ارزیابی عملکرد الگوریتم‌های مختلف فرااکتشافی در کشف کلید رمز الگوریتم رمزنگاری ویجینر

مهدی احمدی پری و میثم مرادی
گروه کامپیوتر، دانشکده فنی و مهندسی، دانشگاه ملایر، ملایر، ایران
Mahmadip@Yahoo.Com
En.M.Moradi.Co@Gmail.Com

چکیده

در سال‌های اخیر، استفاده از الگوریتم‌های فرااکتشافی در مسائل مختلف، مورد توجه قرار گرفته است. الگوریتم‌های فرااکتشافی در حل مسائل مختلف، کارایی و عملکرد متفاوتی از خود نشان می‌دهند. یک الگوریتم فرااکتشافی ممکن است برای حل یک مسئله خاص از دیگر الگوریتم‌ها، عملکرد بهتر و در یک مسئلهٔ دیگر، عملکرد ضعیف‌تری از خود نشان دهند. در این پژوهش عملکرد الگوریتم‌های مختلف فرااکتشافی برای یک مسئله خاص که کشف کلید رمز در الگوریتم رمزنگاری ویجینر است مورد بررسی قرار گرفته و عملکرد الگوریتم‌های مختلف فرااکتشافی از نظر دقت نتایج حاصل و سرعت هم‌گرایی، مورد تحلیل رمز قرار خواهد گرفت و بهترین الگوریتم انتخاب می‌شود.

واژگان کلیدی: تحلیل‌رمز، الگوریتم رمزنگاری ویجینر، الگوریتم‌های فرااکتشافی ، کشف کلید رمز.

۱- مقدمه

با توجه به افزایش حجم مستندات متنی و تبادل آن بین افراد در سطح‌های مختلف ارتباطاتی، نیازمند الگوریتم‌های قدرتمند در زمینه رمزنگاری هستیم. الگوریتم‌های رمزنگاری از دو بخش رمزنگاری و رمزگشایی تشکیل شده‌اند [۶]. مجموعه مراحل تبدیل یک متن ساده به معادل رمزشدهٔ آن را رمزنگاری می‌گویند و معکوس آن یعنی مراحل تبدیل یک متن رمزشده به متن ساده را رمزگشایی می‌گویند. سامانهٔ الگوریتم‌های رمزنگاری به دو دستهٔ کلی متقارن[1] و نامتقارن[2] تقسیم می‌شوند [۷]. در الگوریتم‌های متقارن برای رمزنگاری و رمزگشایی پیغام‌ها، از یک کلید استفاده، درحالی‌که در الگوریتم‌های نامتقارن از دو کلید متفاوت جهت رمزنگاری استفاده می‌شود [۸]. الگوریتم‌های متقارن نیز خود به دو دسته کلاسیک و مدرن تقسیم می‌شوند[۱].

الگوریتم‌های رمزنگاری متقارن کلاسیک براساس جابه‌جایی یا جایگزینی حروف، متن‌ها را رمز می‌کنند. تفاوت رمزهای جابه‌جایی با رمزهای جایگزینی در این است که رمزهای جابه‌جایی، حروف سادهٔ متن را عوض نمی‌کنند؛ بلکه فقط جای آن‌ها را تغییر می‌دهند و آن‌ها را جابه‌جا می‌کنند؛ درحالی‌که رمزهای جانشینی، هر حرف از متن ساده را با حرفی دیگر جایگزین می‌کنند. الگوریتم ویجنر یک الگوریتم رمزنگاری کلاسیک است که از روش جانشینی برای رمزکردن متن‌ها استفاده می‌کند [۱]. این الگوریتم جزء نخستین الگوریتم‌های رمزنگاری چندالفبایی شناخته شده است [۹].

تحلیل رمز به مطالعه روش‌ها و اصولی می‌پردازد که براساس آن می‌توان بدون در اختیار داشتن کلید رمز داده‌های رمزنگاری‌شده را از رمز خارج کرد یا کلید رمز را به‌دست آورد [۱].

[1] Symmetric
[2] asymmetric

تاکنون روش‌های مختلفی برای شکستن الگوریتم رمزنگاری ویجینر کشف شده‌اند. پژوهش‌گران در [۱۰] از الگوریتم ژنتیک، جهت شکستن الگوریتم ویجینر استفاده کرده‌اند. در [۱۱] از الگوریتم بهینه‌سازی پرتو ذرات جهت تحلیل رمز الگوریتم ویجینر استفاده شده است. یکی از روش‌ها آزمایش همه حالت‌های مختلف کلید است. درصورت کم‌بودن حالت‌های مختلف کلید روش موثری می‌تواند باشد. در الگوریتم ویجینر تعداد کلیدهای مختلف 26^n است که n به‌عنوان طول کلید می‌باشد. به‌عنوان مثال، تعداد کلیدهای مختلف با طول چهار، ۴۵۶۹۷۶ کلید مختلف است. هر چه قدر طول کلید بزرگتر باشد، این عدد بزرگتر و آزمون همه این کلیدها بسیار زمان‌بر می‌تواند شود. برای رویارویی با این مشکل می‌توان از الگوریتم‌های جستجوی فرااکتشافی استفاده کرد. این الگوریتم‌ها به جای جستجوی تمام فضای حالت، جستجو را به بخشی از فضای کلید محدود می‌کنند و در زمان کمتری به جواب می‌رسند. الگوریتم‌های فرااکتشافی مختلف، عملکرد متفاوتی برای حل مسئله کشف کلید رمز در الگوریتم رمزنگاری ویجینر دارند. برخی هم‌گرایی سریع‌تری دارند و برخی نتایج دقیق‌تری ارائه می‌دهند. در این پژوهش سعی شده است، عملکرد این الگوریتم‌ها برای کشف کلید رمز در الگوریتم رمزنگاری ویجینر مورد بررسی قرار گیرد و بهترین الگوریتم را از نظر دقت نتایج حاصل و سرعت هم‌گرایی مورد تحلیل رمز قرار گیرد.

ساختار پژوهش به این صورت سازماندهی شده است. در بخش دوم کارهای مرتبط، در بخش سوم الگوریتم رمزنگاری ویجینر و الگوریتم‌های جستجو معرفی می‌شوند. در بخش چهارم به طراحی آزمایش‌ها پرداخته شده است. در بخش پنجم نتایج آزمایش‌ها و در بخش ششم نیز نتیجه‌گیری بیان شده است.

۲- کارهای مرتبط

در سال‌های اخیر پژوهش‌های متعددی در بررسی الگوریتم‌های جستجو در حوزه‌های مختلف انجام شده است.

