DNA Computing

ارائه شده به:

استاد رامین حلاوتی

توسط:

محمد شفیعی

هوش مصنوعی

ترم دوم، سال ۸۵-۱۳۸۴

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

فهرست منابع

مقدمه ۳

کامپیوترهای فعلی ۳

محاسبات بیولوژیکی ۴

محاسبات مبتنی بر DNA ۵

تراشه‌های DNA ۵

برنامه‌ها و روبات‌های ژنتیکی ۵

تلفیق کامپیوترهای سیلیکونی و موجودات زنده ۶

اندام‌های (موجودات) زنده کنترل شده (مهندسی) ۶

DNA در مقابل سیلیکون ۶

تاریخچه ۷

DNA و ساختار آن ۸

حل مسئله‌ی مسیر همیلتونی توسط آدلمن ۱۱

گیت‌های منطقی DNA ۱۵

مدل‌های محاسبات مبتنی بر DNA ۱۶

مدل surface-based ۱۷

مدل sticker-based ۱۸

کارهای انجام شده ۲۱

فهرست منابع ۲۲

مقدمه

پیشرفت علم و تکنولوژی چیزی نیست که بر کسی پوشیده باشد. تحرک و تحول در عرصه کامپیوتر هم همپای دیگر عرصه‌ها به سوی آینده‌ای بهتر ادمه دارد. اما این پیشرفت در علم کامپیوتر با توجه به موجودیت سخت‌افزاری آن از محدودیت‌هایی هم رنج می‌برد. اما همواره تنها یک رویه در رسیدن به یک مقصد وجود ندارد و انسان در مواجه با موانع راه‌های مختلف و امکاناتی که هر یک به دست می‌دهند و جوانب کار را بررسی و در صورت رسیدن به یک جمع‌بندی در تغییر مسیر، روند حرکت خود را تغییر می‌دهد و این شاید از خصوصیات یک موجود هوشمند(!) است.

در این تحقیق ابتدا به بعضی محدودیت‌های موجود فعلی در ادامه راه پیشرفت در عرصه کامپیوتر می‌پردازیم و به بعضی از راه‌حل‌ها اشاره خواهیم کرد و بحث را با محاسبات برمبنای DNA ادامه می‌دهیم. در این بحث ابتدا به مقایسه‌ای بین DNA و سیلیکون و در ادامه به بررسی تاریخچه مختصری خواهیم پرداخت و سپس به مشی‌ای مشابه رویه‌های فعلی که در آن از گیت‌های منطقی متشکل از DNA در ساخت کامپیوتر و محاسبات استفاده می‌شود اشاره می‌کنیم. در نهایت دو مدل برای محاسبات با DNA و امکاناتی که در این مدل‌ها برای انجام محاسبات و پیاده‌سازی الگوریتم‌ها وجود دارد می‌پردازیم و حل مسائل نمونه بار بررسی خواهیم کرد.

کامپیوترهای فعلی

اکثر [همه] ما با شرایط کنونی در محاسبات و دنیای الکترونیک و کامپیوتر به قدری خو گرفته‌ایم که با ذکر نام کامپیوتر، کلماتی مانند Keyboard، RAM، ROM، Electronic و Silicon به ذهنمان تدایی می‌شود؛ اما این سوال که «آیا همیشه باید با همین ابزار و روش به کارمان ادامه دهیم؟» جای بحث و تأمل بیشتر دارد. همانطور که در گذشته‌ای نه چندان دور شاید همین سوالات در محاسبه با چرتکه[!] مطرح بود و زمانی که کامپیوترهای فعلی پا به عرصه ظهور گذاشتند هیچکس تصور نمی‌کرد که به این سرعت چنین تأثیری در زندگی در جنبه‌های مختلف داشته باشند.

توسعه با سرعت زیاد در کوچک شدن ابعاد تراشه‌ها و افزایش سرعت آنها باعث مطرح شدن قانون مور[1] شد [1] و این توسعه همچنان ـ اگرچه با سرعت کمتر ـ ادامه می‌یابد، چنانچه عده‌ای هنوز معتقدند که قانون مور همچنان صادق و برقرار است [2].

اما این توسعه با این سرعت در آینده‌ای نه چندان دور به حد نهایت خود خواهد رسید که در نتیجه برای ادامه توسعه باید به دنبال راه‌های جایگزین بود. اما موانع ادامه راه توسعه از دو جنبه قابل بررسی‌اند.

یک جنبه این محدودیت، محدودیت‌هایی است که به دلیل قوانین فیزیکی حاکم بر مواد سازنده کامپیوترهای فعلی یا همان سیلیکون تحمیل می‌شود [3]، [4]. قدرت پردازش کامپیوترها معمولاً بر اساس سرعت آنها سنجیده می‌شود. سرعت در اینجا یعنی اطلاعات با چه سرعتی می‌توانند در مدار از نقطه "A" به "B" منتقل شوند و با چه سرعتی وقتی به نقطه "B" رسیدند، می‌توانند پردازش شوند. مدل سنتی طراحی کامپیوتر بر کم کردن فاصله مقصدی که اطلاعات باید به آن منتقل شود (به صورت سیگنال‌های الکتریکی) تمرکز دارد؛ به عبارت دیگر کم کردن فاصله بین دو نقطه "A"

---

[1] Moore's Law

B". این مسئله می‌تواند به معنای گنجاندن تعداد بیشتری واحدهای محاسبه‌گر و ترانزیستور در یک تراشه پردازشگر مرکزی در کامپیوتر باشد. امروزه این مسئله تا حد زیادی به پیش رفته و به مرزهای فیزیکی خود نزدیک شده است و برای بهبود هر چه بهتر نتیجه، اندازه تراشه‌های کامپیوتری با گذشت زمان کوچکتر شده‌اند، رفته رفته اندازه ترانزیستورها به اندازه‌های قابل مقایسه در مقابل اندازه اتم تبدیل شده‌اند. بدبختانه اثرات کوانتومی که در ابعاد کوچک رخ می‌دهد می‌تواند در آینده‌ای نه چندان دور مانع انتشار موثر سیگنال‌ها و یا تداخل‌هایی بین آنها شود. به طور مثال اصل عدم قطعیت هایزنبرگ بیان می‌کند که اجزا مثلاً الکترون‌های سازنده سیگنال اطلاعات در درون یک کامپیوتر می‌توانند رفتار عجیب بودن در مکانی دیگر غیر از مکانی که باید در آن باشند را نشان دهند. که خود می‌تواند باعث به هم ریختن مدارات انتقال اطلاعات شود. اشاره شده است که در ۱۰ تا ۱۵ سال آینده تکنولوژی به این محدودیت فیزیکی خواهد رسید [5].

مشکلات دیگری مانند سمی بودن اجزای تراشه‌های ساخته شده از سیلیکون که چه در ساخت و چه در از بین بردن آنها باعث تحمیل خطراتی برای محیط زیست می‌شود [6] و یا مسائلی مانند اینکه کامپیوترهای سیلیکونی خیلی از لحاظ مصرف انرژی بهینه نیستند [7] از نقاط منفی دیگر سیلیکون می‌تواند باشد.

جنبه دیگر محدودیت که به «تنگنای وُن نیومن [۱]» معروف است [4] به معماری کامپیوترهای فعلی که در آنها واحد پردازش مرکزی [۲] نیاز به انتقال دستورات و داده‌ها از حافظه اصلی دارد برمی‌گردد. اگر ارتباط بین CPU و حافظه اصلی را به صورت یک جاده دوطرفه و واحدهای اطلاعاتی را به صورت خودرو در نظر بگیریم وجود این تنگنا کاملاً محسوس است چرا که اغلب محاسبات به صورت واکشی تعدادی دستور توسط CPU از حافظه اصلی و سپس اجرای آنها (احتمالاً بعد از واکشی عملوندهای آن دستور) و در دوباره ذخیره نیتجه (در صورت وجود) در حافظه می‌باشد. در نتیجه تقریباً واضح است که سرعت پردازش با حد سرعت ارتباط بین CPU و حافظه اصلی محدود شده است.

با توجه به مواردی که مورد بررسی قرار گرفت بعضی از پژوهشگران به دنبال جوابی ورای این مرزها هستند، جوابی که در آن از مواد و یا تکنیک‌های متفاوتی برای فائق آمدن بر این محدودیت‌ها بهره بجویند. کامپیوترهای کوانتومی، نوری و مبتنی بر DNA به عنوان راه‌های مختلف مطرح شده‌اند [4] که ما به بررسی راه سوم خواهیم پرداخت.

