وزارة التعليم العالي والبحث العلمي
جامعة ديالى
كلية التربية الاساسية
قسم الحاسبات

مشروع بحث بعنوان :

استخدام الخوارزميات الجينية في ذكاء اصطناعي وتطبيقاتها

بحث متقدم الى كلية التربية الاساسية جامعة ديالى قسم الحاسبات وهو جزء من متطلبات نيل شهادة البكالوريوس

الطالبتان
شيلان ابراهيم خليل هدى عامر طارق

بأشراف : م . م ياسر علي

للفصل الدراسي ٢٠١٧ – ٢٠١٨

بسم الله الرحمن الرحيم

[قَالُوا سُبْحانَكَ لَا عَلمَ لنا إلَّا مَا عَلَّمتَنَا إنَّكَ أنتَ العَليم الحَكيمُ]

صدق الله العظيم

(البقرة : ٣٢)

ب

الإهـــداء

- إلهي لا يطيب الليل الا بشكرك ، ولا يطيب النهار الا بطاعتك ، ولا تطيب اللحظات الا بذكرك ، ولا تطيب الآخرة الا بعفوك ، ولا تطيب الجنة الا برؤيتك الى ربي وموجدي وخالقي

- الى من بلغ الرسالة ، وأدى الأمانة ، ونصح الأمة ، الى نبي الرحمة ونور العالمين ... الى سيدنا محمد (صلى الله عليه واله وسلم)

- الى من تعهداني بالتربية في الصغر ، وكانا اي نبراساً يُضيئ فكري بالنصح ، والتوجيه في الكبر..... الى امي ، وابي حفظهما الله

- الى من شملوني بالعطف ، وأمدوني بالعون ، وحفزوني بالتقدم ... الى اخوتي واخواتي ... رعاهم الله

- الى كل من علمني حرفاً ، واخذ بيدي في السبيل تحصيل العلم والمعرفة

اليهم جميعاً أهدي ثمرة جهدي ، ونتاج بحثي المتواضع ...

ت

الشكـــر والتقـــدير

الحمد الله الذي مَنَّ بنعمته عينا فتمَّ إنجاز هذا البحث

لا بدّ لي ان اتقدم بالشكر والتقدير الى مشرف بحثي الأستاذ (ياسر علي)

كما أتوجه بخالص الشكر إلى كل من ساندني خلال فترة كتابة بحثي وهم كُثر ويعجز القلم عن ذكرهم ، ولكني أرفع كفّي لهم وادعو الله تعالى أن يحفظهم ويسدّد خطاهم .

الخلاصة :-
خلاصة هذا البحث لمعرفة كيفية استخدام الخوارزميات الجينية في الذكاء الاصطناعي وتطبيقاته حيث ذكرنا تعريف الخوارزميات الجينية التي هي عبارة عن تقنية في مجال الذكاء الاصطناعي وحل المشاكل اذ يعتمد أيجاد افضل الحلول باعتماد على العشوائية في البحث تستخدم الخوارزميات الجينية في عدة تطبيقات وكذلك تم ذكر تاريخ الخوارزميات الجينية التي تسمى هذا الايام بالخوارزميات التطورية , وتم التطرق في الى استخدام الخوارزميات الجينية في مطابقة النصوص وهي من المواضيع المهمة في مجال الخوارزميات حيث الهدف منه هو البحث عن النص داخل مجموعة كبيرة او يمكن البحث عن نمط قريب لنمط المراد البحث عنه او مشابه له حيث موضوع مطابقة النصوص لها خاصية البحث والاستبدال كذلك يستخدم الخوارزميات في برامج الأنتي فايروس . وفي مطابقة النصوص يمكن دمج هذه الخوارزميات مع خوارزميات اخرى لتكوين مشاريع رائعة .
الكلمات المفتاحية :-
الذكاء الاصطناعي, الخوارزميات الجينية.

قائمة المحتويات

ت	المواضيع	الصفحات
	الواجهة	أ
	الآية	ب
	الاهداء	ت
	الشكر والتقدير	ث
	الخلاصة	ج
	١ ـ مشكلة البحث ٢ـ اهداف البحث ٣ ـ اهمية البحث ٤ـ حدود البحث ٥ ـ الجدول الزمني	ح
١	مقدمة البحث	١ـ٢
	الفصل الاول	٣
١ـ١	تعريف الخوارزميات الجينية	٤
٢ـ١	المصطلحات الأساسية في الخوارزمية الجينية	٥
٣ـ١	مجالات تطبيق الخوارزميات الجينية	٦ـ٧
	الفصل الثاني	٨
٢	طريقة عمل الخوارزمية الوراثية	٩
١ـ٢	تاريخ الخوارزميات الوراثية	١٠
	الفصل الثالث	١١
٣	خوارزميات مطابقة النصوص PATTERN MATCHING ALGORITHM	١٢
١ـ٣	خوارزميات المطابقة مهمه للغاية لأي مبرمج	١٣
٢ـ٣	نوع أخر من الخوارزميات تستخدمه برامج الأنتي فايروس	١٣
٣ـ٣	٣ـ٣ الخوارزمية NAIVE SEARCHING ALGORITHM	١٤
٤ـ٣	كود سي++ لتطبيق خوارزمية NAÏVE SEARCH	١٥
٥ـ٣	كود جافا لتطبيق خوارزمية NAÏVE SEARCH	١٦
٦ـ٣	الخوارزمية KNUTH – MORRIS – PRATT STRING MATCHING	١٧
٧ـ٣	الخوارزمية BOYER-MOORE ALGORITHM	١٨ـ١٩
٨ـ٣	التوصيات	٢٠
٩ـ٣	الاستنتاج	٢٠
١٠ـ٣	الخاتمة	٢٠
	المصادر	٢٠

