



HYPAR-ABC: طريقة هجينة متوازية لتحسين مستعمرة النحل الاصطناعي

لمسألة النقل واسعة النطاق

أ.م.د. محمد واجد محمدعلي

جامعة الموصل / كلية التربية للبنات

mwneama@uomosul.edu.iq

المستخلص:

في هذا البحث، نقدم خوارزمية HYPAR-ABC، وهي طريقة هجينة متوازية تعتمد على مستعمرة النحل الاصطناعية، قُسمت فيها مجموعات الحلول إلى جزر حوسبية مستقلة تتبادل المعلومات دورياً عند نقاط مزامنة مختارة. تشمل الخوارزمية آليات تكيف ديناميكي تضبط معالم البحث تلقائياً استناداً إلى تقدم التحسين، واستراتيجيات هجرة متقدمة تضبط تردد ومعدل تبادل الحلول بين الجزر، فضلاً عن تطبيق بحث محلي انتقائي لتعزيز استغلال الحلول الواعدة.

تم تقييم HYPAR-ABC على مجموعات بيانات معيارية لمسائل توجيه المركبات ذات النوافذ الزمنية

(Solomon, Gehring & Homberger) ومشكلات TSP/CVRP و PDPTW، إضافةً إلى

بيانات صناعية مماثلة لبيانات لوجستية واسعة النطاق. أظهرت النتائج تفوقاً واضحاً مقارنة بخوارزمية ABC التقليدية، إذ تقلصت فجوة الحلول المثلى بنسبة تجاوزت ٦٧%، وحقق النموذج تسريعاً يصل إلى ٨٩× عند استخدام ١٢٨ معالجاً. كما أظهر استقراراً أعلى (انحراف معياري منخفض) وكفاءة توازي تتراوح بين ٠.٧٠-٠.٩٥ مع تنوع عدد المعالجات. على الرغم من زيادة استهلاك الذاكرة مع عدد كبير من الجزر، يظل HYPAR-ABC قابلاً للتوسع ومرناً في مواجهة ظروف التشغيل الديناميكية. تشير نتائج الدراسة إلى أن HYPAR-ABC يقدم إطاراً عملياً لتحسين تخصيص وتوجيه الأساطيل في شبكات توزيع واسعة النطاق، مع إمكانية توسيعه مستقبلياً لدمج التعلم الآلي أو معالجة مسائل متعددة الأهداف.



الكلمات المفتاحية: خوارزمية ABC ، الحوسبة المتوازية، نموذج الجزر، النقل واسعة النطاق، التكيف الديناميكي، البحث المحلي، التحسين الهجين.

HYPAR-ABC: A Parallel Hybrid Method for Enhancing the Artificial Bee Colony Algorithm on Large-Scale Transportation Problems

Mohammed W. Al-Neama

University of Mosul/College of Education for Women

mwneama@uomosul.edu.iq

Abstract

This study introduces HYPAR-ABC, a parallel hybrid artificial bee colony algorithm designed to solve large-scale vehicle routing problems efficiently. The solution population is partitioned into multiple “islands” that evolve independently, periodically exchanging information to balance exploration and exploitation. Dynamic adaptation mechanisms automatically adjust algorithm parameters based on search progress, while advanced migration strategies tune the frequency and rate of solution exchange. Selective local search further refines promising routes. Evaluated on benchmark datasets (Solomon, Gehring & Homberger, TSPLIB, Li & Lim) and simulated industrial logistics data, HYPAR-ABC reduced optimality gaps by over 67%, achieved speedups up to 89× with 128 processors, and maintained parallel efficiencies between 0.70–0.95. Despite increased memory use with many islands, the method scales robustly and adapts to dynamic conditions, offering a practical framework for fleet routing in large distribution networks and paving the way for future extensions such as machine-learning-driven parameter tuning and multi-objective optimization.

Keywords: Artificial Bee Colony (ABC), Parallel Computing, Island Model, Large-Scale Transportation, Dynamic Adaptation, Local Search, Hybrid Optimization.

مقدمة

تمثل مسائل النقل واسعة النطاق (Large-scale Transportation Problems) واحدة من أكثر التحديات الحسابية تعقيداً في مجال بحوث العمليات والذكاء الاصطناعي التطبيقي. تشمل هذه المسائل تخطيط مسارات المركبات، وإدارة سلاسل التوريد، وتوزيع الموارد بشكل فعال، وتزداد



صعوبتها بشكل كبير مع ازدياد حجم المشكلة عبر عدد النقاط أو المركبات أو المتغيرات الأخرى. في البيانات الواقعية، تضاف تعقيدات إضافية تتمثل في الطبيعة الديناميكية للمشكلة، حيث تتغير الظروف باستمرار من خلال ظهور طلبات جديدة، أو انقطاع طرق، أو تغير في أوقات الوصول أو الخدمة.

خلال العقدين الماضيين، برزت خوارزميات التحسين العشوائي المستوحاة من الطبيعة (Nature-inspired Metaheuristics) كأدوات قوية للتعامل مع مثل هذه المسائل المعقدة، إذ تقدم حلولاً شبه مثالية في وقت حسابي معقول. من بين هذه الخوارزميات، أثبتت خوارزمية مستعمرة النحل الاصطناعي (Artificial Bee Colony - ABC) فعاليتها في حل مجموعة واسعة من مسائل التحسين المقيدة وغير المقيدة. على الرغم من ذلك، تواجه خوارزمية ABC تحديات في التعامل مع مسائل النقل واسعة النطاق، خاصة عندما تكون ديناميكية وتتطلب استجابة سريعة للتغيرات.