احمدی و همکاران [۲] شکستن الگوریتم رمزنگاری ویجینر را با استفاده از الگوریتم بهینه‌سازی پرتو ذرات بررسی کردند. اسدی و همکاران [۳] کشف کلیدرمز در الگوریتم رمزنگاری ویجینر را با استفاده از الگوریتم بهینه‌سازی پرتو ذرات مشارکتی (MPSO) مورد تحلیل رمز قرار داده‌اند. رستگار و همکاران [۴] یک الگوریتم تکاملی جهت تخمین توزیع جدید با استفاده از آتوماتای یادگیر را بررسی کردند. هدیه و همکاران [۵] الگوریتمی جهت یافتن بهینه سراسری در مسائل را مورد پژوهش قرار دادند. *Gopalakrishnan* و همکاران [۱۲] تحلیل رمز ویجینر را با الگوریتم ژنتیک انجام دادند. *Sivagurunathan* و همکاران [۱۴] کاهش فضای کلید جستجوی الگوریتم رمزنگاری ویجینر را با استفادهٔ بهینه‌سازی پرتو ذرات بررسی کردند. مرادی و همکاران [۱۵] شکستن الگوریتم رمزنگاری SDES را با استفاده از استاندارد بهینه‌سازی پرتو ذرات بهینه شده بررسی کردند. *Karaboga* و همکاران [۲۰] یک ایده مبتنی بر بهینه‌سازی عددی را با استفاده از الگوریتم زنبور عسل مطرح کردند. Garg [۲۳] الگوریتم ترکیبی[1] و الگوریتم ژنتیک را جهت تحلیل‌رمز الگوریتم SDES موردارزیابی قرار داد. *Hasan Husein* و همکاران [۲۴] الگوریتم ژنتیک را جهت تحلیل رمز الگوریتم DES8 استفاده کردند. *Salabat* و همکاران [۲۵] الگوریتم کلونی مورچه‌ها را جهت تحلیل رمز الگوریتم DES استفاده کردند. *Sathya* و همکاران[۲۶] یک روش جدید در الگوریتم ژنتیک در تحلیل رمز الگوریتم DES16 معرفی کردند. *Laskari* و همکاران[۲۷] روش هوش تکاملی را جهت تحلیل رمز متون به کار گرفتند.

۳- بررسی الگوریتم ویجینر و الگوریتم‌های جستجو

در این بخش به معرفی الگوریتم ویجینر و الگوریتم‌های جستجو پرداخته می‌شود.

۳-۱ الگوریتم رمزنگاری ویجینر(Vigenere)

الگوریتم رمزنگاری ویجینر نخستین‌بار در سال ۱۹۵۳ ارائه شد. این الگوریتم یک روش رمزنگاری جانشینی است. رمزنگاری جانشینی روشی است که در آن هر واحد از متن اصلی بر طبق یک سامانه معین با رمزشدهٔ آن جایگزین می‌شود. یک واحد ممکن است یک تا، سه حرف یا حتی ترکیبی از آن‌ها و شکل‌های مشابه آن باشد. در این الگوریتم به هر یک از حروف یک عدد نسبت داده می‌شود.

[1] Memetic

به‌عنوان مثال برای حروف زبان انگلیسی اعداد صفر تا ۲۵ به‌ترتیب به حروف a تا z نسبت داده می‌شود؛ سپس متنی که قرار است رمز شود دسته‌بندی می‌شود. تعداد حروف هر دسته با تعداد حروف کلید برابر است. فرآیند رمزنگاری به‌صورت زیر انجام می‌شود:

$$Ci = Di + ki \pmod{26} \quad (۱)$$

که در آن Di برابر است با i اُمین حرف هر دسته، ki برابر است با i اُمین حرف کلید و Ci نیز i اُمین حرف از متن رمز شده است. فرآیند رمزگشایی نیز به‌صورت زیر انجام می‌شود.

$$Ci = Di - ki \pmod{26} \quad (۲)$$

به‌عنوان مثال برای رمزکردن متن how are you today با استفاده از کلید test، ابتدا فاصله بین حروف حذف شده، سپس کلید به اندازهٔ طول متن پشت سر هم تکرار می‌شود. بعد با استفاده از رابطهٔ (۱) متن رمز می شود. مراحل در جدول (۱) آمده است.

(جدول ۱): مراحل رمز متن در الگوریتم ویجینر

متن اصلی	**howareyoutoday**
کلیدرمز	**testtesttestte**
متن رمزشده	**asotkiqhnxgwtc**

در این مثال حرف h با عدد معادل ۷ با حرف t که عدد معادل آن ۱۹ است جمع به پیمانه ۲۶ شده که نتیجه آن عدد صفر یا همان حرف a است. بقیه حروف نیز به همین ترتیب رمز می‌شوند.

۳-۲ الگوریتم‌های جستجو

در حل مسائل کاربردی نیاز به جستجو، امری غیر قابل اجتناب و در عین حال دشوار است. به همین جهت تعداد زیادی از الگوریتم‌های جستجو با فلسفه‌ها و دامنهٔ استفادهٔ متفاوت به‌وجود آمده‌اند. این الگوریتم‌های جستجو را می‌توان به دو دستهٔ کلی جستجوهای کامل و جستجوهای مکاشفه‌ای تقسیم کرد. تفاوت اساسی بین الگوریتم‌های این دو دسته به این صورت است که در جستجوهای کامل، تمام فضای جستجو به‌طور کامل مورد جستجو و ارزیابی قرار می‌گیرد تا جواب مورد نظر یافته شود؛ درحالی‌که در جستجوهای مکاشفه‌ای، تنها بخشی از فضا که احتمال یافتن جواب در آن بیشتر است، مورد توجه قرار می‌گیرد. جستجوهای مکاشفه‌ای به دو دسته الگوریتم‌های قطعی و غیرقطعی تقسیم می‌شوند. ویژگی اصلی الگوریتم‌های قطعی در این است که تحت شرایط یکسان، جواب‌های یکسان می‌دهند. نمونه‌ای از این الگوریتم‌ها را به تپه‌نوردی می‌توان اشاره کرد. ایراد اساسی این الگوریتم‌ها احتمال گیرافتادن در کمینه‌های محلی است. در مقابل، الگوریتم‌های مکاشفه‌ای غیرقطعی با استفاده از احتمالات و جستجوهای تصادفی، در شرایط یکسان، جواب‌های متفاوتی به‌دست می‌آورند. همین‌طور درصورت افتادن در کمینه‌های محلی، از آن‌ها می‌گریزند. الگوریتم‌های مکاشفه‌ای غیرقطعی براساس تعداد جواب‌هایی که در هر تکرار بررسی و ذخیره می‌کنند به دو دسته تقسیم می‌شوند. بعضی مانند ذوب فلزات تنها یک جواب را در هر تکرار مورد بررسی قرار داده و ذخیره و بعضی دیگر در هر تکرار، دسته‌ای از جواب‌ها را ذخیره می‌کنند، که به این الگوریتم‌ها، الگوریتم‌های مبتنی بر جمعیت می‌گویند [۴]. الگوریتم‌های مبتنی بر جمعیت نیز به الگوریتم‌های تکاملی (EA) و الگوریتم‌های مبتنی بر هوش جمعی تقسیم‌بندی می‌شوند. در این پژوهش از هر نمونه یک یا چند الگوریتم را انتخاب کرده و با هم از لحاظ عملکرد مقایسه خواهند شد تا بهترین نمونه برای مسئلهٔ مد نظر انتخاب شود. از الگوریتم‌های قطعی، الگوریتم تپه‌نوردی انتخاب شده است. از الگوریتم‌های غیرقطعی تک‌نقطه‌ای الگوریتم ذوب فلزات، از الگوریتم‌های تکاملی الگوریتم ژنتیک و از الگوریتم‌های هوش جمعی سه الگوریتم زنبور عسل، الگوریتم بهینه‌سازی پرتو ذرات و الگوریتم بهینه‌سازی پرتو ذرات مشارکتی انتخاب شده است. در ادامه ابتدا الگوریتم‌های جستجوی منتخب تشریح می‌شود.