١ ـ مشكلة البحث :-

مع التقدم العلني الهائل والتسارع الذي تشهده التكنلوجيا في الوقت الراهن ظهر أنماط اخرى جديدة من الانظمة سميت بالأنظمة الذكية وبالتالي فتح المجال امام العالم نحو نوع جديد من التكنلوجيا اطلق عليه اسم (تكنلوجيا الذكاء الاصطناعي) التي سرعان ما طورت واستخدمت في عدد من التطبيقات ومن هنا جاءت مشكلة عدم التعرف من قبل المطلعين على ما هية هذه التكنولوجيا.

٢ – أهداف البحث

هو مساعدة الاشخاص لمعرفة ما هيه الخوارزميات الجينية وكيفية استخدامها في الذكاء الاصطناعي وتطبيقاته ومن هذا التطبيقات التي تستخدم فيها الخوارزميات مطابقة النصوص والتي هي من المجالات المهمة للمبرمج ويتم تطبيقه بواسطة خطوات علميه مدروسة وكل هذا يتم عن طريق استخدام الخوارزميات التي تقوم بحل مشكلة مطابقة النصوص وهناك برامج أنتي فايروس تستخدم نوع اخر من الخوارزميات.

٣ – اهمية البحث

يأتي أهمية البحث من الناحية العلمية التطبيقية في الواقع العملي وتطبيقه في عمليات الحاسوب وفي كل المجالات وبالاخص اكتشاف الفايروسات باستخدام هذه الطرق حيث يتم تقديم افضل البرامج مطابقة النصوص ومعرفة كيفية استخدام الخوارزميات في الذكاء الاصطناعي وتطبيقاته وغيرها .

٤ – حدود البحث

أ ـ حدود مكانية

جامعة ديالى\ كلية التربية الاساسية.

ب ـ حدود زمانية

العام الدراسي ٢٠١٧-٢٠١٨ .

٥ـ منهجية البحث

الوصفي .

٦ـ أدوات جمع البيانات

المكتبة , الانترنت.

الفصل الأول

١ ـ المقدمة

مع التقدم العلني الهائل والتسارع الذي تشهده التكنلوجيا في الوقت الراهن ظهر انماط اخرى جديده من الانظمه سميت بالانظمه الذكية وبالتالي فتح مجال امام العالم نحو نوع جديد من التكنولوجيا اطلق عليها اسم (تكنولوجيا الذكاء الاصطناعي) التي سرعان ما طورت واستخدمت في العديد من التطبيقات ومن هنا جاءت الخورزميات الجينيه كأليه تعامل مع كائن ما وجعله يتمتع بالذكاء .

