

## Suitable drug structure prediction with Opposition-Based Differential Evolution

M. Koohimoghadam<sup>1</sup>, A. Torkaman-Rahmani<sup>2\*</sup>

<sup>1</sup>M.Sc Student, Department of Computer Engineering, Iran University of Science and Technology

m\_koohi\_m@comp.iust.ac.ir

<sup>2</sup>Assistant Professor, Department of Computer Engineering, Iran University of Science and Technology

rahmani@iust.ac.ir

---

### Abstract

Discovery of new drugs and study of their side effects has been an important research field in recent years. Because of direct effect of the pharmaceutical products on human health usually the drug design projects are challenging and technically demanding.

The incorporation of computer simulations into drug design projects is one of the best ways to optimize drugs' potency. In this approach, researchers try to find the best interaction between protein structure and drug in a virtual environment; this procedure is called "molecular docking". The molecular docking problem can be considered as a search problem. The search space in this problem is defined with all possible protein-ligand interactions and the best interaction is the solution of problem.

In this paper, a new approach for finding the best interaction is proposed. The proposed method is based on opposition based differential evolution algorithm. Also the proposed method is enhanced by a local search algorithm and a pseudo-elitism operator. Like other metaheuristic algorithms, our method uses a population of possible solution and AutoDock scoring function is used to evaluate each vector in the population. Six different protein-ligand complexes are used to verify the efficiency of the proposed algorithm. The experimental results show that the proposed algorithm is more robust and reliable than other algorithms such as simulated annealing and Lamarckian genetic algorithm.

**Keywords:** Drug, receptor, molecular docking, differential evolution algorithm, scori

---

\* Corresponding author

Address: Adel Torkaman-Rahmani, Department of Computer Engineering, Iran University of Science and Technology, Tehran, Iran

Tel: +98 21 73225324

Fax: +98 21 73225324

E-mail: rahmani@aust.ac.ir

## پیش‌بینی ساختار مناسب دارو با استفاده از الگوریتم تکامل تفاضلی مبتنی بر نقطه‌ی مخالف

محمد کوهی مقدم<sup>۱</sup>، عادل ترکمان رحمانی<sup>۲\*</sup>

<sup>۱</sup> دانشجوی کارشناسی ارشد مهندسی کامپیوتر گرایش هوش مصنوعی و رباتیک-دانشگاه علم و صنعت ایران

m\_koohi\_m@comp.iust.ac.ir

<sup>۲</sup> استادیار دانشکده مهندسی کامپیوتر-دانشگاه علم و صنعت ایران

rahmani@iust.ac.ir

### چکیده

کشف داروهای جدید و بررسی اثرات جانبی آن‌ها یکی از زمینه‌های مهم پژوهشی است که دانشمندان داروساز در آن به فعالیت مشغولند. به دلیل اثر مستقیم محصولات دارویی بر سلامت انسان‌ها، تحقیقات داروسازی از حساسیت بالایی برخوردار بوده و رسیدن به جوابی مطلوب در این تحقیقات اغلب زمان زیادی احتیاج خواهد داشت. پیش‌بینی ساختار دارو به کمک نرم‌افزارهای شبیه‌سازی، راهکاری است که در سال‌های اخیر مورد توجه محققین داروسازی بوده است. در این مسئله دانشمندان به دنبال یافتن بهترین برهم‌کنش بین ساختار دارو و گیرنده می‌باشند. این مسئله در منابع علمی با نام پهلوگیری مولکولی شناخته می‌شود و می‌توان آن را به عنوان یک مسئله جستجو در نظر گرفت که فضای جستجو در آن حالت‌های مختلف برهم‌کنش دارو و گیرنده می‌باشد. هدف نهایی از حل این مسئله انتخاب بهترین برهم‌کنش از میان این فضای جستجو است.

در این مقاله از الگوریتم تکامل تفاضلی مبتنی بر نقطه مقابل برای یافتن بهترین حالت برهم‌کنش دارو و گیرنده استفاده شده است. برای بهبود نتایج، الگوریتم مذکور با یک روش جستجوی محلی و یک عملگر نخبه‌گرا تلفیق شده است. الگوریتم ارائه شده، مانند دیگر الگوریتم‌های فرااکتشافی یک الگوریتم تکرار شونده می‌باشد که به کمک جمعیتی از بردارهای جواب سعی در یافتن بهترین برهم‌کنش دارد. همچنین تابع ارزیاب استفاده شده در این پژوهش تابع ارزیاب AutoDock می‌باشد.

برای ارزیابی الگوریتم پیشنهادی شش ساختار متفاوت گیرنده-دارو استفاده شده است. نتایج حاصل از پیش‌بینی ساختار دارو برای هر یک از این شش گیرنده با نتایج الگوریتم ژنتیک لامارکی و الگوریتم سردسازی شبیه‌سازی شده و الگوریتم تکامل تفاضلی معمولی مقایسه شده است. بر اساس نتایج الگوریتم ارائه شده نسبت به الگوریتم‌های دیگر از عملکرد بهتری برخوردار است.

کلیدواژگان: دارو، گیرنده، پهلوگیری مولکولی، الگوریتم تکامل تفاضلی، تابع امتیازدهی.

عهده‌دار مکاتبات

نشانی: تهران، نارمک، دانشگاه علم و صنعت ایران، دانشکده مهندسی کامپیوتر، کدپستی: ۱۳۱۱۴-۱۶۸۴۶

تلفن: ۰۲۱۷۳۲۲۵۳۲۴، دورنگار: ۰۲۱۷۳۲۲۵۳۲۴، پیام نگار: rahmani@iust.ac.ir

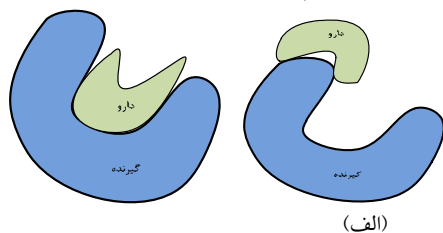
## ۱- مقدمه

امروزه یکی از مهمترین زمینه‌های تحقیقاتی، مطالعه پیرامون ساخت داروهای جدید می‌باشد. نتایج حاصل از تحقیقاتی که در این زمینه انجام می‌شود تاثیر مستقیمی بر میزان سلامت جامعه خواهد داشت، به طوریکه دانشمندان داروساز همواره در تلاش‌اند تا ترکیبات دارویی با کمترین اثرات جانبی بر روی بیماران را شناسایی کنند. نقش مهمی که دارو بر سلامت بشر دارد سبب شده‌است تا صنعت داروسازی یکی کلیدی‌ترین صنایع در دنیای امروز به شمار آید و شرکت‌های تولید کننده دارو همواره در پی یافتن روش‌هایی باشند تا محصولات خود را با کیفیت بهتر و در زمان کمتر به بازار عرضه کنند.

از سویی دیگر با پیشرفت علوم کامپیوتر روز به روز بر وسعت کاربران این علم در زمینه‌های مختلف تحقیقاتی افزوده شده‌است. این پیشرفت آنچنان سریع و تأثیرگذار بوده که دیگر تصور دنیایی بدون کامپیوتر دور از ذهن به نظر می‌آید. در این بین، دانشمندان علوم کامپیوتر نقش اصلی را در این پیشرفت چشمگیر ایفا کرده و همواره در تلاش‌اند تا برای حل مسائل موجود در دنیای واقعی، الگوریتم‌هایی با کارایی بالا ارائه دهند. بیوانفورماتیک<sup>۱</sup> از جمله زمینه‌های پژوهشی است که در سال‌های اخیر بسیار مورد توجه دانشمندان بوده است. این علم نتیجه استفاده از الگوریتم‌های کامپیوتری در محدوده‌ی مسائل زیستی می‌باشد. همپا با گسترش بیوانفورماتیک شاخه‌ی جدیدی از علم در حوزه‌ی مسائل شیمی به وجود آمد که با نام شیمی‌انفورماتیک<sup>۲</sup> شناخته می‌شود. این دو شاخه از علم در بسیاری از زمینه‌ها با یکدیگر در تعامل هستند [۱، ۲].

پیش‌بینی ساختار مناسب دارو به کمک شبیه‌سازی‌های کامپیوتری یکی از مسائلی است که در سال‌های اخیر بسیار مورد توجه محققین بوده‌است. این مسئله محققین در هر دو شاخه‌ی علوم بیوانفورماتیک و شیمی‌انفورماتیک را درگیر خود می‌کند. الگوریتم‌های کامپیوتری که در این حوزه مورد استفاده قرار می‌گیرند اغلب با فراهم آوردن یک محیط

مجازی سعی در شبیه‌سازی آزمایش‌های تجربی دارند. دارو ساختار شیمیایی است که اغلب بر روی گیرنده<sup>۳</sup> بیولوژیکی خاص می‌نشیند. اغلب گیرنده‌های دارویی از نوع پروتئین‌ها هستند. اتصال دارو به پروتئین خاص سبب ایجاد تغییراتی دلخواه در سازوکار آن پروتئین و در نتیجه تغییرات بیولوژیکی در بدن موجود زنده می‌شود. میزان کارایی دارو و اثری که از خود نشان می‌دهد تا حد زیادی به نحوه‌ی برهم‌کنش گیرنده و دارو مربوط می‌شود. از این رو دانشمندان داروساز، همواره در تلاش‌اند تا ساختار دارو را به گونه‌ای طراحی کنند که در محلی مناسب بر روی گیرنده پهلوی بگیرد [۳]. در منابع علمی به فرآیند اتصال یک مولکول کوچک<sup>۴</sup> به گیرنده، پهلوگیری مولکولی<sup>۵</sup> اطلاق می‌شود. همچنین اصطلاح لیگاند<sup>۶</sup> به انواع مختلف مولکول‌های کوچکی اطلاق می‌شود که به گیرنده متصل می‌شوند<sup>۷</sup> [۴]. برای نمونه در شکل (۱) دو برهم‌کنش متفاوت برای مولکول دارو و گیرنده نمایش داده شده‌است. در این شکل، برهم‌کنش (ب) در حالت بهتری از برهم‌کنش (الف) قرار دارد از اینرو ترکیب (ب) از پایداری بیشتری برخوردار خواهد بود. در گذشته حل مسئله‌ی پهلوگیری مولکولی و یافتن بهترین برهم‌کنش گیرنده-دارو به صورت آزمایش و خطا صورت می‌گرفت. در این فرآیند زمان‌بر، دانشمندان در موقعیت‌های متفاوتی از گیرنده، ساختارهای متفاوتی از دارو را می‌آزمودند تا در نهایت به بهترین برهم‌کنش دست یابند.



