IJEIT
المجلة الدولية للهندسة وتقنية المعلومات
journal homepage:www.ijeit.misuratau.edu.ly
2022 IECMU International Engineering conference-Misurata University

الأَمْثَلَة باستخدام خوارزمية أسراب العناصر

نجوى محمد عبد الصادق
قسم الرياضيات التطبيقية، الأكاديمية الليبية مصراتة، مصراتة، ليبيا.
najwaalsadeg@bs.lam.edu.ly

علي محمد عبد الشاهد
قسم الهندسة الكهربائية والالكترونية، كلية الهندسة، جامعة مصراتة، مصراتة، ليبيا.
a.abdulshahed@mu.edu.ly

الملخص— **يعتبر التحسين أو بمعنى أدق "الأمثلة" (Optimization) أمرًا بالغ الأهمية في العديد من التطبيقات الهندسية والرياضية المختلفة، حيث أن من الأهداف الرئيسية للأمثلة هو تقليل قيمة الخطأ وزيادة دقة النموذج الرياضي في العديد من التطبيقات. تُحسن طرق الأمثلة جودة الحلول بصورة متكررة حتى الوصول للحل الأمثل، أو على الأقل الممكـــن. ونظراً لهذه الأهمية البالغة لهذا المجال، تم في هذه الدراسة استخدام مفهوم خوارزمية أسراب العناصر (Particle Swarm Optimization PSO) وتطبيقاتها وذلك بتوضيح حالتين دراسيتين بشكل عددي لبيان آلية عمل هذه الخوارزمية أمام البحاث والدارسين في هذا المجال. ولقد أظهرت النتائج أن تطبيق اسلوب أمثلية أسراب العناصر (PSO) قد أعطى نتائج عالية الدقة في عملية الوصول إلى الامثلية من خلال قصر الفترة الزمنية وكذلك طبيعة وخصائص هذا الاسلوب في الوصول إلى حالة الامثلية. كما تبين النتائج أيضًا أن طريقة أمثلة أسراب العناصر من الطرائق المهمة التي تستخدم لإيجاد الحل الأمثل للعديد من المسائل الهندسية لما امتازت به هذه الطريقة من سهولة في الاستخدام، وسرعة التنفيذ.**

الكلمات المفتاحية: **خوارزمية أسراب العناصرPSO، الأمثلة Optimization دالة الهدف، البايثون** Python.

1. المقدمة

تطبيقات الأمثلة (التحسين) تتواجد في كل مكان، من التصميم الهندسي إلى علوم الاتصالات والحاسب، ومن التخطيط لوضع أقل أبراج اتصالات والحصول على أفضل تغطية للهاتف المحمول إلى الروتين اليومي حيث تسعى الشركات إلى تعظيم الأرباح وتقليل كلفة التركيب وبأقل مجهود. فالمصمم الهندسي عند تصميمه لمنتج معين يراعي أن يتم تعظيم أداء المنتج وتقليل تكلفته في الوقت نفسه، لذا فإن دراسة الأمثلة (التحسين) تساعد في حل الكثير من المشاكل التي تواجه المهندس في أعماله الروتينية اليومية. في الوقت الحاضر، أصبحت عمليات المحاكاة الحاسوبية أداة لا غنى عنها لحل مشكلات التحسين باستخدام خوارزميات بحث فعالة ومتنوعة [3، 4].

منذ أكثر من خمسين عاما أثبت علماء البيولوجيا وجود أنواع متعددة من أشكال الذكاء منبثقة من مجتمع الحشرات والأسماك والطيور أو الثدييات، داخل كثيب النمل وأسراب النمل الأبيض ومستعمرات النحل وأسراب الطيور والأسماك. على الرغم من ذلك فإن التفاعل المجرد البسيط بين عدد كبير من المخلوقات البسيطة يمكن أن يقود إلى نشوء ذكاء متفاعل ومتأقلم مع البيئة المحيطة. وفي مجتمعات الحشرات يكون النظام بأكمله منظما في نموذج لامركزي. الكثير من الوحدات المستقلة بذاتها التي تملك سلوكا بسيطا احتماليا نسبيا يجري توزيعها في البيئة. كل وحدة مزودة فحسب بالمعلومات المحلية. ولا تمتلك هذه الوحدات أي تمثيل أو معرفة واضحة بالتركيبة الشاملة التي من المفترض أن تقوم بإنتاجها أو تطويرها. ولا أية خطة على الإطلاق. أي بكلمات أخرى إن المهمة الشاملة ليست مبرمجة بشكل واضح من خلال الأفراد ولكنها تنبثق بعد نجاح عدد كبير من التفاعلات الأحادية بين الأفراد أو بين الأفراد والبيئة. هذا النموذج من الذكاء الجماعي الذي يتم بناؤه من قبل العديد من الكيانات المنفردة البسيطة كان هو الملهم لنظام جديد في علوم الحاسوب وهو ذكاء السرب [4].

خوارزميات الأمثلة (التحسين) المستوحاة من الطبيعية هي فرع من فروع الذكاء الاصطناعي التي تتعامل مع اكتشاف أو تحري الحلول المثلى لمسألة معينة ضمن مجموعة من البدائل، أو يمكن أن يُنظر إليها على أنها إحدى الأدوات الكمية الرئيسية في مجال صنع القرار، بحيث يجب أن تؤخذ القرارات لتحسين هدف واحد على الأقل من أهداف في مجموعة محددة من الظروف. بشكل عام تعمل خوارزميات الامثلة كما يلي: إنتاج مجتمع من الأفراد وتقويم صلاحيتهم، وإنتاج مجتمع جديد محسن، وتكرار هذه العملية عددا من المرات حتى الوصول للحل الأمثل أو الممكن على أقل تقدير. ومع بداية التسعينات من القرن الماضي بدأت الأبحاث باتجاه محاكاة الكائنات الحية الأقل ذكاء من الإنسان، مثل مستعمرات النمل والأسماك والطيور، أي ذاك النوع من الذكاء الاجتماعي للحيوانات الذي يظهر في سلوكها، وبالتالي تكون نتيجة مثل هذه المحاكاة نقل الذكاء الاجتماعي أو التعاوني إلى الحاسب. ومن الأمثلة على ذلك تلك السلسلة الحديثة لخوارزمية الأمثلة المستوحاة من الطبيعة [4] والتي تكون عادة مبنية على فئة بسيطة من القواعد، لتقوم بتحسين الحلول بصورة متكررة حتى الوصول إلى الحل الأمثل.