ان اختراع تقنيات الذكاء الاصطناعي فتح بوابات جديدة وشكل قفزه نوعيه في علوم الحاسوب كيف يمكننا تعريف الذكاء الاصطناعي ان كنا حتى الان لا نتفق على تعريف واضح لذكاء البشري لذلك تتنوع تعاريف الذكاء الاصطناعي وتتعدد لكن مهما اختلفت التعاريف فان مضمونه واحد وجوهره وخبره والمعرفة والذاكرة فكل تطبيقاته تتمحور حول الامثلة ومعرفة الحل الانسب للمشاكل المطروحه بطريقه فعاله الامثله هي عمليه معالجه مسأله بطريقه مظبوطه بهدف توصل الى افظل نتائج ممكنه وانطلاق من اهميتها في حل المشاكل تعد الامثله من اهم فروع علم الحاسوب والهندسه والعديد من الانظمه الاخرى والتي تعددت وسائل تطبيقها وتطورت مع الزمن وصولا الى تقنيات التطور والخوارزميات الجينيه اللتان اصبحتا الان معا تشكلان مايسمى الحوسبه التطوريه في بدايه العقد السادس من القرن العشرين (١٩٦٢م)تم اختراع الخوارزميات الجينيه من قبل (جون هولاند) هو وزملائه في الولايات المتحدة الامريكية التي تم تطويرها في جامعه ميتشيغان في عام (١٩٧٥). تعد الخوارزميات الجينيه تمثيل للمعتقد السائد بان ذكاء البشري يخلق مع الانسان ويتم اكتسابه عن طريق الوراثه بشكل كبير فهي محاكاة لعمليه التزاوج بين الكائنات الحيه من نفس النوع وقد استخدمت لها العديد من مصطلحات علم الوراثه مثل:
الجيل والوالدين والعبور والطفرة ....الخ.
تحاول الخوارزميات الجينيه الوصول الى الحل الانسب لمشكله ما وذلك اعتماد على مبدا العالم (دارون)في الاصطفاء الطبيعي القائم على الاحتفاظ
بالميزات والصفات الجيدة الموجودة في الجيل الاباء ونقل الى جيل ابناء بهدف حصول على ذريه قويه تتمع بافظل الصفات الجيل السلف على اقل تقدير(البقاء للاقوى). جميع الكائنات الحيه تتالف من خلايا تحتوي العدد نفسه من سلاسل DNAالتي تسمى الكروموسومات مما يضفي طابعا شخصيا للكائن. كل كروموسوم يمكن تقسيمه الى جينات المكونه له والتي تعطي الكائن سمه محدده. لون العين فعندها يختلف ترتيب الجينات وتختلف السمه التي تعطيها مما يؤثر على الكائن ككل هناك ترابط وثيق بين معلومات السابقه والخوارزميات الجينيه التي تعمل بنفس المبدا انقسام الخليه وتكاثر خلايا الحيه حيث تشير الكروموسومات الى الصفات المحتملة بحيث تكون مجموعه احتمالات كل بت اما (٠،١) وبناء عليه اتخذ قرار فيما يتعلق بجينات الجل التالي وصولا الى الحل الانسب. وبعد ان تبلورت فكرت الخوارزميات الجينيه بدات تطبيقاته تتوسع في منتصف العقد الثامن من القرن العشرين حيث شملت نطاق واسع م المواضيع ففي عام ١٩٩٢ استخدم العالم (جون كوزا)الخوارزميات الجينيه الي حيز البرمجه واطلق عليه البرمجه الجينيه.

١-١ تعريف الخوارزمية الجينية

هي تقنية بحث، تستخدم في مجال الذكاء الاصطناعي (artificial intelligence) وتحديدًا في فرع البحث وحل المشاكل (Problem solving and search) اذ تقوم بإيجاد أفضل الحلول التحسين الحلول للمشاكل (optimization problems) بالاعتماد على العشوائية في البحث . كما تستخدم الخوارزمية الجينية في التطبيقات المعلوماتية الإحيائية (bioinformatics) و علوم الحاسوب والهندسة و الاقتصاد و الكيمياء و الصناعات التحويلية (manufacturing) و الرياضيات والفيزياء وغيرها من الميادين.

١-٢ المصطلحات الأساسية في الخوارزمية الجينية

يوضح الجدول (1) قسما من المصطلحات الأساسية لعلم الوراثة وما يقابلها في الخوارزمية الجينية.

ألجين(Gene)	**القيمة(Value) تمثل الوحدة الأساسية في هيكلية بيانات الخوارزمية الجينية التي قد تكون قيمة عددية أو حرفية أو ثنائية.**
الكروموسوم (Chromosome)	**المقطع(String): هومجموعة من القيم (Gene) الأساسية التي تمتلك مجموعة من الحلول لتطبيق معين.**
الفرد(Individual)	**يمثل الكروموسوم.**
حجم المجتمع (population size)	**عدد الأفراد في الجيل(عدد الكروموسومات في الجيل).**
الجيل(Generation)	**يمثل مجموعة من الأفراد المتكونة في وقت معين.**
التعبير(phenotype)	**تمثل فضاء الحلول للأفراد، يترجم قيم الكروموسومات.**
الأليل(Genotype)	**يمثل التمثيل الجيني للأفراد.**

١-٣ مجالات تطبيق الخوارزميات الجينية

١- مسائل الأمثلة بشكل عام:
بما فيها الأمثلة العددية, والحسابية مثل مسألة البائع المتجول التصميم الصناعي Tsp,مثل مسألة آلة تقطيع الخشب, جدولة أعمال التسوق , أمثلة جودة الصوت والفيديو
٢ -البرمجة الأوتوماتيكية
حيث تم استخدام الخوارزميات الجينية لتطوير برامج حاسوبية بهدف تنفيذ مهام محددة, ولتصميم بني حاسوبية أخرى ,مثل شبكات الفرز sorting network .
٣ -تعليم الروبوتات والآلات

تم استخدام الخوارزميات الجينية في كثير من تطبيقات التعلم التلقائي machine –learning ,ومن ضمنها التصنيفclassification و التنبأPrediction .. وقد تم استخدام الخوارزميات الجينية في تصميم الشبكات العصبونية neural networks design

٤- النماذج الاقتصادية Economic models

تم استخدام الخوارزميات الجينية لنمذجة آليات ابتكار وتطوير استراتيجيات المزايدة. .وفي مجال نشوء الأسواق الاقتصادية emergence of economic markets.