(ب)

شکل ۱- دو برهم‌کنش متفاوت مولکول فرضی دارو و گیرنده

استفاده از روش‌های آزمون و خطا در طراحی دارو سبب می‌شد تا انجام برخی از پروژه‌های طراحی دارو ۱۲ تا ۱۶

<sup>۱</sup> Bioinformatics    <sup>۲</sup> Chemoinformatics    <sup>۳</sup> Receptor    <sup>۴</sup> Small molecule    <sup>۵</sup> Molecular docking    <sup>۶</sup> Ligand

<sup>۷</sup> در حقیقت یک مولکول دارو را می‌توان به عنوان یک مولکول کوچک در نظر گرفت، از اینرو در بعضی بخش‌های این نوشتار از واژه لیگاند به جای واژه دارو استفاده نمودیم.

الگوریتم ارائه شده در این پژوهش نسخه‌ی تغییر یافته‌ی الگوریتم تکامل تفاضلی مبتنی بر نقطه مقابل<sup>۲۰</sup> می‌باشد. الگوریتم تکامل تفاضلی یک الگوریتم تکرارشونده مبتنی بر جمعیت است که امروزه استفاده از آن برای حل مسائل دنیای واقعی<sup>۲۱</sup> بسیار رایج است. سادگی و قابل اعتماد بودن از مزیت‌های مهم این الگوریتم می‌باشند [۲۱]. همچنین مفهوم نقطه‌ی مقابل<sup>۲۲</sup> نیز یکی از مفاهیمی است که در سال‌های اخیر کاربرد وسیعی در الگوریتم‌های یادگیری ماشین و بهینه‌سازی داشته‌است [۲۲].

مانند دیگر الگوریتم‌های فرااکتشافی، ارزیابی جواب‌های تصادفی که در هر تکرار الگوریتم تولید می‌شوند از چالش‌های مهم الگوریتم پیشنهادی می‌باشد. هرچه تابع ارزیاب<sup>۲۳</sup> استفاده شده در الگوریتم از دقت بالاتری برخوردار باشد، جواب نهایی نیز دقیق‌تر خواهد بود. در این مقاله از تابع ارزیاب ارائه شده در بسته‌ی نرم‌افزاری AutoDock برای ارزیابی بردارهای جواب استفاده شده‌است [۲۰].

در نهایت برای ارزیابی کارایی الگوریتم پیشنهادی، نتایج پیش‌بینی ساختار دارو به کمک این الگوریتم با الگوریتم‌های سردسازی شبیه‌سازی شده<sup>۲۴</sup>، الگوریتم ژنتیک لامارکی<sup>۲۵</sup> و الگوریتم تکامل تفاضلی معمولی مقایسه شده‌است. در ادامه‌ی این نوشتار، در بخش بعدی جزئیات مسئله‌ی پهلوگیری مولکولی تشریح خواهد شد. در بخش سوم الگوریتم پیشنهادی و نحوه‌ی پیاده‌سازی آن ارائه شده‌است، در بخش چهارم نحوه‌ی نگاشت مسئله پهلوگیری مولکولی برای حل به کمک الگوریتم پیشنهادی مورد بررسی قرار گرفته‌است. در نهایت در بخش پنجم نتایج حاصل از اجرای الگوریتم ارائه و با دیگر الگوریتم‌های موجود مقایسه شده‌است.

## ۲- تعریف مسئله

دارو ماده‌ای است با منشأ طبیعی یا شیمیایی که برای

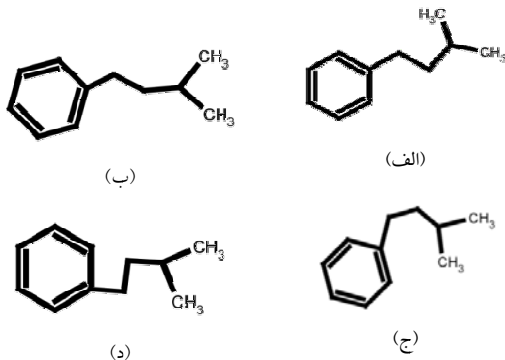
سال به طول بیانجامد [۵]. کشف بهترین برهم‌کنش دارو-گیرنده، به عنوان یک گلوگاه<sup>۸</sup> در پروژه‌های طراحی دارو به حساب می‌آید و کاستن از زمان انجام آن تأثیر بسزایی در کاهش زمان تولید نهایی یک دارو خواهد داشت [۳]. از اینرو نرم‌افزارهای شبیه‌سازی بسیاری برای سرعت بخشی به این مرحله از تولید دارو عرضه شده‌اند. DOCK [۶]، AutoDock [۷، ۸]، GOLD [۹، ۱۰] و FLEXX [۱۱] از جمله نرم‌افزارهای شبیه‌سازی هستند که برای این منظور ارائه شده‌اند.

از مسئله‌ی پهلوگیری مولکولی می‌توان به عنوان یک مسئله جستجو<sup>۹</sup> یاد کرد، که حالت‌های مختلف برهم‌کنش پروتئین و دارو در آن به عنوان فضای جستجوی مسئله می‌باشند و جواب نهایی پایدارترین برهم‌کنش در این فضای جستجو است [۵]. تاکنون الگوریتم‌های متفاوتی برای حل مسئله پهلوگیری مولکولی در منابع علمی ارائه شده است. تطبیق شکل سریع<sup>۱۰</sup>، شبیه‌سازی مونت‌کارلو<sup>۱۱</sup>، ساخت‌افزایشی<sup>۱۲</sup> و الگوریتم فرااکتشافی<sup>۱۳</sup> از جمله‌ی این روش‌ها می‌باشند [۱۱-۱۴].

یافتن بهترین حالت برای کنارهم قرار گرفتن ساختار دارو و گیرنده یک مسئله NP-hard<sup>۱۴</sup> می‌باشد [۱۵]. از اینرو تاکنون الگوریتم‌های فرااکتشافی زیادی برای حل این مسئله مورد استفاده قرار گرفته‌است. الگوریتم سردسازی شبیه‌سازی شده<sup>۱۵</sup> از اولین الگوریتم‌های فرااکتشافی استفاده شده برای حل این مسئله می‌باشد که اولین بار توسط توسعه‌دهندگان نرم‌افزار AutoDock ارائه شد [۸، ۱۶]. همچنین الگوریتم ژنتیک، از دیگر الگوریتم‌های استفاده شده برای حل این مسئله است. توسعه دهندگان AutoDock برای بالا بردن دقت این الگوریتم در یافتن بهترین برهم‌کنش از یک الگوریتم جستجوی محلی در کنار الگوریتم ژنتیک استفاده کردند [۷]. الگوریتم‌های بهینه‌سازی تجمع ذرات<sup>۱۶</sup>، بهینه‌سازی چندهدفه<sup>۱۷</sup>، بهینه‌سازی کلونی مورچه<sup>۱۸</sup> و بهینه‌سازی تکامل تفاضلی<sup>۱۹</sup> از دیگر الگوریتم‌های فرااکتشافی ارائه شده برای حل این مسئله می‌باشند [۱۷-۲۰].

<sup>8</sup>Bottleneck <sup>9</sup> Search Problem <sup>10</sup> Fast shape matching <sup>11</sup> Monte Carlo simulation <sup>12</sup> Incremental Construction  
<sup>13</sup> Metaheuristic <sup>14</sup> Non-Polynomial <sup>15</sup> Simulated Annealing Algorithm <sup>16</sup> Particle Swarm Optimizatio <sup>17</sup> Multi-Objective Optimization  
<sup>18</sup> Ant Colony Optimization <sup>19</sup> Differential Evolution Optimization <sup>20</sup> Opposition Based Differential Evolution <sup>21</sup> Real-world problem  
<sup>22</sup> Opposite point <sup>23</sup> Function evaluation <sup>24</sup> Simulated Annealing Algorithm <sup>25</sup> Lamarckian Genetic Algorithm

مولکول دارو به محل گیرنده به علت وجود نیروهای الکترواستاتیکی بین اتم‌های دارو و اتم‌های پروتئین شکل پروتئین تغییر می‌کند [۲۵].



شکل ۳- چهار ساختار مختلف برای یک مولکول فرضی دارو (تصاویر توسط نرم‌افزار MarvinSketch تهیه شده‌اند)

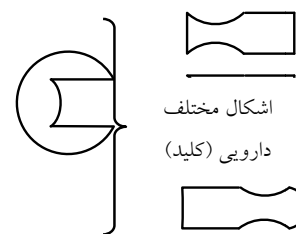
همچنین در صورتی که هیچ اطلاعاتی در مورد ساختار کلی ساختمان دارو وجود نداشته باشد درجه‌ی سختی این مسئله بیشتر خواهد شد. این مسئله به طراحی De novo در منابع علمی شهرت دارد [۲۶]. کلمه‌ی De novo یک کلمه‌ی لاتین است که به معنی "از ابتدا" می‌باشد. در این حالت برای حل مسئله لازم است تا علاوه بر یافتن بهترین مکان دارو بر روی گیرنده و تعیین زاویه‌ی پیوندهای آزاد دارو، تکه‌های سازنده‌ی یک ساختار دارو نیز پیش‌بینی شود که در این حالت فضای جستجوی مسئله چند برابر خواهد کرد. در [۲۷] پیچیدگی حل مسئله‌ی پهلوگیری مولکولی در حالت‌های مختلف مورد بحث قرار گرفته است.