في العام 1859 م قدم العالم تشارلز داروين نظريته في التطور والتي تتلخص بأنه في ظل وجود موارد محدودة يتنافس الأفراد مع بعضهم البعض من أجل البقاء. ويحظى الأفراد الذين يمتلكون أفضل الصفات بفرصة أكبر في البقاء على قيد الحياة والتكاثر وتوريث صفاتهم إلى الجيل التالي. وبمرور الوقت تصبح تلك الصفات الجيدة أكثر شيوعا في الأجيال اللاحقة ويطلق على هذه العملية مصطلح "البقاء للأصلح" وتشير نظرية داروين أيضا إلى إمكانية حدوث بعض التغييرات العشوائية التي تؤدي إلى الحصول على مزيج من الصفات الجديدة. تزداد فرصة الكائن الحي في البقاء على قيد الحياة إذا كانت التغيرات الجديدة تعود عليه بالفائدة وهذا ما يسمى بالانتقاء الطبيعي [3، 4].

تعد الخوارزميات التطورية أحد فروع الذكاء الصناعي وهي تحاكي التوزع الطبيعي لبعض الكائنات الحية. تعتمد هذه الخوارزميات في آلية عملها على التقنيات الحيوية كإعادة الانتاج (Reproduction)، إعادة التجميع (Recombination) وآلية الاصطفاء (Selection) وتستخدم مفاهيم التطور البيولوجي مثل الانتقاء الطبيعي (Natural Selection) والبقاء للأصلح (Survival of Fittest) من أجل حل المسائل المختلفة للحصول على أفضل النتائج (أو القريبة للأفضل). تعبر الخوارزميات التطورية عن مصطلح شامل لأنواع عديدة من خوارزميات الذكاء الاصطناعي التي تعتمد في مبدأ عملها على محاكاة بعض العمليات الحيوية التي تحدث في الطبيعة. تأخذ هذه الخوارزميات أهميتها بسبب قدرتها على إيجاد حلول فعّالة لمختلف المسائل دون الحاجة لمعرفة الكثير من المعلومات عن طبيعة المسألة المراد حلها. يعد هذا النوع من الخوارزميات أحد فروع التقنيات الحديثة وهي تحاكي الذكاء الطبيعي لبعض الكائنات الحية.

هناك عدة نماذج من الخوارزميات التطوريّة نذكر منها ما يلي [3]:

خوارزمية أمثلية أسراب العناصر Particle Swarm Optimization (PSO)، الخوارزمية الجينية Genetic Algorithm (GA)، خوارزمية طائر الوقواق Cuckoo Search (CS) ، خوارزمية مستعمرات النمل Ant Colony Optimization (ACO)، خوارزمية المناعة الاصطناعية Artificial Immune Algorithm (AIA). اقترح الباحثان [5] في هذه الدراسة إظهار استخدام تقنيات

استلمت الورقة بالكامل في 26 ديسمبر 2022 وروجعت في 3 اغسطس 2022
وقبلت للنشر في 30 سبتمبر 2022
ونشرت ومتاحة على الشبكة العنكبوتية في 1 ديسمبر 2022.

الحساب التطوري مثل أمثلة أسراب الطيور (PSO) وأمثلة مستعمرة النمل (ACO) في حل مشكلة الانتشار الداخلي لشبكات الحاسوب اللاسلكية. حيث أنه وعلى الرغم من أن هذه الخوارزميات تستخدم استراتيجيات وجهود حسابية مختلفة، إلا أنها تشترك أيضا في بعض أوجه التشابه، حيث تم مقارنة أدائها مع الخوارزميات الجينية (GA)، والتي تستخدم كمرجع في هذه الحالة. وتم أيضا توضيح قدرة هذه الخوارزميات على تحسين مواقع نقاط الوصول باستخدام البيانات المستمدة من نموذج الشبكة العصبية لشبكة محلية لاسلكية معينة. واستنتج الباحثان أن النتائج التي تم الحصول عليها بواسطة خوارزمية PSO أفضل مقارنة بخوارزمية ACO. ومع ذلك فإن النهج القائم على خوارزمية مستعمرة النمل يكون مفيدا في حل مشاكل الانتشار، على الرغم من أن خوارزمية ACO تتطلب مزيدا من العمل لتحسين معلماتها وتحسين تحليل بيانات مادة الفيرمون (Fheromone) وتقليل وقت الحساب. وفي عام (2018) قام الباحثان (فراس احمد و نور سليم) [1] بدراسة كيفية استخدام المحاكاة للمفاضلة بين بعض الطرائق الحديثة لحل نموذج النظرية الرمادية من الرتبة الأولى (GM(1,1، كما تم استخدم طريقة المربعات الصغرى الاعتيادية (LS) لتقدير معلمات النموذج مع تطبيق عملي، بينت هذه الدراسة أن أفضل طريقة في تقدير معلمات النموذج الرمادي هي طريقة (POS) حيث أثبتت حصولها على أفضل نتائج مقارنتاً بالطرق المستخدمة في الدراسة المشار إليها.