٥- التفاعل بين التطور والتعلم

حيث تم استخدامها لدراسية التأثير المتبادل بين تعلم الأفراد وتطور الأنواع.

٦- كنماذج للأنظمة الاجتماعية:

تم استخدامها لدراسة جوانب تطور النظم الاجتماعية, مثل تطور التعاون evolution of cooperation و تتطور الاتصال ,evolution of communication سلوك القافلة لدى النمل trail-following behavior . in ants والكثير الكثير من المجالات التي استخدمت فيها الخوارزميات الجينية.

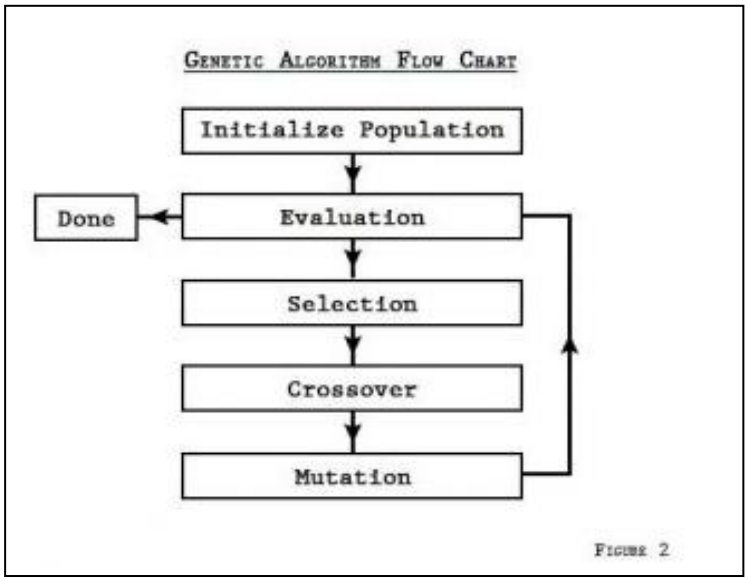
مما سبق نلاحظ بأن الخوارزميات الجينية ما هي إلا طريقة ذكية لحل طيف واسع من المسائل بعضها لم يجد له حلا ممكنا أو فعالا بالطرق التقليدية ,ولكننا بالوقت نفسه لا نستطيع القول بأنها طريقة ,إيجابية بشكل مطلق في الحل, فهي مثلها مثل بقية الخوارزميات ,لها ايجابياتها وسلبياتها إيجابياتها من حيث كونها سهلة النتجيز, وأقل عرضة لأن تقع في نهاية محلية, ومن سلبياتها كونها مكلفة حسابيا بشكل عام, وأبطأ من بعض الطرائق الأخرى, وعلى كل الأحوال , فإنه مع الامكانيات الرهيبة للحواسيب الحالية , فإن السلبيات السابقة لم تعد بتلك الأهمية الكبيرة.

وفي النهاية لا يسعنا إلا القول بأن المعلومات السابقة عن الخوارزميات الجينية ماهي إلا قطرة من فيض هائل لا يزال يتطور كل يوم, ولا يمكننا الاحاطة بها كلها في ورقة بحث واحد مقالة واحدة. وهناك اكثر من تطبيقات .

الفصل الثاني

٢ ـ طريقة عمل الخوارزمية الوراثية

طريقة عمل الخوارزميات الوراثية مستوحاة من العمليات التي تحدث أثناء عملية التطور في الكائنات الحية، وهذه العمليات هي الانتخاب الطبيعي (Natural Selection) والوراثة (Inheritance) والخلط (Crossover) والتبديل أو الطفرة (Mutation)، والخوارزميات الوراثية هي عبارة عن نظام برمجي حيث هناك مجموعة من الحلول (Population) لمشكلة معينة، تتطور وتتقدم لتصبح أفضل. تبدأ عملية التطور من مجموعة عشوائية من الأفراد (الحلول) وتحدث عملية التطور على مستوى أجيال (generations)، في كل جيل يتم حساب كفاءة كل فرد في هذا الجيل، ثم يتم اختيار مجموعة من الأفراد من الجيل الحالي بناءً على كفاءتهم، هؤلاء الأفراد يتم تعديلهم وإحداث الطفرة فيهم وخلطهم لإنتاج جيل جديد والذي يتم تكرار نفس العمليات السابق ذكرها عليه وهكذا. عادة يكون شكل كل حل من الحلول عبارة عن سلسلة مكونة من رقمين الصفر والواحد، مثال: (١٠١٠١١٠٠)، وتسمى هذه السلسلة كروموسوم (Chromosome) ويسمى كل رقم في السلسلة جين (Gene)، ومع ذلك يمكن أن يتم تحويله إلى شكل مختلف على حسب نوع المشكلة التي تقوم الخوارزمية بحلها. وعملية الخلط (Crossover) تكون عبارة عن أخذ حلين من الحلول وخلطهم لإنتاج حلين مختلفين، مثال: لنفرض أن لدينا حل هو (١٠١٠١١٠٠) ولدينا حل آخر هو (١١١١٠٠٠٠)، تتم عملية الخلط عن طريق أخذ النصف الأول من الحل الأول وهو (١١٠٠)، وأخذ النصف الثاني من الحل الثاني وهو (١١١١)، ثم إنتاج حل جديد مكون من هذين النصفين وهو (١١١١١١٠٠)، وبهذا الشكل يحتوي الكروموسوم للحل الجديد علي جينات من كلا الكروموسومين للحلين القديمين. أما عملية التبديل أو الطفرة فتتم عن طريق تبديل أحد الأرقام الموجودة في الحل (تبديل أحد الجينات الموجودة في الكروموسوم)، مثال: لنفرض أن لدينا حل وهو (١٠١١١٠٠١٠)، فإن عملية الطفرة ممكن أن تُبدل أول رقم من اليمين وهو الصفر وتحوله إلى واحد، فيصبح شكل الحل الجديد كالتالي (١٠١١١٠٠١١).