در این مقاله فرض شده است که ساختار کلی مولکول دارو و محل اتصال<sup>۲۸</sup> بر روی پروتئین به صورت تقریبی مشخص است. در حقیقت الگوریتم پیشنهادی محل دقیق دارو را پیش‌بینی کرده و زاویه‌ی پیوندهای آزاد آن را مشخص می‌کند. مقادیر مختلفی که برای زاویه‌ی پیوندها در ساختار دارو و مکان قرارگیری آن می‌توان در نظر گرفت، فضای جستجوی مسئله را می‌سازد. برای نمونه در شکل (۴) چهار حالت مختلف برهم‌کنش یک مولکول فرضی دارو و

درمان بیماری خاص وارد بدن بیمار می‌شود. پس از ورود دارو به بدن، این ماده به یکی از مولکول‌های زیستی موجود در بدن متصل<sup>۲۶</sup> می‌شود و سبب ایجاد تغییراتی در عملکرد گیرنده<sup>۲۷</sup> می‌گردد. گیرنده ساختار زیستی است که دارو به آن می‌چسبد و با آن برهم‌کنش نشان می‌دهد. این ساختار در داروهای مختلف می‌تواند متفاوت باشد، اما اغلب گیرنده‌های دارویی از نوع پروتئین‌ها هستند [۲۳].

مسئله‌ی پهلوگیری مولکولی نمونه‌ای از مسئله‌ی قفل-کلید می‌باشد [۲۴]. همان‌طور که تنها یک کلید قادر به باز کردن یک قفل خاص می‌باشد در اینجا نیز جواب مسئله حالتی است که داروی (کلید) مورد نظر به بهترین شکل با پروتئین (قفل) اتصال پیدا می‌کند (شکل ۲).

پروتئین (قفل)



شکل ۲- مدل قفل - کلید برای بیان مسئله پهلوگیری مولکولی

فضای جستجوی مسئله شامل تمامی حالات برهم‌کنش دارو و گیرنده می‌باشد. علاوه بر موقعیت‌های مختلفی از گیرنده که دارو به صورت بالقوه قادر به اتصال به آن می‌باشد، ساختار دارو نیز دارای تعدادی پیوند آزاد است و به تنهایی قادر است اشکال متعددی به خود بگیرد. از اینرو محققان در حل این مسئله با یک فضای جستجوی بسیار بزرگ سر و کار خواهند داشت.

تعداد ساختارهای متفاوتی که یک مولکول دارو به خود می‌گیرد به تعداد پیوندهای آزاد آن وابسته است. برای نمونه در شکل (۳) چهار ساختار متفاوت برای یک مولکول فرضی دارو نمایش داده شده است. با در نظر گرفتن این نکته، شرایط متفاوتی برای تعریف یک مسئله‌ی پهلوگیری مولکولی به وجود می‌آید. در سخت‌ترین حالت برای تعریف این مسئله، ساختار پروتئین قابل تغییر است و با نزدیک شدن

<sup>26</sup> Bind

<sup>27</sup> Receptor

<sup>28</sup> Binding site

گیرنده نمایش داده شده است.  $a_j$ ،  $b_j$  به ترتیب مشخص کننده حد پایین و بالا برای متغیر  $j$  هستند. همچنین  $rand(0,1)$  عددی تصادفی در بازه  $[0,1]$  می باشد. بدین ترتیب پس از انجام مقدار دهی اولیه، جمعیتی از بردارهای اولیه خواهیم داشت که هر یک از بردارها نماینده یک جواب برای مسئله می باشد.

در الگوریتم تکامل تفاضلی جمعیتی که در تکرار بعدی تولید می شود، حاصل اجرای دو عملگر همبری<sup>۲۹</sup> و جهش<sup>۳۰</sup> بر روی جمعیت فعلی است. تاکنون روش های متعددی برای عملگرهای همبری و جهش در مقالات ارائه شده است [۳۰]. علاوه بر این دو عملگر یک عملگر انتخاب نیز وجود دارد که وظیفه انتخاب بردار مناسب برای تولید جمعیت بعدی را بر عهده دارد، در ادامه به تشریح این عملگرها می پردازیم.

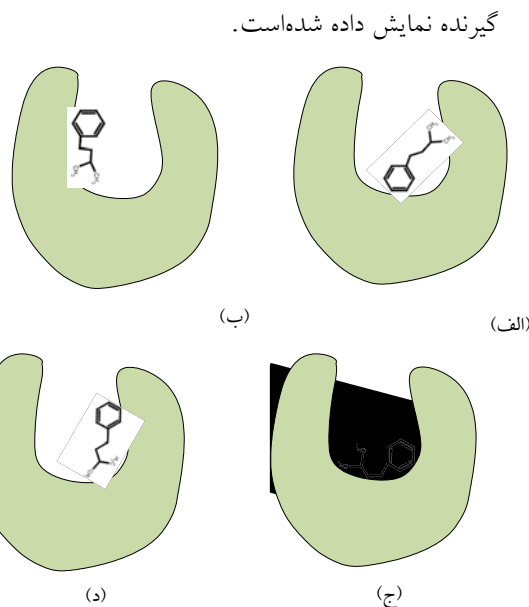
### ۱-۳- عملگر جهش

پس از مقداردهی اولیه بردارهای جمعیت، نوبت به اعمال عملگر جهش بر روی بردارهای جمعیت می رسد. اگر فرض کنیم  $X_{i,G} (i = 1, 2, 3, \dots, N_p)$  یک بردار تصادفی تولید شده در تکرار  $G$  الگوریتم باشد که  $N_p$  در آن اندازهی جمعیت است، بردار جهش که با  $V_{i,G}$  نمایش داده می شود به صورت زیر تعریف می شود.

$$V_{i,G} = X_{a,G} + F \times (X_{c,G} - X_{b,G}) \quad (2)$$

در این معادله  $X_{a,G}$ ،  $X_{b,G}$  و  $X_{c,G}$  سه بردار تصادفی مختلف در جمعیت هستند که در تکرار  $G$  جمعیت حضور دارند. برای اینکه بتوان در هر مرحله، سه بردار کاملاً متفاوت در جمعیت انتخاب کرد لازم است تا  $N_p \geq 4$  برقرار باشد. در حقیقت می توان علت نام گذاری این الگوریتم را به تکامل تفاضلی به این مرحله نسبت داد. زیرا بردار جهش، حاصل اضافه کردن تفاضل دو بردار تصادفی به بردار تصادفی سوم است. همان طور که اشاره شد، برای عملگر جهش تعاریف دیگری نیز وجود دارد که به علت استفادهی این مقاله از تعریف (۲) به ذکر این تعریف بسنده می شود.

همچنین  $F$  یک عدد ثابت است که ضریبی برای اضافه کردن  $(X_{c,G} - X_{b,G})$  به بردار تصادفی سوم می باشد.



شکل ۴-حالات های مختلف اتصال یک مولکول فرضی دارو به گیرنده با فرض آگاهی در مورد محل تقریبی اتصال دارو به پروتئین

### ۳- الگوریتم پیشنهادی

در این پژوهش برای حل مسئله ی پهلوی گیری مولکولی از الگوریتم تکامل تفاضلی مبتنی بر نقطه ی مقابل استفاده شده است. الگوریتم تکامل تفاضلی یک الگوریتم بهینه ساز مبتنی بر جمعیت است، که اولین بار در سال ۱۹۹۵ توسط پرایس و استورن ارائه شد [۲۸، ۲۹]. سادگی  $i = \{1, 2, 3, \dots, N_p\}$  اجرای این الگوریتم و کم بودن تعداد پارامترهایی که باید برای اجرای آن تنظیم کرد، سبب شد تا این الگوریتم به سرعت برای حل مسائل مختلف گسترش یابد.

مانند دیگر الگوریتم های تکاملی، اولین مرحله ی در اجرای الگوریتم تکامل تفاضلی تولید جمعیتی اولیه از بردارهای جواب می باشد. در ابتدای اجرای الگوریتم، به علت نبود اطلاعات کافی از فضای جستجوی مسئله، جمعیت اولیه به صورت کاملاً تصادفی تولید می شود. در معادله (۱) نحوه ی تولید بردارهای جمعیت اولیه آورده شده

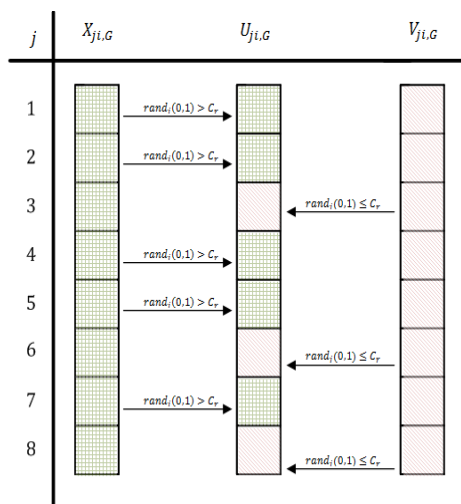
$$X_{i,j} = a_j + rand(0,1) \times (a_j - b_j) \quad (1)$$

$$j = 1, 2, 3, \dots,$$

در این معادله تعداد بُعد مسئله با  $D$  مشخص می شود و

<sup>29</sup> Crossover

<sup>30</sup> Mutation



شکل ۵- ساخت یک بردار همبری از روی بردار جهش و بردار اصلی جمعیت

### ۳-۳- عملگر انتخاب

تا این قسمت از الگوریتم به ازای هر بردار در جمعیت  $(X_{i,G})$  یک بردار همبری  $(U_{i,G})$  تولید شده است. در ادامه لازم است تا از بین دو بردار  $U_{i,G}$  و  $X_{i,G}$  یک بردار انتخاب شود و در جمعیت بعدی قرار گیرد. برای حل مسئله ی پهلویی مولکولی به دنبال کمینه کردن مقدار انرژی برهم کنش هستیم، از این رو برداری با انرژی کمتر برای ساخت جمعیت بعدی انتخاب می شود.

در هر بار تکرار الگوریتم سه مرحله ی جهش، همبری و انتخاب به ازای هر بردار عضو جمعیت اجرا می شود. این روند آنقدر تکرار می شود تا زمانی که یکی از شرط های خاتمه برقرار شود. رسیدن یکی از بردار های جواب به کمترین میزان انرژی تعریف شده و اجرای الگوریتم به تعداد مشخصی تکرار از شرط های خاتمه برای الگوریتم می باشد.

