] Sarah L. Sweatlock. *Asymptotic Weight Analysis of Low-Density Parity Check (LDPC) Code Ensembles*. PhD thesis, California Institute of Technology Pasadena, 2008. 1, 20

[22] M. H. Taghavi, A. Shokrollahi, and P. H. Siegel. Lp decoding. In *in Proc.46th Ann. Allerton Conf. Commun., Control Computing, Monticello, IL*, 2008. 4

[23] Akin Tanatmis, Stefan Ruzika, Horst W. Hamacher, Mayur Punekar, and and Wehn Norbert Kienle, Frank. A separation algorithm for improved lp-decoding of linear block codes. *IEEE Transactions on*, 56:3277–3289, 2010. 117, 137

[24] R. M. Tanner, D. Sridhara, and T. Fuja. A class of group-structured ldpc codes. In *in Proc. Int. Conf. Inf. Syst. Technol. Appl., Kharkiv,Ukraine*, 2001. , 119, 120, 121, 122, 123, 124, 125, 126, 127, 133, 134, 135, 136, 137

[25] Nana Traore, Shashi Kant, and Tobias Lindstrøm Jensen. Message passing algorithm and linear programming decoding for ldpc and linear block codes. Technical report, Electronic Systems Signal and Information Processing in Communications, 2007.

فهرست الفبایی

واژه‌نامه فارسی به انگلیسی

آستانه (مرز) ..... Theorshold
احتمال پسین ..... A posteriori probability
احتمال پیشین ..... A priori probability
احتمال همگذری ..... Cross over probability
آر. دابلیو . هامینگ ..... R.W.Hamming
ارسال ماهواره‌ای تلویزون دیجیتالی ..... Satelling transmission of digital television
اطلاع خارجی ..... Exterinsic information
افزونگی ..... Redundancy
الگوریتم عبور پیام ..... Message passing algorithm
الیاس ..... Elias
بی‌حافظه ..... Memoryless
برش بررسی‌زوجیت افزونگی ..... Redundant parity ckeck cut
برش قابل‌پذیرش ..... Valid cut
برش گوموری ..... Gomory cut
پوشش ..... Cover
تابع نشانگر ..... Indicator function
تصدیق ML ..... ML certificate
تکامل چگالی ..... Density-evolution
توزیع‌های درجه ..... Degree distributions
جواب کسری ..... Fractional solution
حاشیه‌ای ..... Marginal
چندسقفی ..... Polytop
چند سقفی سراسری ..... Global polytop
چند سقفی کدواژه ..... Codeworde polytop
چندسقفی کدواژه محلی ..... Local codeword polytop
چندسقفی تصویر ..... projection polytop
چندسقفی محض ..... Proper polytop
چندوجهی ..... Polyhedron
دسته ..... Enssemble
راس فاکتور ..... Factor node (vertex)
راس متغییر ..... Variable node (vertex)
شکاف ..... Gap

ضرب نقطه به نقطه ..... Pointwise multiplication
ظرفیت کانال ..... Capacity of channel
فاکتورگیری ..... Factorization
فاصله کسری ..... Fractional distance
کانال پارازیت نرمال سفید جمعی ..... Additive White Gaussian Noise Channel
کانال پاک‌شدگی دودویی ..... Binary Erasure Channel
کانال متقارن دودویی ..... Binary Symmetric Channel
کد تشخیص 2e خطا ..... 2e-error detecting code
کدهای تصحیح خطا ..... Theory or Error-Correcting codes
کد تصحیح کننده e خطا ..... e-error correcting code
کد توربو ..... Turbo code
کد دوگان ..... Dual code
کد رید - مدلر ..... Reed-Muller code
کد سراسری ..... Global code
کدگذاری ..... Encoding
کدگشایی ..... Decoding
کدگشایی تکراری ..... Iteration decoding
کدگشایی جمع - ضرب ..... Sum-Product decoding
کدگشایی گسترش سنت ..... Belief propagation decoding
کدگشایی ماکزیمم درستنمایی ..... Maximum likelihood decoding
کد محلی ..... Local code
کدواژه ..... Codeword
کدواژه دوگان ..... Doual codeworde
کدواژه ماکزیمم درستنمایی ..... maximum likelihood codeworde
کدواژه محلی ..... Local codeword
کدواژه‌های کاذب ..... Pseudo-codewords
کدهای بررسی‌زوجیت کم چگال ..... Low-Density Parity-Check codes
کدهای بلوکی خطی ..... Linear block codes
کدهای LDPC باقاعده ..... Regular LDPC codes
کدهای LDPC بی‌قاعده ..... Irregular LDPC codes
کدهای پیچشی ..... Convolutional codes
کد هامینگ ..... Hamming distance
کلود شانون ..... Clud Shannon
کمر ..... Girth
گالاگر ..... Galager
گراف تنر ..... Tanner graph
گراف تنر ۴- دور آزاد ..... 4-Cycle free tannar graph
گراف فاکتور ..... Factor graph
گراف فاکتور سبک فورنی ..... Fornay-style factor graph
گراف نرمال ..... Normal graph
لگاریتم نسبت درستنمایی ..... Log liklihood ration

LP کاهشی ..... LP relaxation
ماتریس بررسی‌زوجیت ..... Parity check matrix
ماتریس مولد ..... Genarator matrix
ماکزیمم احتمال پسین ..... Maximum aposteriory probability
متغییر حالت ..... State Variable
متغییر خارجی ..... External variable
مجموعه ممنوع ..... Forbiden set
مجموعه توقف ..... Stopping set
محدودیت پایداری ..... Stability constraint
مک‌کی ..... MacKay
مکمل متعامد ..... Othogonal complement
نرخ طرح ..... Desingn rate
نرخ واقعی ..... True rate
نسل دوم استاندارد رادیو تلویزون دیجیتال ..... Secend generation digital video broadcating
نشانگر ..... Indicator
نظریه اطلاعات ..... Informaton theory
نظریه کدگذاری ..... Coding theory
نظریه کدگذاری کانال ..... Channel coding theory
نظریه کدگذاری مدرن ..... Modern coding theory
نظریه کدگذاری منبع ..... Source coding thory
نقطه نسبی ..... Relative pointe
نیل ..... Neal
وزن ..... Weight

Abstract

Resently, utilizing LDPC codes have been magnificently of interest for many resarchers and applicants. In this project we first study basic and elementary concepts in the coding theory and then investigate some message passing algorithms using factor graphs, moreover we study linear programming decoding and some of its improvements. finally we compare some of the decoding algorithms.

Keywords: *LDPC Codes, Factor graph, Tanner graph, message passing algorithm, Decoding, Encoding, Linear code, Linear programming and LP decoding.*

Shahrood University of Technology
Facualty of Methematical Sciences
Department of Mathematics

MS. Thesis

On linear and nonlinear codes and their application(LDPC codes)

By:
Mahnaz Eleain

Supervisor:
Dr. Sadegh Rahimi Sharbaf

Advisor:
Dr. Jafari Rad

May 2011