محاسبات بیولوژیکی [۳]

قبل از پرداختن به محاسبات مبتنی بر DNA که حالت خاص محاسبات بیولوژیکی است ابتدا نگاهی کوتاه به انواع محاسبات بیولوژیکی که بی‌ارتباط به محاسبات مبتنی بر DNA هم نیستند می‌پردازیم. بعضی از این انواع تکنولوژی این امکان را دارند که در انواع مختلفی از مسائل به کار روند، درحالی که بقیه تنها به عنوان ابزاری برای اهداف خاص کاربرد دارند. این انواع مختلف را می‌توان به صورت زیر دسته‌بندی کرد [8]:

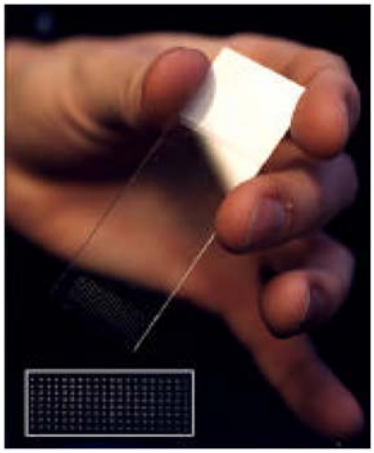
---

[۱] Von Neumann Bottleneck

[۲] Central Processing Unit (CPU)

[۳] Biological Computing

- **محاسبات مبتنی بر DNA**[1]: در این حالت که در واقع موضوع بحث ما نیز می‌باشد از رشته‌های DNA در کامپیوترهای مبتنی بر DNA برای انجام اعمال استفاده می‌شود [9]. توضیح بیشتر و بررسی کاملتر در مورد این نوع را در طول بحث خواهیم داشت.
- **تراشه‌های DNA**[2]: یک تکنولوژی بسیار مرتبط با محاسبات مبتنی بر DNA است. این ابزار در ساختار بسیار مشابه کامپیوترهای مبتنی بر DNA است. ایده اصلی استفاده از تعداد زیادی رشته DNA که بر روی تراشه شیشه‌ای کوچکی چسبانده شده‌اند می‌باشد [10]. هر یک از این رشته‌ها این قابلیت را دارند که به یک رشته ژنتیکی نمونه وابسته شوند. این تراشه‌ها این امکان را به پژوهشگران می‌دهند که هزاران رشته ژنتیکی را به صورت همزمان برای استفاده در پیشبرد درمان امراض ارزیابی کنند. تحقیقات در این زمینه در بسیاری از دانشگاه‌ها مانند دانشگاه هوستن[3] ادامه دارد و شرکت سیترن[4] هم تراشه‌های DNA برای پژوهشگران بیولوژی را تولید می‌کند [11]. در شکل ۱ نمونه‌ای از این تراشه‌ها آمده است.



شکل ۱ نمونه‌ای از تراشه‌های DNA

- **برنامه‌ها و روبات‌های ژنتیکی:** پژوهشگران برنامه‌های کامپیوتری ژنتیکی‌ای توسعه داده‌اند که می‌توان آنها را به سلولهای زنده ارائه کرد تا فرآیندهای آنها را کنترل کنند و قابلیت تکثیر توسط سلول را نیز دارند. تحقیقات تاکنون منجر به تولید رشته‌های کنترل شده ژنتیکی شده است که می‌توانند موجب تولید یکی از دو ژن ممکن (شکل ۲) در سلولی که در آن قرار می‌گیرند، شوند. این مسئله کاملاً قابل قیاس

---

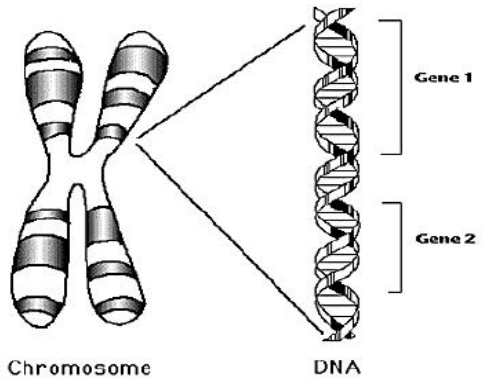
[1] DNA Computing

[2] DNA Chips

[3] Houston

[4] Xeotron

با برنامه‌های کامپیوتر است و می‌توان از این تکنولوژی به عنوان سویچ در کنترل مواد شیمیایی که موجودات زنده سنتز می‌کنند بهره برد.



شکل ۲ کروموزوم، DNA، ژن

تلاشهایی برای افزودن عناصر پردازش داده، حافظه و ارتباطی برای ساخت روبات‌های کوچک ژنتیکی که می‌توانند در سلول‌ها به انجام کار بپردازند ادامه دارد. این امکان می‌توانند در درمان خودکار سلول در صورت بروز مشکل کمک کند.

- **تلفیق کامپیوترهای سیلیکونی و موجودات زنده:** هدف این روش استفاده از موجودات زنده در کنترل تکنولوژی است که از طریق ارتباط اعصاب سلول‌های زنده و اجزای کامپیوترهای سیلیکونی میسر می‌شود. دلیل این کار این است که مغز انسان و در درجات ضعیف‌تر موجودات دیگر، قابلیت فهم مسائل پیچیده‌ای را دارد که با هیچمقدار از قدرت پردازشی مبتنی بر سیلیکون قابل اداره نیست. همچنین این اعصاب می‌توانند مسائل را با بخشی از اطلاعات به درستی پاسخ دهند. به طور مثال از اعصاب برای انجام اعمال ریاضی [12] استفاده شده است. کارهایی برای استفاده از مغز نوعی مارماهی در کنترل روبات انجام شده است و در آزمایشی نیز یک تراشه کامپیوتر به عصب بازو پروفسور کوین وارویچ متصل شد و ظرف چهار ماه به وسیله آن یک روبات را کنترل می‌کرد [8].
- **اندام‌های (موجودات) زنده کنترل شده (مهندسی):** افراطی‌ترین شکل در تحقیقات محاسبات بیولوژیکی یک قدم دیگر نسبت به مدل‌های تلفیقی با قصد مهندسی اندام‌های زنده به پیش رفته‌اند. تحقیق در کشت شبکه‌های عصبی از بافت‌های عصبی زنده در حال انجام است. البته این دسته در مراحل بسیار مقدماتی قرار دارد و نمی‌توان به حصول نتیجه در آن در آینده‌ای نزدیک امیدوار بود.

DNA در مقابل سیلیکون

بعد از بررسی شیوه‌های مختلف محاسبات بیولوژیکی در این قسمت به مقایسه‌ای بین DNA و سیلیکون در استفاده در ساخت کامپیوترها و انجام محاسبات می‌پردازیم و به نقاط قوت و ضعف هر یک به اجمال اشاره‌ای خواهیم کرد.

همانطور که در قبل هم اشاره شد در ادامه کار با کامپیوترهای سیلیکونی و توسعه آنها با مشکل اثرات کوانتومی روبه‌رو خواهیم شد؛ همینطور اشاره شد که در استفاده از سیلیکون با مواد سمی سر کار داریم و در ضمن این کامپیوترها از لحاظ مصرف انرژی هم زیاد بهینه نیستند در مقابل این نکات منفی در مورد DNA وجود ندارد یعنی نه با اثر کوانتومی مواجه هستیم و نه با مواد سازنده سمی سروکار داریم و نیتجه حاصل هم در مصرف انرژی بهینه‌تر است.

معیار مقایسه	مبتنی بر DNA	مبتنی بر سیلیکون
اجزای سازنده	مواد بیولوژیکی مانند اسیدهای آمینه	سیلیکون و مواد غیرآلی دیگر
طرح پردازشی	پردازش موازی به صورت گسترده	ترتیبی و پردازش موازی به صورت محدود
حداکثر تعداد اعمال در ثانیه	10^{14} (در سال ۲۰۰۲)	10^{12} (در سال ۲۰۰۲)
مشکل اثر کوانتومی	ندارد	دارد
اجزای سمی	ندارد	دارد
بهینگی انرژی	دارد	ندارد

جدول ۱ مقایسه بین DNA و سیلیکون

یکی از نقاط منفی DNA کندی در انجام هر عمل است به طوری که شاید این کندی تا ۱۰ برابر در انجام یک عمل به تنهایی به وسیله DNA مشهود باشد ولی به لطف امکان موازی‌سازی بالا و انجام اعمال به طور موازی و نبودن تنگنای ون نیومن که در قبل به آن اشاره شد می‌توان تعداد انجام محاسبات در ثانیه را به مقادیری بالاتر از توان سیلیکون در ثانیه رساند و در واقع نکته قوت DNA در انجام محاسبات حجیم در واقع همین امکان موازی‌سازی و طرح پردازشی موازی آن است. خلاصه مقایسه‌ای که در این بخش انجام شد برای وضوح بیشتر در جدول ۱ آمده است.