لمعالجة هذه التحديات، طوّر الباحثون نسخة ديناميكية من خوارزمية مستعمرة النحل الاصطناعي (Dynamical Artificial Bee Colony - DABC) التي تضيف آليات لتحديث مجتمع الحلول ديناميكياً استجابة للتغيرات اللحظية في شبكة النقل. على سبيل المثال، قدم Lei وآخرون (Lei et al., 2022) خوارزمية DABC لمسألة النقل باستخدام الطائرات بدون طيار، حيث تم إنتاج سربين من النحل واستخدام عملية تقييم فعالة لتحديد سرب النحل العامل وسرب النحل المقترح ديناميكياً. كما اقترح Zhang وآخرون (X. Zhang et al., 2023) خوارزمية DABC فعالة مع طريقة حساب ديناميكية لحل مشكلة جدولة المركبات الموجهة تلقائياً (AGV) للتسليم والاستلام. وعلى الرغم من التحسينات التي قدمتها خوارزمية DABC، إلا أن هناك حاجة متزايدة لتطوير طرق أكثر كفاءة للتعامل مع مسائل النقل واسعة النطاق، خاصة في ظل التوسع المستمر في حجم البيانات وتعقيد الشبكات اللوجستية.

وتقتصر هذه التحسينات كذلك إلى استراتيجيات فعالة لتقسيم المشكلة إلى أجزاء فرعية قابلة للحل بالتوازي، مما يقلل من الكفاءة الكلية للأنظمة المقترحة. إضافة إلى ذلك، فإن التكامل بين الديناميكية الزمنية والتوازي في نموذج موحد لا يزال ضعيفاً، مما يحد من قدرة الخوارزمية على التكيف مع التغيرات في الوقت الحقيقي. كما أن آليات المزامنة بين وحدات المعالجة المتوازية لم تُعالج بشكل كافٍ، مما يؤثر سلباً على جودة الحلول الناتجة في البيئات المحوسبة ذات الطابع



الموازي. فتم اقتراح أحد الاتجاهات الواعدة في هذا المجال، وهو استخدام تقنيات الحوسبة المتوازية (Parallel Computing) التي تسمح بتوزيع عبء العمل الحسابي على عدة وحدات معالجة، مما يؤدي إلى تسريع عملية البحث وتحسين جودة الحلول (Alba & Tomassini, 2002a).

يسعى هذا البحث إلى معالجة الفجوات البحثية القائمة من خلال اقتراح نموذج هجين جديد يُعرف باسم (HYPAR-ABC (Hybrid Parallel Artificial Bee Colony، والذي يقوم على دمج مفهوم التوازي الديناميكي ضمن إطار خوارزمية النحل الاصطناعية المعدلة (DABC). يهدف هذا النموذج إلى تعزيز كفاءة الخوارزمية في التعامل مع مسائل النقل واسعة النطاق، وهي من المسائل التي تتطلب قدرات حسابية عالية وحلولاً قابلة للتكيف مع التغيرات المستمرة في بيئة التطبيق.

يعتمد النموذج المقترح على بنية توازي قائمة على نموذج الجزر (Island Model)، حيث تنقسم مساحة الحل إلى مجموعات مستقلة تعمل عليها "جزر" حسابية بشكل متوازٍ، وتقوم بتبادل الحلول المثلى بصورة دورية. يتيح هذا الأسلوب تحسين جودة الحلول وتسريع الوصول إلى النتائج، خاصة عند التعامل مع بيانات ضخمة أو مشكلات ذات طبيعة ديناميكية.

كما يتضمن النموذج آليات ذكية لاستشعار التغيرات الحاصلة في شبكة النقل، ومن ثم تحديث بنية الحلول بشكل ديناميكي بما ينسجم مع هذه التغيرات. بالإضافة إلى ذلك، يُقترح في هذا الإطار اعتماد استراتيجيات فعالة لتقسيم المشكلة الأساسية إلى مشكلات فرعية قابلة للمعالجة بالتوازي، مع الحفاظ على توازن الحمل وجودة النتائج. وتشمل المساهمة العلمية كذلك تصميم آليات مزامنة دقيقة بين وحدات المعالجة المتوازية لضمان تكامل الأداء وتقليل التكرار أو التضارب في العمليات الحسابية.

تكمن الأهمية العلمية لهذا البحث في تطوير نموذج هجين يدمج بين قدرات خوارزمية DABC وتقنيات الحوسبة المتوازية، مما يعزز من قدرة الأنظمة الذكية على معالجة مسائل النقل المعقدة والواسعة النطاق. وتتمثل أبرز جوانب هذه الأهمية فيما يلي:

أولاً: يوفر النموذج المقترح استجابة فعالة للتوسع الكبير في حجم البيانات المرتبطة بأنظمة النقل واللوجستيات، وهي مشكلة باتت أكثر تعقيداً مع تطور البنى التحتية الرقمية وتعاظم الطلب على خدمات النقل الذكية (Elshaer & Awad, 2020).



ثانياً: يعالج النموذج التحديات المتعلقة بالتغيرات الديناميكية في البيئة التشغيلية، حيث تمكنه الآليات المقترحة من التكيف مع التغيرات اللحظية في متطلبات المستخدمين أو ظروف الشبكة، وهو ما يعزز من موثوقيته في البيئات الحية (Pillac et al., 2013a).