۳-۲-۱ الگوریتم تپه‌نوردی(Hill claimate)

نحوهٔ عملکرد این الگوریتم بدین صورت است که ابتدا جوابی به شکل تصادفی برای مسألهٔ، تولید می‌شود؛ سپس در یک حلقه و تا زمانی که شرط توقف الگوریتم برقرار نشده است، به طور مکرر تعدادی همسایه برای حالت فعلی تولید و از میان حالت‌های همسایه، بهترین آن‌ها انتخاب شده و جایگزین حالت فعلی می‌شود. شبه‌کد این الگوریتم به‌صورت زیر است:

```
Generate a solution ( s' )
 Best = S'
 Loop
    S = Best
    S' = Neighbors (S)
    Best = SelectBest (S')
Until stop criterion satisfied
```

۳-۲-۲ الگوریتم ذوب فلزات (simulate Annealing)

الگوریتم ذوب فلزات یک الگوریتم جستجوی غیرقطعی است که نخستین‌بار در سال ۱۹۸۳ مطرح شد[۲۸] که ایدهٔ اصلی آن از عمل سردکردن تدریجی فلزات برای استحکام بیشتر آن‌ها سرچشمه گرفته است. در این روش همانند روش‌های تپه‌نوردی نیز مسئله از یک حالت مانند S در فضای حالت مسئله شروع کرده و با گذر از حالتی به حالت دیگر به جواب بهینه مسئله نزدیک می‌شود. انتخاب حالت شروع هم می‌تواند به‌صورت تصادفی انجام پذیرد و هم می‌تواند براساس یک قاعده، حالت اولیهٔ مسئله را انتخاب کند. روش کلی کار به این صورت است که در هر تکرار، الگوریتم ذوب فلزات حالت همسایه‌ای مانند 's ایجاد می‌کند و براساس یک احتمال، مسئله از حالت s به حالت 's می‌رود و یا اینکه در همان حالت s باقی می‌ماند. این روند تا زمانی تکرار می‌شود که به یک جواب تاحدودی بهینه برسد یا اینکه ماکزیمم تعداد تکرارها ا انجام شود.

شبه‌کد الگوریتم به صورت زیر است:

```
C = Choose an initial solution
T = Choose an initial temperature
  REPEAT
    S' = Generate a neighbor of the solution C
    ΔE = fitness( S' ) – fitness(C)
    IF (ΔE > 0 ) THEN // S' better than c
       C = S'
    ELSE with probability EXP( ΔE/ T )
       C = S'
    END IF
    T = lower the T using linear/ non-linear
    techniques
  UNTIL meet the stop criteria
```

در هر مرحله، اگر همسایهٔ تولیدشده بهتر از جواب فعلی باشد، پذیرفته می‌شود. درصورتی‌که حالت همسایه از حالت فعلی بدتر باشد، مقدار پارامتر T، تعیین‌کنندهٔ احتمال قبولی جواب است. در ابتدای امر، مقدار پارامتر T طوری انتخاب می‌شود که اکثر حالت‌های همسایه را مورد پذیرش قرار دهد، پارامتر T نشان‌گر دما بوده و مقدار این پارامتر به‌تدریج کاهش می‌یابد. مقدار پارامتر T باید طوری انتخاب شود که قبل از انجام گرفتن بیشینهٔ تعداد تکرارها، مقدار آن به‌طورتقریبی صفر شود.

۳-۲-۳ الگوریتم ژنتیک (Gentic algorithm)

الگوریتم ژنتیک، یکی از روش‌های اکتشافی در مسائل بهینه‌سازی است که ریشه آن از قانون بقای اصلح نشأت می‌گیرد و در واقع این الگوریتم یک شبیه‌سازی مجازی از نظریه تکامل تدریجی داروین است. این الگوریتم در هر تکرار محاسباتی (نسل) روی جمعیتی از کروموزوم‌ها عمل کرده و تغییرات تصادفی روی مجموعه کروموزوم‌ها از طریق اعمال عملگرهای ژنتیکی (جهش و ترکیب) انجام می‌دهد. پس از اعمال این عملگرها دنباله کروموزوم‌ها از نظر عملکرد بر اساس تابع هدف ارزیابی شده و انتخاب برای نسل‌های بعدی بر مبنای این ارزیابی انجام می‌شود. این الگوریتم دارای سه عمل‌گر مهم است که هر کدام به‌طور خلاصه توضیح داده می‌شود.

- **انتخاب والد**

عمل‌گر انتخاب، یک جفت از اعضای کنونی نسل را جهت شرکت در عملیات تکاملی ادغام و جهش انتخاب می‌کند. در این مرحله دو والد برای تولید مثل انتخاب می‌شوند که بایستی منجر به تولید فرزندانی تکامل یافته شوند. روش‌های انتخاب می‌توانند به شکل‌های مختلف اجرا شوند که با توجه به شرایط مسائل نتایج مختلفی را نیز می‌توانند ارائه دهند. در ادامه روش‌های انتخابی که به‌طورمعمول مورد استفاده قرار می‌گیرند به‌اختصار شرح داده می‌شود.

- **روش چرخ رولت**

انتخاب چرخ رولت[1] که نخستین‌بار توسط «هلند» پیشنهاد شد یکی از مناسب‌ترین انتخاب‌های تصادفی بوده که ایدهٔ آن، احتمال انتخاب است. احتمال انتخاب متناظر با هر کروموزوم، براساس برازندگیِ آن محاسبه می‌شود. اگر F_K مقدار برازندگی کروموزوم k ام باشد، احتمال بقای متناظر با آن کروموزوم عبارت است از:

$$P_k = \frac{f_k}{\sum_{i=1}^{n} f_i} \qquad (۳)$$

روش پیاده‌سازیِ چرخ رولت به این صورت است که یک دایره را در نظر گرفته و آن به تعداد کروموزوم‌ها، طوری تقسیم می‌شود که هر بخش متناظر با مقدار برازندگی

[1] roulette wheel

کروموزوم مربوط باشد؛ حال چرخ را چرخانده و هر کجا که چرخ متوقف شد به شاخص چرخ نگاه کرده، کروموزوم مربوط به آن بخش انتخاب می‌شود.

- **روش رقابتی یا تورنمنت**

در روش رقابتی[1] که شبیه رقابت در طبیعت است، یک زیرمجموعه کوچکی از کروموزوم‌ها (به‌طورمعمول دو یا سه عدد) به‌صورت تصادفی انتخاب شده و به رقابت می‌پردازند. سرانجام در این رقابت، براساس تابع ارزیابی یکی از آن‌ها به پیروزی رسیده و به‌عنوان والد جدید انتخاب شده و این فرآیند تا تولید همهٔ والدها در جمعیت جدید تکرار می‌شود.

- **روش بولتزمن**

ایدهٔ اصلی این روش از حرارت‌دادن فلزات سرچشمه گرفته است. در این روش ابتدا دمای بالایی را برای شروع، انتخاب کرده و رفته‌رفته از مقدار دما کاسته می‌شود. دمای بالا بدین معنی است که در ابتدای اجرای الگوریتم کرموزوم‌های بهینه و غیربهینه از شانس به‌طورتقریبی یکسانی برای انتخاب برخوردارند. این در حالی است که رفته‌رفته و با کاهش دما احتمال انتخاب کروموزوم‌های بهینه، افزایش و شانس انتخاب کروموزم‌های غیربهینه کاهش می‌یابد.

- **روش تصادفی**

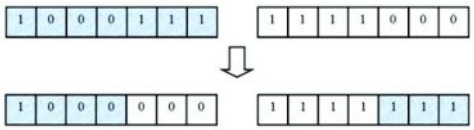
در این روش عمل انتخاب به‌طورکامل تصادفی صورت می‌گیرد و همهٔ کروموزوم‌ها از شانس یکسانی برای انتخاب برخوردارند.