تاریخچه

در انتهای دهه‌ی ۵۰ یک فیزیکدان به نام ریچارد فینمن[۱] برای نخستین بار ایده استفاده از سلول‌های زنده و مجموعه‌های پیچیده مولکولی در ساخت کامپیوترهای (محاسبه‌گر) میکروسکوپی را مطرح کرد. در سخنرانی معروفش به نام «فضای زیادی در کف وجود دارد[۲]» [13]، فینمن درباره‌ی مشکل کار کردن و کنترل کردن اجزا در ابعاد و مقیاس کوچک بحث کرد و رشته نانوتکنولوژی را بیان نهاد. اگرچه او به صورت عمده روی ذخیره اطلاعات و کار در سطح

[۱] Richard Feynman

[۲] There's Plenty of Room at the Bottom

مولکولی تمرکز کرد، به پتانسیل سیستم‌های بیولوژیکی به عنوان پردازنده‌های اطلاعات در مقیاس کوچک (از نظر اندازه) اشاره کرد [13].

بعد از طرح دیدگاه فینمن هیاهوی زیادی برای انجام محاسبات در سطح مولکولی ایجاد شد. پیشرفت‌های اولیه بیشتر به صورت تئوری بودند و برای به حقیقت پیوستن آنها باید تا توسعه روش‌ها و ابزارهای عملی کردن آن صبر می‌کرد. در سال ۱۹۹۴ آدلمن[۱]، در نهایت نشان داد که چگونه یک جستجوی تصادفیِ موازیِ حجیم ممکن است با استفاده از اعمال استاندارد بر روی رشته‌های DNA پیاده‌سازی شود [14]. کارهای قبلی عموماً از پروتئین‌ها استفاده می‌کردند ولی ادلمن در کارش از DNA استفاده کرد. به جزئیات این کار در بخش‌های بعدی خواهیم پرداخت.

در سال ۱۹۹۷ در دانشگاه راچستر[۲] گیت‌های منطقیِ DNA ای ساخته شد [6] که به جای سیگنال‌های الکتریکی، ورودی و خروجی آنها DNA است در ادامه به این مورد هم خواهیم پرداخت.

بعدها اِهود شپیرو[۳] امکان برنامه‌ریزی مولکول‌ها با اطلاعات پزشکی و دارویی و تزریق آنها به افراد را بررسی کرد و در سال ۲۰۰۱ توانست کوچکترین کامپیوتر که به اندازه‌ی یک قطره آب بود و از مولکول‌های DNA و آنزیم‌ها به عنوان ورودی و خروجی و نرم‌افزار و سخت‌افزار استفاده می‌کرد، بسازد [15].

تحقیقات در زمینه‌ی کامپیوترهای DNA ای همچنان ادامه دارد و مدل‌هایی هم برای این کامپیوترها ارائه شده است که در آنها به کمک بیولوژی و نه الکترونیک اعمالی بر روی DNA تعریف و پیاده‌سازی شده است که محاسبات و برنامه‌ریزی برای کامپیوترهای DNA ای بر اساس آنها انجام می‌شود و حتی بر طبق آنها برای این کامپیوترها کامپایلر هم ارائه شده است که در ادامه بحث به بعضی از این مدل‌ها و امکانات هر یک خواهیم پرداخت.

DNA و ساختار آن

برای ادامه بحث نخست لازم است که اطلاعات کمی در مورد DNA و ساختار آن داشته باشیم؛ بنابراین در این بخش به این موضوع خواهیم پرداخت.

DNA (DeoxyriboNocleic Acid) یک مولکول حجیم با ساختار دورشته‌ای و مارپیچ است (شکل ۳). پیوندهای بین دو رشته از نوع پیوندهای هیدوژنی و ضعیف‌اند به طوری که معمولاً با استفاده از حرارت و گرما شکسته می‌شوند؛ در مقابل پیوند بین بازهای سازنده، در هر رشته که از طریق فسفات‌ها برقرار می‌شود از نوع کووالانس بوده که پیوندهای قوی‌ای محسوب می‌شوند.

بازهای تشکیل دهنده DNA عبارتند از Adenine، Cytosine، Guanine و Thymine که به ترتیب با A، C، G و T به آنها اشاره خواهیم کرد. خصوصیت جالب توجه این است که در تشکیل مارپیچ‌های دوگانه‌ی DNA فقط A با T و C با G تشکیل پیوند می‌دهند و از این خصوصیت رشته‌های DNA در محاسبات و تولید رشته‌های جدید با استفاده از یک طرح اولیه استفاده می‌شود.

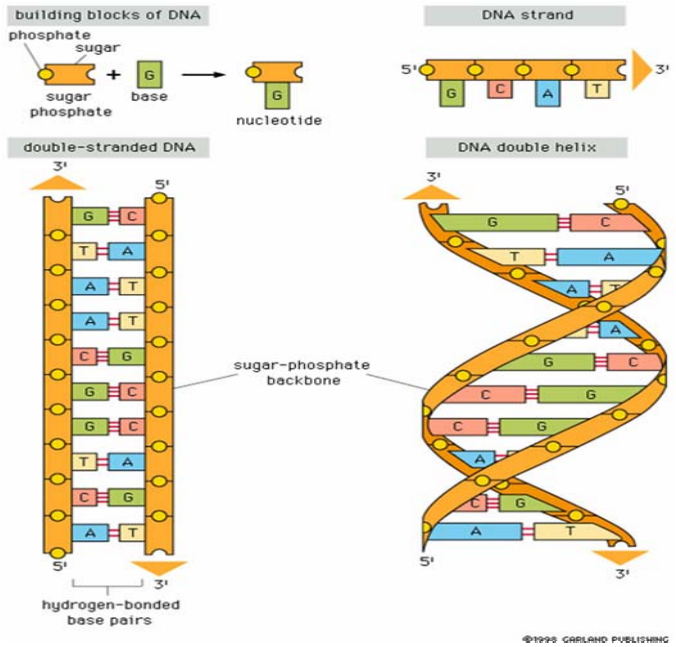
---

[۱] Adleman

[۲] Rochester

[۳] Ehud Shapiro

حال که به ساختار مولکول DNA تا حدی پرداختیم در ادامه به اعمالی که می‌توانیم بر روی این رشته‌ها انجام دهیم و با استفاده از آنها محاسبات DNA ای و در واقع کامپیوترهای DNA ای را به پیش ببریم، می‌پردازیم. اَدلمن هم برای اولین بار محاسبات خود را توسط همین اعمال انجام داد و به نتیجه رسید که در بخش بعد به آن می‌پردازیم.



شکل ۳ DNA و ساختار آن

مواردی که برای کار با DNA موجودند عبارتند از [16]:

- **جفت شدن[1] Watson-Crick**: همانطور که پیشتر اشاره شد در هر رشته‌ی DNA فقط باز A با T و C با G جفت می‌شوند؛ بنابراین هر رشته DNA یک مکمل مشخص دارد که در آن به ازای هر باز متناظر جفت‌شونده آن باز وجود دارد. به این رشته مکمل، مکمل Watson-Crick می‌گویند. اگر مثلاً در مجموعه جوابی که با آن کار می‌کنیم رشته‌ی مکمل یک رشته موجود باشد، این دو رشته با هم جفت می‌شوند و ساختار مارپیچ تشکیل می‌دهند. اما همانطور که گفتیم این پیوند از نوع هیدروژنی و شکننده است. در ضمن اگر در مجموعه جواب دو رشته‌ی مکمل و یا دو رشته با زیررشته مکمل قابل ملاحظه با هم موجود نباشد هیچ جفت‌شدگی‌ای رخ نخواهد داد.
- **پلیمریس‌ها[2]**: پلیمریس‌ها اطلاعات را از یک مولکول به دیگری کپی می‌کنند. به طور مثال پلیمریس DNA مکمل Watson-Crick یک رشته DNA را از روی یک الگوی DNA می‌سازد. در واقع پلیمریس DNA یک سیگنال شروع لازم دارد که به بگوید از کجا شروع به ساختن کپی مکمل کند. این سیگنال

---

[1] pairing

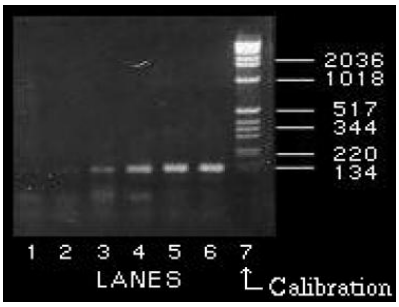
[2] polymerases

[1] ـ یک تکه (معمولاً کوتاه) از DNA که با مکمل Watson-Crick توسط الگوی مشخص جفت می‌شود ـ تأمین می‌شود. هر جا یک چنین چاشنی الگوی مکملی یافت شود، پلیمریس DNA شروع به افزودن بازها به چاشنی برای ساختن کپی مکمل بر اساس آن الگو (چاشنی) می‌کند.