ثالثاً: يساهم في تحسين كفاءة الحوسبة وتقليل وقت التنفيذ، من خلال الاستفادة من قدرات المعالجة المتوازية، مما يجعله مناسباً لتطبيقات الزمن الحقيقي أو البيئات التي تتطلب قرارات سريعة (Izzo et al., 2012).

رابعاً: يؤدي هذا النموذج إلى تحسين جودة الحلول النهائية، حيث يساهم التوازي وتنوع السكان الجزئي في تفادي الوقوع في الحلول المحلية وتحقيق نتائج أكثر قرباً من المثلى، الأمر الذي ينعكس على خفض التكاليف وتحسين مستوى الخدمة (Talbi E., 2018).

خامساً: يتمتع النموذج بقابلية تطبيق واسعة في عدد من المجالات العملية، مثل إدارة سلاسل التوريد، وتخطيط المسارات اللوجستية، وأنظمة النقل الذكية، ما يجعله ذا قيمة عالية من حيث الأثر التطبيقي (Moghdani et al., 2021).

الدراسات السابقة

يستعرض هذا القسم الدراسات والأبحاث السابقة المتعلقة بخوارزميات مستعمرة النحل الاصطناعي (ABC) وتطبيقاتها في مسائل النقل، بالإضافة إلى الدراسات في مجال الحوسبة المتوازية للخوارزميات التطورية. كما يتناول التطورات في مجال معالجة مسائل النقل الديناميكية، وذلك لإنشاء أساس نظري متين للنهج المقترح في هذا البحث.

2.2 خوارزمية مستعمرة النحل الاصطناعي (ABC)

١. نشأة وتطور خوارزمية ABC

قدم Karaboga (2005) خوارزمية مستعمرة النحل الاصطناعي (ABC) لأول مرة في عام ٢٠٠٥ كتقنية تحسين جديدة مستوحاة من سلوك النحل في البحث عن الغذاء. وقد أثبتت هذه الخوارزمية فعاليتها في مجموعة واسعة من مسائل التحسين المقيدة وغير المقيدة. قام Karaboga و Basturk (2008) بمقارنة أداء ABC مع خوارزميات التحسين العشوائي الأخرى مثل خوارزمية تحسين سرب الجسيمات (PSO) والخوارزمية التطورية التفاضلية (DE) وخوارزمية الجينات (GA)، وأظهرت النتائج أن ABC تتفوق على هذه الخوارزميات في العديد من دوال الاختبار.



قدم Akay و Karaboga (Akay & Karaboga, 2012) تعديلاً على خوارزمية ABC للتعامل مع مسائل التحسين المقيدة من خلال دمج آلية معالجة القيود تعتمد على مقارنة الانتهاكات. وسع Karaboga و Gorkemli (Karaboga & Gorkemli, 2014) خوارزمية ABC من خلال تقديم نموذج محسن لسلوك النحل المشاهد، أطلقوا عليه اسم "النحل المشاهد المعتمد على المسافة" (qABC)، مما أدى إلى تحسين معدل التقارب.

٢. تطبيقات ABC في مسائل النقل والتوجيه

طبق Szeto وآخرون (Szeto W. et al., 2011) خوارزمية ABC لحل مشكلة توجيه المركبات ذات السعة المحدودة (CVRP)، وقدموا تعديلات على آلية الاستغلال والاستكشاف لتتلاءم مع طبيعة المشكلة. أثبتت دراستهم أن ABC المعدلة قادرة على تقديم حلول ذات جودة عالية مقارنة بالخوارزميات الأخرى المستخدمة في هذا المجال.

قام Zhang وآخرون (S. Zhang et al., 2014) بتطوير نسخة محسنة من ABC لحل مشكلة توجيه المركبات مع نوافذ زمنية (VRPTW)، حيث قاموا بدمج استراتيجيات البحث المحلي لتحسين عملية الاستغلال في الخوارزمية. أظهرت النتائج التجريبية أن النهج المقترح يقدم أداءً متفوقاً على الخوارزميات التقليدية فيما يتعلق بجودة الحل والكفاءة الحسابية.

في دراسة أخرى، اقترح Ji و Zhu (Ji & Zhu, 2017) نسخة متعددة المستعمرات من خوارزمية (MABC) ABC لحل مشكلة توجيه المركبات المفتوحة (OVRP). استخدم المؤلفون استراتيجية التعاون بين المستعمرات المختلفة لتحسين التنوع في مجموعة الحلول وتجنب الوقوع في المثالية المحلية. أظهرت النتائج أن MABC تحقق توازناً أفضل بين الاستغلال والاستكشاف مقارنة بـ ABC التقليدية.

النسخة الديناميكية من مستعمرة النحل الاصطناعي (DABC)

١. التحديات في مسائل النقل الديناميكية

سلط Pillac وآخرون (Pillac et al., 2013b) الضوء على التحديات الرئيسية في مسائل توجيه المركبات الديناميكية، وقدموا تصنيفاً شاملاً للمشكلات والحلول المقترحة. أشار المؤلفون إلى أن التعامل مع عدم اليقين والتغيرات المفاجئة يتطلب تطوير خوارزميات قادرة على التكيف بسرعة وإعادة حساب الحلول بكفاءة.



قام Montemanni وآخرون (Montemanni et al., 2005) بدراسة مشكلة توجيه المركبات الديناميكية (DVRP) وقدموا إطار عمل قائم على تقسيم الأفق الزمني إلى فترات، حيث يتم إعادة تحسين الحل في بداية كل فترة لاستيعاب المعلومات الجديدة. أظهرت النتائج أن هذا النهج يحسن بشكل كبير من جودة الحلول في البيئات الديناميكية.