### ۳-۴- الگوریتم تکامل تفاضلی مبتنی بر نقطه ی

#### مقابل

این الگوریتم سازوکاری شبیه به الگوریتم تکامل تفاضلی دارد با این تفاوت که در آن از مفهوم نقطه مقابل برای رسیدن به جواب بهینه به کار رفته است. اولین بار تیزهوش در سال ۲۰۰۵ از مفهوم نقطه ی مقابل برای بالابردن کارایی [۲۲، ۳۱]. نقطه ی مقابل را به صورت زیر می توان تعریف

انتخاب مقادیر بزرگ برای  $F$  سبب می شود تا جمعیت تولید شده در مرحله ی بعدی تنوع بیشتری داشته باشد و مقادیر کوچک سبب می شود تا جمعیت سریع تر همگرا شود. این عدد برای حل مسائل مختلف متفاوت بوده و معمولاً مقدار آن بین  $[۲,۰]$  تغییر می کند.

### ۲-۳- عملگر همبری

پس از اجرای عملگر جهش، نوبت به اعمال عملگر همبری بر روی بردار های جمعیت می باشد. این عملگر برای ساخت بردار جدید و اضافه کردن آن به جمعیت استفاده می شود. برای الگوریتم تکامل تفاضلی کلاسیک عملگر همبری به صورت زیر تعریف می شود.

$$U_{i,G} = (U_{1i,G}, U_{2i,G}, \dots, U_{Di,G}) \quad (۳)$$

$$U_{ji,G} = \begin{cases} V_{ji,G} & \text{if } rand_j(0,1) \leq C_r \text{ or } j = k \\ X_{ji,G} & \text{Otherwise} \end{cases} \quad (۴)$$

همانند عملگر جهش در اینجا نیز با یک بردار همبری سروکار داریم. این بردار در تکرار  $G$  الگوریتم با  $U_{i,G}$  نمایش داده می شود که  $i = \{1, 2, 3, \dots, N_p\}$  است. همچنین  $C_r \in (0,1)$  به عنوان نرخ همبری شناخته می شود و  $rand_j(0,1)$  یک عدد تصادفی بین  $[۰,۱]$  است. برای مقادیری به هر متغیر از بردار همبری لازم است تا در هر بار شرط  $rand_j(0,1) \leq C_r$  بررسی شود، در صورتی که این شرط برقرار باشد عددی که در بردار همبری قرار خواهد گرفت از بردار جهش  $V_{i,G}$  گرفته می شود و در غیر این صورت مقدار متغیر  $X_{i,G}$  در بردار همبری قرار خواهد گرفت. همچنین  $k \in \{1, 2, 3, \dots, D\}$  یک عدد تصادفی است که برای اطمینان از انتخاب حداقل یک مقدار از بردار جهش تعریف شده است. مقدار معمول برای  $C_r$  اعداد بین ۰.۴ تا ۱ می باشد. روند اجرای تولید بردار همبری در شکل (۵) آمده است.

الگوریتم یادگیری تقویتی و الگوریتم ژنتیک استفاده کرد

همچنین همان‌طور که اشاره شد، می‌توان در زمان تولید جمعیت بعدی در طی اجرای الگوریتم نیز از این مفهوم استفاده کرد. در شکل (۶) شبه‌کد الگوریتم تکامل‌تفاضلی مبتنی بر نقطه مقابل آورده شده‌است. در این الگوریتم  $J_r$  فاکتور پرش می‌باشد و اگر  $J_r < rand(0,1)$  برقرار باشد بردارهای مقابلی که برای بردارهای موجود در جمعیت کنونی تولید می‌شوند با بردارهای اصلی برای تولید جمعیت بعدی رقابت می‌کنند. پارامترهای دیگر این الگوریتم همانند الگوریتم تکامل‌تفاضلی کلاسیک است.

با توجه به مراحل اجرای الگوریتم، می‌توان به این نکته پی برد که برای هر بردار حاضر در جمعیت دو بار تابع ارزیابی فراخوانی می‌شود. در نگاه اول فراخوانی مجدد تابع ارزیابی برای بردار مقابل سبب کند شدن الگوریتم و افزایش تعداد فراخوانی‌های تابع ارزیابی در رسیدن به جواب نهایی خواهد شد. رهنمایان در [۳۲] نشان داده‌است ارزیابی بردارهای مقابل در این الگوریتم نه تنها باعث افزایش تعداد کل فراخوانی‌های تابع ارزیابی نمی‌شود، بلکه به الگوریتم در جستجوی بهتر فضای جواب و یافتن سریع‌تر نقطه‌ی بهینه کمک نیز خواهد کرد.

در این مقاله برای یافتن بهترین برهم‌کنش به کمک الگوریتم تکامل‌تفاضلی مبتنی بر نقطه‌ی مقابل، تغییراتی در آن ایجاد شده‌است. این تغییرات شامل اضافه کردن یک الگوریتم جستجوی محلی و یک عملگر شبه‌نخبه‌گرا به الگوریتم می‌باشد. در ادامه به توضیح پیرامون این تغییرات می‌پردازیم.

کرد.

اگر فرض کنیم  $x$  یک عدد حقیقی در فاصله‌ی  $[a, b]$  باشد آنگاه  $x'$  را نقطه مقابل  $x$  می‌نامیم و داریم.

$$x' = a + b - x \quad (5)$$

همچنین برای ابعاد بالاتر نیز این تعریف قابل تعمیم است. اگر فرض کنیم  $P(x_1, x_2, \dots, x_D)$  که  $x_i \in [a_i, b_i], i = 1, 2, \dots, D$  یک نقطه در فضای  $D$  بعدی باشد آنگاه نقطه مقابل این نقطه که با  $P(x'_1, x'_2, \dots, x'_D)$  نشان داده می‌شود به صورت زیر تعریف می‌شود.

$$x'_i = a_i + b_i - x_i \quad (6)$$

از مفهوم نقطه‌ی مقابل می‌توان در دو مرحله از الگوریتم تکامل‌تفاضلی استفاده کرد، این دو مرحله عبارتند از:

- تولید جمعیت اولیه
- مرحله‌ی تولید جمعیت بعدی در هر تکرار الگوریتم

در مرحله‌ی تولید جمعیت اولیه، برای هر بردار در جمعیت اولیه یک بردار مقابل<sup>۳۱</sup> تولید می‌شود. بردار مقابل در حقیقت نقطه‌ی مقابل بردار فعلی در فضای  $D$  بعدی جستجو می‌باشد. بردار اصلی و بردار مقابل با یکدیگر از لحاظ برازندگی مقایسه می‌شوند و هرکدام که از لحاظ تابع ارزیابی بهتر بودند در جمعیت بعدی قرار می‌گیرند. در زیر جزئیات این روند آورده شده‌است.

**گام نخست:** تولید تصادفی جمعیت اولیه

$P_{i,j}$  where  $i$

$$j = \{1, 2, \dots, D\}$$

**گام دوم:** تولید بردارهای مقابل  $OP_{i,j}$  برای هر بردار  $P_{i,j}$  نمایش‌دهنده‌ی مقدار متغیر  $J^m$  از بردار  $J^m$  جمعیت می‌باشد و  $a_j$  و  $b_j$  به ترتیب حد پایین و بالا برای متغیر  $J^m$  هستند.

$$OP_{i,j} = a_j + b_j - P_{i,j}$$

**گام سوم:** انتخاب بهترین بردارها از بین مجموعه  $\{P_{i,j}\}$

$\{OP_{i,j}\}$

<sup>31</sup>Opposite vector



### ۵-۳- الگوریتم جستجوی محلی

الگوریتم جستجوی محلی استفاده شده در این مقاله، الگوریتم Solis-Wets شهرت دارد که نوعی الگوریتم تپه‌نوردی<sup>۳۲</sup> است که در اطراف یک بردار به جستجوی دقیق‌تر جواب می‌پردازد [۳۳]. شبه‌کد این الگوریتم در شکل (۷) آورده شده است.

الگوریتم جستجوی محلی، یک بردار از جواب را به عنوان ورودی دریافت می‌کند و متغیرهای موجود در این بردار را بر اساس مقداری که توسط یک تابع تصادفی گوسی تولید می‌شوند تغییر می‌دهد. در ابتدا مقادیر تولید شده به بردار اضافه می‌شوند و اگر بردار جدید دارای برآیندگی بهتری نسبت به بردار اصلی بود مجدداً مقادیر قبلی به بردار اضافه می‌شوند این کار به تعداد numSuccess تکرار می‌شود. اگر در همان ابتدا با اضافه کردن مقادیر به جواب بهتری دست نیافتیم مقادیر تصادفی از بردار جواب کسر خواهند شد تا نتیجه‌ی تغییرات این بار با کسر مقادیر سنجیده شود. در صورت بهتر بودن بردار جواب این فرآیند نیز به تعداد numSuccess بار تکرار می‌شود.

### ۶-۳- عملگر شبه‌نخبه‌گرا<sup>۳۳</sup>

استفاده از یک عملگر شبه‌نخبه‌گرا یکی دیگر از مزیت‌های الگوریتم پیشنهادی می‌باشد. در حقیقت وظیفه‌ی عملگر شبه‌نخبه‌گرا حفظ تعداد مشخصی از بهترین بردارهای جمعیت فعلی در جمعیت بعدی است. در روش پیشنهادی بهترین بردار جمعیت در هر تکرار انتخاب می‌شود و به کمک توزیع کوشی<sup>۳۴</sup> تغییر اندکی در آن ایجاد می‌شود و این بردار در جمعیت بعدی قرار می‌گیرد. نحوه‌ی تولید بردار جدید در زیر آمده است.