- **لیگیس‌ها** [2]: لیگیس‌ها مولکول‌ها را به هم می‌چسبانند. برای مثال لیگیس DNA دو رشته‌ی DNA نزدیک هم را می‌گیرد و آنها را با پیوندهای کووالانسی به هم پیوند می‌دهد. لیگیس DNA توسط سلول برای ترمیم پاره‌گی در رشته‌های DNA بعد از قرار گرفتن پوسته سلول در معرض اشعه ماوراءبنفش استفاده می‌شود.

- **نوکلیس‌ها** [3]: نوکلیس‌ها، اسیدهای نوکلوئیک را می‌برند (در مقابل عمل چسباندن لیگیس‌ها). مثلاً محدود کننده‌ی درون‌نوکلیس‌ها [4] یک رشته DNA را برای یک رشته از پیش مشخص شده از بازها جستجو می‌کند و وقتی چنین رشته‌ای را یافت، مولکول را به دو تکه می‌شکند. EcoRI (از Escherichia coli) یک آنزیم محدودکننده است که DNA را بعد از G در دنباله‌ی GAATTC می‌شکند (تقریباً هیچ جای دیگر رشته‌ی DNA را نمی‌شکند). در کار اَدلمن از این نوع آنزیم‌ها استفاده نشده است ولی در کارهای بعدی از آنها استفاده شده است.

- **الکتروفوریسیس ژل** [5]: این مورد در سلول وجود ندارد. یک محلول ناهمگن مولکول‌های DNA در یک انتهای ورقه‌ای از ژل قرار داده می‌شود و جریان الکتریکی به آن متصل می‌شود. مولکول‌های DNA که دارای بار منفی شده‌اند به سمت آند حرکت می‌کنند؛ اما رشته‌های کوتاه با سرعت بیشتری از رشته‌های بلند حرکت می‌کنند. این فرآیند رشته‌های DNA را بر اساس طول جدا می‌کند. با استفاده از مواد شیمیایی ویژه و اشعه ماوراءبنفش این امکان وجود دارد که نوارها را در ژل، در جایی که مولکول‌های DNA متوقف می‌شوند دید.



شکل ۴ تصویرگری ژل

---

[1] primer

[2] ligases

[3] nucleases

[4] restriction endonucleases

[5] gel electrophoresis

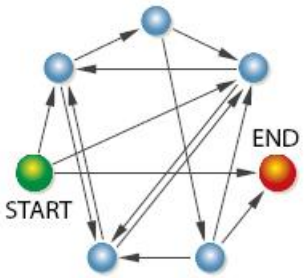
- **سنتز DNA**[1]: این امکان وجود دارد که یک رشته DNA را روی یک تکه کاغذ نوشت و آنرا به شرکت‌های سنتز تجاری داد و چند روز بعد یک لوله آزمایش که تقریباً حاوی 10^{18} مولکول DNA که همگی یا تقریباً همه منطبق بر رشته توصیف شده هستند، دریافت کرد. در زمان آزمایش اَدلمن رشته‌های با طول تقریبی ۱۰۰ می‌توانستند با این روش تولید شوند و برای هر رشته به طول ۲۰ هزینه‌ای تقریباً ۲۵ دلار نیاز بود (برای دریافت 10^{18} مولکول!). مولکول‌ها به صورت خشک در یک لوله کوچک به شکل یک جسم جامد سفید رنگ کوچک دریافت می‌شوند.

حل مسئله‌ی مسیر همیلتونی توسط اَدلمن

بعد از پرداختن به ابزارها و امکانات موجود برای کار با رشته‌های DNA در این قسمت به بررسی نخستین مسئله‌ای که توسط اَدلمن حل شد یعنی مسئله مسیر همیلتونی خواهیم پرداخت.

مسئله‌ی مسیر همیلتونی عبارت است از تعیین اینکه آیا در یک گراف با رئوس و یال‌های مشخص، مسیری همیلتونی (مسیری که از تمام رئوس و از هر رأس فقط یک بار عبور کند) از یک رأس مشخص به یک رأس معین وجود دارد یا خیر؛ حالت کلی‌تر این مسئله در نظر گرفتن یال‌های گراف به صورت جهتدار است. این مسئله NP-complete است و تاکنون یک الگوریتم بهینه برای حل کردن آن ارائه نشده است.

مسئله‌ی مسیر همیلتونی که توسط اَدلمن مورد توجه و بررسی قرار گرفت، از یک گراف با ۷ رأس و ۱۴ یال جهتدار تشکیل شده بود [16] (شکل ۵). اگرچه این مسئله از لحاظ حل، مسئله‌ای نسبتاً ساده (به دلیل تعداد کم رئوس آن) می‌باشد اما از این جنبه حائز اهمیت است که حل یک مسئله‌ی عام را با استفاده از DNA نشان می‌دهد و امکان استفاده از کامپیوترهای مبتنی بر DNA به عنوان کامپیوترهای همه‌منظوره را نشان می‌دهد و بعد از آن کارهای بسیاری در این زمینه و بر اساس این کار انجام شده است.



شکل ۵ مسئله‌ی مسیر همیلتونی حل شده توسط اَدلمن به وسیله‌ی DNA

الگوریتم مورد استفاده توسط اَدلمن برای حل مسئله توسط DNA به قرار زیر بود:

برای یک گراف با n رأس:

۱. یک مجموعه از مسیرهای تصادفی در گراف تولید کن.

---

[1] DNA synthesis

۲. برای هر مسیر در مجموعه:

أ. بررسی کن که آیا مسیر با رأس مورد نظر شروع و به رأس مشخص شده ختم می‌شود. اگر چنین نیست، آن مسیر را از مجموعه حذف کن.

ب. بررسی کن که مسیر دقیقاً از n رأس می‌گذرد. اگر چنین نیست آن مسیر را از مجموعه حذف کن.

ج. به ازای هر رأس، بررسی کن که آن مسیر از آن رأس بگذرد. اگر نمی‌گذرد آن مسیر را از مجموعه حذف کن.

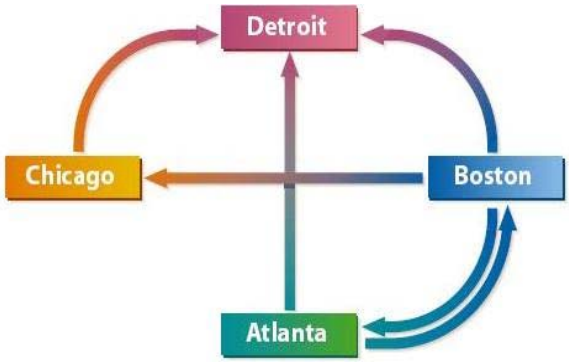
۳. اگر مجموعه تهی نیست، یک مسیر همیلتونی وجود دارد؛ در غیر این صورت مسیر همیلتونی‌ای وجود ندارد.

اگرچه این الگوریتم صددرصد کامل نیست، ولی اگر ساختن مسیر به اندازه کافی تصادفی باشد و مجموعه حاصل هم به اندازه کافی بزرگ باشد، احتمال زیادی وجود دارد که جواب درست حاصل شود.

برای سادگی بیشتر در اینجا کارهایی که اَدلمن برای حل مسئله انجام داد [16] را روی یک گراف جهتدار با ۴ رأس و ۶ یال (شکل ۶) دنبال می‌کنیم.

CITY	DNA NAME	COMPLEMENT
ATLANTA	ACTTGCAG	TGAACGTC
BOSTON	TCGGACTG	AGCCTGAC
CHICAGO	GGCTATGT	CCGATACA
DETROIT	CCGAGCAA	GGCTCGTT

FLIGHT	DNA FLIGHT NUMBER
ATLANTA - BOSTON	GCAGTCGG
ATLANTA - DETROIT	GCAGCCGA
BOSTON - CHICAGO	ACTGGGCT
BOSTON - DETROIT	ACTGCCGA
BOSTON - ATLANTA	ACTGACTT
CHICAGO - DETROIT	ATGTCCGA



شکل ۶ مسئله‌ی مسیر همیلتونی با ۴ رأس و ۶ یال و کد کردن مسئله به وسیله‌ی DNA

مسئله‌ی مورد بررسی در اینجا تعیین وجود یک مسیر همیلتونی از آتلانتا به عنوان رأس مبدأ، به دیترویت به عنوان رأس مقصد می‌باشد. همانطور که در شکل ۶ واضح است تنها یک چنین مسیری یعنی «آتلانتا ← بوستون ← شیکاگو ← دیترویت» وجود دارد و مثلاً از دیترویت به آتلانتا چنین مسیری وجود ندارد.