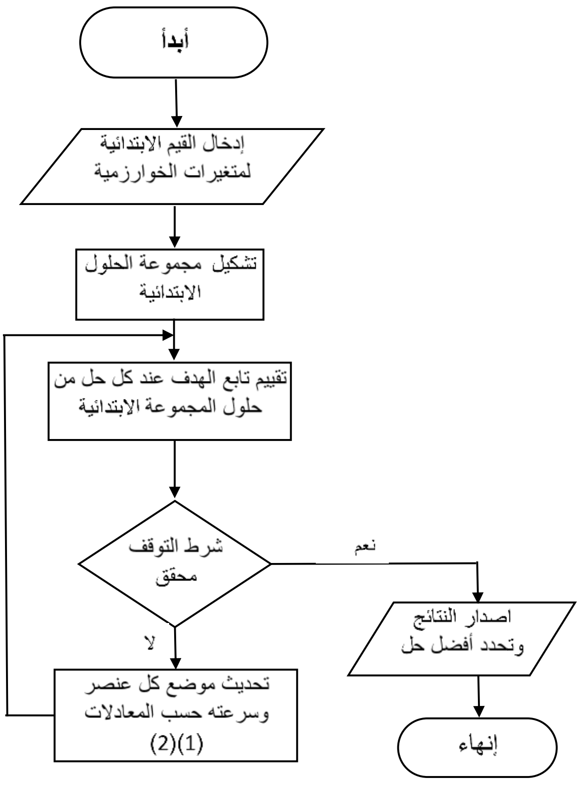
في هذا البحث تم اختيار خوارزمية (PSO) بسبب أنها تمتاز بسهولة في التطبيق وسرعة في التنفيذ باستخدام الحاسوب، وكذلك بسبب عدم احتوائها على خطوات برمجية إضافية تحاكي الاصطفاء الطبيعي مثل التهجين (Crossover) والطفرة (Mutation) الموجودة في الخوارزمية الجينية (GA) وبعض الخوارزميات التطورية الأخرى. بالإضافة إلى أن (PSO) تستخدم الأعداد الحقيقية على أنها الحلول، وهذا يختلف عن الخوارزميات التطورية الأخرى والتي تحتاج خطوات حسابية إضافية [6، 7].

2. خوارزمية أسراب العناصر (PSO)

تم تطوير نموذج الخوارزمية من قبل العالمين Eberhart و Kennedy في عام 1995، وهذا النموذج مستوحى من السلوك الاجتماعي لأسراب الطيور وجماعات الأسماك أثناء تنقلها من مكان لآخر. تقوم فكرة خوارزمية PSO على محاكاة سلوك أسراب الطيور، ولتوضيح الفكرة سنطرح المثال التالي [8]:

لدينا سرب من الطيور ينتشر في منطقة محددة بغرض البحث عن الطعام، وينتشر الطعام في هذه المنطقة عشوائياً، بالإضافة إلى أن الطيور لا تعلم مواضع الطعام على نحو واضح، فما هي الطريقة المثلى للبحث عن الطعام؟ الطريقة المثلى هي بانتشار عدد من الطيور في كافة المنطقة، مع إخبار الطيور في كل مرة بعضها البعض عن مواضع الطعام، فالطيور تبدأ رحلة البحث من مناطق عشوائية ومع كل إعادة تقترب الطيور من الطعام (المنطقة الممتلئة بالطعام) والذي يكافئ الحل الأمثل والاكثر دقة .

تحاكي خوارزمية (PSO) السيناريو السابق، واستخدمته لحل مسائل الأمثلة (Optimization Problems)، ضمن خوارزمية الأسراب (PSO) كل طير (Bird) يكافئ حل ضمن فضاء الحلول المسماة بالعنصر (Particle)، ولكل عنصر من العناصر قيمة ملائمة تسمى (Fitness Value) أي تدل على مدى ملائمة هذا الجزء للحل، ويتم تقييم قيم الملائمة هذه عبر تابع يدعى بتابع الملائمة (Fitness Function) والتقييم هنا يهدف إلى حساب مقدار قرب هذا الجزء من الحل الأمثل، وكذلك تملك العناصر سرعات (Velocities) وتقود هذه السرعات بدورها هذه العناصر الطائرة. يتم تهيئة خوارزمية (PSO) بمجموعة من العناصر العشوائية (حلول)، ومن ثم يتم البحث عن الحل الأفضل عبر تحديث هذه الحلول، ضمن كل تكرار (Iteration)، ويتم تحديث كل عنصر من العناصر ضمن التجمع عبر إتباع القيم المثلى التالية: أفضل قيمة ملائمة سجلها العنصر (P_{best})، بالإضافة إلى أفضل قيمة ملائمة مسجلة ضمن السرب (g_{best}) ، وأخيرا قيمة أفضل تموضع محلي لعنصر بالمقارنة مع العناصر المجاورة محليا (LBest) [2]، والشكل (1) يوضح آلية عمل خوارزمية (PSO) . بشكل عام، تستمر عملية تغيير سرعات الطيور حتى الوصول للحل الأمثل الذي يعتمد على معيار التوقف. في الواقع يختلف هذا المعيار باختلاف الحالة قيد الدراسة ومنها: ظهور حل أمثلي ممكن للمشكلة المطروحة، أو الوصول إلى عدد أعظمي تم تحديده مسبقاً، أو عدم تحسين نتائج الخوارزمية مع تعديل السرعات.



شكل 1. آلية عمل الخوارزمية PSO

ويمكن تمثيل خوارزمية الـPSO رياضيًا بالصيغة التالية:

$$v_i^{k+1} = w \times v_i^k + c_1 \times rand()_1 \times (P_{best_i} - x_i^k) + c_2 \times rand()_2 \times (g_{best} - x_i^k) \qquad (1)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \qquad (2)$$

حيث:

v_i^{k+1} : سرعة العنصر عند التكرار $k+1$.

v_i^k : سرعة العنصر عند التكرار k.

w : عامل الوزن.

x_i^{k+1} : موضع العنصر i عند التكرار $k+1$.

x_i^k : موضع العنصر i عند التكرار k .

c_1 : الثابت الفردي ويتعلق بـP_{best}.

c_2 : الثابت الجماعي ويتعلق بـg_{best}.

$rand()_1$: عدد عشوائي بين الصفر والواحد.

$rand()_2$: عدد عشوائي بين الصفر والواحد.