$$I_{best_i} = I_{best_i} + x_{Cauchy} \text{ where } (7)$$

$$i = \{1, 2, 3, \dots, D\}$$

Generate uniformly distributed random population  $P_0$   
**for**  $i=0$  to  $N_p$  **do**  
**for**  $j=0$  to  $D$  **do**  
 $OP_{0i,j} = a_j + b_j - P_{0i,j}$   
**end for**  
**end for**  
 Select  $N_p$  fittest individuals from set the  $\{P_0, OP_0\}$  as initial population  $P$   
**while** termination conditions not satisfied  
**for**  $i=0$  to  $N_p$  **do**  
 Select three random individual  $X_a, X_b, X_c$  where  
 $X_i \neq X_a \neq X_b \neq X_c$   
 $V_i \leftarrow X_{ia} + F \times (X_{ib} - X_{ic})$   
**for**  $j=0$  to  $D$  **do**  
**if**  $rand_j(0,1) \leq C_r$  **then**  
 $U_{i,j} \leftarrow V_{i,j}$   
**else**  
 $U_{i,j} \leftarrow P_{i,j}$   
**end if**  
**end for**  
 Evaluate  $U_i$   
**If** ( $fitness(U_i) \leq fitness(P_i)$ ) **then**  
 $P'_i \leftarrow U_i$   
**else**  
 $P'_i \leftarrow X_i$   
**end if**  
**end for**  
 $X \leftarrow X'$   
**If**  $rand(0,1) < J_r$  **then**  
**for**  $i=0$  to  $N_p$  **do**  
**for**  $j=0$  to  $D$  **do**  
 $OP_{i,j} \leftarrow MIN_j^p + MAX_j^p - P_{i,j}$   
**end for**  
**end for**  
**end if**  
 Select  $N_p$  fittest individuals from set the  $\{P, OP\}$  as initial population  $P$   
**end while**  
 شکل ۶- شبه‌کد الگوریتم تکامل تفاضلی مبتنی بر نقطه‌ی مقابل

<sup>32</sup>Hill Climbing

<sup>33</sup>Pseudo-elitism

<sup>34</sup>Cauchy Distribution

```

1. Initialize a new population  $P$ ;
   for each individual  $I \in P$  do
       Make the opposite Individual
        $I_{Opp}$  based on Equation (1)
   end for
   Generate a new population with selecting the
   best individual between  $I$  and  $I_{Opp}$ 
2. Continue the algorithm with the initialized
   population
   while the termination conditions are not
   satisfied
       for each individual  $I \in P$  do
           Selected Three Individuals randomly
            $I_{r1}, I_{r2}, I_{r3}$  where  $I \neq I_{r1} \neq I_{r2} \neq I_{r3}$ 
           A mutant vector  $V$  is made based on
           the
           Equation (2)
           for  $i=0$  to  $D$ 
               if  $rand(0,1) \leq C_r$ 
                    $I_{temp_i} = V_i$ 
               else
                    $I_{temp_i} = I_i$ 
               end if
           end for
           if  $fitness(I_{temp})$  is better than  $fitness(I)$ 
                $I_{next} = I_{temp}$ 
           else
                $I_{next} = I$ 
           end if
       end for
       Select the best individual  $I_{Best}$ 
       for  $i=0$  to  $D$ 
            $I_{best_i} = I_{best_i} + x_{Cauchy}$ 
       end for
       Pass  $I_{Best}$  to the next population

       for each individual  $I \in P$  do
           if  $rand(0,1) \leq S_r$ 
               Apply Solis Wets local search on  $I$ 
           end if
       end for
       if  $rand(0,1) \leq J_r$ 
           for each individual  $I \in P$  do
               Make the opposite individual  $I_{Opp}$ 
               based on Equation (1)
               Select the best individual  $\{I, I_{Opp}\}$  to
               make the next generation
           end for
       end if
   end while

```

شکل ۸- شبه کد الگوریتم پیشنهادی

```

Function SolisWets(Sol,N,bias,rho)
    numEval  $\leftarrow$  0
    numSuccess  $\leftarrow$  0
    numFailed  $\leftarrow$  0
    For numEval  $\leq$  maxEval
    For i=1 to N do
         $diff[i] \leftarrow RandomGsussian(0, rho)$ 
    End for
    End for
     $sol' \leftarrow sol + bias + diff$ 
    If  $sol'$  is better than  $sol$ 
         $sol \leftarrow sol'$ 
         $bias \leftarrow 0.2 \times bias + 0.4 \times (dif + bias)$ 
        numSuccess  $\leftarrow$  numSuccess + 1
        numFailed  $\leftarrow$  0
    Else
         $sol'' \leftarrow sol - bias - diff$ 
    If  $sol''$  is better than  $sol$ 
         $sol \leftarrow sol''$ 
         $bias \leftarrow bias - 0.4 \times (dif + bias)$ 
        numSuccess  $\leftarrow$  numSuccess + 1
        numFailed  $\leftarrow$  0
    Else
        numFailed  $\leftarrow$  numFailed + 1
        numSuccess  $\leftarrow$  0
    End if
    End if
    If (numSuccess > 5)
         $rho \leftarrow 2 \times rho$ 
        numSuccess  $\leftarrow$  0
    Else (numFailed > 3)
         $rho \leftarrow \frac{rho}{2}$ 
        numFailed  $\leftarrow$  0
    End if
    numEval  $\leftarrow$  numEval + 1
End for
End Function

```

شکل ۷- شبه کد الگوریتم جستجوی محلی Solis-West

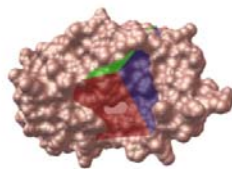
تغییر اندک در بهترین بردار به کمک توزیع کوشی سبب می‌شود تا از حضور یک بردار یکسان در تکرارهای مختلف الگوریتم جلوگیری شود. با توجه به این توضیحات شبه کد الگوریتم پیشنهادی را به صورت شکل (۸) است.

X	Y	Z	Q <sub>w</sub>	Q <sub>x</sub>	Q <sub>y</sub>	Q <sub>z</sub>	O <sub>1</sub>	O <sub>2</sub>	----
---	---	---	----------------	----------------	----------------	----------------	----------------	----------------	------

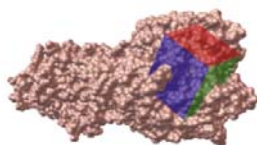
شکل ۹- بازنمایی وضعیت یک برهم‌کنش

## ۲-۴- کنترل محل قرار گیری دارو در گیرنده

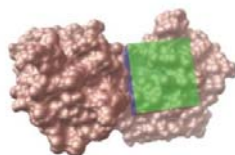
در این مقاله فرض شده‌است محل اتصال دارو به پروتئین به صورت تقریبی مشخص است. از این رو برای یافتن بهترین محل اتصال، لازم نیست تمام فضای پروتئین جستجو شود. برای نمونه در شکل (۱۰) چند ساختار مختلف پروتئین نمایش داده شده‌است که در آن‌ها محل قرار گیری دارو به صورت تقریبی به وسیله‌ی یک مکعب مشخص شده‌است. برای حل مسئله فرض می‌شود ساختار دارو تنها در این مکعب قابل حرکت و تغییر مکان می‌باشد. از این رو فضای جستجوی کل مسئله به صورت قابل ملاحظه‌ای کاهش می‌یابد و حل مسئله آسان‌تر خواهد شد.



(الف)



(ب)



(ج)

شکل ۱۰- نمایش محل تقریبی اتصال لیگاند. الف) پروتئین 1hvr

ب) پروتئین 4hmg (ج) پروتئین 4dfr (تصاویر به کمک

نرم‌افزار AutoDockTools تهیه شده‌است)

اضافه کردن یک الگوریتم جستجوی محلی و یک عملگر شبه‌نخبه‌گرایی به الگوریتم تکامل تفاضلی سبب افزایش دقت الگوریتم در یافتن بهترین برهم‌کنش خواهد شد. در ادامه روش حل مسئله‌ی پهلویی مولکولی را به کمک الگوریتم پیشنهادی شرح می‌دهیم.

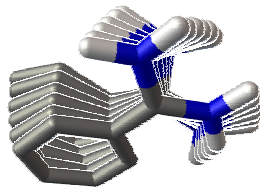
## ۴- حل مسئله پهلویی مولکولی

همان‌طور که اشاره شد اولین گام برای حل مسئله‌ی پهلویی مولکولی به کمک الگوریتم‌های فرااکتشافی مبتنی بر جمعیت، ارائه یک روش برای بازنمایی مسئله می‌باشد. در ادامه، به شرح بازنمایی استفاده شده برای وضعیت یک برهم‌کنش پروتئین-دارو توسط یک بردار جواب می‌پردازیم.

### ۱-۴- بازنمایی برهم‌کنش

برای نمایش وضعیت قرارگیری دارو در گیرنده پروتئینی لازم است تا مکان و نحوه‌ی قرارگیری مولکول دارو در گیرنده مشخص باشد. همچنین اگر دارو دارای پیوندهای آزاد باشد لازم است تا وضعیت این پیوندها نیز مشخص شود. نمایشی که برای یک عضو جمعیت به کار رفته‌است در شکل (۹) نمایش داده شده‌است. سه متغیر  $X, Y, Z$  در این بازنمایی، مکان قرار گیری دارو را در گیرنده پروتئین مشخص می‌کنند. همچنین مولکول دارو قادر است در یک نقطه حرکت چرخشی داشته‌باشد. نمایش چرخش دارو در محل خود را می‌توان با اعداد کوآرتنیون<sup>۳۵</sup> نمایش داد.  $Q_w, Q_x, Q_y, Q_z$  چهار عددی هستند که به کمک آن‌ها می‌توان وضعیت دارو را در یک نقطه تعیین کرد. همچنین وضعیت قرار گرفتن پیوندهای آزاد دارو را می‌توان با متغیرهای  $O_1, O_2, O_3 \dots$  نمایش داد. در ادامه به شرح جزئیات این بازنمایی می‌پردازیم.

<sup>35</sup> Quaternion



شکل ۱۲- چرخش ساختار Benzamidine حول نقطه‌ی مرکزی

$$Q = W = Xi + Yj + Zk \quad (8)$$

$$\text{Where } w^2 + x^2 + y^2 + z^2 = 1$$

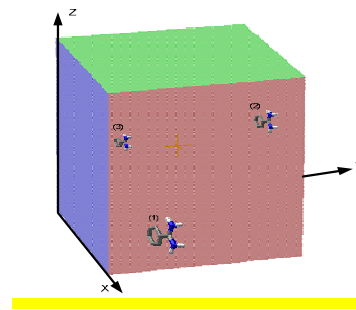
یک عدد کوآترینیون به صورت معادله (۸) نمایش داده می‌شود. در بازنمایی نحوه‌ی برهم‌کنش دارو توسط بردارهای الگوریتم تکامل تفاضلی از اعداد کوآترینیون به شکلی که در الگوریتم ژنتیک استفاده شده‌است، نمی‌توان استفاده کرد. مشکل از اینجا به وجود می‌آید که در الگوریتم تکامل تفاضلی بعد از تفریق کردن مقادیر دو بردار از یکدیگر، اعدادی که برای نمایش عدد کوآترینیون تولید خواهند شد شرط زیر را نقض می‌کند.

$$W^2 + X^2 + Y^2 + Z^2 = 1$$

با نقض این شرط چهارتایی مورد نظر دیگر یک عدد کوآترینیون معتبر نخواهد بود. از این رو باید به دنبال راهکاری جدید برای بازنمایی چرخش دارو بود. برای حل این مشکل از سه عدد تصادفی بین ۰ تا ۱ برای نمایش یک چرخش استفاده کرده‌ایم. با استفاده از این سه عدد و روشی که در [۳۵] ارائه شده‌است می‌توان اعداد کوآترینیون تصادفی با توزیع نرمال تولید کرد. اگر  $Q = (w, x, y, z)$  یک عدد کوآترینیون و  $R_1, R_2, R_3$  سه عدد تصادفی در بازه‌ی [۰, ۱] باشند، آنگاه عدد کوآترینیون متناظر با این سه مقدار در معادله (۹) نمایش داده شده‌است.