کار با الصاق دنباله‌های DNA تصادفی به هر شهر شروع می‌شود. برای مثال به آتلانتا دنباله‌ی ACTTGCAG داده می‌شود؛ دنباله نسبت داده شده به هر شهر در شکل ۶ آمده است. نیمه‌ی اول هر دنباله را به عنوان قسمت اول و نیمه دوم را به عنوان قسمت دوم نام شهر در نظر می‌گیریم (همانند اسم و فامیل! مثلاً فامیلی آتلانتا GCAG و اسم بوستون TCGG خواهد بود!). اکنون به هر پرواز مستقیم از یک شهر به شهر دیگر (هر یال) یک شماره پرواز DNA ای که از تلفیق قسمت دوم نام شهر مبدأ با قسمت اول نام شهر مقصد (فامیلی مبدأ + اسم مقصد) به دست می‌آید، می‌دهیم. مثلاً شماره پرواز از آتلانتا به بوستون GCAGTCGG می‌باشد. کلیه شماره پروازها که به این روش کد شده‌اند نیز در شکل ۶ موجودند.

همانطور که قبلاً بیان کردیم هر رشته‌ی DNA مکمل Watson-Crick خاص خود را دارد. بنابراین هر شهر مکمل DNA ای خاصِ نام خود دارد. مثلاً مکمل نام آتلانتا به صورت TGAACGTC خواهد بود. مکمل هر نام هم در شکل ۶ آمده است.

بعد از انجام کد کردن مسئله به صورتی که در پیش آمد، اَدلمن مکمل نام شهرها و شماره پروازها را به صورت رشته DNA تولید کرد (البته طول نام شهرها به نظر می‌رسد که می‌تواند بسیار کوتاهتر باشد؛ ولی همانطور که قبلاً اشاره کردیم تولید این رشته‌ها بسیار ارزان است!). او تقریباً $۱۰^{۱۴}$ مولکول از هر رشته متفاوت دریافت کرد و آنها را در یک لوله آزمایش قرار داد. برای شروع محاسبه، مقداری آب همراه لیگیس، نمک و تعدادی جزء دیگر برای شبیه‌سازی شرایط درون سلول افزود. با این حال تنها از یک پانزدهم یک قاشق چایخوری از محلول حاصل را استفاده کرد و در حدود یک ثانیه بعد جواب مسئله را در دست داشت (در آن زمان حل این مسئله با روش‌های معمول به طور تقریبی ۵۴ ثانیه طول می‌کشید).

برای اینکه ببینیم چگونه به جواب رسید، آنچه درون لوله آزمایش رخ داد را بررسی می‌کنیم. با توجه به شیوه کد کردن، مثلاً شماره پرواز آتلانتا ـ بوستون (GCAGTCGG) و مکمل نام بوستون (AGCCTGAC) ممکن است به طور اتفاقی با هم روبه‌رو شوند. بر اساس طراحی کدها، رشته اول به TCGG ختم و رشته دوم با AGCC شروع می‌شود. از آنجائیکه این رشته‌ها مکمل یکدیگرند، به هم خواهند چسبید. اگر نتیجه‌ی حاصل با شماره‌ی پرواز بوستون ـ شیکاگو (ACTGGGCT) رودررو شود، این رشته نیز به مجموعه‌ی حاصل از تلفیق قبل خواهد چسبید؛ چون انتهای نتیجه‌ی حاصل یعنی TGAC مکمل شروع این رشته که ACTG می‌باشد، است. بدین ترتیب طول ترکیبات با چسبیدن شماره‌های پرواز به هم با استفاده از مکمل نام شهرها، افزایش می‌یابد. لیگیس موجود در مخلوط هم باعث چسبیدن دائم زنجیرهای DNA ی شماره پروازها می‌شود. در نتیجه لوله‌ی آزمایش حاوی مولکول‌هایی که کد شده‌ی مسیرهای تصادفی بین شهرهای مختلف هستند، خواهد بود (مطابق آنچه در مرحله‌ی اول الگوریتم لازم بود).

از آنجایی که تعداد مولکول‌های اولیه DNA بسیار زیاد بوده است و در مسئله تنها تعداد کمی شهر وجود دارد، اطمینان تقریبی از وجود حداقل یک رشته که مسیر همیلتونی را کد کرده است وجود دارد.

نکته‌ی جالب توجه این است که کلیه مسیرها در یک لحظه و با تراکنش همزمان بین صدها تریلیون مولکول ساخته می‌شوند. این واکنش بیولوژیکی نشانگر یک پردازش موازی حجیم است.

برای مسئله‌ای که بررسی می‌کردیم تنها یک جواب که «آتلانتا ← بوستون ← شیکاگو ← دیترویت» می‌باشد وجود دارد. بنابراین مولکولی که کد شده جواب خواهد بود رشته‌ی GCAGTCGGACTGGGCTATGTCCGA خواهد بود.

در مورد مسئله‌ای هم که اَدلمن حل کرد، متأسفانه اگرچه مطمئن بود که جواب بین رشته‌های حاصل است ولی در حدود ۱۰۰ تریلیون مولکول هم که مسیرهای غیر همیلتونی را کد کرده بودند، وجود داشت که باید حذف می‌شدند. برای حذف مولکول‌هایی که نه با شهر مبدأ شروع و نه با شهر مقصد خاتمه می‌یافتند، اَدلمن از واکنش زنجیری پلیمریس ((PCR (Polymerase Chain Reaction) استفاده کرد. این تکنیک مهم نیازمند تعداد زیادی کپی از دو تکه کوچک از DNA به عنوان چاشنی است که سیگنال شروع رونوشت Watson-Crick را به پلیمریس DNA بدهد. چاشنی‌هایی که استفاده شدند قسمت دوم نام شهر مبدأ (GCAG برای آتلانتا در مثالی که در اینجا بررسی می‌کنیم) و مکمل Watson-Crick قسمت اول نام شهر مقصد (GGCT برای دیترویت در این مثال) بودند. این دو چاشنی با هم

DNA مکمل رشته‌ای که شهر مبدأ صحیح را دارد کپی کند و دومی باعث تکرار و تکثیر مولکول‌هایی که شهر مقصد درست را کد کرده‌اند می‌شود.

PCR با چرخش حرارتی، بالا و پایین بردن مداوم دمای مخلوط در لوله آزمایش به پیش می‌رود. شرایط گرم باعث ترغیب پلیمریس DNA در شروع تکثیر می‌شود. یک محیط داغ باعث جدا شدن رشته‌های به هم چسبیده در ساختار مارپیچی دوگانه، و تکثیر دوباره هر تکه می‌شود.

در نتیجه حاصل از این عملیات، مولکول‌هایی که شهرهای مبدأ و مقصد درست دارند به صورت نمایی تکثیر شده‌اند. در مقابل مولکول‌هایی که حاوی مبدأ صحیح ولی مقصد نادرست یا برعکس می‌باشند، بسیار آرامتر و به صورت خطی تکثیر می‌شوند. رشته‌های DNA که نه مبدأ و نه مقصد درست دارند اصلاً تکثیر نمی‌شوند. در نتیجه اَدلمن با برداشتن حجم کمی از مخلوط حاصل بعد از تکمیل PCR ، یک محلول ـ که حاوی کپی‌های زیادی از مولکول‌ها با مبدأ و مقصد درست ولی تعداد کمی از مولکول‌هایی که دارای این شرایط نبودند، بود ـ در اختیار داشت. در نتیجه قسمت اول از مرحله دوم الگوریتم هم کامل شد.

سپس اَدلمن از الکتروفورسیس ژل برای مشخص کردن مولکول‌هایی که طول صحیح داشتند استفاده کرد (در مثال ما این طول برابر ۲۴ است). کلیه مولکول‌های دیگر دور انداخته شدند. با این کار قسمت (ب) از مرحله ۲ الگوریتم هم به اتمام رسید.

برای بررسی رشته‌های باقیمانده برای اینکه مسیر کد شده در آنها از تمام شهرهای میانی عبور می‌کند، اَدلمن از جفت شدن Watson-Crick در یک رویه به نام جداسازی وابستگی [۱] بهره برد. این رویه از چندین کپی از مولکول‌های کاوش [۲] DNA که مکمل نام یک شهر مشخص (مثلاً بوستون) را کد کرده‌اند استفاده می‌کند. این کاوشگرها به توپ‌های مکروسکوپی آهن که تقریباً قطری در حدود یک میکرون دارند، متصل می‌شوند.