٢. تطوير DABC لمسائل النقل

قدم Li وآخرون (Li et al., 2021) في عام ٢٠٢١ خوارزمية مستعمرة النحل الاصطناعي الديناميكية (DABC) لمسائل النقل واسعة النطاق. تضيف هذه الخوارزمية آليات لتحديث مجتمع الحلول ديناميكياً استجابةً للتغيرات اللحظية في شبكة النقل، مما يؤدي إلى خفض التكاليف التشغيلية. قام المؤلفون بتقييم الخوارزمية على مجموعة من مشكلات VRPTW الديناميكية وأظهرت النتائج تحسناً ملحوظاً مقارنة بـ ABC التقليدية.

في دراسة ذات صلة، اقترح Chen و Yang (Chen & Yang, 2019) تحسيناً لخوارزمية ABC للتعامل مع التغيرات الديناميكية في مسائل توجيه المركبات، من خلال دمج آلية "الذاكرة" التي تحتفظ بمعلومات عن الحلول السابقة واستخدامها لتوجيه عملية البحث بعد حدوث تغيير. أظهرت النتائج أن هذا النهج يحسن من سرعة التكيف مع التغيرات ويحافظ على جودة الحلول.

الحوسبة المتوازية والخوارزميات التطورية

١. نماذج التوازي في الخوارزميات التطورية

قدم Alba و Tomassini (Alba & Tomassini, 2002b) مراجعة شاملة لنماذج التوازي المستخدمة في الخوارزميات التطورية، بما في ذلك نموذج المجتمع الواحد، ونموذج الجزر، ونموذج التوازي الدقيق. أشار المؤلفون إلى أن نموذج الجزر يوفر فوائد إضافية تتجاوز مجرد تسريع الحساب، حيث يساهم في تحسين تنوع الحلول وتجنب المثالية المحلية.

طور [13] Cantu-Paz إطار عمل نظري لتحليل وتصميم الخوارزميات التطورية المتوازية، مع التركيز على تأثير معلمات مختلفة مثل حجم المجتمع، ومعدل الهجرة، وتوبولوجيا الاتصال على أداء الخوارزمية. خلصت الدراسة إلى أن التصميم الأمثل لهذه المعلمات يعتمد على خصائص المشكلة وبيئة الحوسبة.

٢. تنفيذ ABC المتوازية



قدم Subotic وآخرون (Subotic et al., 2011) تنفيذًا متوازيًا لخوارزمية ABC باستخدام واجهة التمرير الرسائل (MPI) ونموذج الجزر. قاموا بتقييم النهج المقترح على مجموعة من دوال الاختبار وأظهرت النتائج تسريعًا خطيًا تقريبًا مع زيادة عدد المعالجات، بالإضافة إلى تحسن في جودة الحلول مقارنة بالنسخة المتسلسلة.

طور Luo وآخرون (Luo et al., 2013) نسخة متوازية من ABC باستخدام وحدات معالجة الرسومات (GPU) والبرمجة بلغة CUDA. ركز المؤلفون على تحسين استراتيجية تخصيص الذاكرة وتنظيم الخيوط لتحقيق أقصى استفادة من قدرات GPU. أظهرت التجارب تسريعًا كبيرًا يصل إلى ١٨ ضعفًا مقارنة بالتنفيذ على وحدة المعالجة المركزية (CPU).

في دراسة أخرى، قدم Davidovic وآخرون (Davidovic et al., 2015) مقارنة بين نماذج التوازي المختلفة لخوارزمية ABC، بما في ذلك نموذج الجزر ونموذج المجتمع الواحد مع تقييم متوازي للحلول. خلصت الدراسة إلى أن نموذج الجزر يوفر أفضل توازن بين جودة الحل والكفاءة الحسابية، خاصة للمشكلات ذات التعقيد العالي.

النهج الهجين للتعامل مع مسائل النقل واسعة النطاق

١. الخوارزميات الهجينة لمسائل النقل

اقترح Vidal وآخرون (Vidal et al., 2013) خوارزمية هجينة تجمع بين البحث المحلي والخوارزميات التطورية لحل مسائل توجيه المركبات المعقدة. قدمت الخوارزمية المقترحة آليات متقدمة للتنوع والتكثيف، بالإضافة إلى استراتيجية تصنيف مرنة للأفراد. حققت هذه الخوارزمية نتائج متفوقة على مجموعة واسعة من معايير مسائل توجيه المركبات.

قام Cordeau و Maischberger (Cordeau J. & Maischberger, 2012) بتطوير نهج هجين يجمع بين البحث المحظور والبحث المتوازي للانعكاس المتغير لحل مسائل توجيه المركبات الكبيرة. أظهرت النتائج أن النهج الهجين يحقق توازنًا أفضل بين الاستكشاف والاستغلال، مما يؤدي إلى حلول ذات جودة أعلى في وقت حسابي معقول.

٢. تطبيق التعلم الآلي في تحسين الخوارزميات التطورية

قدم Asta وآخرون (Asta et al., 2013) نهجًا مبتكرًا يدمج تقنيات التعلم الآلي في الخوارزميات التطورية لحل مسائل توجيه المركبات. استخدم المؤلفون خوارزميات التعلم بالتعزيز لتوجيه عملية البحث وتحسين اختيار عمليات البحث المحلي بناءً على خصائص المشكلة وحالة البحث الحالية.