- **عمل ترکیب**

ترکیب[2]، فرآیندی است که در آن نسل قدیمی کروموزوم‌ها با یکدیگر مخلوط و ترکیب می‌شوند تا نسل تازه‌ای از کروموزوم‌ها به‌وجود بیاید. درعمل ترکیب جفت‌هایی که در قسمت انتخاب، به عنوان والد انتخاب شدند، در این قسمت ژن‌هایشان را با هم مبادله می‌کنند و اعضایی جدید را به‌وجود می‌آورند. تاکنون روش‌های مختلفی برای عمل ترکیب استفاده شده‌اند که برخی از آنها را در ادامه شرح داده می‌شود.

- **ترکیب تک‌نقطه‌ای**

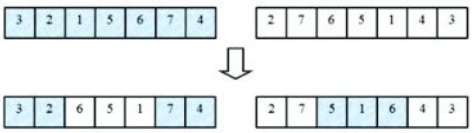
ترکیب تک‌نقطه‌ای دو کروموزوم را با انتخاب تصادفی موقعیتی مانند P، ترکیب می‌کند، P مقداری کمتر یا مساوی طول کزوموزوم‌هاست. اگر طول کزوموزوم‌ها N باشد از دو کروموزوم والد، دو فرزند بدین‌صورت به‌وجود می‌آید. یک فرزند با رونوشت‌کردن ژن‌های $(P_1\text{-}1)...1$ از کروموزوم والد نخست و ژن‌های N .. P از کروموزوم دوم، ساخته می‌شود و فرزند دیگر به‌طور مشابه، این‌بار با رونوشت‌کردن ژن‌های $(P_1\text{-}1)...1$ از والد دوم و ژن‌های N .. P از والد نخست، به‌وجود می‌آید. در این نوع ترکیب از دو والد، دو فرزند به‌وجود می‌آید. به‌عنوان مثال، این نوع ترکیب در شکل(۱) نشان داده شده است. در این مثال P=4 است.



(شکل ۱): ترکیب تک‌نقطه‌ای

- **ترکیب دونقطه‌ای**

در ترکیب دونقطه‌ای، دو موقعیت P_1 و P_2 به‌عنوان موقعیت‌های ترکیب، به‌طور تصادفی بین ۱ و طول کروموزوم‌ها (N) انتخاب می‌شود. روش ایجاد فرزندان مانند ترکیب تک‌نقطه‌ای است. فرزند نخست، ژن‌های $(P_1\text{-}1)...1$ را از والد نخست، ژن‌های $P_1...(P_2\text{-}1)$ را از والد دوم و ژن‌های $P_2...N$ را دوباره از والد نخست، به ارث می‌برد. فرزند دوم، ژن‌های $(P_1\text{-}1)...1$ را از والد دوم، ژن‌های $P_1...(P_2\text{-}1)$ را از والد نخست و ژن‌های $P_2...N$ را دوباره از والد دوم، به‌دست می‌آورد. در این روش ترکیب نیز، از یک جفت، دو فرزند به‌وجود می‌آید، در این روش احتمال اینکه والدها بدون تغییر به جمعیت بعد منتقل شوند، کمتر است. در شکل(۲) نمونه‌ای از این ترکیب با موقعیت‌های ترکیب $P_1=2$ و $P_2=5$ نشان داده شده است.

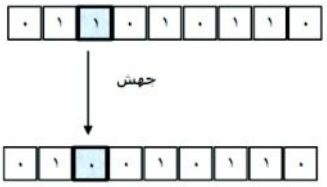


(شکل ۲): ترکیب دونقطه‌ای

- **عمل جهش**

جهش[3]، عمل‌گری است که بر روی یکی از کروموزوم‌های جمعیت عمل کرده و خصوصیات آن را در نقطهٔ جهش تغییر می‌دهد. نمونه‌ای از آن در شکل (۳) آمده است.

[1] tournament
[2] crossover
[3] Mutation



(شکل ۳): عمل جهش

جهش ژنی و جهش تصادفی دو راه حل اساسی برای جهش است. در روش جهش تصادفی، یک ژن به‌صورت تصادفی از کروموزوم‌های فرزند انتخاب شده و مقدار آن را براساس نوع کدگذاری‌ای که در آن استفاده شده، تغییر داده می‌شود. در روش تصادفی، نرخ جهش برابر تعداد کروموزوم‌هایی است که عمل جهش روی آن‌ها صورت می‌گیرد.

۳-۲-۴ الگوریتم بهینه‌سازی پرتو ذرات(PSO)

PSO یک تکنیک بهینه‌سازی مبتنی بر قواعد احتمال است که توسط دکتر راسل ابرهارت، دانشمند علوم رایانه و دکتر جیمزکندی روان‌شناس مسائل اجتماعی در سال ۱۹۹۵ ارائه شد و از رفتار اجتماعی پرندگان یا ماهی‌ها در پیداکردن غذا الهام گرفته شده است [۱۶]. پرندگان تنها با تنظیم حرکت فیزیکی خود و اجتناب از تصادم به دنبال غذا می‌گردند و به‌طور تئوری هر پرنده به‌عنوان یکی از اعضای گروه از تجربه قبلی خود و یافته‌های سایر اعضا برای یافتن غذا بهره می‌برد. این مشارکت یک مزیت قطعی برای یافتن غذا است. پایهٔ اصلی نظریه PSO همین تسهیم اطلاعات بین اعضا در یک گروه است.

در هر مرحله از حرکت جمعیت هر ذره با دو مقدار بهترین، به‌روز می‌شود؛ نخستین مقدار بهترین جواب از لحاظ شایستگی است که تاکنون برای هر ذره به‌طور جداگانه به‌دست آمده است و Pb نامیده می‌شود. مقدار بهترین دیگر، بهترین مقداری است که تاکنون توسط تمام ذره‌ها در میان جمعیت به‌دست آمده است. این مقدار بهترین کلی است و Pg نامیده می‌شود. مقادیرPb و Pg براساس یک تابع ارزیابی انتخاب می‌شود. این تابع برای مسائل مختلف متفاوت است. بعد از یافتن دو مقدار Pb و Pg هر ذره به‌صورت چندبعدی با دو مقدار Xid و Vid که به‌ترتیب معرف وضعیت مکانی و سرعت مربوط به بعد dام از iاُمین ذره است سرعت و مکان جدید خود را بروز می‌کند. این فرآیند با استفاده از رابطه‌های (۴) و (۵) زیر صورت می‌گیرد:

(۴)

$$Vi[] = wVi[t] + C1 \times rand()(Pb[t] - Xi[t]) + C2 \times rand()(Pg[t] - Xi[t])$$

(۵)

$$Xi[t+1] = Xi[t] + Vi[t+1]$$

در رابطه‌های بالا w وزن اینرسی، c1 و c2 عوامل یادگیری که به آن‌ها ضرایب شتاب نیز گفته می‌شود و rand() عددی تصادفی بین صفر و یک است. مقدار این پارامترها در حل مسائل مختلف متفاوت است. مقدار این پارامترها در هم‌گرایی مسئله بسیار موثر هستند. در[۱۷] مقدار بین ۰/۸ تا ۱/۲ برای w، در [۱۸] مقدار c1=c2=0.5 و در [۱۹] مقدار c1+c2≤4 پیشنهاد شده است. به هر حال انتخاب مقدار این پارامترها به نوع مسئله بستگی دارد.