اَدلمن این توپ‌ها را در لوله آزمایش حاوی مولکول‌ای باقیمانده در شرایط ترغیب جفت شدن Watson-Crick آویزان کرد. فقط آن مولکول‌هایی که دارای نام شهر مورد نظر (بوستون) هستند به کاوشگرها می‌چسبد. سپس با استفاده از یک آهنربا در دیواره لوله آزمایش توپ‌های آهنی را در حالی که بقیه محلول را بیرون می‌ریخت، روی دیواره نگه داشت. آنچه باقیمانده بود دارای نام شهر مورد نظر بود.

سپس دوباره با استفاده از حلال و بالا بردن دما، در حالی که آهنربا را دور کرده بود رشته‌های جدا شده را از توپ آهنی جدا کرد و توپ را که دیگر رشته‌ای به آن نچسبیده بود با استفاده از یک آهنربا از لوله آزمایش خارج کرد (رشته‌های حاصل همگی مسیرهایی را کد کرده‌اند که از بوستون عبور می‌کنند). این کار را برای شهرهای دیگر هم تکرار کرد و در انتهای کلیه جداسازی‌ها قسمت (ج) از مرحله دوم الگوریتم هم کامل شده بود. اما مولکول‌هایی که اکنون در لوله آزمایش بودند حتماً باید مسیر همیلتونی مورد نظر را کد کرده باشند و از آنجایی که لوله آزمایش تهی نبود پس مسئله جواب داشت (مرحله ۳). با استفاده از الکتروفورسیس ژل، رشته‌ی کد شده نیز بررسی و مسیر مورد نظر بدست آمد.

---

[۱] affinity separation

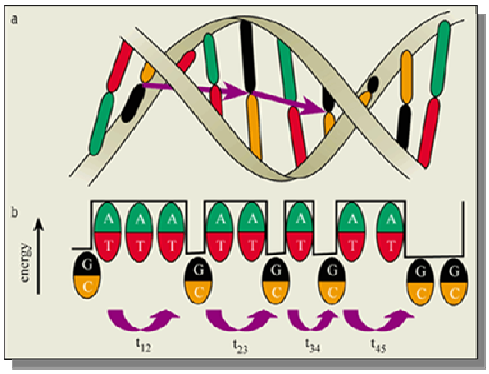
[۲] probe

چند نکته در این حل مسئله جالب توجه است: یکی امکان انجام محاسبات به صورت موازی در حجم وسیع که یک نقطه قوت قابل توجه به حساب می‌آید و دیگری زمان نسبتاً زیاد برای تفکیک جواب و سختی جداسازی است (انجام کار جداسازی توسط اَدلمن ۷ روز به طول انجامید). شاید همین نکته دوم و انجام کار به صورت دستی موجب شده است که هنوز کامپیوترهای همه منظوره مبتنی بر DNA نتوانند به تولید انبوه برسند و جایگزین کامپیوترهای مبتنی بر سیلیکون شوند. در ضمن در کار اَدلمن از ماشین تورینگ که دارای نوار ورودی و خروجی و اعمال و الفبایی برای کار بر روی این نوار است الهام گرفته شده و اَدلمن از متناظر ساختن رشته‌های DNA به عنوان نوار و A، C، G و T به عنوان الفبا و موارد ذکر شده در قسمت قبل به عنوان اعمال بهره جسته است.

گیت‌های منطقی DNA

توضیح کامل چگونگی ساخت گیت‌های منطقی از DNA که بر روی رشته‌های DNA عمل می‌کنند و چگونگی فرآیند کار آنها از حوصله بحث خارج است. در این قسمت تنها به دو رویه‌ی مختلف در ساخت گیت‌های منطقی از DNA می‌پردازیم.

یک رویه در ساخت گیت‌های منطقی استفاده از انتقال حفره در DNA است [17]. انتقال الکترون در DNA عموماً به دلیل فرآیند اکسید شدن گوانین است و در واقع انتقال حفره در DNA بیانگر وارد شدن تخریت اکسیدی به گوانین است. گوانین یک حفره بدست می‌دهد که می‌تواند در DNA با فرآیندهای پرش حرکت کند (شکل ۷).



شکل ۷ انتقال حفره در DNA

متوکسی‌بنزودیزادنین [۱] یک نوکلیوباز [۲] مصنوعی است که اخیراً برای انتقال بهینه حفره در DNA توسعه داده شده است.

رویه‌ی دیگر استفاده از سنجاق سر [۳] DNA در ساخت گیت‌های منطقی است [18]. سنجاق سر DNA ساختاری از DNA است که در آن یک رشته‌ی DNA به دلیل وجود مکمل Watson-Crick قابل توجه قسمتی از آن رشته در خود

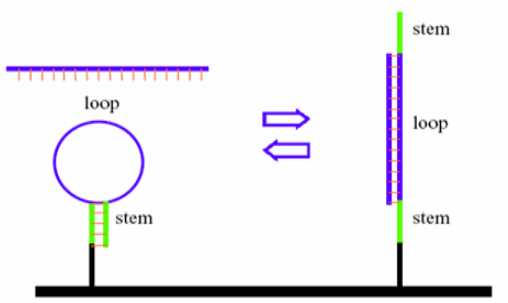
---

[۱] methoxybenzodeazaadenine

[۲] nucleobase

[۳] hairpin

[1]، موجب چسبیدن رشته به خودش در آن ناحیه می‌شود ولی در ناحیه بین این قسمت به نام حلقه [2] رشته DNA به صورت جفت نشده باقی می‌ماند (شکل ۸).



شکل ۸ سنجاق سر DNA

ساختار سنجاق سر در مولکول DNA در بیوسنسورهای متنوعی به طور گسترده استفاده شده است و همچنین بر اساس این ساختار مدل‌هایی برای ساخت گیت‌های منطقی که در آنها از کاشت این ساختار روی یک سطح استفاده می‌شود، ارائه شده است [18]. اما این مدل‌ها یکبار مصرف بوده و بعد از یک استفاده از بین می‌روند که یک نقطه ضعف به حساب می‌آید.

تاکنون گیت‌های متفاوتی مانند and، or، not، nor و nand توسط این دو رویه ساخته و مدارات منطقی شبیه‌سازی شده‌اند. ساخت این گیت‌ها که گیت‌های پایه می‌باشند امکان ساخت کامپیوترهای همه‌منظور مبتنی بر مدل‌های منطقی کنونی را فراهم می‌آورد ولی همانطور که اشاره شد بعضی از این گیت‌ها یکبار مصرفند و با تولید یک خروجی خود از بین می‌روند که در این صورت کامپیوترهای یکبار مصرف خواهیم داشت. در ضمن مسئله بهینگی استفاده از مدل منطقی و گیت‌های منطقی و توجه به امکان پردازش موازی در DNA از مسائل دیگر پیش رو پیشرفت این کامپیوترها می‌باشد.

مدل‌های محاسبات مبتنی بر DNA

همانطور که قبلاً هم اشاره کردیم محاسبات مبتنی بر DNA مطابق روش اَدلمن دارای نقاط ضعفی مانند جدا کردن جواب به صورت دستی است و یا به عبارت دیگر جنبه ابتکار نیز در انجام کار بسیار دخیل است. برای آنکه یک چارچوب استاندارد در انجام محاسبات با استفاده از DNA وجود داشته باشد مدل‌هایی برای این گونه محاسبات ارائه شده است که در این قسمت به دو مدل معروفتر اشاره ولی از توضیح نکات پیاده‌سازی خودداری خواهیم کرد و فقط به عملگرهای پایه‌ای که هر مدل در اختیار ما می‌گذارد خواهیم پرداخت.

---

[4] stem

[1] loop

مدل surface-based

در مورد این مدل تنها به ذکر عملگرهای پایه اکتفا می‌کنیم. عملگرهای پایه‌ای که این مدل در اختیار ما قرار می‌دهد عبارتند از [19]:

- **mark(i, b):** مکان i در تمام رشته‌های DNA در مجموعه جواب را با b علامت‌گذاری می‌کند.
- **mark((i_1, b_1), (i_2, b_2), ..., (i_k, b_k)):** حالت کلی‌تر حالت قبل است و k مکان و هر مکان i را با b_i علامت‌گذاری می‌کند.
- **destroy-marked:** رشته‌های علامت‌گذاری شده را از مجموعه جواب حذف می‌کند.
- **destroy-unmarked:** رشته‌های علامت‌گذاری نشده را از مجموعه جواب حذف می‌کند.
- **unmark:** علامت‌های تمام رشته‌های علامت‌گذاری شده در مجموعه جواب را پاک می‌کند.
- **test-if-empty:** برای بررسی اینکه آیا مجموعه جواب حاوی رشته‌ای است یا خیر کاربرد دارد (برای تشخیص وجود جواب).