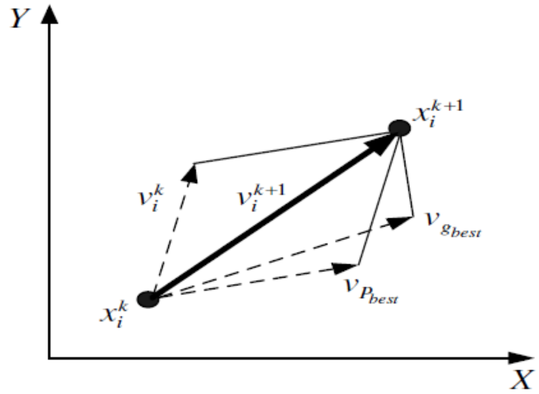
P_{best_i} : موضع **Pbest** للعنصر i (أفضل موقع سابق).

g_{best} : موضع السرب (أفضل موقع من أفضل مواقع العناصر المتحققة).

توضح المعادلتين (1) و (2) السرعة والموضع الجديد على التوالي.

تحسب المعادلة (**1**) السرعة الجديدة لكل عنصر اعتمادً على سرعة العناصر السابقة وموضعها عند الحل الأمثل المحقّق حتى الآن.

يبين الشكل (2) مفهوم تعديل نقاط البحث الموضحة بالمعادلات (**1**) و(**2**).



الشكل 2. مفهوم تعديل نقطة البحث

متغيرات التحكم لـPSO:

هناك خطوتين أساسيتين يجب القيام بهما عند استخدام خوارزمية الـ**PSO** في إيجاد الحل الأمثل هما: تمثيل الحل والتابع المناسب. حيث أن عملية البحث هي عملية متكررة ومعيار التوقف هو أن يتم الوصول إلى عدد التكرارات الأقصى أو أن يحقق أقل خطأ (شرط الحل). لا يوجد الكثير من المتغيرات التي تحتاج إلى تغيير في (**PSO**) وهذه قائمة بالمتغيرات وقيمها المثالية (النموذجية) وفقاً للمرجع [9]:

- عدد العناصر: يتراوح ما بين 20–40. والتي تكون مناسبة للعديد من التطبيقات البسيطة. بينما يمكن استخدام 100 أو 200 عنصر في بعض المشاكل الصعبة أو الخاصة [9].
- مجال العناصر: يحدّد حسب المشكلة المراد إيجاد الحل الأمثل لها، ويمكن تحديد مجالات مختلفة من أجل أبعاد مختلفة للعناصر.
- الثوابت: C_1، C_2 عادةً تساوي 2. على أية حال تستخدم قيم اخرى في أبحاث مختلفة، وفي أغلب الأحيان تكون $C_1 = C_2$ وتتراوح من $(4 \leftarrow 0)$ [10].
- شرط التوقف: العدد الاقصى من التكرارات المطلوبة لتنفيذ الـ **PSO** أو الوصول إلى أقل خطأ (شرط الحل).
- عامل الوزن **w**: تعطي القيم المثالية للمتغير **w** في المجال $(1.2 \leftarrow 0.9)$ وفقًا للمرجع [10].

3. التطبيق العددي

فيما يلي سيتم تقديم حالتين دراسيتين لتوضح كيفية عمل خوارزمية PSO للبحث عن الحل الامثل سواء كانت مسألة تعظيم (**Maximization**) أو تقليل (**Minimization**) للمشكلة المعطاة.

أ. الحالة الدراسية الأولى:

بفرض أنه في أحد فروع الهندسة كان المطلوب إيجاد (Maxf(x)) للدالة:

$$f(x) = x_1^2 - x_1x_2 + x_2^2 + 2x_1 + 4x_2 + 3$$

حيث $-5 \le x_1, x_2 \le 5$ ، باستخدام عدد 05 عناصر (طيور) وعدد 05 تكرارات.

الحل الجبري:

من البديهي أن أعلى قيمة للدالة **f(x)** هي عند وضع $x_1=5$, $x_2=5$ كالتالي:

$$f(x) = (5)^2 - (5)(5) + (5)^2 + 2(5) + 4(5) + 3 = 58$$

وذلك بسبب طبيعة المسألة قيد الدراسة وسهولة حلها جبريا، إلا أن العديد من المسائل يصعب حلها بهذه الطريقة. تم اختيار هذه الحالة البسيطة لسهولة عرضها أمام القارئ وتوضيح آلية عمل خوارزمية **PSO**.

الحل بطريقة PSO:

سوف يتضمن الحل جانب تطبيقي لما تم عرضه من معادلات في الجانب النظري من هذا البحث الذي بدوره يؤدي إلى إيجاد الحل الأمثل وهو القيمة العظمى Maxf(x) للدالة داخل الفترة $-5 \le x1, x2 \le 5$ باستخدام خوارزمية أسراب العناصر (PSO) لحل هذه المشكلة تم اختيار متغيرات خوارزمية الـ PSO كما يلي:

$$C_1 = C_2 = 1.5\,, w = 0.9\ \ , V \in [0,1]$$
$$X = L + rand.* (U - L)$$
حيث $X \in (-5, 5)$

التكرار الأول (1):

الجدول (1) يوضح القيم الابتدائية للحلول، الجدير بالذكر هنا أن هذه الحلول مولدة بشكل عشوائي داخل الفترة المعطاة.

الجدول 1. يوضح القيم الابتدائية للحلول

X(position)	
x1	x2
3.1472	-4.0246
4.0579	-2.2150
-3.7301	0.4688
4.1338	4.5751
1.3236	4.6489

الجدول (2) يوضح القيم الابتدائية لسرعات العناصر (الطيور)، الجدير بالذكر هنا أن هذه الحلول مولدة كذلك بشكل عشوائي.

الجدول 2. يوضح القيم الابتدائية لسرعة كل طائر

V(velocity)	
v_1	v_2
0.2194	0.2449
0.1908	0.2228
0.3828	0.3232
0.3976	0.3547
0.0934	0.3773

الجدول (3) يوضح نتائج عملية تعويض قيم **X** في الجدول (1) للحصول على قيم الدالة **f(x)**.

