کدگذاری مبتنی بر علامت- رقم برای نگاشت داده‌های دیجیتال در حافظه ذخیره‌سازی زیستی مبتنی بر DNA

میثم اللهی رودپشتی و سعیده علی‌نژاد

چکیده: امروزه به دلیل افزایش داده‌های مهم موجود در جهان به ذخیره‌سازهایی با تراکم ذخیره‌سازی بیشتر نیاز است و از این جهت استفاده از حافظه‌های مولکولی زیستی در پژوهش‌های اخیر مورد توجه قرار گرفته است. DNA به عنوان یک ذخیره‌ساز مولکولی می‌تواند حجم زیادی از داده را در فضای محدود و با ماندگاری بالا ذخیره کند. انتخاب یک نگاشت مناسب از داده دیجیتال به الفبای DNA اهمیت زیادی دارد. در این مقاله، یک روش جدید برای نگاشت داده دیجیتال به الفبای DNA با هدف سادگی کدگذاری و کدگشایی، حذف خطای کدگشایی، ذخیره‌سازی داده‌های دیجیتال و علامت- رقم با فشردگی مناسب و سرعت کدگذاری بالا برای داده‌های کلان پیشنهاد شده است. بررسی‌های انجام‌گرفته نشان می‌دهد که روش پیشنهادی در مقایسه با روش‌هایی پیشین می‌تواند بازیابی اطلاعات از DNA را در مدت طولانی تضمین نماید. همچنین به دلیل دستاوردهای حاصل‌شده نسبت به روش‌های پیشین از فشردگی کمتری برای ذخیره داده‌های دیجیتال بهره می‌گیرد.

کلیدواژه: **DNA، سامانه ذخیره‌سازی زیستی، کدگذاری، الگوریتم نگاشت، علامت- رقم، کلان‌داده.**

۱- مقدمه

از سال ۲۰۰۹ تا سال ۲۰۱۸ رشد داده‌های موجود در جهان به صورت نمایی بود که این میزان برای سال‌های ۲۰۱۹ و ۲۰۲۰ حدوداً ۷۰ زتابایت پیش‌بینی شده است [۱]. در گذشته استفاده از نوارهای مغناطیسی و دیسک‌های مغناطیسی به منظور ذخیره‌سازی داده، بسیار متداول بود [۲] و این در حالی است که تکنولوژی جدید این ذخیره‌سازها اجازه ذخیره‌سازی تا حداکثر ۱۰۰ گیگابایت داده را در یک متر مکعب می‌دهد [۳] و [۴]. این میزان تراکم ذخیره‌سازی داده با توجه به حجم داده موجود و هزینه ذخیره‌سازی در واحد حجم برای نیاز امروزی مناسب نیست، لذا امروزه بیشتر از ذخیره‌سازهای دیسک سخت با حجم ذخیره‌سازی حدوداً ۱۰ ترابایت بر متر مکعب و ذخیره‌سازهای حالت جامد با حجم ذخیره‌سازی حدوداً ۱۰ پتابایت بر متر مکعب استفاده می‌شود [۵]. با وجود این همچنان فاصله زیادی بین داده قابل ذخیره‌سازی و تکنولوژی ذخیره‌سازی وجود دارد و بر همین اساس در پژوهش‌های اخیر به حافظه‌های زیستی توجه شده است. ذخیره‌سازهای مولکولی (مانند DNA)، حافظه‌های زیستی هستند که حجم زیادی از اطلاعات زیستی را در فضایی محدود و با ماندگاری بالا ذخیره می‌نمایند، لذا استفاده از حافظه‌های زیستی به منظور ذخیره‌سازی کلان‌داده‌ها می‌تواند در بسیاری از هزینه‌های جانبی که صرف ذخیره‌سازی کلان‌داده‌ها می‌شود صرفه‌جویی نماید. به منظور استفاده از حافظه‌های زیستی لازم است یک نگاشت مناسب از حالت‌های زیستی به الفبای دیجیتال (صفر و یک) صورت گیرد [۶] و [۷]. در این مقاله، با تکیه بر یک کدگذاری جدید، امکان استفاده از حافظه زیستی و به طور خاص DNA را تسهیل می‌نماییم.

در ادامه مقاله، ابتدا در بخش ۲ به ادبیات موضوعی و معرفی ساختار حافظه‌های زیستی و چالش‌های مربوط به آن پرداخته شده است. در بخش ۳ کارهای پیشین و الگوریتم‌های نگاشت داده دیجیتال به DNA معرفی گردیده و در بخش ۴، الگوریتم نگاشت پیشنهادی معرفی و تشریح شده است. ارزیابی و مقایسه الگوریتم پیشنهادی با الگوریتم‌های پیشین در بخش ۵ ارائه شده است. در پایان، در بخش ۶ نتیجه‌گیری و پیشنهادهایی برای کارهای آتی مطرح می‌شود.

۲- حافظه مبتنی بر ساختار DNA

۲-۱ چالش حافظه‌های فیزیکی

داده‌های موجود در جهان به صورت غیر قابل کنترلی در حال افزایش است. به دلیل افزایش شبکه‌های اجتماعی، فضاهای بارگذاری داده، فضاهای ابری، داده‌های پزشکی و داده‌های اینترنت اشیا، مدیریت داده در ذخیره‌سازهای کنونی را با مشکل روبه‌رو کرده است. همچنین پیش‌بینی شده که در سال ۲۰۲۰ به ۳۵ زتابایت فضا برای نگهداری داده نیاز خواهد بود [۱]. ذخیره این حجم داده یک مسئله نگران‌کننده است چرا که تحقق این مهم از طریق حافظه‌های فیزیکی فعلی به یک بن‌بست نزدیک شده است. تاریخچه ذخیره‌سازی داده‌ها از سنگ‌ها، کاغذ، پانچ‌کارت، نوار مغناطیسی، CD، DVD، فلاپی‌دیسک شروع شده و تا امروزه به حافظه‌های پرسرعتی چون SSD می‌رسد [۸]. استفاده از حافظه‌های فیزیکی در سطح کلان‌داده‌ها، نیاز به هزینه‌های فراوانی از جمله هزینه خرید و نگهداری تجهیزات، هزینه فضایی، هزینه الکتریکی که صرف روشن نگه‌داشتن تجهیزات می‌شود، هزینه محاسبات و پردازش روی داده‌ها و از همه مهم‌تر، هزینه به روز رسانی است که تحمیل ذخیره‌سازی در سطح کلان‌داده می‌شود. برای حل این چالش‌ها استفاده از سامانه ذخیره‌سازی مبتنی بر DNA پیشنهاد شده که از انعطاف بالایی برخوردار است و می‌تواند حجم زیادی از داده را فقط در یک مخزن کوچک و به مدت بسیار طولانی و بدون خرابی ذخیره نماید.