۳-۲-۵ الگوریتم بهینه‌سازی پرتو ذرات مشارکتی(MPSO)

الگوریتم MPSO از تغییرات در الگوریتم PSO ایجاد شده است[۵]. در الگوریتم MPSO ذرات مشارکت‌کننده در فرآیند جستجو به چند گروه تقسیم می‌شوند. در این الگوریتم PSO استاندارد چنان تغییر کرده که سه نوع بهترین موقعیت در فضای چندگروهی در نظر گرفته شود، این کار می‌تواند سرعت رسیدن به هم‌گرایی را افزایش بخشیده و همچنین از هم‌گرایی زودرس جلوگیری کند. به‌علاوه به‌دلیل اینکه یک گروه با تعداد اجزای زیاد به تعدادی گروه کوچک‌تر با تعداد اجزای کمتر تبدیل می‌شود و جستجو در هر یک از گروه‌ها با وجود وابستگی به Pg به‌طور مستقل انجام می‌شود، لذا فضای جستجو با اکتشاف بیشتری مورد جستجو قرار می‌گیرد و امکان یافتن راه‌حل‌های بهتر افزایش می‌یابد. در هر تکرار الگوریتم، حرکت هر ذره با توجه به بهترین موقعیتی که خود آن ذره تاکنون داشته است(Pb)، بهترین موقعیت اجزای گروهی که ذره عضو آن است (Pg) و بهترین موقعیتی که تاکنون توسط کل اجزای گروه‌ها به‌وجود آمده است و درواقع بهترین Pg در کل جمعیت است(mbest) انجام می‌پذیرد. رابطه بردار سرعت اجزا در MPSO به‌صورت رابطه‌های(۶) تعریف می‌شود.

(۶)

$$Vi[t+1] = w\,Vi[t] + C1 \times rand()(Pb[t] - Xi[t]) + C2 \times rand()(Pg[t] - Xi[t]) + (C3 \times rand)(mbest[t] - Xi[t])$$

الگوریتم MPSO نیز پارامترهای مختلفی دارد که باید آن‌ها مشخص شود.

۳-۲-۶ الگوریتم کلونی زنبور عسل مصنوعی(ABC)

الگوریتم کلونی زنبور عسل مصنوعی، برای نخستین‌بار در سال ۲۰۰۵ توسط Karaboga معرفی شد [۲۰]. این الگوریتم از شبیه‌سازی رفتار زنبورهای عسل درطبیعت به دست آمده و یکی از روش‌های بهینه‌سازی مبتنی بر جمعیت است. در این روش اجتماع زنبورها به سه گروه زنبورهای کارگر، ناظر و زنبورهای مأمور اکتشاف تقسیم می‌شود. زنبورهای کارگر به‌صورت تصادفی به دنبال منابع غذایی می‌گردند و اطلاعات خود را به اشتراک می‌گذارند. در این میان، زنبورهای ناظر از بین این منابع غذایی، با توجه به تجربه و موقعیت خود، منبع غذایی مناسب را انتخاب می‌کنند؛ درحالی‌که زنبورهای مأمور اکتشاف، منابع غذایی را به‌طورکامل به‌صورت تصادفی و بدون درنظرگرفتن تجربه برمی‌گزینند. هر منبع غذایی انتخاب‌شده بیان‌گر یک جواب ممکن در حل مسأله است. میزان شهد موجود در منابع غذایی، بیان‌گر میزان برازندگی جواب مسأله است. تعداد زنبورهای کارگر مساوی با تعداد زنبورهای ناظر و برابر با تعداد جمعیت مسأله است.

در این الگوریتم ابتدا توسط زنبورهای کارگر به تعداد NS جواب اولیه به‌صورت تصادفی تولید می‌شود. NS بیان‌گر تعداد منابع غذایی و برابر با تعداد زنبورهای کارگر است. هر جواب$(xi_1, xi_2,....,xi_n) = Xi$ یک بردار n بعدی است. پس از تولید این جواب‌ها هر زنبور ناظر یک منبع غذایی را با احتمال pi انتخاب می‌کنند و pi طبق رابطه (۷) محاسبه می‌شود.

$$Pi = \frac{fit_i}{\sum_{j=1}^{NS} fit_j} \qquad (۷)$$

که در آن Fit_i برابر با تابع ارزیابی جواب iام است. هر زنبور ناظر سعی می‌کند براساس جواب انتخابی یک جواب جدید تولید کند. این جواب جدید طبق رابطه (۸) محاسبه می‌شود.

$$x_{ij}^{new} = x_{ij}^{old} + \lambda_{ij} (x_{ij}^{old} - x_{kj}^{old}) \quad \text{for } j = 1,2,...,n \qquad (۸)$$

در رابطهٔ بالا، k عددی است تصادفی که از بازهٔ [1,NS] است. λ_{ij} نیز عددی تصادفی است که از بازهٔ [-1,1] انتخاب می‌شود. xij مقدار بعد iام از جواب jام است. جواب جدید تولیدشده با جواب قدیمی مقایسه شده و اگر دارای کیفیت بهتری باشد، جایگزین آن خواهند شد؛ و این فرآیند توسط زنبورهای کارگر و ناظر تکرار می‌شود؛ و در هر مرحله بهترین جواب تولیدشده ذخیره می‌شود.

جواب‌هایی که پس از تکرارهای مشخص بهبود پیدا نکنند، متروکه اعلام و با منبع جدیدی توسط زنبورهای اکتشاف مطابق با رابطه(۹) جایگزین خواهد شد.

(۹)

$$x_{ij}^{new} = l_j + rand \times (u_j - l_j)$$

که در آن lj و uj حد پایین و بالای بعد jام و rand نیز عددی تصادفی بین صفر و یک است. شبه کد الگوریتم ABC به صورت زیر است:

```
Initialize the population of solutions
Evaluate the population
cycle = 1
repeat
   Employed bee phase
   Calculate probabilities for onlookers
   Onlooker bee phase
   Scout bee phase
   Memorize the best solution achieved so far
   cycle = cycle+1
until cycle = Maximum Cycle Number
```

۴- طراحی آزمایش‌ها

پارامترها و تابع ارزیابی در عملکرد الگوریتم‌های جستجو نقش مهمی را ایفا می‌کنند؛ لذا این عوامل به‌صورت جداگانه بررسی خواهند شد.

۴-۱- پارامترهای الگوریتم‌های جستجو

در این پژوهش هر کدام از الگوریتم‌های مطرح‌شده دارای پارامترهایی هستند که مقادیر آنها تأثیر زیادی در نتیجهٔ حاصل از جستجوی الگوریتم دارد. در این پژوهش، پارامترها برای هر الگوریتم به‌صورت زیر در نظر گرفته شده است.

الگوریتم ژنتیک: الگوریتم ژنتیکی با جمعیت ۲۰۰، نرخ ترکیب ۰٫۱، نرخ جهش ۰٫۱، روش انتخاب تصادفی، روش ترکیب دونقطه‌ای و روش جهش تصادفی تنظیم شده است.

الگوریتم بهینه‌سازی پرتـو ذرات: در ایـن پـژوهش از پارامترهای مطرح‌شـده در [۲] اسـتفاده شـده اسـت کـه: w=0.1 ، c1=1 ، c2=3 و انـدازهٔ جمعیـت نیـز ۶۴ در نظـر گرفته شده است.

الگوریتم بهینه‌سازی پرتـو ذرات مشـارکتی: در ایـن پژوهش از پارامترهای مطرح شـده در [۵] اسـتفاده شـده اسـت کـه w=0.1، c1=1 ، c2=0.5 ، c3=2.5 و تعـداد گروه‌ها ۴ و اندازهٔ جمعیت نیـز ۶۴ در نظـر گرفتـه شـده است.