الجدول 3. يوضح قيم **f(x)** الابتدائية عند قيم **X** في التكرار الاول

f(x)
31.9645
32.6168
13.2071
48.6753
41.4537

نظرًا لعدم وجود تكرار سابق فإن:

$$P_{best_i} = X$$

الجدول (4) يوضح القيم الابتدائية لأفضل موقع سابق.

الجدول 4. يوضح أفضل موقع سابق (قيم الابتدائية)

$P_{best(i)}$	
3.1472	-4.0246
4.0579	-2.2150
-3.7301	0.4688
4.1338	4.5751
1.3236	4.6489

بالنظر إلى قيم **f(x)** في الجدول (3) نجد أن أقصي قيمة للدالة هي:

$$\boldsymbol{Maxf(x) = 48.6753}$$

عندما كانت $\boldsymbol{x_1 = 4.1338,\ x_2 = 4.5751}$
أي أن أفضل موقع من بين مواقع العناصر هو:

$$\boldsymbol{g_{best} = [4.1338 \quad 4.5751]}$$

التكرار الثاني (2):

لتحديد السرعة نعوض في المعادلة (1) للحصول على سرعة الجسيم الاول، المكون الاول:

$$\boldsymbol{v_i^{k+1} = w \times v_i^k + c_1 \times rand()_1 \times (P_{best_i} - x_i^k) + c_2 \times rand()_2 \times (g_{best} - x_i^k)}$$

حيث:

$$\boldsymbol{V_{11} = 0.9 * 0.2194 + (1.5)(0.5949)(3.1472 - 3.1472)}$$

$$\boldsymbol{+(1.5)(0.0855)(4.1338 - 3.1472)}$$

$$\boldsymbol{V_{11} = 0.3240}$$

بعد أن تم إيجاد **V_{11}** نعوض في المعادلة (2) لنتحصل على موقع العنصر **X_{11}**:

$$\boldsymbol{X_{11} = v_{11} + x_{11}}$$

$$\boldsymbol{= 0.3240 + 3.1472}$$

$$\boldsymbol{= 3.4712 \in (-5, 5)}$$

الجسيم الاول، المكون الثاني:

بالتعويض في المعادلة (1) نحصل على **V_{12}**:

$$\boldsymbol{V_{12} = wv_{12} + c_1 rand()_1 (P_{best_{12}} - x_{12}) + c_2 rand()_2 (g_{best} - x_{12})}$$

$$\boldsymbol{= (0.9)(0.2449) + (1.5)(0.5949)(-4.0246 + 4.0246)}$$

$$\boldsymbol{+(1.5)(0.0855)(4.5751 + 4.0246)}$$

$$\boldsymbol{V_{12} = 1.3233}$$

نعوض عن قيمة **V_{12}** في المعادلة (2) نتحصل على موقع العنصر **X_{12}**:

$$\boldsymbol{X_{12} = v_{12} + x_{12}}$$

$$\boldsymbol{= 1.3233 - 4.0246}$$

$$\boldsymbol{= -2.7013 \in (-5, 5)}$$

الجسيم الثاني، المكون الاول:

بالتعويض في المعادلة (1) للحصول على **V_{21}**:

$$\boldsymbol{V_{21} = wv_{21} + c_1 rand()_1 (P_{best_{21}} - x_{21}) + c_2 rand()_2 (g_{best} - x_{21})}$$

$$\boldsymbol{V_{21} = (0.9)(0.1908) + (1.5)(0.5949)(4.0579 - 4.0579)}$$

$$\boldsymbol{+(1.5)(0.0855)(4.1338 - 4.0579)}$$

$$\boldsymbol{V_{21} = 0.1815}$$

نعوض عن قيمة **V_{21}** في المعادلة (2) لنتحصل على موقع العنصر **X_{21}**:

$$\boldsymbol{X_{21} = v_{21} + x_{21}}$$

$$\boldsymbol{= 0.1815 + 4.0579}$$

$$\boldsymbol{= 4.2394 \in (-5, 5)}$$

وبالمثل يتم الحصول على السرعة **V** والموقع **X** لباقي العناصر باستخدام المعادلة (1) والمعادلة (2) على التوالي كما هو موضح في الجدول (5) التالي:

جدول 5. يوضح السرعة **V** والموقع **X** للعناصر عند التكرار الثاني

V(velocity)		X(position)	
v_1	v_2	x_1	x_2
0.3240	1.3233	3.1472	-2.7013
0.1815	1.0714	4.2394	-1.1436
1.3531	0.8175	-2.3770	1.2863
0.3578	0.3192	4.4916	4.8943
0.4445	0.3301	1.7681	4.9790

يتم استخدم قيم **x_1, x_2** في الجدول (5) للحصول على قيم الدالة **f(x)** كما هو موضح في الجدول (6).

جدول 6. يوضح قيم **f(x)** عند قيم **X** في التكرار الثاني

f(x)
27.8600
31.0325
13.7537
53.7063
45.5655

آلية حساب $P_{best(i)}$:
بالمقارنة بين قيم **f(x)** في التكـــرار الاول والثـاني بحيث يتم اختيـار قيم **X** التي تجعل **f(x)** أكبر ما يمكن:
الجسيم الاول **31.9645>27.8600**
الجسيم الثاني **32.6168>31.0325**
الجسيم الثالث **13.7535>13.2971**
الجسيم الرابع **53.7063>48.6753**
الجسيم الخامس **45.5655>41.4537**
يتم اختيار قيم **X** المناظرة لـ**Maxf(x)** للحصول على $P_{best(i)}$ (أفضل موقع سابق) للعناصر كما هو موضح في الجدول (7) التالي:

جدول 7. يوضح قيم $P_{best(i)}$ عند التكرار الثاني

$P_{best(i)}$	
3.1472	-4.0246
4.0579	-2.2150
2.3770-	1.2863
4.4916	4.8943
1.7681	4.9790