۲-۲ حافظه زیستی مبتنی بر DNA

DNA یک ذخیره‌ساز مولکولی است که اطلاعات ژنتیکی تمام موجودات زنده را در خود نگهداری می‌کند. این مولکول برای نگهداری اطلاعات اا‌ز چهار باز آلی شامل آدنین، تیمین، سیتوزین و گوانین استفاده می‌کند که الفبای تشکیل‌دهنده DNA هستند، همچنین الفبای DNA با حروف A، T، C و G نمایش داده می‌شوند [۹]. DNA یک ذخیره‌ساز

این مقاله در تاریخ ۲۲ شهریور ماه ۱۳۹۷ دریافت و در تاریخ ۱۱ مرداد ماه ۱۳۹۸ بازنگری شد.

میثم اللهی رودپشتی، دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی نوشیرانی بابل، بابل، ایران، (email: m.allahi@stu.nit.ac.ir).

سعیده علی‌نژاد (نویسنده مسئول)، دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی نوشیرانی بابل، بابل، ایران، (email: alinezhad@nit.ac.ir).

متراکم است و توانایی ذخیره یک زتابایت در یک گرم را دارد. در سال ۲۰۱۳ یک گروه از دانشمندان از یک استخوان ۷۰۰ هزار ساله اسب اولیه تمام ژنوم آن را بازسازی کردند که این موضوع به دو نکته درباره DNA تأکید می‌کند که اول ماندگاری و دوم پایداری آن در برابر شرایط محیطی است. با توجه به این دو ویژگی از DNA، استفاده از آن به عنوان یک حافظه زیستی مورد توجه پژوهشگران قرار گرفته است [۱۰].

۲-۳ عملیات ورودی/ خروجی روی DNA

• خواندن

DNA یک حافظه رشته‌ای از الفبای چهارحرفه A، T، C و G است و تمامی اطلاعات یک موجود در آن ذخیره می‌شود. این الفبای چهارحرفه به صورت دانه‌های یک تسبیح با پیوند کووالانسی به یکدیگر وصل می‌شوند و یک رشته از DNA را می‌سازند. DNA از یک ساختار دورشته‌ای مکمل تشکیل می‌شود که با پیوندهای واندروالسی در کنار یکدیگر قرار گرفته‌اند. ساختار زوج رشته‌ای که در DNA وجود دارد به تحمل‌پذیری DNA در برابر اشکال کمک می‌کند (مانند حافظه‌های فیزیکی) و از طرفی باعث تسریع در عملیات کپی‌برداری از DNA می‌شود. عملیات خواندن از DNA به معنی برش بخشی از داده‌های DNA و ایجاد DNA تک‌رشته است. از آنجایی که DNA در یک فضای مایه‌ای قرار دارد، به محض برش قسمتی از آن بازهای آلی موجود در فضای معلق به قسمت برش‌یافته چسبیده و رشته DNA بازسازی می‌شود. اطلاعات خارج‌شده از DNA به صورت یک رشته منفرد با الفبای چهارحرفه آدنین، گوانین، سیتوزین و یور اسید است و با حروف A، U، C و G نمایش داده می‌شود. لازم به ذکر است که به اطلاعات استخراج‌شده RNA گفته می‌شود و همچنین بایستی دقت شود که دلیل تفاوت الفبای DNA با RNA در ماهیت بهره‌برداری آنها است. DNA یک حافظه از داده‌های خام است و بایستی از آن محافظت شود که این مسئله ایجاب می‌کند به صورت یک زوج مکمل باشد اما RNA اطلاعات استخراج‌شده تک‌رشته‌ای از DNA است که برای عملکرد تابعی استفاده می‌شود که در ساخت پروتئین‌ها نقش دارد. از طرفی اگر الفبای یکسان با DNA داشته باشد به سرعت به یک ساختار DNA تبدیل می‌شود. همچنین لازم به ذکر است که تمامی عملیات برش‌برداری (خواندن داده) و ساخت رشته RNA (استخراج اطلاعات) توسط یک پروتئین انجام می‌شود که به آن RNA پولیمراز می‌گویند. این پروتئین یک دنباله عددی از DNA (نقطه آغازین خواندن) را شناسایی می‌کند و به آن متصل می‌شود و تا رسیدن به دنباله پایانی (نقطه پایان خواندن) عملیات برش و ساختن RNA را انجام می‌دهد.

به طور خلاصه اگر بخواهیم این مفاهیم را با یک ساختار کامپیوتری مدل نماییم می‌توانیم DNA را به عنوان حافظه، الفبای DNA را به عنوان صفر و یک دیجیتال، RNA را به عنوان یک دستور کامپیوتری، RNA پولیمراز را هم به عنوان هد خواندن و هم یک کدگشای دستور و پروتئین‌ها را به عنوان خروجی‌های یک دستور کامپیوتری در نظر بگیریم [۱۱].

• حذف‌کردن

عملیات حذف‌کردن یا ویرایش با هدف رفع آسیب در ساختار DNA مورد استفاده قرار می‌گیرد. این عملیات مشابه عملیات خواندن است یعنی دستگاه پولیمراز یک لایه از DNA را برمی‌دارد تا دوباره ساختار DNA که دچار آسیب شده است شکل گیرد. لازم به ذکر است حذف یک لایه از DNA به کمک پروتئین‌ها و دستگاه پولیمراز انجام می‌شود.

• نوشتن

عملیات نوشتن در حافظه DNA چندان معنا ندارد اما قابل تعریف است. عمل نوشتن در DNA مانند ROM است یعنی برای اولین بار یک رشته DNA ایجاد شده و تا پایان ثابت می‌ماند. همچنین نوشتن‌های غیر اختیاری نیز در DNA تعریف شده که آن را جهش ژنی می‌نامند. با ترکیب عمل حذف و خواندن می‌توان به عمل نوشتن دست یافت [۱۱].