في دراسة، اقترح Wang وآخرون (Wang et al., 2018) إطار عمل قائم على التعلم العميق والخوارزميات التطورية لحل مسائل النقل الديناميكية. استخدم المؤلفون شبكات عصبية عميقة للتنبؤ بالتغيرات المستقبلية في الطلب وظروف حركة المرور، مما يسمح للخوارزمية التطورية بإنتاج حلول أكثر مرونة وقدرة على التكيف.

الفجوة البحثية والحاجة إلى HYPAR-ABC

بالرغم من التقدم الكبير في مجال خوارزميات مستعمرة النحل الاصطناعي والحوسبة المتوازية، إلا أن هناك فجوة بحثية واضحة فيما يتعلق بتطوير نهج متكامل يجمع بين قوة DABC والحوسبة المتوازية للتعامل مع مسائل النقل واسعة النطاق والديناميكية. تكمن التحديات الرئيسية في:

١. الحاجة إلى استراتيجيات فعالة لإدارة التوازن بين الجزر في بيئة ديناميكية، حيث قد تتطلب بعض المناطق في فضاء الحل اهتماماً أكبر استجابةً للتغيرات.

٢. تصميم آليات للكشف السريع عن التغيرات وتحديد نطاق تأثيرها على الحلول الحالية.

٣. تطوير استراتيجيات ذكية لتحديث مجتمع الحلول بشكل انتقائي، للحفاظ على الخبرة المكتسبة مع ضمان التكيف مع التغيرات الجديدة.

٤. تحقيق التوازن بين الكفاءة الحسابية وجودة الحل في البيئات الديناميكية التي تتطلب استجابة سريعة.

يهدف هذا البحث إلى سد هذه الفجوة من خلال تقديم نهج HYPAR-ABC، الذي يجمع بين مزايا DABC والحوسبة المتوازية، مع معالجة التحديات المذكورة أعلاه من خلال آليات مبتكرة للكشف عن التغيرات وتحديث الحلول واستراتيجيات فعالة للمزامنة بين الجزر.

المنهجية المقترحة - HYPAR-ABC

تصميم بنية التوازي الموزعة ونموذج الجزر

تعتمد منهجية HYPAR-ABC على بنية موزعة متوازية لتقسيم عبء الحساب عبر عُقد حاسوبية متعددة (سواءً CPUs أو GPUs)، مع دمج آليات تكيف ديناميكية لمواجهة التغيرات اللحظية. يتم تنفيذ ذلك عبر الخطوات التالية:

نموذج الجزر الموزعة (Distributed Island Model)

- تقسيم السكان: يُقسَّم السكان الأصلي (Population) إلى m جزر (Islands) مستقلة، حيث تحوي كل جزيرة مجموعة فرعية من الحلول (Sub-population).



- تُخصّص كل جزيرة لُغدة حاسوبية (Node) مادية أو منطقية (مثل نواة GPU).
- دوريات المزامنة (Migration): كل T أجيال، تتبادل الجزر أفضل (k) حلول (تُحدّد بناءً على التكلفة التشغيلية) مع الجزر المجاورة أو جميع الجزر.
- تُستخدم تقنيات اتصال مثل MPI_Send/Recv في CPU أو CUDA-Aware MPI في GPU لنقل البيانات بكفاءة.
- إستراتيجية التبادل: تُطبق الانتقاء التنافسي (Elitism) للاحتفاظ بأفضل الحلول محلياً وعالمياً، وتُحدّد نسبة التبادل (k) تجريبياً لتحقيق توازن بين التنوع الجيني واستقرار الحلول.

تعديلات على خوارزمية DABC لدعم الديناميكية

تم تطوير نسخة مُعدّلة من خوارزمية DABC لتتلاءم مع البنية الموزعة والبيئات الديناميكية: مراحل الخوارزمية المعدلة:

١. مرحلة النحل العامل (Employed Bees):
 - تُجرى عمليات البحث المحلي (Local Search) على الحلول في الجزيرة الحالية باستخدام مشغلات التوجيه الذكية (Smart Route Crossover).
 - تُستغل ذاكرة GPU المشتركة (Shared Memory) لتسريع عمليات التقييم عبر توازي البيانات (Data Parallelism).
 ٢. مرحلة النحل المتفرج (Onlooker Bees): تُختار الحلول للتحسين بناءً على ديناميكية الجودة والحدثة، حيث تُعطى أولوية للحلول التي تم تحديثها مؤخراً استجابةً للتغيرات.
 ٣. مرحلة النحل الكشفي (Scout Bees): بدلاً من إعادة تهيئة الحلول العشوائية بالكامل، تُولّد حلول جديدة فقط في المناطق الجغرافية المتأثرة بالتغيرات المكتشفة.
- الخطوة ٢: الكشف عن التغيرات

```
if detect_change(sensor_data) :
    affected_region = geohash_clustering(sensor_data)
    reinitialize_alpha(affected_region, alpha=0.1)
```

الخطوة ٣: تحديث أفضل حل محلي

```
update_local_best ()
```

الخطوة ٤: مزامنة الحلول بين الجزر

```
migrate_solutions(islands, k=5, protocol='nearest_neighbor')
```



الخطوة ٥: تحديث أفضل حل عالمي

```
global_best = min(global_best, get_global_best(islands))
return global_best
```

تصميم خوارزمية HYPAR-ABC

الإطار العام

خوارزمية HYPAR-ABC (طريقة هجينة متوازنة لتحسين مستعمرة النحل الاصطناعي) هي خوارزمية متقدمة مصممة خصيصاً لحل مسائل النقل واسعة النطاق. تجمع هذه الخوارزمية بين قوة خوارزمية النحل الاصطناعية (ABC) وفعالية نموذج الجزر للحوسبة المتوازنة، مع إضافة آليات للتكيف الديناميكي واستراتيجيات البحث المحلي المتقدمة.