بالنظر إلى قيم **f(x)** في الجدول (6) نجد أن:

$$Maxf(x) = 53.7063$$

عندما $x_1 = 4.4916,\ x_2 = 4.8943$

أي أن:

$$g_{best} = [4.4916 \quad 4.8943]$$

التكرار الثالث (3):

نقوم بإعادة نفس الخطوات السابقة باستخدام المعادلة (1) لإيجاد سرعة العناصر**V**، والمعادلة (2) لإيجاد قيم موقع العناصر**X**، من ثم نستخدم قيم **X** لإيجاد قيم **f(x)** على التوالي أنظر الجدول (8):

جدول 8. يوضح سرعة وموقع العناصر وقيم **f(x)**

V(velocity) v_1	v_2	X(position) x_1	x_2	f(x)
0.1334	0.9843	3.6045	-1.7170	25.4710
0.0337	0.7826	4.2394	-0.3611	30.0346
2.0987	1.1985	-0.2784	2.4848	19.3259
0.3221	0.2873	4.8137	5.0000	56.7306
0.7493	0.2862	2.5174	5.0000	46.7851

يتم اختيار قيم **X** المناظرة لـ**Maxf(x)** في التكرارات السابقة للحصول على $P_{best(i)}$ (أفضل موقع سابق) للعناصر كما هو موضح في الجدول (9) التالي:

جدول 9. يوضح قيم $P_{best(i)}$ عند التكرار الثالث

$P_{best(i)}$	
3.4712	-2.7013
4.2394	-1.1436
-0.2784	2.4848
4.8137	5.0000
2.5174	5.0000

بالنظر إلى قيم **f(x)** في الجدول (8) نجد أن:

$$Maxf(x) = 56.7306$$

عندما $x_1 = 4.8137,\ x_2 = 5.0000$

أي أن:

$$g_{best} = [4.8137 \quad 5.0000]$$

التكرار الرابع (4):

نقوم بإعادة نفس الخطوات السابقة باستخدام المعادلة (1) لإيجاد سرعة العناصر**V**، والمعادلة (2) لإيجاد قيم موقع العناصر**X**، من ثم نستخدم قيم **X** لإيجاد قيم **f(x)** على التوالي أنظر الجدول (10):

جدول 10. يوضح سرعة وموقع العناصر وقيم **f(x)**

V(velocity) v_1	v_2	X(position) x_1	x_2	f(x)
01561	0.8690	3.7606	-0.848	25.1798
0.0696	0.6935	4.3427	0.3325	30.5409
2.5419	1.4012	2.2635	3.8860	34.4996
0.2899	0.2586	5.0000	5.0000	58.0000
0.9689	0.2576	3.4863	5.0000	49.6953

يتم اختيار قيم **X** المناظرة لـ**Maxf(x)** في التكرارات السابقة للحصول على $P_{best(i)}$ (أفضل موقع سابق) للعناصر كما هو موضح في الجدول (11) التالي:

جدول 11. يوضح قيم $P_{best(i)}$ عند التكرار الرابع

$P_{best(i)}$	
3.6045	-1.7170
4.3427	0.3325
2.2635	3.8860
5.0000	5.0000
3.4863	5.0000

بالنظر إلى قيم **f(x)** في الجدول (10) نجد أن:

$$Maxf(x) = 58.0000$$

عندما $x_1 = 5.0000,\ x_2 = 5.0000$

أي أن:

$$g_{best} = [5.0000 \quad 5.0000]$$

التكرار الخامس (5):

نقوم بإعادة نفس الخطوات السابقة على التوالي نتحصل على جدول (12) كما يلي:

جدول 12. يوضح سرعة وموقع العناصر وقيم **f(x)**

V(velocity) v_1	v_2	X(position) x_1	x_2	f(x)
01601	0.7567	3.9208	-0.0914	26.2152
0.1469	1.2228	4.4896	1.5553	33.7933
2.6386	1.4040	4.9021	5.0000	57.3244
0.2609	0.2327	5.0000	5.0000	58.0000
1.0661	0.2319	4.5524	5.0000	55.0672

يتم اختيار قيم **X** المناظرة لـ**Maxf(x)** في التكرارات السابقة للحصول على $P_{best(i)}$ (أفضل موقع سابق) للعناصر كما هو موضح في الجدول (13) التالي:

جدول 13. يوضح قيم $P_{best(i)}$ عند التكرار الخامس

$P_{best(i)}$	
3.9208	-0.0914
4.4896	1.5553
4.9021	5.0000
5.0000	5.0000
4.5524	5.0000

بالنظر إلى قيم **f(x)** في الجدول (12) نجد أن:

$$Maxf(x) = 58.0000$$

عندما $x_1 = 5.0000,\ x_2 = 5.0000$

أي أن:

$$g_{best} = [5.0000 \quad 5.0000]$$

يتم التوقف عندما يتحقق معيار التوقف، في هذه الحالة لا يوجد تغير في نتيجة التكرار والتكرار السابق له، كالتالي:

$$\boldsymbol{Maxf(x)(n-1) - Maxf(x)(n) = 0}$$

حيث: **n** عدد التكرارات، نتحصل على:

$$\mathbf{58.0000 - 58.0000 = 0}$$

أي أن أفضل قيمة لـ**x_1, x_2** هي:

$$\boldsymbol{x_1 = 5, \qquad x_2 = 5}$$

وهي القيمة التي تحقق**Maxf(x)** :

$$\boldsymbol{f(X) = 58}$$

وهي نفس القيمة المتحصل عليها بالطريقة الجبرية سالفة الذكر.