٢-١ تاريخ الخوارزميات الوراثية

بدأ العمل على ما يسمى هذه الأيام بالخوارزميات التطورية (Evolutionary Algorithms) في ستينيات القرن الماضي، وتحديدًا في الولايات المتحدة وأوربا. كان جون هولاند (John Holland) وزملائه في جامعة ميتشيجان (Michigan) مهتمين بأنظمة الذكاء الاصطناعي القادرة على التأقلم تحت الظروف البيئية المتغيرة. وكانت هذه هي الفكرة، أنه لكي تتمكن مجموعة من الأفراد من التأقلم في بيئة معينة، فإنها يجب أن تَعمل كما يعمل النظام الطبيعي، حيث يتم حذف الحلول عديمي الفائدة ومكافئة الحلول ذات الفائدة. وكانت رؤية جون هولاند هي بناء خوارزميات وراثية تمتلك خصائص التطور الطبيعي، وتحويل هذه الخصائص إلى شكل يمكن معالجته رياضيًا، ثم استخدام تلك الخوارزميات وتطبيقها في حل الكثير من المشكلات الموجودة في الواقع.

الفصل الثالث

٣- خوارزميات مطابقة النصوص Pattern Matching Algorithm

موضوع مطابقة النصوص أو البحث في النصوص من المواضيع المهمه في مجال الخوارزميات ، حيث الهدف هو البحث عن نص ما سوف نطلق عليه Pattern داخل مجموعه كبيره من النصوص Text - ويمكن أن يكون البحث عن النمط بالضبط exact او عن أي نمط قريب للنمط المراد البحث عنه ، وبما أن خوارزميات البحث المتسلسل Linear Search والبحث الثنائي Binary Search تستخدم في البحث عن "مفتاح واحد" داخل مجموعه كبيره من النصوص(أو الأرقام) فإنها لا تصلح استخدامها في حالتنا هذه ، فنحن نريد البحث عن نمط معين (مجموعه من المفاتيح أو الحروف) داخل النص الكبير وليس مفتاح واحد فقط وأحيانا نريد البحث عن أي نمط أخر مشابه.

٣-١- خوارزميات المطابقة مهمه للغاية لأي مبرمج

حيث أن تطبيقات موضوع بحث النصوص كبيره ومتنوعه، مثلا أغلب برامج محررات النصوص توجد فيها خاصيه بحث وأستبدال Find and Replace هل تسألت يوما كيف تجري هذه العمليه وما هي الخوارزميه المستخدمه ؟ -أغلب البرامج تستخدم خوارزميه Boyer-Moore algorithm – حيث تقوم هذه الخوارزمية بالبحث عن جميع الكلمات في الملف لكي تستبدل النص الجديد بجميع النصوص المشابه للنمط المراد البحث عنه. هذه الخوارزمية Boyer-Moore وغيرها من الخوارزميات التي تبحث عن النص المشابه تسمى Exact String Searching وهناك حوالى ٢٠ خوارزميه أو أكثر.

٣-٢ نوع أخر من الخوارزميات تستخدمه برامج الأنتي فايروس

حيث تقوم بالبحث عن التوقيع في قاعده بيانات تحتوي على الألاف التواقيع في مده صغيره جدا قد لا تتجاوز ثانيه واحده وتحتاج الى خوارزميات للبحث عن أكثر من نمط في المره الواحده Multiple pattern searching مثل خوارزمية Aho-Corasick حيث في هذا النوع سوف تستخدم الأشجار Tree كهيكل للبيانات أو الأصح شجرة ال ternary Search Tree والتي تم الحديث عنها هنا، حيث البحث في الأشجار سريع جدا مقارنه مع غيره، او حتى تستخدم بنيه Hash Table (سبق الحديث عنها هنا) مثل طريقة Veld man.