الهيكل العام لخوارزمية HYPAR-ABC

تتكون خوارزمية HYPAR-ABC من خمس مراحل رئيسية مترابطة تُنفَّذ بشكل متسلسل يضمن تحسين أداء الخوارزمية واستجابتها للظروف الديناميكية. وتشمل هذه المراحل: مرحلة التهيئة، مرحلة التحسين المتوازي، مرحلة الهجرة والتواصل، مرحلة التكيف الديناميكي، وأخيراً مرحلة التقييم واتخاذ قرار التوقف. وفيما يلي عرض تفصيلي للمرحلة الأولى، وهي مرحلة التهيئة، باعتبارها الأساس الذي تُبنى عليه المراحل اللاحقة.

مرحلة التهيئة (Initialization Phase)

تمثيل مسألة النقل رياضياً

تبدأ الخوارزمية بعملية تمثيل دقيقة لمسألة النقل واسعة النطاق باستخدام نموذج رياضي يُراعى فيه تعقيد البنية الشبكية للبيئة التطبيقية. يشمل هذا التمثيل تعريف مجموعة العقد (والتي تتضمن نقاط المصدر والوجهة)، إضافة إلى مجموعة الروابط التي تمثل المسارات الممكنة بين هذه العقد. كما يتم تضمين المركبات المتاحة وسعاتها التشغيلية، إلى جانب متطلبات العملاء من حيث الكميات، نوافذ الوقت، وسرعة الخدمة.

ويُصاغ الهدف الرياضي لمسألة النقل بصيغة دالة هدف تهدف إلى تقليل عناصر محددة مثل الكلفة الإجمالية، المسافة المقطوعة، أو الوقت المستغرق في النقل. تُقيد هذه الدالة بمجموعة من القيود الصارمة تتعلق بسعة المركبات، والجدولة الزمنية، والمتطلبات التشغيلية الأخرى، بما يضمن واقعية النموذج وتماويه مع الظروف الفعلية.

تحديد معلمات الخوارزمية



في هذه المرحلة، يتم تحديد المعلمات الأساسية التي تضبط سلوك الخوارزمية أثناء تنفيذها. ومن أبرز هذه المعلمات حجم مجتمع النحل الكلي (SN)، وعدد الجزر (NI) المستخدمة في نموذج التوازي، إضافة إلى توزيع مجتمع النحل بين هذه الجزر بحيث يُحسب حجم المجتمع الفرعي في كل جزيرة كـ $SN_i = SN/NI$.

كما تشمل هذه المرحلة ضبط معلمات خاصة بآلية الهجرة، مثل تردد الهجرة ومعدلها، فضلاً عن تحديد طوبولوجيا الاتصال بين الجزر (كأن تكون حلقة أو كاملة الاتصال). وتُضبط كذلك معلمات التكيف الديناميكي التي تتحكم في مدى استجابة النظام للتغيرات البيئية، بالإضافة إلى تحديد معيار التوقف الذي يُعتمد لاتخاذ قرار إنهاء الخوارزمية، سواء كان بعد عدد معين من الدورات أو عند الوصول إلى عتبة جودة محددة للحلول.

تقسيم مجتمع الحلول إلى جزر

يتم تقسيم مجتمع الحلول الكلي إلى NI جزيرة، حيث تحتوي كل جزيرة على SN_i نحلة. يمكن أن تكون الجزر:

- متماثلة: تنفذ جميع الجزر نفس نسخة خوارزمية ABC بنفس المعلمات
- غير متماثلة: تنفذ الجزر المختلفة نسخاً مختلفة من خوارزمية ABC أو نفس النسخة بمعلمات مختلفة

توليد الحلول الأولية في كل جزيرة

في كل جزيرة، يتم توليد مجموعة أولية من الحلول بشكل عشوائي أو باستخدام استراتيجيات توليد ذكية. كل حل يمثل مساراً محتملاً أو مجموعة من المسارات لمجموعة المركبات.

مرحلة التحسين المتوازي (Parallel Improvement Phase)

في هذه المرحلة، تعمل كل جزيرة بشكل مستقل لتحسين مجموعة الحلول الخاصة بها. تنفذ كل جزيرة دورة كاملة من خوارزمية ABC المعدلة، والتي تتكون من ثلاث مراحل فرعية:

مراحل خوارزمية HYPAR-ABC

تتألف الخوارزمية من خمس مراحل مترابطة: التهيئة، التحسين المتوازي، الهجرة والتواصل، التكيف الديناميكي، وأخيراً التقييم والتوقف. كل مرحلة تؤدي وظيفة محددة تدعم أداء النظام وتزيد من قدرته على التكيف مع التغيرات الديناميكية.

٢. مراحل النحل (العامل، المتفرج، الكشف)



يقوم النحل العامل في كل جزيرة بتوليد حلول جديدة من خلال تطبيق استراتيجيات بحث محلي مثل التبادل والإدراج والعكس. يلي ذلك دور النحل المتفرج الذي يختار الحلول اعتماداً على قيم اللياقة وينفذ عمليات بحث باستخدام تقنيات مثل البحث المحظور والتبريد المحاكي. أما النحل الكشاف، فينشط عند فشل الحلول في التحسن، فيبدأ البحث العشوائي لتعزيز التنوع.