در زیر دو الگوریتم جستجو متفاوت برای حل مسئله SAT (با فرض اینکه فرمول بولی به صورت CNF است) بر اساس این مدل آمده است. در این دو الگوریتم که به صورت شبه کد ارائه شده‌اند هدف نشان دادن چگونگی استفاده از عملگرهای پایه می‌باشد و به نکات پیاده‌سازی و جزئیات کار پرداخته نشده است (1 بیانگر true و 0 بیانگر false است).

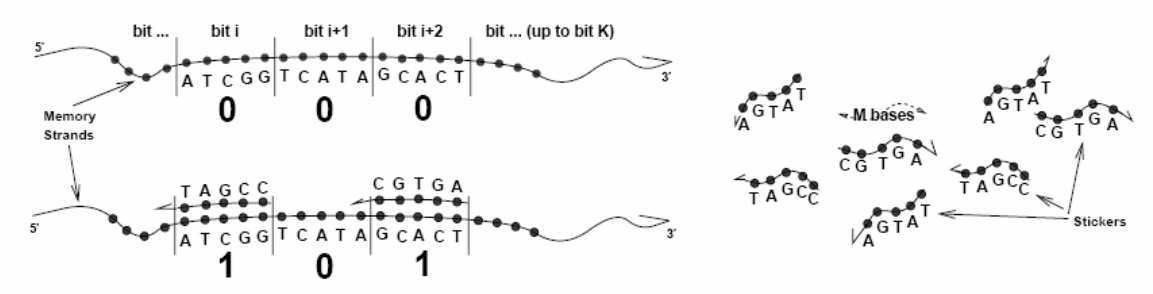
```
First Algorithm:
for each clause C of the form (x_i1 ∨ … ∨ x_ik ∨ x̄_j1 ∨ … ∨ x̄_jw) do
   mark((i_1, 0), …, (i_k, 0), (j_1, 1), …, (j_w, 1))
   comment: all remaining solutions that set C to “false” are marked
   destroy-marked
test-if-empty

Second Algorithm:
for each clause C do
   for each unnegated variable x_i in C do
      mark(i, 1)
   for each negated variable x_i in C do
      mark(i, 0)
   comment: all remaining solutions that set C to “true” are marked
   destroy-unmarked
test-if-empty
```

هر نوع الصاق درست و نادرست به متغیرها توسط یک رشته‌ی DNA نشان داده می‌شود و تشخیص یک وضعیت هم با استفاده از شیوه کد کردن به خصوص میسر می‌شود. همانطور که ملاحظه می‌شود در انتهای هر الگوریتم از "test-if-empty" برای بررسی وجود جواب استفاده شده است.

مدل sticker-based

در این مدل [21] از دو گروه پایه از مولکول‌های DNA استاندارد تک رشته‌ای برای نمایش یک رشته‌ی بیتی استفاده می‌شود. مثلاً یک رشته‌ی حافظه با طول N باز را در نظر بگیرید. این رشته به K ناحیه بدون همپوشانی هر کدام با طول M باز ($N \geq MK$) تقسیم می‌شود. هر ناحیه با دقیقاً یک مکان بیتی (یا به طور معادل یک متغیر بولی) در طول محاسبات مشخص می‌شود. همچنین K رشته‌ی sticker و یا به طور خلاصه stickers هم طراحی می‌کنیم. هر sticker از M باز تشکیل شده و مکمل یک و فقط یکی از K ناحیه حافظه است. اگر یک sticker با ناحیه جفت‌شونده‌اش روی یک رشته حافظه، جفت شده بود، در آن صورت آن بیت در آن رشته on است و اگر هیچ sticker ای با یک ناحیه جفت نشده بود، بیت متناظر آن ناحیه off است (شکل ۹).



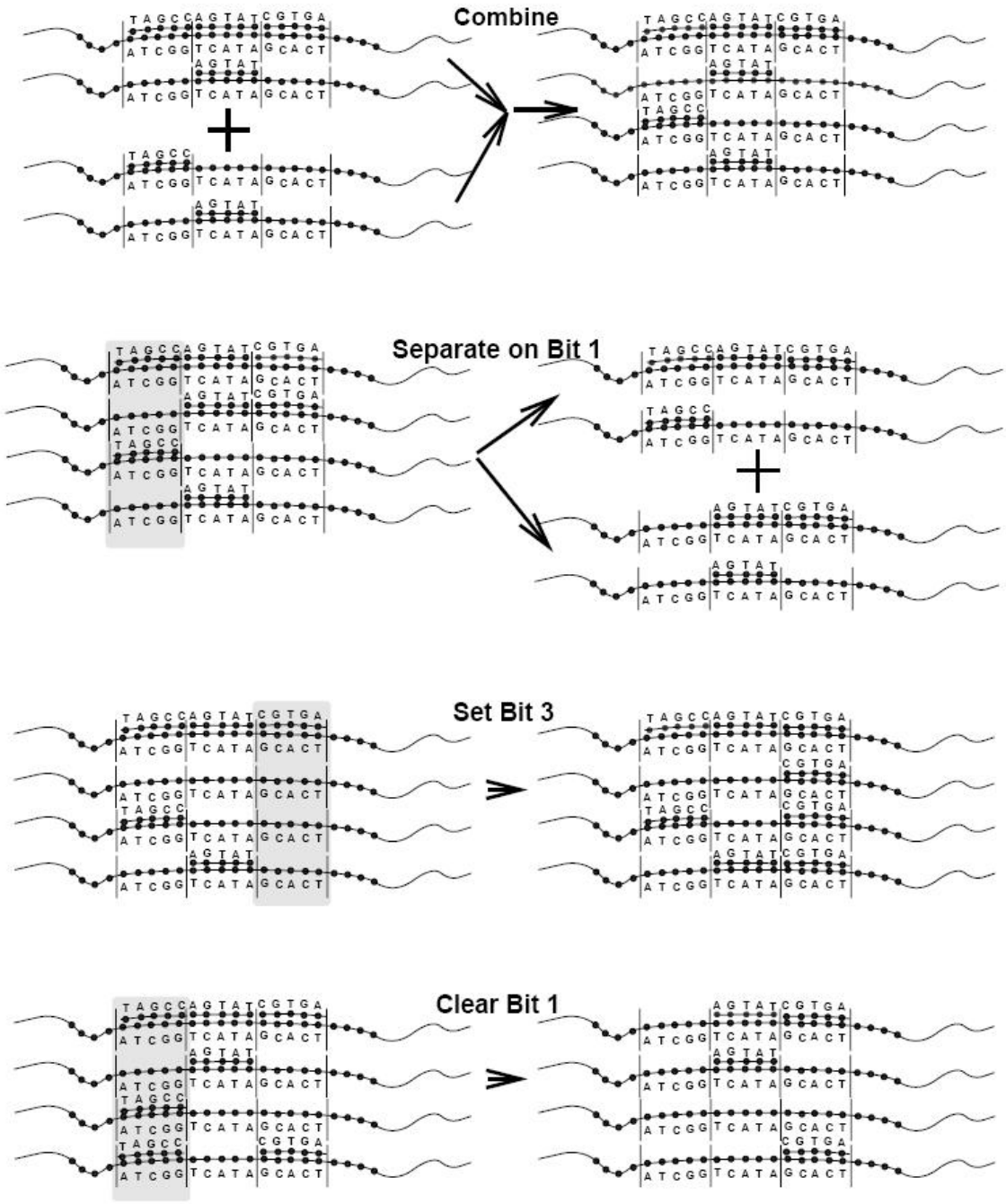
شکل ۹ رشته حافظه و sticker های متناظر آن

همانطور که ملاحظه شد این مدل در نمایش اطلاعات تا حدی با مدل قبل که برای نمایش اطلاعات از ساختار خاصی پیروی نمی‌کرد و برای نمایش از کد کردن اطلاعات با استفاده از رشته‌های DNA مانند روش اَدلمن بهره می‌جست متفاوت است.

عملگرهای پایه این مدل عبارتند از:

- **combine**
- **separate**
- **set**
- **clear**

نتیجه حاصل از اعمال هر یک از این عملگرها بر روی یک رشته DNA در شکل ۱۰ به صورت شماتیک نشان داده شده است.