الأمثله على التطبيقات في مطابقه النصوص لا تنتهي ويمكن أن تدمج هذه الخوارزميات وخوارزميات أخرى لتكون لدينا مشاريع رائعه للغايه وعمليه أيضا ، فمثلا يمكن عمل برنامج لكشف الغش في الأختبارات، حيث يقوم البرنامج باستقبال اجابات طالبين ومن ثم يقوم بتظليل جميع الكلمات المتطابقه وفي النهايه يعرض تقرير يوضح عدد الكلمات المتطابقه وفي حال زاد العدد عن عدد معين ، فنستنتج أن الطالبين غاشين . مثال أخر وهو المحرر الأملائي في برامج محررات النصوص ، حيث أنك بمجرد كتابه الكلمه فيقوم الجزء المسؤول عن التأكد من الأخطاء (عادة هو Thread أخر يعمل بمجرد بدء البرنامج) بالبحث في القاعده التي تحتوي على جميع الكلمات في اللغه العربيه فاذا وجد أنها موجوده فهذا يعني أن الكلمه صحيحه ، والا فيقوم بعرض جميع الكلمات القريبه ويكون هذا باستخدام خوارزميات لايجاد النصوص الأقرب للنمط وذلك بالأعتماد على طريقه نطقها Sound وأشهر خوارزميات هذا النوع هو Soundix Searching.

٣ ٤ الخوارزمية Naive Searching Algorithm

تعريفات: عندما يتم ذكر مصطلح النمط أو Pattern فهذا يعني النص الذي نريد البحث عنه، وعندما يتم ذكر النص الكبير أو Text فنحن نقصد النص الذي يحتوي على جميع الحروف .

نبدأ الأن في أحد أقدم وأبسط الخوارزميات وفكرتها تكون عن طريق مقارنه حرف من ال pattern مع حرف مع ال Text، فاذا تطابق الحرفين ، فنقوم بالذهاب الى الحرف التالى في كل من النمط pattern وال text. اما في حال لم يتطابقا فنقوم بتحريك مؤشر الحرف في ال text الى الحرف التالي ونقوم بارجاع مؤشر الحرف في pattern الى البدايه ، وسوف نبدأ عمليه المقارنه مره أخرى . وسوف نستمر هكذا الى أن نجد التطابق في كل حروف ال pattern، أو أن نصل لنهايه ال text. وسوف نشير الى طول ال pattern بالحرف M أما طول ال text سوف نشير له بالحرف N. وسوف نتوقف في البحث عندما نصل ل N-M لأننا عندما نصل لتلك الخانه وحتى اذا كان الحرف التالي متطابق فسوف نتوق ف لأن ال Text سوف يكون أصغر من ال pattern.

هناك من طبق الخوارزميه بأكثر من وجه ولكنها في النهايه هي نفس النتيجه
الكود التالى يوضح لنا تطبيق الخوارزميه بعده طرق ممكنه (قد يصعب تتبع مثل هذه الحلقات لذلك لا أفضل من الورقه والقلم في حال لم تستطيع فهم الحلقه بشكل جيد).

٣-٤ كود سي++ لتطبيق خوارزمية Naïve Search

Naive String Search C++

```
// implement Naive String Search (also known as Brute-Force) Using Several Method
// By : Wajdy Essam

#include <iostream>
using std:: string ;

// C implementation
// return the first index when match string
int BruteForceSearch1 (char* pattern , char* text) {
   int i , j , m = strlen(pattern) , n = strlen(text) ;

   for (i=0 , j=0 ; j<m && i<n ; i++,j++ ) {
      while ( pattern[j] != text[i]) {
         i -= j-1 ;
         j = 0 ;
      }

      if ( j == m )
         return i-m ;
      else
         return i ;
   }
}
```

```
// C implementation
// printing index for all matching string in text
void BruteForceSearch2 (char* pattern , char* text ) {
   int i , j , m = strlen(pattern) , n = strlen(text) ;

   for (j=0 ; j<=n-m ; j++) {
      for (i=0 ; i<m && pattern[i] == text[i+j] ; i++)  ;    // note ; here
      if ( i >= m)
         std::cout << j << " " ;
   }
}

// C++ Implementation
// return first index when mathc string , else return -1
class NaiveSearch {
   public :
      NaiveSearch() { mText = "" ;}
      NaiveSearch(const string& text) : mText(text) {  }

         return -1 ;
      }
   private :
      string mText ;
};

int main (int argc , char* argv[]) {
   std:: cout << "index is : " << BruteForceSearch1("ss","wajdy essam is assembly programmer") <<
std::endl ;
```