متخصصان داروسازی قادر به یافتن محل تقریبی اتصال دارو بر پایه‌ی تجربیات خود هستند و می‌توانند این مکعب‌ها را به صورت تجربی تعیین کنند، اما کار دشوار یافتن محل دقیق اتصال و تعیین زوایای پیوندهای آزاد دارو می‌باشد که توسط الگوریتم پیشنهادی پیش‌بینی خواهد شد.

محل قرارگیری دارو در گیرنده به کمک سه متغیر  $(X, Y, Z)$  در بردار جواب کنترل می‌شود. در حقیقت مقادیر این سه متغیر مختصات نقطه‌ی مرکزی دارو است که در درون مکعب تغییر مکان می‌دهد. برای مثال در شکل (۱۱) تغییر مکان یک ساختار دارو در فضای مکعب نمایش داده شده‌است.

شکل ۱۱- تغییر مکان دارو در فضای گیرنده به کمک سه پارامتر  $(X, Y, Z)$  در بردار جواب کنترل می‌شود.

### ۳-۴- کنترل چرخش دارو در یک نقطه

پس از تعیین محل قرارگیری دارو، نوبت به تعیین وضعیت دارو در آن نقطه می‌رسد. هر ساختار دارو که در یک نقطه‌ی مشخص قرار دارد می‌تواند حول نقطه‌ی مرکزی خود چرخش داشته‌باشد. برای نمونه در شکل (۱۲) چرخش مولکول بنزامیدین<sup>۳۶</sup> حول نقطه‌ی مرکزی نمایش داده شده‌است.

برای کنترل چرخش یک مولکول دارو از اعداد کوآترینیون استفاده می‌کنیم [۳۴]. اعداد کوآترینیون یک سیستم اعداد جابجایی ناپذیر می‌باشند که گسترش یافته‌ی سیستم اعداد مختلط<sup>۳۷</sup> هستند.

<sup>36</sup> Benzamidine<sup>37</sup> Complex Number

پیوندهای آزاد، متغیر به انتهای بردار جواب اضافه کرد. در شکل (۱۳) این متغیرها با  $O_i$  نمایش داده شده‌اند. مقادیر این متغیرها در بازه‌ی  $[0, 2\pi]$  تغییر می‌کند. بدین ترتیب طول یک بردار جواب متغیر است و بر اساس معادله (۱۰) بدست می‌آید. برای مثال طول بردار بازنمایی برای داروی فسفوکلین که چهار پیوند آزاد دارد، برابر ۱۰ می‌باشد.

(۱۰) تعداد پیوندهای آزاد دارو + ۶ = طول بردار جواب

#### ۵-۴- ارزیابی یک برهم‌کنش

برای حل مسئله‌ی پهلوگیری مولکولی لازم است تا یک تابع امتیازدهی برای ارزیابی هر برهم‌کنش تعریف کنیم. تابع ارزیابی که در این مقاله از آن استفاده شده تابع امتیازدهی AutoDock می‌باشد که یکی از توابع زیرمجموعه AMBER<sup>39</sup> می‌باشد [۷]. این تابع ارزیاب به دلیل استفاده از مدل یکپارچه اتم<sup>40</sup> دارای هزینه‌ی محاسباتی پایین بوده و از دقت قابل قبولی برخوردار است. این تابع در معادله‌ی (۱۱) نشان داده شده‌است. در این معادله  $i$  به اتم‌های مولکول لیگاند و  $j$  به اتم‌های مولکول پروتئین اشاره دارد.

$$E = W_{vdw} \sum_{i,j} \left( \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^6} \right) + \text{محاسبه انرژی واندروالس}$$

$$W_{hbond} \sum_{i,j} E(t) \left( \frac{C_{ij}}{r_{ij}^{12}} - \frac{D_{ij}}{r_{ij}^{10}} \right) + \text{محاسبه انرژی پیوند هیدروژنی}$$

$$W_{elec} \frac{q_i q_j}{\epsilon(r_{ij}) r_{ij}} + \text{محاسبه انرژی الکتریکی} \quad (11)$$

$$W_{sol} \sum_{i,j} (S_i V_j + S_j V_i) e^{-\left( \frac{r_{ij}^2}{2\sigma^2} \right)} + \text{محاسبه انرژی انحلال}$$

برای هر برهم‌کنش پروتئین-دارو یک مقدار انرژی بر اساس معادله بالا بدست می‌آید. هرچه میزان انرژی محاسبه شده برای یک برهم‌کنش کم‌تر باشد، برهم‌کنش مورد نظر پایدارتر خواهد بود. از این رو می‌توان مسئله‌ی پهلوگیری

$$R_1, R_2, R_3 \in \mathbb{R} \text{ and } R_1, R_2, R_3 \in [0, 1]$$

$$Q = \begin{pmatrix} \sqrt{1 - R_1 \sin 2\pi R_2}, \sqrt{1 - R_1 \cos 2\pi R_2}, \\ \sqrt{R_1 \sin 2\pi R_3}, \sqrt{R_1 \cos 2\pi R_3} \end{pmatrix} \quad (1)$$

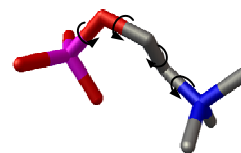
X	Y	Z	R <sub>1</sub>	R <sub>2</sub>	R <sub>3</sub>	O <sub>1</sub>	O <sub>2</sub>	...
---	---	---	----------------	----------------	----------------	----------------	----------------	-----

شکل ۱۳- بازنمایی یک برهم‌کنش در الگوریتم پیشنهادی

با استفاده از این روش، بازنمایی جدید برای حل مسئله به صورت شکل (۱۳) خواهد بود. به کمک این بازنمایی دیگر مشکل تفریق دو بردار جمعیت از هم وجود نخواهد داشت زیرا اعداد  $R_1, R_2, R_3$  سه عدد تصادفی بین ۰ تا ۱ هستند که در صورت تفریق دو بردار نیز چنین خاصیتی را حفظ خواهند کرد و بردار جدید حاصل از تفریق، حاوی یک عدد کوآرتنیون معتبر خواهد بود.

#### ۴-۴- کنترل زاویه‌ی پیوندهای آزاد دارو

تاکنون توانستیم با استفاده از شش متغیر در بردار بازنمایی، مکان قرارگیری دارو و همچنین چرخش دارو حول نقطه‌ی مرکزی را کنترل کنیم. اکنون نوبت به کنترل زاویه‌ی پیوندهای آزاد دارو است. در شکل (۱۴) پیوندهای آزاد ساختار فسفوکلین<sup>38</sup> نمایش داده شده‌است.



شکل ۱۴- پیوندهای آزاد Phosphocholine

برای کنترل زاویه‌ی پیوندهای آزاد دارو می‌توان به تعداد

<sup>39</sup> Assisted Model Building with Energy Refinement

<sup>40</sup> United Atom Model<sup>38</sup> Phosphocholine

برای ترکیب‌های مختلف در جدول (۲) آمده‌است.

جدول ۱. مشخصات ساختارهای استفاده شده برای ارزیابی الگوریتم

ابعاد مسئله	تعداد پیوندهای آزاد	PDB	ساختار لیگاند	ساختار پروتئین
۶	۰	3ptb	Benzamide	$\beta$ -Trypsin
۱۰	۴	2mcp	Phosphocholine	McPC-603
۱۱	۵	1stp	Biotin	Streptavidin
۱۶	۱۰	1hvr	XK263	HIV-1 protease
۱۷	۱۱	4hmg	Sialic acid	Influenza hemagglutinin
۱۳	۷	4dfr	Methotrexate	Dihydrofolate reductase

جدول ۲. مشخصات گسسته‌سازی محل تقریبی اتصال لیگاند

تعداد نقاط شبکه	نرخ گسسته‌سازی (A)	ابعاد فضای مکعب (A)	ترکیب
۸۲۸۶۱	۰.۳۷۵	۴۲×۴۰×۴۶	3ptb
۶۸۹۲۱	۰.۳۷۵	۴۰×۴۰×۴۰	1stp
۱۳۲۶۵۱	۰.۳۷۵	۵۰×۵۰×۵۰	2mcp
۲۲۶۹۸۱	۰.۳۷۵	۶۰×۶۰×۶۰	4dfr
۳۵۷۹۱۱	۰.۲۰۰	۷۰×۷۰×۷۰	4hmg
۲۲۶۹۸۱	۰.۳۷۵	۶۰×۶۰×۶۰	1hvr

در این بخش نتایج حاصل از الگوریتم پیشنهادی با سه الگوریتم دیگر مقایسه شده‌است. این سه الگوریتم عبارتند از الگوریتم ژنیتیک لامارکی (LGA)، الگوریتم سردسازی شبیه‌سازی شده (SA)، و الگوریتم تکامل تفاضلی معمولی (DE). الگوریتم‌های مقایسه شده همگی از نوع الگوریتم‌های فرااکتشافی تکرارشونده هستند. شرط خاتمه برای این الگوریتم‌ها رسیدن به تعداد معینی تکرار می‌باشد.