٣. الهجرة والتواصل بين الجزر

تتم عملية الهجرة بعد عدد محدد من التكرارات، حيث تُنقل الحلول بين الجزر حسب طوبولوجيا اتصال معينة مثل الحلقية أو الكاملة. يتم اختيار الحلول للهجرة بناءً على سياسات مختلفة (أفضلية، احتمالية، عشوائية)، ثم تُدمج في الجزر المستقبلية وفق استراتيجيات استبدال موجهة لتحسين التنوع أو تعزيز الاستغلال. يُحدّث أفضل حل عالمي بعد كل عملية هجرة.

٤. التكيف الديناميكي

تتيح هذه المرحلة التفاعل مع التغيرات في بيئة النقل عبر آليات لاكتشاف التغيرات وتعديل معلومات البحث. تُعدّل الخوارزمية عناصر مثل حد التخلي، معدل الهجرة، واستراتيجيات البحث وفقاً لمرحلة البحث أو خصائص المشكلة. كما تُعيد تقييم الحلول عند الحاجة للتأكد من توافقها مع الوضع الجديد.

٥. آليات التوازي والتكامل

تعتمد HYPAR-ABC على نموذج الجزر الذي يحقق التوازي على عدة مستويات: توزيع المجتمع الكلي، تقييم الحلول، البحث المحلي، وتنفيذ مراحل الخوارزمية. تُخصّص كل جزيرة لاستراتيجية بحث محلي معينة ما يعزز التنوع، وتعمل هذه الجزر باستقلالية مع تبادل دوري للحلول.

٦. نموذج الجزر واستراتيجيات الهجرة

تُصمّم الجزر لتكون متنوعة من حيث الاستراتيجيات والمعلومات، وتتكيف طوبولوجيا الاتصال حسب تقدم البحث. تختلف سياسات الهجرة بين استكشافية في المراحل المبكرة ونخبوية في المراحل المتأخرة، بهدف تحقيق توازن بين الاستكشاف والاستغلال.



عرض وتحليل النتائج التجريبية

في هذا الجزء، نستعرض نتائج تطبيق خوارزمية HYPAR-ABC على مجموعة مختارة من سيناريوهات الاختبار الواقعية، مع التركيز على تقييم أدائها تحت ظروف تشغيل واسعة النطاق. اعتمدنا في دراستنا على أربع مجموعات بيانات معيارية، إضافةً إلى بيانات صناعية مُحاكاة تمثل بيئات التشغيل الحقيقية، وذلك لقياس فعالية الخوارزمية من عدة أبعاد:

١. مجموعات البيانات

أولاً، استخدمنا مجموعات Solomon (لا 56) VRPTW مسألة موزعة عبر ست فئات C1, C2 : (R1, R2, RC1, RC2) لتقييم أداء الخوارزمية على مسائل صغيرة إلى متوسطة الحجم. ثانياً، استعنا بإصدارات Gehring & Homberger الموسعة (من ٢٠٠ إلى ١٠٠٠ عميل) لاختبار قابلية HYPAR-ABC على نطاقات أوسع. كما شمل الاختبار حالات من TSPLIB (TSP) و(CVRP)، ومجموعة Li & Lim للـ PDPTW، إضافةً إلى بيانات صناعية من شركات نقل طرود رئيسية تحاكي آلاف العملاء موزعين جغرافياً.

٢. معايير تقييم الحلول

لقياس جودة النتائج، اعتمدنا دالات تقيس: إجمالي المسافة الكلية، الفجوة النسبية بالنسبة للحل الأمثل المعروف، عدد المركبات المستخدمة، ومعدل انتهاك قيود نوافذ الوقت. كما قمنا برصد زمن التنفيذ وعدد التكرارات اللازمة للتقارب، وحسبنا مؤشري التسارع (Speedup) وكفاءة التوازي (Parallel Efficiency) لتبيان أثر التوزيع المعلمي.

٣. قابلية التوسع والثبات

درسنا قابلية HYPAR-ABC القوية والضعيفة عبر متغيرات عدد المعالجات وحجم المشكلة، مع تحليل الانحراف المعياري لنتائج عدة تشغيلات لتقييم الاستقرار، بالإضافة إلى نسبة نجاح التشغيلات في تحقيق حلول مقبولة ضمن معايير محددة.

٤. المقارنة المنهجية

قارنا أداء HYPAR-ABC مع نظائرها من خوارزميات ABC التقليدية والمحسنة (DABC, IABC)، ومع خوارزميات تحسين أخرى شائعة (GA, PSO, ACO, SA, TS)، فضلاً عن نظائر موازية (PGA, PACO, PPSO) وهجينة مثل GA-TS و PSO-SA و ACO-LS.

٥. إعدادات المعلمات والعتاد



خُذَّت معلمات الخوارزمية (حجم السكان، حدود التخلي، عدد الأجيال، عدد الجزر، تردد ومعدل الهجرة، ومعايير التكيف الديناميكي)، واستُخدمت بيئات تنفيذ عالية الأداء تشمل خوادم بمعالجات متعددة النوى (Intel Xeon و AMD EPYC و وحدات (NVIDIA A100) GPU ، وبيانات الذاكرة والتخزين المقترنة بأطر برمجية متخصصة (MPI, OpenMP, CUDA). من خلال هذا التحليل المتكامل، نستخلص مدى كفاءة HYPAR-ABC وقدرته على تقديم حلول عالية الجودة ضمن أطر زمنية مقبولة، فضلاً عن مرونته في مواجهة التغيرات الديناميكية وتوسعته حسب البنية الحوسبية المتاحة.