٣-٥ كود جافا للتطبيق خوارزمية Naïve Search

```
1
2  // Java Implementation for Naive Search Algorithm
3
4  public class Demo{
5     public static void main (String args[]) {
6        // test first method
7        System.out.println("Index is : " + BruteForceSearch1("ss","wajdy essam is as
   sembly programmer") ) ;
8
9        // test second method
10       System.out.print("index is : ") ;  BruteForceSearch2("ss","wajdy essam is as
   sembly programmer");
11       System.out.println();
12
13       // test third method
14       NaiveSearch ns = new NaiveSearch("wajdy essam is assembly programmer");
         System.out.println("index is : " + ns.match("ss") );
15    }
16
17    // return the first index when match string
18    static int BruteForceSearch1 (String pattern , String text) {
19       int i , j , m = pattern.length() , n = text.length()  ;
20
21       for (i=0 , j=0 ; j<m && i<n ; i++,j++ ) {
22          while ( pattern.charAt(j) != text.charAt(i) ){
23             i -= j-1 ;
24             j = 0 ;
25          }
26
32
33       return -1 ;
34    }
35
36    // printing index for all matching string in text
37    static void BruteForceSearch2 (String pattern , String text ) {
38       int i , j , m = pattern.length()  , n = text.length()  ;
39
40       for (j=0 ; j<=n-m ; j++) {
41          for (i=0 ; i<m && pattern.charAt(i) == text.charAt(i+j) ; i++)  ;    // n
   ote ; here
43          if ( i >= m)
44             System.out.print(j + " ");
45       }
46    }
47 }
48
49 // return first index when mathc string , else return -1
50 class NaiveSearch {
51    public NaiveSearch() { mText = "" ;}
52    public NaiveSearch(String  text) { mText = text ; }
53
54    public boolean matchAt (String pattern , int position) {
55       for (int i=0 ; i<pattern.length() ; i++) {
56          if ( pattern.charAt(i) != mText.charAt(i+position) )
57             return false ;
68       return -1 ;
69    }
70
71    private String mText ;
72 }
73
74
67
```

٣-٦ الخوارزمية Knuth – Morris – Pratt String Matching

في عام ١٩٧٧ نشر الثلاثي Knuth (صاحب كتاب Art of programming الشهير) و Morris و Pratt مقاله Fast Pattern Matching in Strings وتحدثوا فيها عن ايجاد خوارزميه أسرع من الطريقه التي تعتمد على Brute-Force وسميت هذه الخوارزميه بأسم هؤلاء الأشخاص وأختصارا KMP .

للننظر قليلا الى ال Brute-Force وسنجد أنها عندما تجد حرف في pattern لا يطابق الحرف في text فإنها تقوم بالذهاب الى الحرف التالي من الtext وتعيد المقارنه من أول الPattern . خوارزميه KMP جائت لتحسين تلك الخوارزميه حيث تم التخلص من عمليات المقارنه المتكرره، طريقه عملها كالتالي:

أولا تقارن الحرف الأول في الpattern مع الحرف الأول من ال text في حاله كانا مختلفين فتقوم بالذهاب الى الموقع التالى من text (كما في Brute-Force بالضبط) ولكن الأختلاف يكون في حال قمنا بمقارنه ثلاثه حروف من الtext مع ال pattern وكانت النتيجه صحيحه ولكن الحرف الرابع يختلف ، هنا في هذه الحاله لن نقوم كما في الbrute-force باعاده مقارنه جميع الحروف مع الحرف التالى من text ، لكن بما أننا نعلم بأن هذه الحروف قمنا بمطابقتها سابقا فنقوم بالذهاب الى الموقع التالى من text الذي يشابه هذه الحروف التي طابقناها . وبالتالى ستكون هناك عمليات أقل وبالتالى الخوارزميه أسرع بكثير من السابقه.

في البدايه يجب أن نقوم بحساب Prefix للpattern وسوف نحتاجه عندما نقوم بمقارنه الpattern مع text ، حيث هذه prefix array تحتوي على عدد الأحرف المشابه من أول الpattern للحرف الحالى من الpattern ، الصوره التاليه توضح كيف يمكن حسابه :

٣-٧ الخوارزمية Boyer-Moore Algorithm

نتحدث هذه المره عن أحد أسرع الخوارزميات المستخدمه في عمليه البحث عن النصوص وهي Boyer-Moore وسميت بذلك بالطبع نسبه لمخترعي هذه الخوارزميه حيث قدموها في عام ١٩٧٧ كبديل للطريقه البحث البدائيه أو ما يسمى ب Naive Searching.