مولکولی را به گونه‌ی دیگری نیز مطرح کرد. این مسئله در واقع به دنبال نقطه‌ی کمینه‌ی تابع امتیازدهی معادله (۱۱) می‌باشد. مقدار این تابع زمانی به کمترین مقدار خود می‌رسد که بتوان پایدارترین برهم‌کنش را از میان فضای جواب بدست آورد.

## ۵- نتایج

برای ارزیابی کارایی الگوریتم‌های پهلوگیری مولکولی، می‌توان از چند ترکیب پروتئین-دارو که به روش تجربی در آزمایشگاه کشف شده‌اند استفاده کرد. این ساختارها را می‌توان از پایگاه اینترنتی PDB<sup>41</sup> [۳۶] دریافت کرد.

$$RMSD = \sqrt{\frac{1}{N} \times \sum_{i=1}^N (x_{ci} - x_{di})^2 + (y_{ci} - y_{di})^2 + (z_{ci} - z_{di})^2} \quad (12)$$

میزان خطا بین ساختار پیش‌بینی شده توسط الگوریتم شبیه‌سازی و ساختار واقعی دارو که در آزمایشگاه کشف شده‌است به کمک فرمول (۱۲) قابل محاسبه است [۳۷]. در این فرمول مختصات اتم‌های دارو که در آزمایشگاه کشف شده‌اند با  $(x_c, y_c, z_c)$  نمایش داده می‌شود و  $(x_d, y_d, z_d)$  مختصات اتم‌های ساختار پیش‌بینی شده توسط الگوریتم شبیه‌سازی می‌باشد. هرچه میزان عدد محاسبه شده توسط این فرمول به صفر نزدیک باشد، جواب بدست آمده توسط الگوریتم دقیق‌تر بوده و کارایی الگوریتم بالاتر می‌باشد.

از شش ساختار مختلف پروتئین-دارو برای بررسی کارایی الگوریتم پیشنهادی استفاده شده‌است. اسامی این شش ساختار پروتئینی به همراه اسامی لیگاندهای آن‌ها در جدول (۱) آمده‌است، خوانندگان محترم می‌توانند برای آشنایی بیشتر با این شش ساختار به [۷] مراجعه کنند.

در اولین گام برای حل مسئله، فضای پهلوگیری دارو بر روی پروتئین گسسته‌سازی می‌شود. هرچه تعداد نقاط گسسته‌سازی بیشتر باشد فضای جستجوی مسئله بزرگتر خواهد بود. حاصل گسسته‌سازی مکعب، یک شبکه خواهد بود که نقطه‌ی مرکزی دارو بر روی نقاط این شبکه جابجا می‌شود. نحوه گسسته‌سازی فضای پهلوگیری مسئله

<sup>41</sup> Protein Data Bank

حل این مسئله، پیش‌بینی بهترین برهم‌کنش دارو و گیرنده می‌باشد. استفاده از الگوریتم‌های کامپیوتری یکی از راه‌های سرعت بخشی برای حل این مسئله می‌باشد که در سال‌های اخیر بسیار مورد توجه محققین قرار گرفته‌است. این مسئله در دسته‌ی مسائل NP قرار دارد، از اینرو الگوریتم‌هایی که برای حل این مسئله ارائه شده‌اند اغلب از نوع فرااکتشافی می‌باشند. در این مقاله از نسخه‌ی تغییر یافته‌ی الگوریتم تکامل تفاضلی مبتنی بر نقطه‌ی مقابل برای حل این مسئله استفاده شده‌است. این الگوریتم مانند دیگر الگوریتم‌های فرااکتشافی، یک الگوریتم تکرار شونده متنی بر جمعیت‌است. پیاده‌سازی الگوریتم پیشنهادی در بستر سیستم عامل لینوکس و با زبان برنامه‌نویسی ++C انجام شده‌است. همچنین از تابع امتیازدهی AutoDock برای ارزیابی انرژی ترکیب‌های مختلف استفاده شده‌است.

جدول ۳. انرژی بدست‌آمده حاصل از اجرای الگوریتم‌های SA, LGA,

mODE برای ساختارهای متفاوت

PDB کد	انرژی ( Kcal/mol )			
	الگوریتم	بهترین	میانگین	انحراف معیار
3ptb	mODE	-5.77	-5.76	0.01
	LGA	-5.67	-5.53	0.09
	SA	-5.10	-4.60	0.22
	DE	-5.72	-5.69	0.13
2mcp	mODE	-6.32	-6.04	0.10
	LGA	-6.15	-5.90	0.14
	SA	-4.81	-4.38	0.15
	DE	-6.24	-5.92	0.12
1stp	mODE	-8.49	-8.40	0.06
	LGA	-8.55	-8.20	0.20
	SA	-7.44	-5.94	0.74
	DE	-8.23	-8.11	0.20
1hvr	mODE	-20.41	-20.38	0.03
	LGA	-20.11	-15.03	3.10
	SA	-10.22	-7.57	0.90
	DE	-20.23	-19.4	1.1
4hmg	mODE	-11.52	-10.87	0.64
	LGA	-10.69	-10.13	0.45
	SA	-6.80	-6.54	0.29
	DE	-11.6	-10.9	0.4
4dfr	mODE	-15.87	-14.59	1.37
	LGA	-15.11	-12.98	1.64
	SA	-8.10	-6.36	0.75
	DE	-15.31	-15.01	1.24

در شکل (۱۵) نتایج حاصل از اجرای این چهار الگوریتم نمایش داده شده‌است. ستون افقی در این نمودارها نشان دهنده‌ی تعداد تکرار الگوریتم می‌باشد و ستون عمودی انرژی ترکیب در هر تکرار را نمایش می‌دهد. برای مقایسه‌ی دقیق‌تر نتایج، هر الگوریتم برای هر ترکیب به تعداد ده مرتبه اجرا شده و میانگین انرژی ساختارها در این ده اجرا گزارش شده‌است. اندازه‌ی جمعیت استفاده شده برای الگوریتم ژنتیک لامارکی و الگوریتم پیشنهادی ۱۵۰ می‌باشد. همچنین در الگوریتم LGA نرخ همبری برابر ۰.۸ و نرخ جهش برابر ۰.۰۲ می‌باشد. در الگوریتم پیشنهادی که با mODE نام‌گذاری شده‌است، اندازه‌ی فاکتور مقیاس (F) و نرخ همبری برابر ۰.۵ می‌باشد. در الگوریتم تکامل تفاضلی معمولی (DE) پارامترها همانند الگوریتم mODE انتخاب شده‌اند.

در جدول (۳) میزان انرژی نهایی هر ترکیب برای چهار الگوریتم مختلف باهم مقایسه شده‌است. همان‌طور که از نتایج برمی‌آید الگوریتم پیشنهادی توانسته است برهم‌کنش‌هایی با انرژی درونی کمتر را پیش‌بینی کند. هرچند در یافتن بهترین برهم‌کنش، نتایج mODE با الگوریتم LGA و DE تقریباً مشابه می‌باشد اما این نتیجه برای میانگین جواب‌ها اختلاف محسوسی داشته و نشان‌دهنده‌ی قابل اعتماد بودن الگوریتم پیشنهادی در اجراهای مختلف است.

همچنین مقایسه‌ی خطای ساختار بدست آمده از این چهار الگوریتم در جدول (۴) آمده‌است. در این جدول اندازه‌ی RMSD بهترین ترکیبی که کمترین اختلاف را با ساختار مرجع دارد به کمک رابطه‌ی (۱۲) بدست می‌آید. بر اساس این نتایج الگوریتم پیشنهادی توانسته در مواردی که ساختار دارو دارای پیوندهای آزاد بیشتری است بهتر از دیگر الگوریتم‌ها عمل کند. جدول (۵) نیز زمان حل مسئله به کمک الگوریتم‌های مختلف را نمایش می‌دهد.

## ۶- نتیجه‌گیری

مسئله‌ی پهلویی مولکولی یکی از مسائل مهم در حوزه‌ی بیوانفورماتیک و شیمی‌انفورماتیک می‌باشد. هدف از

نتایج حاصل از پیش‌بینی ساختار دارو به کمک الگوریتم ارائه‌شده با سه الگوریتم ژنتیک لامارکی، سردسازی شبیه‌سازی شده و الگوریتم تکامل تفاضلی معمولی مقایسه شده‌است. بر اساس نتایج بدست آمده الگوریتم پیشنهادی قادر خواهد بود ساختارهای دارویی با دقت بالاتر را در زمان کمتر کشف کند. همچنین mODE از قابلیت اعتماد بالاتری برخوردار بوده و انحراف معیار جواب‌ها در اجرای مختلف الگوریتم از دیگر الگوریتم‌ها کم‌تر می‌باشد.