الگوریتم زنبور عسل: تعداد اعضـای کلـونی ۶۴، مقـدار پارامتر limit=100 در نظر گرفته شده است.

۴-۲- تابع ارزیابی

هرکدام از الگوریتم‌های مطرح‌شده احتیـاج بـه یـک تـابع ارزیابی دارند تا کلیدهایی که تولید می‌کنند ارزیابی شـود و از بین آن‌ها بهترین راه انتخاب شـود. در ایـن پـژوهش برای ارزیابی کلیدهای تولیدشده توسـط هـر الگـوریتم از درصد تکرار حروف در متون استاندارد استفاده شده است. ایـن مقـادیر از محاسـبه درصـد تکـرار حـروف در متـون مختلف به‌دست آمده‌اند و در منابع مختلف قابل دسترسی هستند[۲۱, ۲۲]. با توجه به این مقادیر تابع ارزیابی مورد استفاده در این پژوهش به‌صورت رابطهٔ (۱۰) است:

(۱۰)

$$F = 1 - (\sum |sf[i] - sdf[i]| + \sum\sum |sdf[i,j] - ddf[i,j]|)/1000$$

که در آن sf[i] درصد تکرار حرف i ام در متـون اسـتاندارد اسـت و sdf[i] درصـد تکـرار حـرف i ام در مـتن رمزگشایی‌شده توسط کلید نامزد است. همچنـین sdf[i,j] و ddf[i,j] به‌ترتیـب درصـد تکـرار دو حرفـی‌هـا در متـون استاندارد و متن رمزگشایی شده است. مقادیر i و j از صفر تا ۲۵ تغییر می‌کنند. هر چقدر که مقدار این تابع بـه یـک نزدیکتر باشد، کلید نامزد انتخاب‌شده به کلیـد اصـلی کـه متن توسط آن رمز شده نزدیک‌تر است.

۴-۳- ابزار و محیط پیاده سازی

در این پژوهش جهت پیاده‌سازی از یک لپ‌تاب با پردازنده Corei5 و چهار گیگا بایت حافظه Ram استفاده شده است. برای نوشتن برنامه‌ها هم از زبان ویژوال بیسیک استفاده شده است.

۵ - نتایج و بحث

در این آزمایش‌هـا، الگـوریتم ویجینـر بـه‌عنـوان الگـوریتم رمزنگـاری در نظـر گرفتـه شـده اسـت و الگـوریتم‌هـای جستجوی مورد استفاده، الگوریتم ژنتیک، الگوریتم زنبـور عسـل، الگـوریتم بهینـه‌سـازی پرتـو ذرات، الگـوریتم بهینه‌سازی پرتو ذرات مشارکتی، الگـوریتم ذوب فلـزات و الگوریتم تپه‌نوردی است.

۵-۱ انجام آزمایش‌ها

در این پژوهش برای مقایسه الگوریتم‌های معرفی شده، عملکرد آن‌ها را در کشف کلید دو فایل مختلف که با کلیدهای متفاوتی رمز شده‌اند، بررسی و نتایج در چند جدول نشان داده شده است. از آنجا که این الگوریتم‌ها غیرقطعی هستند، در هر آزمایش هر کدام را ده مرحله اجرا کرده و میانگین نتایج حاصل از ده مرحله اجرا، برای هر الگوریتم در نظر گرفته شده است. فایل نخست از ۴۹۰ حرف و فایل دوم از ۱۸۵۰ حرف تشکیل شده است. در جدول (۲) نتایج حاصل از اجرای الگوریتم‌ها روی فایل نخست که با کلیدی به طول ۱۲ حرف رمز شده مشاهده می‌شود.

(جدول ۲): طول فایل ۴۹۰ حرف و طول کلید ۱۲

الگوریتم	بهترین نتیجه(تعداد حروف کشف شده از کلید)	تعداد کلیدهای آزمون شده برای رسیدن به بهترین نتیجه
الگوریتم ژنتیک	۱۲	۵۰۰۰
الگوریتم بهینه‌سازی پرتو ذرات	۸/۵	۶۳۰۰
الگوریتم زنبور عسل	۹/۳	۱۸۰۰۰
الگوریتم بهینه‌سازی پرتو ذرات مشارکتی	۱۰/۱	۱۸۴۰۰
الگوریتم ذوب فلزات	۱۰/۹	۱۳۵۰
الگوریتم تپه نوردی	۱۰/۹	۱۲۷۰

همان‌طور که در جدول (۲) مشاهده می‌شود، الگوریتم ژنتیک توانسته کلید رمز را به‌طور کامل کشف کند و برای

این کار پنج‌هزار کلید مختلف را آزمایش کرده است. از بین الگوریتم‌ها، الگوریتم تپه نوردی سریع‌ترین هم‌گرایی را دارد و با آزمایش ۱۲۷۰ کلید مختلف بهترین نتیجه خود را بدست آورده است. در جدول (۳) نتایج حاصل از اجرای الگوریتم‌ها روی فایل نخست که با کلیدی به‌طول دوازده حرف رمزشده مشاهده می‌شود.

(جدول ۳): طول فایل ۴۹۰ حرف و طول کلید ۱۹

الگوریتم	بهترین نتیجه(تعداد حروف کشف شده از کلید)	تعداد کلیدهای آزمون شده برای رسیدن به بهترین نتیجه
الگوریتم ژنتیک	۱۹	۱۱۶۰۰
الگوریتم بهینه‌سازی پرتو ذرات	۱۲/۵	۹۰۰۰
الگوریتم زنبور عسل	۱۳/۷	۲۴۷۰۰
الگوریتم بهینه‌سازی پرتو ذرات مشارکتی	۱۳	۲۳۷۰۰
الگوریتم ذوب فلزات	۱۷/۴	۴۴۰۰
الگوریتم تپه نوردی	۱۷/۳	۲۶۰۰

در این آزمایش، همان‌طور که در جدول (۳) مشاهده می‌شود، الگوریتم ژنتیک نیز بهترین نتیجه را حاصل می‌کند و الگوریتم تپه‌نوردی سریع‌ترین هم‌گرایی را دارد. در جدول (۴) نتایج حاصل از اجرای الگوریتم‌ها روی فایل نخست که با کلیدی به طول ۳۱ حرف رمزشده مشاهده می‌شود.

(جدول ۴) : طول فایل ۴۹۰ حرف و طول کلید ۳۱

الگوریتم	بهترین نتیجه(تعداد حروف کشف شده از کلید)	تعداد کلیدهای آزمون شده برای رسیدن به بهترین نتیجه
الگوریتم ژنتیک	۲۵/۶	۲۲۰۰۰
الگوریتم بهینه سازی پرتو ذرات	۱۳/۸	۱۵۳۰۰
الگوریتم زنبور عسل	۱۷/۵	۳۴۴۰۰
الگوریتم بهینه سازی پرتو ذرات مشارکتی	۱۵/۹	۳۸۰۰۰
الگوریتم ذوب فلزات	۲۵/۵	۶۸۰۰
الگوریتم تپه نوردی	۲۵/۱	۵۴۰۰

در این آزمایش همان‌طور که در جدول (۷) نمایان است، الگوریتم ژنتیک نیز بهترین نتیجه را حاصل می‌کند و الگوریتم تپه‌نوردی سریع‌ترین هم‌گرایی را دارد. در ادامهٔ پژوهش، آزمایش‌ها روی فایل دوم با اندازه ۱۸۵۰ بایت انجام می‌شود. در جدول (۵) نتایج حاصل از اجرای الگوریتم‌ها روی فایل دوم که با کلیدی به طول دوازده حرف رمزشده مشاهده می‌شود.