شکل ۱۰ عملگرهای پایه در مدل sticker-based و نتیجه حاصل از اعمال هر یک

نتیجه حاصل از اعمال set و clear همانطور که واضح است جفت کردن رشته با sticker متناظر و جدا کردن sticker از رشته است.

نتیجه حاصل از combine و separate هم تلفیق دو مجموعه و یا جدا کرد یک مجموعه به دو مجموعه می‌باشد. این اعمال در واقع به نوعی معادل destroy-marked و destroy-unmarked در مدل surface-based می‌باشند.

در مورد نکات جالب توجه درباره این مدل باید گفت که این مدل هم از رشته‌های DNA به عنوان مواد فیزیکی در محاسبه استفاده می‌کند اما با توجه به خصوصیاتش امکان دسترسی تصادفی حافظه را فراهم می‌کند. در ضمن در این مدل نیازی به توسعه رشته‌های DNA مثلاً همانند توسعه و تغییر در آنها در انتقال حفره یا استفاده از ساختارهای سنجاق سری در تولید گیت‌های منطقی و همچنین استفاده از آنزیم‌ها همانند روش اَدلمن نیست. اما جالب توجه‌ترین نکته این است که مواد در این مدل حداقل در تئوری قابل استفاده مجدد هستند.

به نظر می‌رسد این مدل از مدل surface-based قوی‌تر باشد. در ضمن شایان ذکر است که بر اساس این مدل کامپایلری مشابه زبان C برای کامپیوترهای DNA ای مبتنی بر این مدل ارائه شده است [21].

برای نشان دادن قدرت عملگرهایی که مشخص شد الگوریتمی برای حل مسئله‌ی NP-complete ، Minimal Set Cover با استفاده از مدل sticker-based ارائه می‌کنیم. (این مسئله را می‌توان به صورت مجموعه‌ی B کوله‌پشتی که هر یک حاوی تعدادی شی است در نظر گرفت. اشیاء دارای A نوع هستند. هدف پیدا کردن کوچکترین زیرمجموعه از کوله‌پشتی‌ها است که در آنها حداقل یک شی از هر نوع موجود باشد. تعریف رسمی مسئله به این صورت است که با داشتن مجموعه $C = \{C1, ..., CB\}$ از زیرمجموعه‌های $\{1, ..., A\}$ کوچکترین زیرمجموعه‌ی I از $\{1, ..., B\}$ به طوری که $\cup_{i\in I} C_i = \{1, ..., A\}$ باشد، کدام است؟) حل مسئله در این مدل کاملاً سرراست است. ترکیبات حافظه که نشانگر کلیه 2^B انتخاب ممکن از کوله‌پشتی‌ها است را می‌سازیم. تمام رشته‌هایی که شامل کوله‌پشتی i که حاوی تمام انواع موجود در زیرمجموعه C_i است را علامت می‌گذاریم. سپس ترکیباتی را که حاوی تمام انواع A علامتگذاری شده‌اند را جدا می‌کنیم و رشته‌ها(یی) که از کمترین کوله‌پشتی استفاده کرده‌اند را می‌خوانیم. این الگوریتم برای حل مسئله‌ی minimal set cover بر اساس مدل sticker-based در زیر آمده است.

```
Design a memory strand with K = B + A bit regions.
Initialize a (K, B) library set in a tube called T0.
comment: bits 1…B represent which bags are chosen, bits B+1…B+A which
object types are present.

for i=1 to B
   separate T0  into Ton and Toff based on bit i
   for j=1 to |Ci|
      set bit N + Ci[j] in Ton
   combine Ton and Toff into T0
comment: mark the final A positions of each complex to record which
object types it contains.

for i=B+1 to B+A
   separate T0 into T0 and Tbad based on bit i
   discard Tbad
comment: get rid of ones which do not have all A types.

for i=0 to B-1
```

```
   for j=I down to 0
      separate Tj into T(j+1)' and Tj based on bit i+1
      combine Tj+1 and T(j+1)' into Tj+1
comment: count how many bags were used. At the end of the outer loop,
tube Ti contains all complexes which used exactly i bags.

read T1;
else if it was empty then   read T2;
else if it was empty then   read T3;
. . .
```

$|C_i|$ تعداد اشیاء در زیرمجموعه C_i و $C_i[j]$، j امین شیء در زیرمجموعه C_i است. به این نکته توجه کنید که این الگوریتم O(AB) مرحله طول می‌کشد و ورودی آن O(AB) بیت است.

کارهای انجام شده

کارهای زیادی در زمینه محاسبات با DNA انجام شده است. برای انجام محاسبات از مدل‌های متفاوت استفاده شده است و هر روش تا حدی با روش دیگر متفاوت است و تعداد زیادی از آنها مطابق روش اَدلمن انجام شده‌اند اما در همه آنها استفاده مشخص از رشته‌های DNA و امکان پردازش موازی در حد وسیع، به چشم می‌خورد.

کارهایی مانند حل مسئله‌ی 3-SAT با ۲۰ متغیر [22] و یا شکستن الگوریتم DES (Data Encryption Standard) [23] توسط محاسبات با DNA انجام شده است. به عنوان مثال و برای اشاره به نمونه‌ای از استفاده وسیع از امکان پردازش موازی، حل مسئله‌ی 3-SAT با ۲۰ متغیر در یک کامپیوتر مبتنی بر سیلیکون به $O(k \times 2n)$ زمان احتیاج دارد که n تعداد متغیرها و K تعداد جملات است ولی حل این مسئله با استفاده از محاسبات DNA ای تنها به $O(n + k)$ زمان احتیاج دارد [22].

فهرست منابع

1. Moore, Gordon E. "Cramming More Components onto Integrated Circuits", 1965.
2. Kanellos, Michael. "Moore's Law Will Continue to Rule", July 10th, 2002.
3. Lopez, Alexander. "Computers are becoming faster and cheaper, but their speed is still limited by the physical restrictions of an electron moving through matter. What technologies are emerging to break through this speed barrier?", Scientific American, 1999.
4. Amos, Martyn. Theoretical and Experimental DNA Computation, Springer,
5. Junnakar, Sandeep. "Tomorrow's Tech: The Domino Effect", The New York Times, October 24th, 2002.
6. Bonsor, Kevin. "How DNA Computers Will Work".
7. Forbes, Nancy. "Life after Silicon: Ultrascale Computing", The Industrial Physicist, December 1997, pp. 20-23.
8. Fulk, Kevin. "Future Technology Briefing - Biological Computing", ISRC, 2002.
9. Levin, David. "DNA Computing", IEEE Computing in Science & Engineering, May-June, 2002, pp. 5-8.
10. "DNA Chips", Technology Review, January-February, 2001, pp. 118-119.
11. Xeotron Corporation's Homepage.
12. Innovations at Georgia Tech News release.
13. Feynman, Richard P. "There's Plenty of Room at the Bottom", Gilbert, In D, editor, Miniaturization, pp. 282-296, Reinhold, 1961.
14. Adleman, Leonard M. "Molecular Computation of Solutions to Combinatorial Problems", Science, 266:1021-1024, 1994.
15. "DNA Basis for New Generation of Computers", CNN.com, August 18th, 2003.
16. Adleman, Leonard M. "Computing with DNA", Scientific American, August, 1998.
17. Okamoto, A. and Tanaka, K. and Saito, I. "DNA Logic Gates", Kyoto University, Japan, April, 2004.

18. Liu, W. and Shi, X. and Zhang, S. and Liu, X. and Xu, J. "A New DNA Computing Model for the NAND Gate Based on Induced Hairpin Formation", Wenzhou Normal College, April, 2004.

19. Liu, Q. and Guo, Z. and Fei, Z. and Codon, A. E. and Corn, R. M. and Lagally, M. G. and Smith, L. "A Surface-Based Approach to DNA Computation", University of Wisconsin, August, 1997.

20. Roweis, S. and Winfree, E. and Burgoyne, R. and Chelypov, N. V. and Goodman, M. F. and Rothemund, P. W. K. and Adleman L. M. "A Sticker Based Model for DNA Computation", University of Southern California, May, 1996.

21. Carroll, Steven. "A Complete Programming Environment for DNA Computation", University of Illinois at Urbana-Champaign, 2002.

22. Braich, R. S. and Chelyapov, N. and Johnson, C. and Rothemund, P. W. K. and Adleman, L. M. "Solution of a 20-Variable 3-SAT Problem on a DNA Computer", University of Southern California, April, 2002.

23. Boneh, D. and Dunworth, C. and Lipton, R. J. "Breaking DES Using a Molecular Computer", Princeton University.