جدول ۵. میانگین زمان اجرای الگوریتم‌های LGA, SA, mODE برای

ساختارهای متفاوت

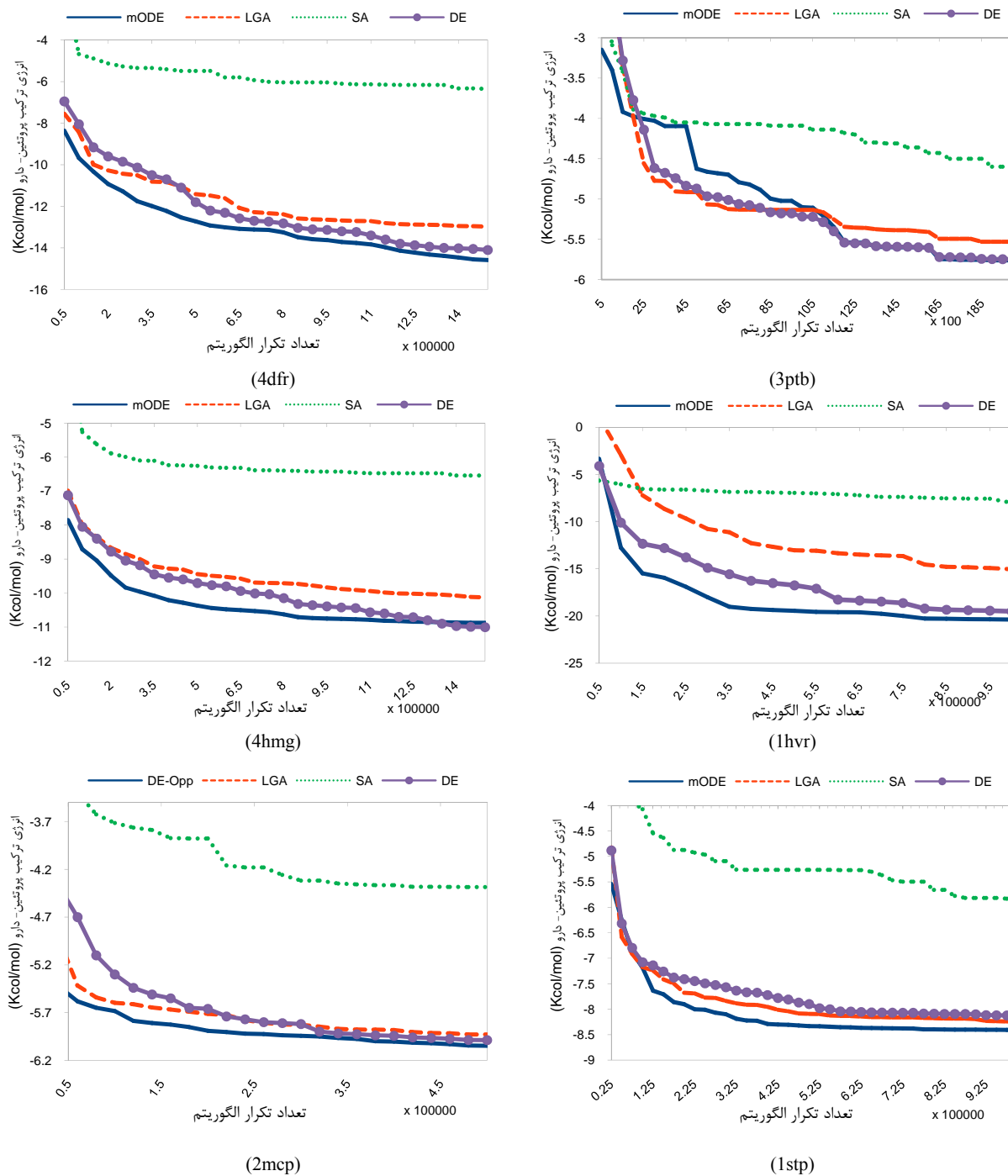
کد PDB	میانگین زمان (ثانیه)			
	DE	SA	LGA	mODE
3ptb	0.4	2	0.3	0.24
2mcp	7	21	7	7
1stp	22	53.2	21.8	21.2
1hvr	111	170.2	85.5	101.8
4hmg	70.2	130.6	60.1	63.3
4dfr	120	192	111.1	109.2

جدول ۴. RMSD بدست آمده حاصل از اجرای الگوریتم‌های LGA, SA, mODE

برای ساختارهای متفاوت

کد PDB	الگوریتم	RMSD (Å)		
		بهترین	میانگین	انحراف معیار
3ptb	mODE	0.19	0.24	0.04
	LGA	0.40	1.31	1.14
	SA	0.59	1.52	1.17
	DE	0.21	0.23	0.43
2mcp	mODE	0.67	0.94	0.17
	LGA	0.83	1.04	0.09
	SA	0.89	1.33	0.31
	DE	0.72	1.11	0.21
1stp	mODE	0.5	0.72	0.13
	LGA	0.68	0.83	0.13
	SA	0.82	1.80	1.11
	DE	0.71	0.89	0.34
1hvr	mODE	0.44	0.56	0.05
	LGA	0.44	2.85	2.32
	SA	3.97	5.60	0.83
	DE	0.67	1.1	1.01
4hmg	mODE	0.97	1.22	0.37
	LGA	1.11	1.98	0.92
	SA	1.57	2.57	0.95
	DE	0.91	1.21	0.67
4dfr	mODE	0.83	1.08	0.34
	LGA	0.91	2.10	1.55
	SA	3.08	6.25	2.39
	DE	0.9	1.32	1.16





شکل ۱۵- میانگین انرژی حاصل از اجرای چهار الگوریتم در ده اجرای متفاوت برای شش ترکیب مختلف

- Proteins: Structure, Function, and Bioinformatics*, vol. 8, pp. 195-202, 1990.
- [17] S. Janson, *et al.*, "Molecular docking with multi-objective Particle Swarm Optimization," *Applied Soft Computing*, vol. 8, pp. 666-675, 2008.
- [18] A. Oduguwa, *et al.*, "Multi-objective optimisation of the protein-ligand docking problem in drug discovery," 2006, pp. 1793-1800.
- [19] O. Korb, *et al.*, "PLANTS: Application of ant colony optimization to structure-based drug design," *Ant Colony Optimization and Swarm Intelligence*, pp. 247-258, 2006.
- [20] R. Thomsen, "Flexible ligand docking using differential evolution," 2003, pp. 2354-2361 Vol. 4.
- [21] S. Rahnamayan, "Opposition-based differential evolution," Doctor of Philosophy, Systems Design Engineering, University of Waterloo, 2008.
- [22] H. R. Tizhoosh, "Reinforcement learning based on actions and opposite actions," 2005.
- [23] L. Yao, *et al.*, "Novel opportunities for computational biology and sociology in drug discovery:: Corrected paper," *Trends in biotechnology*, vol. 28, pp. 161-170, 2010.
- [24] I. S. Moreira, *et al.*, "Protein-protein docking dealing with the unknown," *Journal of computational chemistry*, vol. 31, pp. 317-342, 2010.
- [25] M. Totrov and R. Abagyan, "Flexible ligand docking to multiple receptor conformations: a practical alternative," *Current opinion in structural biology*, vol. 18, pp. 178-184, 2008.
- [26] M. Hartenfeller and G. Schneider, "De novo drug design," *Meth Mol Biol*, vol. 672, pp. 299-323, 2011.
- [27] M. L. Teodoro, *et al.*, "Molecular docking: A problem with thousands of degrees of freedom," 2001, pp. 960-965 vol. 1.
- [28] R. Storn and K. Price, "Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces," *International Computer Science Institute-Publications-TR*, 1995.
- [29] R. Storn and K. Price, "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341-359, 1997.
- [30] K. V. Price, *et al.*, *Differential evolution*: Springer Berlin, Germany, 2005.
- [31] H. R. Tizhoosh, "Opposition-based learning: A new scheme for machine intelligence," 2005, pp. 695-701.
- [32] S. Rahnamayan, *et al.*, "Opposition-based differential evolution," *Evolutionary Computation, IEEE Transactions on*, vol. 12, pp. 64-79, 2008.
- [33] F. J. Solis, "Minimization by random search techniques," *Mathematics of operations research*, pp. 19-30, 1981.
- [34] A. L. Mackay, "Quaternion transformation of molecular orientation," *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 40, pp. 165-166, 1984.
- [35] K. Shoemake, "Uniform Random Rotations," in *Graphics Gems III*, D. Kirk, Ed., ed New York and Orlando: Academic Press, 1992, pp. 124-132.
- [36] W. D. Cornell, *et al.*, "A second generation force field for the simulation of proteins, nucleic acids, and
- [1] Krasky, *et al.*, "A combined bioinformatics and chemoinformatics approach for the development of new antiparasitic drugs," *Genomics*, vol. 89, pp. 36-43, 2007.
- [2] C. Steinbeck, *et al.*, "Computational metabolomics—a field at the boundaries of cheminformatics and bioinformatics," *Journal of Cheminformatics*, vol. 3, pp. 1-1, 2011.
- [3] S. Mandal, *et al.*, "Rational drug design," *European journal of pharmacology*, vol. 625, pp. 90-100, 2009.
- [4] Y. Chen and B. K. Shoichet, "Molecular docking and ligand specificity in fragment-based inhibitor discovery," *Nature Chemical Biology*, vol. 5, pp. 358-364, 2009.
- [5] A. Oduguwa, *et al.*, "An Overview of Soft Computing Techniques Used in the Drug Discovery Process," *Applied Soft Computing Technologies: The Challenge of Complexity*, pp. 465-480, 2006.
- [6] T. J. A. Ewing, *et al.*, "DOCK 4.0: search strategies for automated molecular docking of flexible molecule databases," *Journal of computer-aided molecular design*, vol. 15, pp. 411-428, 2001.
- [7] G. M. Morris, *et al.*, "Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function," *Journal of computational chemistry*, vol. 19, pp. 1639-1662, 1998.
- [8] G. M. Morris, *et al.*, "AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility," *Journal of computational chemistry*, vol. 30, pp. 2785-2791, 2009.
- [9] M. L. Verdonk, *et al.*, "Improved protein-ligand docking using GOLD," *Proteins: Structure, Function, and Bioinformatics*, vol. 52, pp. 609-623, 2003.
- [10] S. Joy, *et al.*, "Detailed comparison of the protein-ligand docking efficiencies of GOLD, a commercial package and ArgusLab, a licensable freeware," *In silico biology*, vol. 6, pp. 601-605, 2006.
- [11] B. Kramer, *et al.*, "Evaluation of the FLEXX incremental construction algorithm for protein-ligand docking," *Proteins: Structure, Function, and Bioinformatics*, vol. 37, pp. 228-241, 1999.
- [12] I. D. Kuntz, *et al.*, "A geometric approach to macromolecule-ligand interactions," *Journal of molecular biology*, vol. 161, pp. 269-288, 1982.
- [13] M. Liu and S. Wang, "MCDOCK: a Monte Carlo simulation approach to the molecular docking problem," *Journal of computer-aided molecular design*, vol. 13, pp. 435-451, 1999.
- [14] R. Meier, *et al.*, "ParaDockS: A Framework for Molecular Docking with Population-Based Metaheuristics," *Journal of Chemical Information and Modeling*, vol. 50, pp. 879-889, 2010.
- [15] B. S. Sadjad and Z. Zsoldos, "Toward a Robust Search Method for the Protein-Drug Docking Problem," *IEEE IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pp. 1120-1133, 2010.
- [16] D. S. Goodsell and A. J. Olson, "Automated docking of substrates to proteins by simulated annealing,"

- organic molecules," *Journal of the American Chemical Society*, vol. 117, pp. 5179-5197, 1995.
- [37] R. Dias, *et al.*, "Molecular docking algorithms," *Current Drug Targets*, vol. 9, pp. 1040-1047, 2008.