۲-٤ کدگذاری حافظه‌های زیستی

برای ذخیره اطلاعات دلخواه در DNA ابتدا بایستی اطلاعات را به داده دیجیتال تبدیل نمود و سپس داده‌ها را به توالی ژنتیکی تبدیل کرد. بعد از تبدیل داده‌های دیجیتال به الفبای چهارحرفه DNA، آنها را به یک شرکت زیست‌شناسی می‌دهند تا داده‌ها را به DNA مصنوعی تبدیل نماید. اطلاعات در یک لوله ذخیره می‌شود و بایستی در محل سرد نگهداری شود. هنگام بازیابی اطلاعات و خارج‌کردن آنها از لوله آزمایشگاهی، از یک ماشین توالی استاندارد DNA استفاده می‌شود که داده‌ها را کدگشایی می‌کند. DNA مصنوعی به دلیل تراکم بالا در کدگذاری دیجیتال و همچنین دوام بالا، یک محیط مناسب برای ذخیره‌سازی داده است [۱۲].

۲-۵ چالش‌های حافظه زیستی

اگرچه DNA مصنوعی تولیدشده دارای تراکم بالا در کدگذاری است اما بازیابی داده‌های ذخیره‌شده در یک مقیاس بزرگ در حال حاضر با چالش جدی تکنولوژی روبه‌رو است. به علت محدودیت تکنولوژی، در حال حاضر بازیابی داده‌های ذخیره‌شده در یک مقیاس بزرگ نیازمند تمام توالی ممکن از DNA موجود در یک مخزن است. با استفاده از ایده دسترسی تصادفی می‌توان فایل‌های مختلف را به صورت جداگانه از مخزن جداسازی نمود. با این وجود نرخ حذف، درج و جایگزینی داده جدید با خطای ۱٪ امکان‌پذیر است که با کدگذاری مناسب می‌توان داده‌ها را در شرایط بسیار سخت بازیابی کرد [۱۳].

۳- کارهای پیشین

در این قسمت برخی کارهای پیشین در زمینه الگوریتم‌های نگاشت داده دیجیتال به DNA و روش‌های مبتنی بر تحمل‌پذیری اشکال روی حافظه‌های زیستی و چالش‌های مربوط مورد بررسی قرارگرفته است.

۳-۱ الگوریتم‌های نگاشت داده دیجیتال به DNA

کدگذاری پایه (مبنای ٤)

برای این که اطلاعات دیجیتال در DNA ذخیره شود لازم است که یک کدگذاری بین الفبای دیجیتال و الفبای DNA صورت گیرد. ساده‌ترین روش کدگذاری مبتنی بر یک نگاشت مستقیم صورت می‌گیرد که در [۶] و [۷] گزارش شده و به کدگذاری مبنای ۴ شناخته می‌شود. در این روش هر دو بیت داده دیجیتال به یک حرف از الفبای DNA نگاشت داده می‌شود. نحوه نگاشت داده دیجیتال به الفبای DNA به صورت ذیل است

$$۰۰ \rightarrow A, ۰۱ \rightarrow C, ۱۰ \rightarrow G, ۱۱ \rightarrow T$$

برای مثال رشته CTCCCCG معادل ۰۱۱۱۰۱۰۱۰۱۰۱۱۰ است. این روش بسیار ساده است و اعداد را تا دوبرابر فشرده می‌کند، اما خطای

ماتریس داده	p 01010000	o 01101111	l 01101100	y 01111001	a 01100001	; 00111011
ماتریس کدشده سه تایی	12011	02110	02101	222111	01112	222021
ماتریس کدشده DNA	GCGAG	TGAGT	ATCGA	TGCTCT	AGAGC	ATGTGA

حروف قبلی

ارقام سه تایی	A	C	G	T
0	C	G	T	A
1	G	T	A	C
2	T	A	C	G

شکل ۱: عملیات نگاشت در روش هافمن [۶].

خواندن یا بازیابی آن بسیار زیاد است. به دلیل محدودیت تکنولوژی، در برابر رشته‌های الفبای تکراری، این روش خطای زیادی دارد و از این جهت باید در کدگذاری تا حد امکان از حروف تکراری جلوگیری شود، هرچند در این روش چنین چیزی امکان‌پذیر نیست و برای رشته‌های دیجیتال تکرارشونده، الفبای تکرارشونده تولید می‌کند.

کدگذاری مبنای ۳

این روش برای رفع مشکل رشته‌های تکراری در روش کدگذاری مبنای ۴ پیشنهاد شده است [۱۴]. در کدگذاری مبنای ۳ باقیمانده صفر، ۱ و ۲ خواهد بود یعنی برای نمایش یک عدد نیاز به سه علامت داشته که اگر بخواهد از نگاشت DNA استفاده کند یک علامت می‌تواند برای جلوگیری از رشته تکراری استفاده شود [۶] و [۷]. کدگذاری مبنای ۳ می‌تواند در کاهش نرخ خطا بسیار مناسب باشد اما مشکلی که این روش دارد این است که نگاشت مستقیم به داده دیجیتال در آن وجود ندارد. زیرا مبنای ۳ توانی از ۲ نیست و همواره بین داده دیجیتال و کد تبدیل‌شده افزونگی وجود دارد. نحوه نگاشت داده دیجیتال به الفبای DNA به صورت ذیل است

$$۰ \rightarrow G, ۱ \rightarrow A/T, ۲ \rightarrow C$$

کدگذاری مبتنی بر الگوریتم هافمن

آخرین نوع کدگذاری از داده‌های دیجیتال به الفبای DNA در [۶] ارائه شده است. این کدگذاری مبتنی بر الگوریتم هافمن [۱۵] برای رفع مشکل نگاشت مستقیم در کدگذاری مبنای ۳ پیشنهاد گردیده است. در این روش با استفاده از کدگذاری هافمن، هر ۸ بیت از داده دیجیتال به ۵ یا ۶ بیت از مبنای ۳ تبدیل شده و سپس کد مبنای ۳ به داده دیجیتال نگاشت داده می‌شود. شکل ۱ عملیات نگاشت را نشان می‌دهد. در این شکل آرایه اول شامل داده، آرایه دوم اعمال روش هافمن روی آرایه داده و آرایه سوم نگاشت آرایه دوم به الفبای DNA است. عملیات نگاشت بر اساس جدولی که در شکل ۱ نشان داده شده انجام می‌شود.