المقارنه في هذه الخوارزميه تعتمد على البدء من اليمين الى اليسار وليس كما هو الحال مع الخوارزميات العاديه ،، بالاضافه الى انها تقلل كثيرا من عمليات المقارنه خصوصا في حال لم يكن الحرف في text موجود في الـ pattern حيث تقفز بمقدار معين نقوم بحسابه كما سيتبين الأن ، هذا المقدار سوف يكون في جدول الازاحه skip (أو بالأسم الصحيح Bad-Character Shift) بمعنى أن هناك عمليه تحليل للـ pattern قبل البدء في البحث ، كما هو الحال مع KMP.
جدول الإزاحه يجب أن يكون حجمه مساوي لحجم ال character set التي نريد أن يتكون منها النص text والنمط pattern، وبما أننا حاليا سوف نبحث عن الحروف الأنجليزيه سوف يكون حجم الجدول بعدد حروف ال ASCII وهي ٢٥٦ . وقبل أن نبدأ في البحث في النص ، يجب أن نقوم بتحليل النمط ونقوم بتعبئه الحرف المقابل في الجدول للحرف المقابل للنمط بمقدار ظهور أخر index للنمط .
المثال التالي يوضح لنا كل شيء ، لنفرض لدينا الcharacter set مكونه من الحروف A,B,C,D,E والنمط هو DECADE ، سوف يكون شكل الجدول كالتالي

A	B	C	D	E
3	-1	2	4	5

تم حساب هذا الجدول كالتالي ، نبدأ من اليسار لليمين ونقوم بوضع ال Index لكل حرف في الpattern في الجدول .أولاً D موقعه هو ٠، وسنضع ٠ في الخانه المناسبه للحرف D في الجدول(الخانه الثالثه -الترقيم من ٠-). ثانياً الحرف E موقعه هو ١ وسنضع ١ في الخانه المناسبه للحروف E في الجدول .وسنستمر كذلك.

لاحظ أن الحرف D سيتكرر مره ثانيه وسنقوم بكتابه الIndex الجديد (٤) في الخانه القديمه ، وبنفس الأمر للحرف E. وهذا هو لب عمليه التحليل في البدايه ، فقط نحن نريد موقع أخر ظهور للحرف ، وهو ما سنحتاجه في عمليه البحث في حال أختلف الحرف من النص مع النمط .لاحظ أيضا بقيه الحروف في الcharacter set والتي لا توجد لها قيم سنضع لها -١. الكود التالي يبين كيف يمكن حساب هذا الجدول:

هناك طريقه أخرى لعمل هذا الجدول حيث يتم تهيئه المصفوفه أولا بطول النمط ، سوف يتم وضح الحرف M-i- ١ بدلا من i في الحلقه الثانيه ، وسوف تختلف عمليه البحث بشكل صغير أيضا ، لكن التطبيق أعلاه أسهل في الفهم والاستعياب من الثاني.

٨-٣ التوصيات :-

تستخدم الخوارزميات في مجالات كثيرة وعديدة والتي لم نتمكن الشرح عنها في البحث بشكل مفصل يتم توصية ان يتم استخدامه والاستفادة منه ومعرفة اكثر من مجال مطابقة النصوص وبرامج الأنتي فايروس لحل مشاكل عده

٩-٣ الاستنتاجات :-

نستنتج في هذا البحث مشكلة مطابقة النصوص التي يقوم البحث عن نص ومطابقة وهذا يتم حله عن طريق الخوارزميات ونستنتج من هذا البحث حل عملية الغش بين الطلاب المدارس بواسطة عمل برنامج التي يعمل بواسطة الخوارزميات

١٠-٣ الخاتمة :-

نصل لنهاية المقالة ، أرجوا أن تكون مفيدة للباحث في هذا المجال، ويمكن ان تحل التمرين التالي لتعزيز المفاهيم في هذا الموضوع. فكرته: نريد عمل تطبيق بسيط نستفيد منه من هذه الخوارزمية ، مثلا نقوم بعمل محرر نصوص صغير جدا ، يحتوي فقط على مربع نص text area ويوجد زر اسمه find & replace نقوم فيه بكتابه الاسم المراد البحث عنه والكلمة المراد تغييرها ،، ومن ثم نضغط بدء العملية ،، أغلب محررات تستخدم BM لأنها الأفضل وهو ما سنطبقه أيضا .. يمكن التطبيق بأي لغة برمجه ..

المصادر :-

* ر. سون &ل. بوكمان، (محررون(.، *البنايات الحوسبية : دمج العمليات العصبية والرمزية* .دار نشر كلوفر الأكاديمية، نيدام، ماساتشوستس. ١٩٩٤

* مارغريت بودن، "العقل كآلة"، دار نشر جامعة أكسفورد، ٢٠٠٦.

*Nilsson ،Nils (1998)، *Artificial Intelligence: A New Synthesis* ،Morgan Kaufmann publishers ، ISBN 978-1-55860-467-4

*Poole ،David؛ Mack worth ،Alan؛ Goebel ،Randy (1998)، *Computational Intelligence: A Logical Approach* ،New York: Oxford University Press

*المصطلحات الأساسية في الخوارزمية الجينية من الموقع أدناه

http://knol.google.com

*الخوارزميات الجينية في مطابقة النصوص من الموقع أدناه

https://informatic-ar.com/string_matching