ب. الحالة الدراسية الثانية:

بفرض أنه في أحد فروع الهندسة كان المطلوب إيجاد أقل قيمة للدالة.

$$\boldsymbol{f(x,y) = (x-3.14)^2 + (y-2.72)^2 + \sin(3x+1.41) + \sin(4y-1.37)}$$

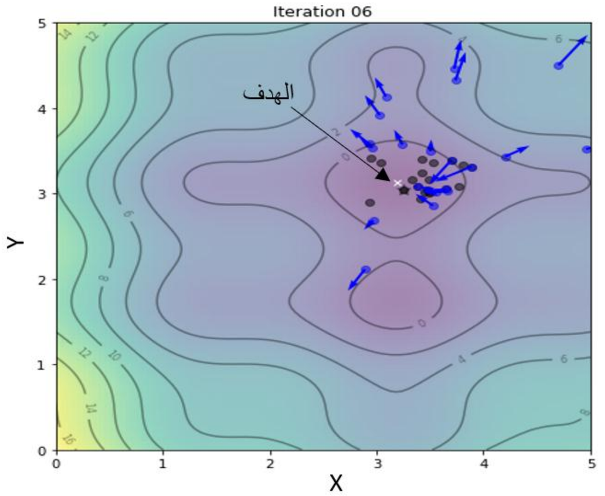
سوف يتضمن الحل جانب تطبيقي لما تم عرضه من أساليب في الجانب النظري، سيتم تطبيق مسألة تقليل **Minf(x)** باستخدام امثليه أسراب العناصر **(PSO)** ، وذلك بتطبيق المعادلتين (1) و (2) وباستخدام عدد 20 عنصر وعامل وزن **(w=0.9)** وقيم مختلفة للثوابت الفردي **(C_1)** والإجتماعي **(C_2)**، وبالاعتماد على برنامج (البايثون) تم الحصول على الجدول (14) التالي:

جدول 14. يوضح قيم **f(x,y)** عند متغيرات مختلفة

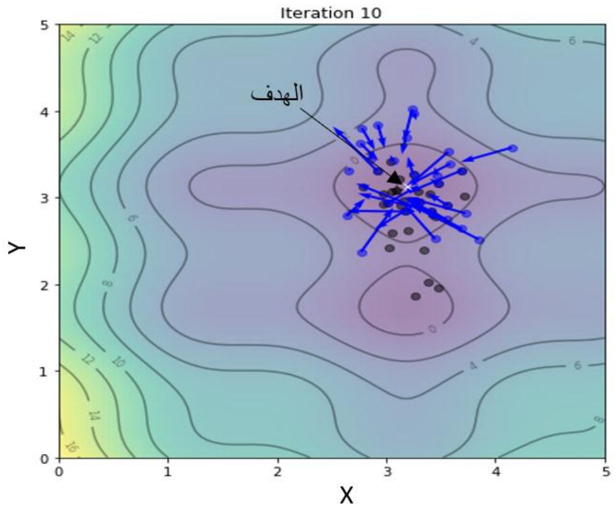
المتغير	**C_1=2** **C_2=2** عدد التكرارات 50	**C_1=2** **C_2=1.5** عدد التكرارات 30	**C_1=1.5** **C_2=2** عدد التكرارات 10	**C_1=1.5** **C_2=1.5** عدد التكرارات 6
X	3.187	3.199	3.068	3.087
Y	3.142	3.101	3.120	3.067
f(x,y)	-1.806	-1.800	-1.734	-1.722

الجدول (14) يوضح أفضل قيم تم الحصول عليها لـ**(x,y)** بقيم مختلفة لمتغيرات الخوارزمية. حيث كانت أفضل تلك القيم التي أدت الى تقليل دالة الهدف، عندما كانت **C_1=2, C_2=2** وعدد التكرارات **(50)** حيث بلغت قيمة **(x=3.187)** اما قيمة **(y=3.142)** ودالة الهدف فقد كانت **f(x,y)=-1.806**.

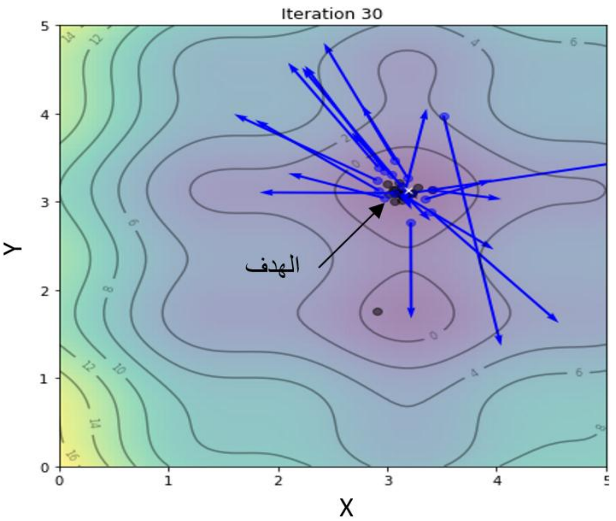
الشكل (3) و الشكل (4) والشكل (5) والشكل (6) توضح مراحل مختلفة من تنفيذ خوارزمية **PSO** بواسطة (البايثون)، حيث نلاحظ في حالة عدد التكرارات كانت 6، كانت العناصر (الطيور) تتجه في اتجاهات مختلفة، أي انها لم تصل إلى قرار جيد (حل مناسب) بعد، وفي حالة عدد التكرارات 10، نلاحظ أن العناصر تتجه إلى الحل، وعند زيادة عدد التكرارات يتحسن الحل إلى أن يصل إلى القيمة المطلوبة (الحل الامثل) كما هو موضح في الشكل (6) عندما كانت عدد التكرارات 50.