سایر کدگذاری‌های معرفی‌شده در پژوهش‌های پیشین به شرح زیر است. این روش‌ها به دلیل مشکلاتی که به آنها اشاره شده است چندان مورد مقبولیت قرار نگرفته‌اند.

روش سه‌گانه

در این روش الفبای DNA به ازای دو حرف متوالی یک عدد دیجیتال را مشخص می‌کند. نحوه نگاشت به صورت ذیل خواهد بود [۷]

$$۰ \rightarrow AA, ۱ \rightarrow TT, ۲ \rightarrow GG, ۳ \rightarrow CC, ۴ \rightarrow AT,$$
$$۵ \rightarrow AG, ۶ \rightarrow AC, ۷ \rightarrow TA, ۸ \rightarrow TG, ۹ \rightarrow TC$$

این روش در حالت عادی الفبای دوحرفه تکراری ایجاد می‌کند و همچنان احتمال خطای خواندن در این روش وجود دارد.

روش مقدار اسکی

روش مقدار اسکی [۱۶] برعکس مبنای ۴ عمل می‌کند یعنی در روش مبنای ۴ هر ۲ بیت داده دیجیتال ارزش یک حرف از الفبای DNA را دارد اما در این روش هر دو حرف از الفبای DNA می‌تواند به جای یک بیت دیجیتال استفاده شود. بدیهی است که این روش با هدف جلوگیری از کنار هم قرار گرفتن الفبای تکراری پیشنهاد شده است. نحوه نگاشت به صورت ذیل است [۷]

$$۰ \rightarrow A/T, ۱ \rightarrow C/G$$

مشکل اول این روش احتمال خطای بالای خواندن است یعنی همچنان در برابر رشته‌های تکراری الفبای تکرار تولید می‌شود. مشکل بعدی این روش فشرده‌سازی ضعیف داده دیجیتال است زیرا در این روش به ازای یک بایت داده دیجیتال نیاز به ۸ الفبای DNA است.

۳-۲ روش‌های مبتنی بر تحمل‌پذیری اشکال

بین سال‌های ۲۰۱۰ تا ۲۰۱۶ کارهای روی حافظه‌های زیستی مبتنی بر DNA با تمرکز روی یک کدگذاری جدید بوده است که با ذخیره چند کیلوبایت و تا حداکثر چند مگابایت اطلاعات در ساختار DNA و در شرایط آزمایشگاهی و سپس کدگشایی آنها با استفاده از ماشین‌های توالی‌یاب، صحت الگوریتم خود را اثبات نموده‌اند که این کارها در [۹] گزارش شده‌اند. برای اولین بار در [۱۷]، ذخیره طولانی مدت اطلاعات با حجم زیاد مورد توجه قرار گرفت. این روش با بهره‌گیری از کدگذاری پایه و استفاده از روش‌های شناسایی خطا توانست یک DNA که در شرایط ناپایدار از نظر محیطی (شرایط دمایی بالا برای DNA باعث باز شدن پیوندهای بین اتمی می‌شود و این مسئله باعث از دست رفتن اطلاعات می گردد) است، در یک محفظه ایزوله شده از سیلیکا بدون خطا بازیابی نماید. در [۱۸] از الگوریتم Reed-Solomon به منظور تشخیص و تصحیح اشکال استفاده می‌نماید. لازم به ذکر است که در این روش و روش‌های مشابه مقدار داده افزونه بسیار زیاد خواهد شد و این مسئله هم از نظر مقیاس‌پذیری و هم از نظر هزینه ذخیره‌سازی بسیار مهم است. همچنین کارهای مشابه دیگر در [۱۹] تا [۲۳] آمده‌اند. از آنجایی که هدف این مقاله ارائه یک کدگذاری جدید است بنابراین تمرکز ما بر روی روش‌های کدگذاری می‌باشد. هر کدام از روش‌های کدگذاری با بهره‌گیری از روش‌های تحمل‌پذیر در برابر اشکال می‌توانند مورد استفاده قرار گیرند.

۳-۳ چالش کدگذاری‌های پیشین

چالش‌های موجود در کدگذاری‌های پیشین مورد بررسی قرار گرفت و همان طور که اشاره شد، چالش‌های اصلی کدگذاری‌های پیشین عبارتند از:

جدول ۱: مقایسه روش‌های مختلف کدگذاری.

ویژگی / کدگذاری	مبنای ۴	مبنای ۳	هافمن	سه‌گانه	اسکی	پیشنهادی
میزان فشرده‌سازی	۱	۲	۳	۴	۵	۵
کاهش خطای خواندن	-	۱	۱	۴	۴	۱
پیچیدگی کدگذاری	۱	۲	۴	۲	۲	۲
نگاشت مستقیم	۱	-	۲	۲	۲	۲

(۱ عالی، ۲ خیلی خوب، ۳ خوب، ۴ بد و ۵ خیلی بد)

قرارگیری الفبای تکراری کنار هم: این مشکل در همه روش‌ها غیر از مبنای ۴ تا حدودی حل گردیده و در بخش ارزیابی روش پیشنهادی درباره این مشکل صحبت شده است.

عدم نگاشت مستقیم: این چالش در نگاشت مبنای ۳ به داده دیجیتال وجود داشت که کدگذاری هافمن برای حل این مشکل مورد استفاده قرار گرفت. هرچند این مسئله باعث شد علاوه بر پیچیدگی محاسبه مبنای ۳، پیچیدگی کدگذاری هافمن نیز در محاسبات اضافه شود.

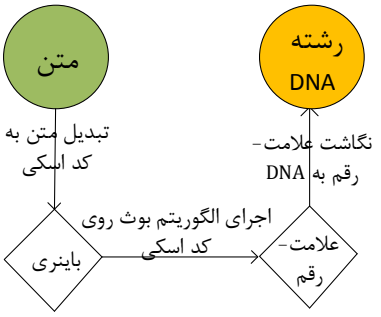
میزان فشردگی داده دیجیتال در DNA: این مشکل در روش مقدار اسکی بیشترین است و در روش مبنای ۴ کمترین مقدار خود را دارد. در جدول ۱ مقایسه‌ای بین روش‌های پیشین انجام گرفته و روش پیشنهادی در این مقاله یک راهکار میانه برای رفع این مشکلات است.