(جدول ۵): طول فایل ۱۸۵۰حرف و طول کلید ۱۲

الگوریتم	بهترین نتیجه(تعداد حروف کشف شده از کلید)	تعداد کلیدهای آزمون‌شده برای رسیدن به بهترین نتیجه
الگوریتم ژنتیک	۱۲	۴۵۰۰
الگوریتم بهینه سازی پرتو ذرات	۹/۵	۶۷۰۰
الگوریتم زنبور عسل	۱۰/۳	۱۸۹۰۰
الگوریتم بهینه سازی پرتو ذرات مشارکتی	۱۰/۹	۱۳۸۰۰
الگوریتم ذوب فلزات	۱۱/۱	۱۷۵۰
الگوریتم تپه نوردی	۱۰/۹	۱۱۵۰

در این آزمایش هم همان‌طور که مشاهده می‌شود، الگوریتم ژنتیک نیز بهترین نتیجه را حاصل می‌کند و الگوریتم تپه‌نوردی سریع‌ترین هم‌گرایی را دارد. در جدول (۶) نتایج حاصل از اجرای الگوریتم‌ها روی فایل دوم که با کلیدی به طول نوزده حرف رمزشده مشاهده می‌شود.

(جدول۶): طول فایل ۱۸۵۰ طول کلید ۱۹

الگوریتم	بهترین نتیجه(تعداد حروف کشف شده از کلید)	تعداد کلیدهای آزمون شده برای رسیدن به بهترین نتیجه
الگوریتم ژنتیک	۱۸/۵	۸۸۰۰
الگوریتم بهینه سازی پرتو ذرات	۱۴/۹	۸۲۰۰
الگوریتم زنبور عسل	۱۳/۳	۲۰۵۰۰
الگوریتم بهینه سازی پرتو ذرات مشارکتی	۱۶/۹	۲۱۳۰۰
الگوریتم ذوب فلزات	۱۸	۳۰۰۰
الگوریتم تپه نوردی	۱۷/۷	۲۳۰۰

در آزمایش۶ الگوریتم ژنتیک نیز بهترین نتیجه را حاصل می‌کند و الگوریتم تپه‌نوردی سریع‌ترین هم‌گرایی را دارد. در جدول (۷) نتایج حاصل از اجرای الگوریتم‌ها روی فایل دوم که با کلیدی به طول ۳۱ حرف رمزشده مشاهده می‌شود.

(جدول ۷) : طول فایل ۱۸۵۰ حرف و طول کلید ۳۱

الگوریتم	بهترین نتیجه(تعداد حروف کشف شده از کلید)	تعداد کلیدهای آزمون شده برای رسیدن به بهترین نتیجه
الگوریتم ژنتیک	۲۹/۸	۱۴۰۰۰
الگوریتم بهینه‌سازی پرتو ذرات	۱۹/۵	۱۱۵۰۰
الگوریتم زنبور عسل	۱۸/۶	۲۵۰۰۰
الگوریتم بهینه‌سازی پرتو ذرات مشارکتی	۲۰/۳	۲۴۸۰۰
الگوریتم ذوب فلزات	۲۹/۴	۵۷۵۰
الگوریتم تپه‌نوردی	۲۹/۱	۳۲۵۰

در آخرین آزمایش نیز همان‌طور که در جدول (۷) نمایان است، الگوریتم ژنتیک نیز بهترین نتیجه را حاصل می‌کند و الگوریتم تپه‌نوردی سریع‌ترین هم‌گرایی را دارد.

۵-۲ نتایج آزمایش‌ها

در مجموع نتایج زیر از انجام آزمایش‌ها به‌دست می‌آید:

- الگوریتم ژنتیک در همه آزمایش‌ها، دقیق‌ترین جواب را تولید کرده است. در سه آزمایش، کلید را به‌طور کامل کشف کرده است. بعد از الگوریتم ژنتیک، الگوریتم‌های ذوب فلزات و تپه‌نوردی بهترین دقت را دارند. به‌طور میانگین ترتیب الگوریتم‌ها از نظر دقت جواب تولیدشده به‌صورت جدول (۸) است.

(جدول ۸): ترتیب الگوریتم‌ها از نظر دقت جواب

رتبه	الگوریتم
۱	الگوریتم ژنتیک
۲	الگوریتم ذوب فلزات
۳	الگوریتم تپه‌نوردی
۴	الگوریتم بهینه‌سازی پرتو ذرات مشارکتی
۵	الگوریتم زنبور عسل
۶	الگوریتم بهینه‌سازی پرتو ذرات

- از نظر سرعت هم‌گرایی، الگوریتم تپه‌نوردی سریع‌تر از بقیه الگوریتم‌هاست. بعد از این الگوریتم، الگوریتم ذوب فلزات قرار دارد. به‌طور میانگین ترتیب الگوریتم‌ها از نظر سرعت هم‌گرایی در جدول (۹) آمده است.

(جدول ۹): ترتیب الگوریتم‌ها از نظر سرعت همگرایی

رتبه	الگوریتم
۱	الگوریتم تپه‌نوردی
۲	الگوریتم ذوب فلزات
۳	الگوریتم ژنتیک
۴	الگوریتم بهینه‌سازی پرتو ذرات
۵	الگوریتم زنبور عسل
۶	الگوریتم بهینه‌سازی پرتو ذرات

- الگوریتم‌های هوش جمعی، نسبت به بقیهٔ الگوریتم‌ها از نظر دقت، ضعیف‌تر عمل می‌کنند. از نظر سرعت هم‌گرایی نیز الگوریتم‌های هوش جمعی عملکرد ضعیف‌تری نسبت به بقیه دارند.
- الگوریتم‌های تک‌جوابی (تپه‌نوردی و ذوب فلزات) نسبت به بقیهٔ الگوریتم‌ها سریع‌ترین هم‌گرایی را دارند.
- الگوریتم ژنتیک می‌تواند بهترین گزینه برای شکستن الگوریتم رمزنگاری ویجینر باشد، زیرا دقیق‌ترین نتایج را در زمانی مناسب تولید می‌کند.

۶- نتیجه‌گیری

الگوریتم‌های جستجوی مختلف، عملکرد متفاوتی جهت حل مسائل مختلف دارند. در این پژوهش سعی بر آن شد که کارایی الگوریتم‌های جستجوی مختلف برای کشف کلید رمز در الگوریتم رمزنگاری ویجینر بررسی شود. از بین الگوریتم‌های مختلف، الگوریتم ژنتیک، دقیق‌ترین نتایج را تولید کرد و الگوریتم تپه‌نوردی سریع‌ترین هم‌گرایی را از خود نشان داد. از نتایج به‌دست آمده می‌توان نتیجه گرفت که الگوریتم ژنتیک بهترین گزینه برای کشف کلید رمز در الگوریتم رمزنگاری ویجینر می‌تواند باشد. با این روش در آینده الگوریتم‌های

پیچیده‌ای چون AES و DES مورد پـژوهش قـرار خواهـد گرفت.

۷- مراجع

[۱]ملکیان، ا.، ذاکرالحسینی، ع. " امنیت داده¬ها". ویرایش سوم، تهران، انتشارات علمی فرهنگی نص، ۱۳۹۰.