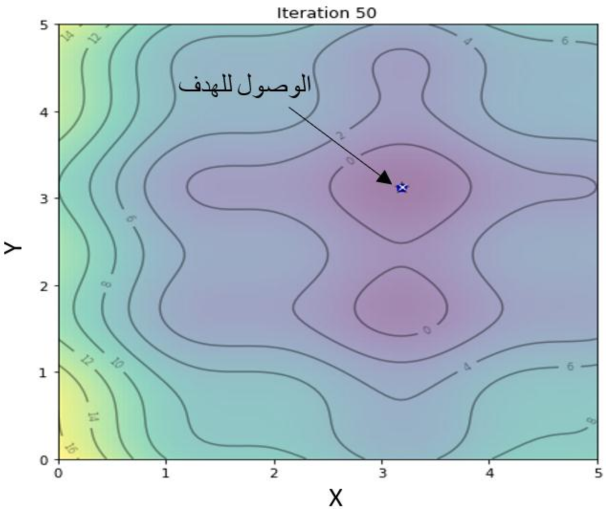
الشكل 3. مراحل تنفيذ خوارزمية PSO بواسطة البايثون (6 تكرارات)



الشكل 4. مراحل تنفيذ خوارزمية PSO بواسطة البايثون (10 تكرارات)



الشكل 5. مراحل تنفيذ خوارزمية PSO بواسطة البايثون (30 تكرار)



الشكل 6. مراحل تنفيذ خوارزمية PSO بواسطة البايثون (50 تكرار)

4. الخلاصة

في هذا البحث تم اختيار خوارزمية (PSO) لما تمتاز به من سهولة في التطبيق وسرعة في التنفيذ، وذلك بسبب عدم احتوائها على متغيرات إضافية مثل التهجين (Crossover) والطفرة (Mutation) كما هو الحال في الخوارزميات الجينية (GA). بالإضافة إلى أن PSO تتخذ الأعداد الحقيقية على أنها العناصر أو الحلول الابتدائية، وهذا يختلف عن الخوارزميات الجينية (GA) والتي تحتاج خطوة إضافية وهي عملية التحول إلى الترميز الثنائي (Binary Encoding) أو إلى استخدام المعاملات الجينية (Genetic Operators). إن أهم الاستنتاجات التي ثم الوصول إليها هي:

- تستند الخوارزميات التطورية في مبدئها على الإدراك والذكاء وهذا ما يجعلها مناسبة تماماً للاستخدامات الهندسية التي كانت سابقاً تُحل بطرق تقليدية.
- من خلال مراقبة وتحليل سلوك الكائنات الحية والتي تعتبر أقل ذكاء من الإنسان، تم الحصول على خوارزميات مستوحاة من الطبيعة أثبتت نجاحها في حل العديد من المشاكل التي كان يصعب حلها بالطرق التحليلية.
- لا تحوي هذه الخوارزمية في حساباتها أي تداخل أو انحراف في القيم بالتالي سرعة البحث تعتمد على سرعة العنصر، وعند تطوير الأجيال فإن قيم العنصر الأفضل في السرب هي فقط من سيتم إرسالها للعناصر الأخرى مما يعني سرعة في عملية البحث عن الهدف.
- الحسابات الرياضية في هذه الخوارزمية هي حسابات بسيطة مقارنة بغيرها من الخوارزميات كما أن لها قدرة كبيرة في الحصول على الحل الأمثل.
- تتبنى خوارزمية (PSO) ترميز المشكلة بالأعداد الحقيقية وهي حازمة مباشرة بالحل.
- استخدام قيم مختلفة لمعامل مكون التمييز الذاتي **(C_1)** ومعامل المكون الاجتماعي **(C_2)** عند ايجاد الحل الامثل للمسائل العددية للوصول الى حالة الامثلية وعدم الاقتصار على قيمة واحدة فقط.
- إن أمثلية اسراب العناصر (**PSO**) تعتمد بصورة أساسية على حساب قيمة الملائمة لكل عنصر من العناصر ضمن السرب وكذلك على تحديث أفضل قيم الملائمة **(P_{best})** لكل عنصر، وتحديث أفضل قيمة ملائمة عامة **(g_{best})**.

من خلال هذا العمل نوصي بإجراء المزيد من الأبحاث والدراسات على كيفية استخدام خوارزمية (**PSO**) وتوظيفها للعمل مع دوال هدف أخرى للحصول على نتائج الأمثلة المطلوبة مختلفة، كما نوصي ايضًا بإجراء دراسة بحثية لمحاولة تطبيق أمثليه اسراب العناصر (**PSO**) في ايجاد الحلول لنماذج رياضية أكثر تعقيداً.

المراجع

أ. المراجع العربية

[1] فراس احمد، نور سليم، استخدام المحاكاة للمفاضلة بين بعض الطرائق الحديثة لنموذج GM(1,1) لايجاد القيم المفقودة و تقدير المعلمات مع تطبيق عملي، مجلة العلوم الاقتصادية والإدارية، العدد (102) المجلد (24)، 404-422.

[2] كريمة بازينة، فاطمة الكريك، علي عبد الشاهد .(2021). أمثلية شبكات Wi-Fi باستخدام تقنيات مستوحاة من الطبيعة. مجلة البحوث الأكاديمية، العدد (17)، 86-94

ب . المراجع الأجنبية

[3] X. Yu and M. Gen, Introduction to evolutionary algorithms: Springer Science & Business Media, 2010.

[4] M. Kaur, et al., "*Binary cuckoo search metaheuristic-based supercomputing framework for human behavior analysis in smart home*," The Journal of Supercomputing, vol. 76, pp. 2479-2502, 2020.

[5] I. Vilović and N. Burum, "*Location optimization of wlan access points based on a neural network model and evolutionary algorithms*," automatika, vol. 55, pp. 317-329, 2014

[6] X.-S. Yang, Nature-inspired metaheuristic algorithms: Luniver press, 2010.

[7] K. Yuxiao, et al., "*Variable order fractional grey model and its application*," Applied Mathematical Modelling, vol. 97, pp. 619-635, 2021.

[8] F. Heppner, "*A stochastic nonlinear model for coordinated bird flocks,*" The ubiquity of chaos, 1990.

[9] J. Kennedy and R. Eberhart, "*Particle swarm optimization*," in Proceedings of ICNN'95-international conference on neural networks, 1995, pp. 1942-1948.

[10] Y. Shi and R. Eberhart, "*A modified particle swarm optimizer*," in 1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360), 1998, pp. 69-73.