٤- الگوریتم پیشنهادی: نگاشت مبتنی بر علامت- رقم

٤-١ فرایند کدگذاری

با توجه به چالش‌های ذکرشده در کدگذاری‌های پیشین در روش پیشنهادی به راهکاری میانه برای مشکلات موجود اشاره خواهیم کرد. در این روش با استفاده از روش کدگذاری بوث مبنای ۲، داده دیجیتال را به علامت- رقم تبدیل می‌کنیم [۲۴] و در این صورت تعداد علامت‌های دیجیتال از دو به سه تبدیل می‌شود. این دقیقاً همان روش مبنای ۳ است با این تفاوت که به جای بازه عدد ۰ تا ۲ بازه را به ۱- تا ۱ تبدیل می‌کنیم. فایده روش این است که می‌توانیم از روش‌های سریع کدگذاری علامت- رقم استفاده کنیم. در این روش سرعت نگاشت به DNA زیاد است زیرا عملیات تبدیل یک تفریق ساده از رقم قبلی است و حتی می‌تواند به صورت موازی در یک گام انجام شود. در شکل ۲ نحوه عملکرد این الگوریتم نشان داده شده است. عملیات نگاشت داده علامت- رقم به الفبای DNA از روی جدول ۲ انجام می‌شود. با توجه به این جدول ردیف الفبا، رشته قبلی را مشخص می‌کند و اعداد داخل جدول رشته فعلی را مشخص می‌نماید. با توجه به شکل ۲، برای تبدیل یک کلمه به رشته DNA، ابتدا بایستی کلمه را توسط حروف اسکی به رشته باینری تبدیل کرد و سپس مقادیر باینری به علامت- رقم تبدیل شود. این کار توسط الگوریتم بوث انجام می‌گردد. در الگوریتم بوث، ابتدا یک بیت فرضی به سمت راست هر بایت اضافه شده، سپس رقم باارزش بیشتر از رقم قبلی خود کم می‌شود و نتیجه در مکان رقم باارزش قرار می‌گیرد. با انجام این کار برای تمام رقم‌ها وضعیت کدگذاری از باینری به علامت- رقم تغییر می‌کند. در حالت کلی الگوریتم بوث را می‌توان به صورت ذیل بیان کرد:

اگر $X_{n-1} \dots X_1 X_0$ ارقام از ۰ تا $n-1$ عدد X باشد آن گاه ارقام علامت- رقم با استفاده از رابطه $A_i = X_{i-1} - X_i$ به دست می‌آید که i از ۰ تا $n-1$ و $X_{-1}=0$ است و A_i ارقام علامت- رقم هستند. به



شکل ۲: عملیات تبدیل داده دیجیتال به الفبای DNA در روش پیشنهادی.

جدول ۲: جدول جستجوی نگاشت علامت- رقم.

	حروف قبلی			
	A	C	G	T
۰	C	G	T	A
۱	G	T	A	C
-۱	T	A	C	G

عنوان مثال کد اسکی حرف h برابر ۰۱۱۰۱۰۰۰ است. با اجرای الگوریتم بوث روی این کد، مقدار آن به ۰۰۰(-۱)(-۱)۱۰(-۱) تبدیل خواهد شد که (-۱) ارزش منفی یک است که در مکان‌های ۳ و ۵ ظاهر شده است. این کار را برای سایر حروف تکرار می‌کنیم تا مقادیر علامت- رقم آن به دست آید.

با توجه به موارد ذکرشده می‌توان بیان نمود که در الگوریتم پیشنهادی دو حالت دیجیتال به سه حالت علامت- رقم تبدیل می‌شود و بنابراین به راحتی می‌توان سه حالت علامت- رقم را (مشابه روش مبنای ۳) به الفبای DNA نگاشت نمود. به کمک جدول ۲ مقادیر علامت- رقم به الفبای DNA تبدیل می‌شود که در آن اولین رقم از داده علامت- رقم با حروف ستون اول جدول ۲ نگاشت می‌شود و سایر ارقام با توجه به حالت فعلی و قبلی خود از روی جدول ۲ نگاشت می‌شوند. به عنوان مثال برای حرف h به این صورت عمل می‌کنیم: مقدار علامت- رقم آن با ۰۰۰(-۱)(-۱)۱۰(-۱) برابر است، بنابراین از سمت چپ شروع به خواندن اعداد آن می‌کنیم. اولین رقم عدد ۱ است که در جدول مقدار عدد ۱ اگر حرفی قبل آن نباشد G خواهد بود. عدد بعدی خوانده می‌شود که عدد صفر است. مقدار عدد صفر را در جدولی که حرف قبلی آن G است جستجو می‌کنیم که حرف T است. به همین ترتیب سایر حروف تبدیل می‌شود و در نهایت رشته GTGATACG برای حرف h به دست می‌آید.

٤-٢ فرایند کدگشایی

در حالت رمزگشایی با استفاده از جدول ۲ می‌توان فرایند معکوس را انجام داد و رشته‌های الفبای چهارحرفه را به علامت- رقم تبدیل کرد. سپس برای تبدیل اعداد علامت- رقم به داده دیجیتال با استفاده از تفریق علامت رقم [۲۴] به صورت موازی در دو گام اعداد دیجیتال را تولید کرد. به عنوان مثال رشته DNA معادل حرف h یعنی رشته GTGATACG را در نظر می‌گیریم. اولین حرف از این رشته حرف G، چون این اولین حرف است و قبل از آن حرف دیگری نیست اولین ستون از جدول ۲ یعنی عدد ۱ معادل عددی آن می‌شود. نکته‌ای که اینجا وجود دارد این است که هیچ وقت رشته ما با حرف A آغاز نمی‌شود و در واقع حرف A به عنوان یک میانجی برای جلوگیری از تکرار حروف است و این به معنی وابستگی رشته‌ها به یکدیگر است یعنی کدشده کلمه hello یک رشته ۴۰ حرف در الفبای DNA است و نمی‌توان گفت یک رشته معادل حرف hello وجود

جدول ۳: مقایسه پارامترهای ارزیابی (n طول داده دیجیتال).