[۲]م. احمدی و ب. اسدی, "شکستن الگوریتم رمزنگاری ویجینر با استفاده از الگورتم بهینه‌سازی گروه ذرات" ، همایش منطقه‌ای علوم کامپیوتر, دانشگاه آزاد درود, ۱۳۹۱.

[۳]ب. اسدی و م. احمدی, "کشف کلید رمز در الگوریتم رمزنگاری ویجینر با استفاده از الگوریتم بهینه‌سازی اجتماع ذرات مشارکتی (MPSO)، "دومین کنفرانس ملی مهندسی نرم افزار, دانشگاه آزاد لاهیجان، ۱۳۹۱.

[۴]ر. رستگار و م. میبدی, "یک الگوریتم تکاملی تخمین توزیع جدید با استفاده ازاتوماتای یادگیر." نشریه مهندسی برق و مهندسی کامپیوتر ایران،۱۳۸۳، شماره دوم، صفحات ۷۳-۸۲ .

[۵]س. هدیه، ح. ثامتی و ح. بیگی." الگوریتمی جهت یافتن بهینه سراسری در مسائل: MPSO"، سیزدهمین کنفرانس انجمن کامپیوتر ایران، ۱۳۸۶.

[6] ishith, S., and Kishore, B. "Improving Security of Vigenere Cipher by Double Columnar Transposition", International Journal of Computer Applications, 2014, Vol 100, No. 14, pp.6-10 .

[7] Stallings, W. "Cryptography and Network Security - Principles and Practice". Fifth edition, Pearson Education, Inc, 2011.

[8] Sangapu, V.A. and Gomatam V.S.A, "Recent Advancements on Symmetric Cryptography Techniques - A Comprehensive Case Study", Global Journal of Computer Science and Technology, 2014, Vol 14, Issue 2, pp 19-30 .

[9] Aliyu, A.M, Olaniyan, A.," Vigenere Cipher: Trends, Review and Possible Modifications", International Journal of Computer Applications (0975 – 8887), 2016, Vol.35, No.11.

[10] Bhateja, A., Kumar, Sh., " Genetic Algorithm with elitism for cryptanalysis of Vigenere cipher", Internationai Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 .

[11] Bhateja, A., Kumar, Sh., Bhateja, A.K.," Cryptanalysis of Vigenere Cipher using Particle Swarm Optimization with Markov chain random walk", International Journal on Computer Science and Engineering (IJCSE), 2013.

[12] Gopalakrishnan, V., et al., "Cryptanalysis of vigenere cipher using genetic algorithm and dictionary analysis" presented at the IASTED International Conference on Technology for Education, 2009.

[13] Ragheb, T. and Subbanagounder, A. "Applying Genetic Algorithms for Searching Key-Space of Polyalphabetic Substitution Ciphers," The International Arab Journal of Information Technology, 2008, vol.5, pp. 87-91.

[14] Sivagurunathan,G. and Purusothaman, T. "Reduction of Key Search Space of Vigenere Cipher Using Particle Swarm Optimization" Journal of Computer Science, 2011, vol. 7, pp.1633-1638.

[15] Moradi, M., Khotanlou, H. & Abbasi, M."Breaking Of Simplified-Data Encryption Standard Using Optimized SPSO", The Scientific Journal Of Advanced Defence Science And Technology, 2014, vol.3, pp.203-210.

[16] Kennedy, J. and Eberhart, R. "Particle Swarm Optimization" presented at the IEEE International Conference on Neural Networks, 1995.

[17] Shi, Y. and Eberhart, R. "A Modified Particle Swarm Optimize" presented at the IEEE Conference on Evolutionary Computation, 1998.

[18] Kumar, A. and Zhang, D. "Palm Print Authentication Using Multiple Represent-ation" Pattern Recognition Journal, 2005, vol. 38, pp.1695-1704.

[19] Carlisle, A. and Dozier, G. "An off-the-Shelf PSO," presented at the Particle Swarm Optimization Workshop, 2001.

[20] Karaboga, D. and Basturk, B. "An Idea Based on Honey Bee Swarm for Numerical Optimization" Erciyes University Technical Report-TR06, 2005.

[21] Jones, M. N. And. Mewhort, D. J. K "Case-sensitive letter and bigram frequency counts from large-scale English corpora" Behavior Research Methods, 2004, vol.36, pp.388-396.

[22] Solso, R. L. And Barbuto, P. F. "Bigram and trigram frequencies and versatilities in the English language" Behavior Research Methods & Instrumentation, 1979, vol.11, pp.475-484.

[23] Garg, P. "A Comparison Between Memetic Algorithm And Genetic Algorithm For The Of Cryptanalysis Simplified Data Encryption Standard Algorithm", International Journal of Network Security & Its Applications, 2009, vol.1, pp.34-42.

[24] Husein, H.M.H., Bayoumi, B.I. Holail, F. S. B., Hasan, E.M., El-Mageed, M.Z.A. "A Gene-tic Algorithm For Cryptanalysis Of DES-8", International Journal of Network Security, 2007, vol.5, pp.213–219.

[25] Salabat, K., Armughan, A., Yahya, D. M. " Ant-Crypto, A Cryptographer For Data Encryption Standard", International Journal of Computer Science Issues, 2013, vol.10, pp.400-406.

[26] Sathya, S. S., Chithralekha, T., Anandakumar, P. "Nomadic Genetic Algorithm For Cryptanalysis Of DES 16"; International Journal of Computer Theory and Engineering, 2010, vol.2, pp.411-415.

[27] Laskari, E.C., Meletiou, G.C.; Stamatiou, Y.C.,Vrahatis, M.N. "Applying Evolutionary Computation Methodsfor The Cryptanalysis Of Feistel Ciphers",Applied Mathematics and Computation, 2007, vol.184, pp.63-72.

[28] Kirkpatrick, S. et al., "Optimization by simulated annealing",1983, Science 220, pp.671-680.

مهدی احمدی پری تحصیلات خود را در مقاطع تحصیلی کارشناسی و کارشناسی ارشد رشته مهندسی کامپیوتر گرایش نرم‌افزار به‌ترتیب در سال ۱۳۸۵ در دانشگاه اصفهان و سال ۱۳۹۰ در دانشگاه آزاد اسلامی واحد اراک به پایان رساند و هم‌اکنون در حال تدریس در دانشگاه ملایر است. از زمینه‌های مورد علاقهٔ وی می‌توان الگوریتم‌های رمزنگاری، شبکه‌های رایانه‌ای، امنیت شبکه‌های رایانه‌ای و الگوریتم‌های هوش جمعی اشاره کرد.

میثم مرادی تحصیلات خود را در مقاطع تحصیلی کاردانی و کارشناسی در رشتهٔ مهندسی کامپیوتر گرایش نرم‌افزار به‌ترتیب در سال ۱۳۸۴ در دانشگاه فنی بروجرد و سال ۱۳۸۶ در دانشگاه آزاد اسلامی واحد ملایر به پایان رساند و مقطع کارشناسی ارشد خود را در سال ۱۳۹۲ در دانشگاه علوم تحقیقات تهران(همدان) در رشته مهندسی کامپیوتر گرایش نرم‌افزار به پایان رساند و هم‌اکنون در حال تدریس در دانشگاه ملایر است. از زمینه‌های مورد علاقهٔ وی می‌توان به امنیت داده‌ها، امنیت شبکه‌های رایانه‌ای، مهندسی نرم‌افزار، اختراع و نوآوری اشاره کرد.