کدگذاری	پیچیدگی کارایی مؤثر	پیچیدگی عملیات تبدیل	الفبای مصرفی برای یک بایت	فشرده‌سازی داده علامت- رقم	فشرده‌سازی داده دیجیتال	رفع خطای خواندن	نگاشت مستقیم	پیچیدگی کارایی مؤثر
مبنای ۴	$O(n^2)$	$O(n)$	۴	n	$\frac{n}{2}$	×	✓	$O(n^2)$
مبنای ۳	$O(n^2)$	$O(n)$	۵	$\frac{5n}{4}$	$\frac{5n}{8}$	✓	×	$O(n^2)$
هافمن	$O(n^2 \log_2 n)$	$O(n \log n)$	۵	$\frac{5n}{4}$	$\frac{5n}{8}$	✓	✓	$O(n^2 \log_2 n)$
سه‌گانه	$O(n^2)$	$O(n)$	۶	$\frac{12n}{4}$	$\frac{6n}{8}$	×	✓	$O(n^2)$
مقدار اسکی	$O(n^2)$	$O(n)$	۸	$2n$	n	×	✓	$O(n^2)$
روش پیشنهادی	$O(n^2)$	$O(n)$	۸	n	n	✓	✓	$O(n^2)$

دارد. به طور مشابه حرف دوم را می‌خوانیم که حرف T است و حرف قبلی آن G بود. مقدار این وضعیت صفر است و به همین ترتیب این کار را برای سایر حروف تکرار می‌کنیم و رشته ۱۰(-۱)۱(-۱)۰۰۰ به دست می‌آید. سپس الگوریتم معکوس علامت- رقم بر روی رشته به دست آمده به این صورت اجرا می‌شود: با حفظ مقادیر صفر و یک، محل‌هایی که ارزش ۱- داریم یک بار صفر قرار داده که در این صورت یک رشته (در این مثال ۱۰۰۱۰۰۰۰) حاصل می‌شود. بار دیگر مقادیر ۱ را با صفر جایگزین کرده و در محل‌هایی با ارزش (۱-) مقدار ۱ قرار می‌دهیم (که در این مثال رشته ۰۰۱۰۱۰۰۰ حاصل می‌شود). سپس با کسر دو رشته به دست آمده مقدار باینری ۰۱۱۰۱۰۰۰ حاصل می‌شود که معادل مقدار اسکی حرف h است.

۵- ارزیابی

در این بخش به ارزیابی کدگذاری پیشنهادی از نظر پارامترهای دقت کدگذاری، تحمل‌پذیری اشکال، کارایی، میزان فشرده‌سازی و کارایی مؤثر پرداخته شده است.

۵-۱ دقت کدگذاری/ دقت خواندن

دقت کدگذاری نشان‌دهنده خطای کم خواندن است. روش کدگذاری مناسب‌تر روشی است که احتمال خطای خواندن در آن کمتر باشد. به طور تجربی مشخص شده هرچه الفبای تکراری کنار هم بیشتر باشد خواندن اطلاعات از DNA (عملیات کدگشایی) با خطای بیشتری مواجه خواهد شد [۲۵].

وقتی یک رشته DNA ساخته می‌شود، مکمل رشته اول به آن اضافه شده و به این ترتیب یک رشته زوج DNA ساخته می‌شود. از نظر مولکولی یک حرف A همیشه با یک حرف T پیوند ایجاد می‌کند و C نیز همیشه با G و برعکس. هنگامی که حروف تکراری کنار هم قرار گیرد این باعث پیوند اشتباه بین زوج DNA می‌شود و بنابراین کدگذاری‌ای مناسب است که کمترین رشته تکرار را داشته باشد. با توجه به این موضوع تنها روشی که از تکرار رشته تکراری جلوگیری می‌کند حالتی است که فقط سه علامت وجود داشته باشد [۶] و [۱۴]. در این حالت با توجه به چهار حرفی بودن الفبای DNA، یک حرف همواره میانجی خواهد بود. با سه علامت و چهار حرف، طبق اصل لانه کبوتری به هر علامت یک حرف اختصاص می‌یابد و یک حرف هم برای این که از یک علامت دو بار تکرار نشود مورد استفاده قرار می‌گیرد. از این جهت تمام روش‌هایی که از سه علامت برای رمزگذاری استفاده می‌کنند دارای دقت یکسانی هستند چرا که همه آنها از تکرار حروف جلوگیری می‌کنند. از این جهت روش پیشنهادی از نظر احتمال خطای خواندن مانند روش مبتنی بر الگوریتم هافمن [۶] عمل می‌کند.

۵-۲ کارایی

کارایی به پیچیدگی محاسباتی کدگذاری و کدگشایی اشاره دارد. در جدول ۳ پیچیدگی محاسباتی الگوریتم‌های مختلف نشان داده شده و مرتبه زمانی الگوریتم‌ها با توجه به [۷] به دست آمده است. در جدول ۳ فرایند تبدیل داده دیجیتال به علامت- رقم سریع‌تر از روش هافمن [۶] به تصویر کشیده شده است چرا که مرتبه زمانی اجرای الگوریتم بوث کمتر از روش هافمن است. همچنین این برتری در کدگشایی نیز وجود دارد زیرا در فرایند کدگشایی ابتدا با توجه به جدول ۲ بایستی الفبای DNA به داده علامت- رقم تبدیل شود. الگوریتم ترتیبی این تبدیل از مرتبه $O(n)$ است در حالی که با استفاده از تکنیک‌های موازی‌سازی و بهره‌گیری از درخت متوازن دودویی می‌تواند از مرتبه لگاریتمی انجام شود. اکنون اگر به داده علامت- رقم نیاز باشد در این صورت کار تمام شده و نیاز به محاسبات اضافی نیست اما اگر نیاز به داده دیجیتال شد فرایند معکوس تبدیل علامت- رقم برای تبدیل به داده دیجیتال را انجام می‌دهیم. مدت زمان لازم برای پردازش این اطلاعات با استفاده از الگوریتم ترتیبی از مرتبه $O(n)$ است و به طور مشابه در گام اول این قسمت نیز می‌تواند با بهره‌گیری از سخت‌افزار علامت- رقم یا برنامه‌نویسی موازی از مرتبه $O(1)$ انجام شود.

۵-۳ میزان فشرده‌سازی/ هزینه ذخیره‌سازی

میزان فشرده‌سازی عبارت است از تعداد الفبای مصرفی (چهار باز آلی) به ازای طول داده دیجیتال. هرچه مقدار حاصل کوچک‌تر باشد، میزان فشرده‌سازی بیشتر است و در نتیجه هزینه ذخیره‌سازی کمتر خواهد بود. نحوه محاسبه فشرده‌سازی از یک تناسب ساده به دست می‌آید. اگر p میزان الفبای مصرفی به ازای یک بایت و n تعداد بیت‌هایی باشد که باید نگاشت شود (طول داده دیجیتال)، آن گاه میزان فشرده‌سازی (C) از رابطه ذیل به دست می‌آید

$$C = \frac{p}{8} \times n \qquad (۱)$$

مقایسه میزان فشرده‌سازی الگوریتم‌های مختلف در جدول ۳ نشان داده شده است. در روش پیشنهادی به ازای هر ۸ بیت داده دیجیتال ۸ باز آلی مصرف می‌شود یعنی نسبت به مبنای سه که به ازای هر ۸ بیت داده دیجیتال ۵ نوکلئوتید مصرف می‌کند، روش پیشنهادی فشرده‌سازی کمتری دارد و تقریباً ۳۵ درصد نوکلئوتید بیشتری مصرف می‌شود.

۵-٤ کارایی مؤثر

هر کدام از روش‌های نگاشت یک نقطه ضعف و یک نقطه قوت دارند، لذا نمی‌توان به صورت دقیق آنها را ارزیابی کرد. برای این مهم نیاز به یک معیار ثابت و کلی است که بتوان برای تمام الگوریتم‌ها آن را استفاده نمود. این معیار ثابت را کارایی مؤثر تعریف می‌کنیم و مقدار آن به صورت حاصل‌ضرب پیچیدگی الگوریتم در میزان فشرده‌سازی که انجام می‌دهد است. نکته‌ای که در جدول ۳ برای مقایسه الگوریتم‌های مختلف وجود دارد این است: تمامی روش‌ها غیر از هافمن و مبنای ۳ به دلیل خطای بالای خواندن مورد ارزیابی قرار نمی‌گیرند اگرچه دارای ویژگی‌های مثبتی نیز هستند. از طرفی روش مبنای ۳ به علت عدم نگاشت مستقیم از مقایسه حذف می‌شود. بنابراین تنها روشی که قابلیت مقایسه با روش پیشنهادی را دارد روش هافمن است. روش هافمن بهترین روش موجود است چرا که هم احتمال خطای کمتری دارد و هم فشرده‌سازی مناسبی انجام می‌دهد هرچند کارایی آن پایین‌تر از سایر الگوریتم‌ها است.

با توجه به یکسان‌بودن پارامتر دقت خواندن در روش پیشنهادی و روش هافمن می‌توان از نظر کارایی مؤثر، آن دو را با هم مقایسه کرد که در این مقایسه روش پیشنهادی در برابر روش هافمن کارایی مؤثر بالاتری دارد. در جدول ۳ به درستی این مفاهیم نشان داده شده است.

٦- نتیجه‌گیری

استفاده از حافظه‌های زیستی مبتنی بر DNA رویکردی جدید است که اخیراً مورد توجه محققین جهت ذخیره‌سازی داده‌های دیجیتال قرار گرفته است. الگوریتم‌های مختلفی برای نگاشت داده دیجیتال به داده زیستی در پژوهش‌های قبلی مورد استفاده قرار گرفته که با چالش‌های مختلف قرارگیری الفبای تکراری کنار هم (خطای خواندن)، نگاشت غیر مستقیم و پیچیدگی محاسباتی روبه‌رو بوده‌اند. از طرفی پژوهش‌های جدید روی حافظه‌های زیستی به سمت اعمال تحمل‌پذیری اشکال به روش‌های پیشین بوده است اما استفاده از روش‌های تحمل‌پذیر اشکال برای کلان‌داده‌ها باعث افزایش حجم محاسبات هم از نظر زمانی و هم از نظر فضایی می‌شود. این یک چالش بسیار مهم برای آینده است چرا که زمان ذخیره‌سازی برای حجم زیادی از داده‌ها بسیار مهم است و بنابراین در این مقاله با تکیه بر کدگذاری سیستم علامت- رقم یک دیدگاه میانه برای کدگذاری زیستی پیشنهاد کردیم. روش‌های پیشین توانستند حجم محدودی از داده را در DNA ذخیره نمایند اما در این روش‌ها به ذخیره داده در سطح کلان توجه نداشته‌اند. یکی از مشکلات داده‌های در سطح کلان زمان مورد نیاز برای پردازش کدگذاری و کدگشایی است. سیستم علامت- رقم ویژگی‌های منحصر به فردی جهت پردازش‌های موازی دارد و این در حالی است که اجرای ترتیبی عملیات آن نیز نسبت به سایر روش‌ها سریع‌تر است. از این جهت سیستم پیشنهادی بر آینده ذخیره کلان‌داده‌ها در DNA بسیار مفید است. علاوه بر سرعت پردازش اطلاعت در کدگذاری و کدگشایی روش پیشنهادی در برابر خطای خواندن دستگاه‌های توالی‌یاب مقاومت می‌کند. به علاوه تمام این ویژگی‌ها روش پیشنهادی یک ساختار دومنظوره است یعنی هم می‌تواند خروجی دیجیتال در هنگام کدگشایی و هم خروجی علامت- رقم جهت محاسبات کامپیوتری تولید نماید و این در حالی است که ذخیره علامت- رقم در کدگذاری‌های پیشین به دو برابر فضا نیاز دارد.

سپاسگزاری

نویسندگان مقاله مراتب قدردانی خود را از حمایت دانشگاه صنعتی نوشیروانی بابل از طریق اعتبار پژوهشی شماره BNUT/394986/98 اعلام می‌دارند.

مراجع

[1] Data storage supply and demand worldwide, from 2009 to 2020, Available at: https://www.statista.com/statistics/751749/worldwide-data-storage-capacity-and-demand, 2018.

[2] Where in the world is storage? Available at: https://www.idc.com/downloads/where_is_storage_infographic_243338.pdf, 2013.

[3] Sony develops magnetic tape technology with the world's highest recording, 2014. Available at: www.sony.net/SonyInf/News/Press/201708/17-070E/index.html.

[4] ExtremeTech. New optical laser can increase DVD storage up to one petabyte, Available at: https://www.extremetech.com/computing/159245-new-optical-laser-can-increase-dvd-storage-up-to-one-petabyte

[5] Sh. Pep, *Storing Your Data on DNA?* Sept. 2018, Available: https://www.capconnect.com.au/single-post/DNA-Storage.

[6] J. Bornholt, R. Lopez, L. Ceze, G. Seelig, and K. Strauss, "A DNA-based archival storage system," in *Proc. of the 21st Int. Conf. on Architectural Support for Programming Languages and Operating Systems, ASPLOS'16*, pp. 637-649, Atlanta, GA, USA, 2-6 Apr. 2016.

[7] A. Garg and M. Choudhary, "Analysing and obtaining the most efficient DNA computing algorithm," *J. of Computational Intelligence in Bioinformatics*, vol. 8, no. 1, pp. 1-6, Jun. 2015.

[8] D. Limbachiya and M. Kumar Gupta, "Natural data storage: a review on sending information from now to then via nature," *ACM J. on Emerging Technologies in Computing Systems*, vol. 18, no. 1, pp. 550-4832, May 2015.

[9] L. Organick, S. Dumas Ang, and Y. J. Chen, "Random access in large-scale DNA data storage," *Nature Biotechnology*, vol. 12, no. 1, pp. 242-248, Feb. 2018.

[10] E. Willerslev, A 700.000 year old horse gets its genome sequenced, https://news.ku.dk/all_news/2013/2013.6/a-700.000-year-old-horse-gets-its-genome-sequenced, 2013.

[11] B. Alberts, et al., Essential Cell Biology, Essential Cell Biology, Garland Science, 3rd Edition, 2009.

[12] Harvard cracks DNA storage, crams 700 terabytes of data into a single gram. https://www.extremetech.com/extreme/ 134672-harvard-cracks-dna-storage-crams-700-terabytes-of-data-into-a-single-gram

[13] R. Heckel, "An archive written in DNA," *Nature Biotechnology*, vol. 36, no. 3, pp. 236-237, Mar. 2018.

[14] N. Goldman, P. Bertone, S. Chen, C. Dessimoz, E. M. LeProust, B. Sipos, and E. Birney, "Towards practical, high-capacity, low-maintenance information storage in synthesized DNA," *Nature Biotechnology*, vol. 494, no. 7435, pp. 77-80, Jan. 2013.

[15] D. Huffman, "A method for the construction of minimum-redundancy codes," in *Proc. of the IRE*, vol. 40, no. 9, pp. 1098-1101, Sept. 1952.

[16] G. M. Church, Y. Gao, and S. Kosuri, "Next-generation digital information storage in DNA," Science 337, 1628 2012.

[17] R. Grass, R. Heckel, M. Puddu, D. Paunescu, and W. Stark, "Robust chemical preservation of digital information on DNA in silica with error-correcting codes," *Angewandte Chemie International Edition*, vol. 54, no. 8, pp. 2552-2555, Feb. 2015.

[18] M. Blawat, *et al.*, "Forward error correction for DNA data storage," *Procedia Computer Science*, vol. 80, Issue: C, pp. 1011-1022, Jun. 2016.

[19] Y. Erlich and D. Zielinski, "DNA fountain enables a robust and efficient storage architecture," *Science*, vol. 355, no. 6328, pp. 950-954, Mar. 2017.

[20] S. Yazdi, R. Gabrys, and O. Milenkovic, "Portable and error-free DNA-based data storage," *Scientific Reports*, vol. 7, no. 1, Article No. 5011, Jul. 2017.

[21] A. Chandrasekaran, O. Levchenko, D. Patel, M. MacIsaac, and K. Halvorsen, "Addressable configurations of DNA nanostructures for rewritable memory," *Nucleic Acids Research*, vol. 45, no. 19, pp. 11459-11465, Jul. 2017.

[22] B. Wang, Y. Xie, S. Zhou, X. Zheng, and C. Zhou, "Correcting errors in image encryption based on DNA coding," *Molecules*, vol. 23, no. 8, p. 1878, Aug. 2018.
[23] H. Lee, R. Kalhor, N. Goela, J. Bolot, and G. Church, "Terminator-free template-independent enzymatic DNA synthesis for digital information storage," *Nature Communications*, vol. 10, no. 1, p. 2283, Jun. 2019.
[24] I. Koren, *Computer Arithmetic Algorithms*, 2nd Edition, Published by A. K. Peters, Natick, MA, 2002.
[25] DNA Replication and Causes of Mutation, Available at: .https://www.nature.com/scitable/topicpage/dna-replicationand-causes-of-mutation-409.

میثم اللهی رودپشتی در سال ۱۳۹۶ مدرک کارشناسی مهندسی کامپیوتر خود را از دانشگاه علم و فنّاوری مازندران- بهشهر و در سال ۱۳۹۸ مدرک کارشناسی‌ارشد مهندسی کامپیوتر خود را از دانشگاه صنعتی نوشیروانی بابل دریافت نمود. زمینه‌های مطالعاتی مورد علاقه نام‌برده شامل حساب کامپیوتری، اینترنت اشیاء، تحمل‌پذیری اشکال، مباحث کوانتومی و موضوعات بین رشته‌ای در علوم است. اینک نیز در حال انجام سیر مطالعاتی در زمینه‌های یادشده می‌باشد.

سعیده علی‌نژاد در سال ۱۳۸۶ مدرک کارشناسی مهندسی کامپیوتر خود را از دانشگاه صنعتی شریف و در سال ۱۳۹۰ مدرک کارشناسی ارشد مهندسی کامپیوتر خود را از دانشگاه صنعتی أصفهان و در سال ۱۳۹۴ مدرک دکتری مهندسی کامپیوتر خود را از دانشگاه صنعتی شریف دریافت نمود. دکتر علی‌نژاد از سال ۱۳۹۵ در دانشکده مهندسی برق و کامپیوتر دانشگاه صنعتی نوشیروانی بابل در مازندران مشغول به فعالیت گردید و اینک نیز عضو هیأت علمی این دانشکده می‌باشد. زمینه‌های تحقیقاتی مورد علاقه نام‌برده عبارتند از سامانه‌های ذخیره‌سازی، سیستم‌های نهفته و بلادرنگ، شبکه‌های حسگر بی‌سیم، تحمل‌پذیری اشکال و اینترنت اشیاء.