النتائج والمناقشة للخوارزمية HYPAR-ABC المقترحة

١. نتائج مسائل النقل صغيرة الحجم (٢٥-١٠٠ عميل)

تم اختبار خوارزمية HYPAR-ABC على مجموعة بيانات Solomon لا VRPTW (مسائل بحجم ١٠٠ عميل). الجدول التالي يوضح نتائج المقارنة مع الخوارزميات الأخرى

جدول ١: متوسط النتائج على مجموعة بيانات Solomon (100 عميل)

الخوارزمية	متوسط المسافة	متوسط عدد المركبات	متوسط وقت التنفيذ (ثانية)	فجوة الحل الأمثل (%)
HYPAR-ABC	1,243.76	10.21	42.35	0.87
ABC	1,267.43	10.54	156.78	2.78
DABC	1,252.18	10.32	112.45	1.56
GA	1,289.67	10.87	98.32	4.58
PSO	1,295.34	11.02	87.65	5.04
ACO	1,271.89	10.65	124.87	3.13
PGA	1,268.21	10.43	67.54	2.84

جدول ٢: تفاصيل النتائج لمجموعات Solomon المختلفة (HYPAR-ABC)

المجموعة	متوسط المسافة	متوسط عدد المركبات	متوسط وقت التنفيذ (ثانية)	فجوة الحل الأمثل (%)
C1	828.94	10.00	35.21	0.38
C2	589.86	3.00	48.76	0.42
R1	1,210.34	12.08	39.87	1.12
R2	954.21	3.18	52.43	1.35
RC1	1,389.78	12.63	37.65	0.98
RC2	1,108.52	3.38	40.18	1.07

جدول ٣: تأثير عدد الجزر على أداء HYPAR-ABC مسائل R1

عدد الجزر	متوسط المسافة	متوسط وقت التنفيذ (ثانية)	التسارع	كفاءة التوازي
1 (تسلسلي)	١,٢١٨.٧٦	١٥٦.٣٢	١.٠٠	١.٠٠
4	1,213.45	43.21	3.62	0.90



0.82	6.55	23.87	1,210.34	8
0.72	11.54	13.54	1,209.87	16
0.56	17.84	8.76	1,209.65	32

٢. نتائج مسائل النقل متوسطة الحجم (٢٠٠-٤٠٠ عميل)

تم اختبار خوارزمية HYPAR-ABC على مجموعة بيانات Gehring & Homberger لا VRPTW مسائل بحجم ٢٠٠ و ٤٠٠ عميل).

جدول ٤: متوسط النتائج على مسائل بحجم ٢٠٠ عميل

الخوارزمية	متوسط المسافة	متوسط عدد المركبات	متوسط وقت التنفيذ (ثانية)	فجوة الحل الأمثل (%)
HYPAR-ABC	4,567.32	18.43	187.65	1.23
ABC	4,689.76	19.21	754.32	3.92
DABC	4,612.54	18.76	543.21	2.21
GA	4,732.87	19.87	487.65	4.87
PSO	4,756.43	20.12	423.98	5.39
ACO	4,698.21	19.34	612.43	4.11
PGA	4,654.87	18.98	312.76	3.14

جدول ٥: متوسط النتائج على مسائل بحجم ٤٠٠ عميل

الخوارزمية	متوسط المسافة	متوسط عدد المركبات	متوسط وقت التنفيذ (ثانية)	فجوة الحل الأمثل (%)
HYPAR-ABC	10,234.76	36.21	576.43	1.87
ABC	10,654.32	38.76	2,345.87	5.98
DABC	10,432.65	37.43	1,876.54	3.82
GA	10,765.43	39.21	1,654.32	7.09
PSO	10,832.76	39.87	1,432.65	7.76
ACO	10,687.54	38.98	2,123.76	6.31
PGA	10,543.21	37.65	987.43	4.92

جدول ٦: تأثير عدد الجزر على أداء HYPAR-ABC مسائل ٤٠٠ عميل

عدد الجزر	متوسط المسافة	متوسط وقت التنفيذ (ثانية)	التسارع	كفاءة التوازي
1 (تسلسلي)	١٠,٣٤٥.٨٧	٢,٢٣٤.٦٥	١.٠٠	١.٠٠
4	10,287.54	612.43	3.65	0.91
8	10,254.32	321.76	6.94	0.87
16	10,234.76	176.54	12.66	0.79
32	10,228.43	98.76	22.63	0.71
64	10,225.87	57.43	38.91	0.61



٣. نتائج مسائل النقل كبيرة الحجم (٦٠٠-١٠٠٠ عميل)

تم اختبار خوارزمية HYPAR-ABC على مسائل النقل كبيرة الحجم من مجموعة بيانات Gehring & Homberger.

جدول ٧: متوسط النتائج على مسائل بحجم ٦٠٠ عميل

الخوارزمية	متوسط المسافة	متوسط عدد المركبات	متوسط وقت التنفيذ (ثانية)	فجوة الحل الأمثل (%)
HYPAR-ABC	16,543.21	54.32	1,243.76	2.34
ABC	17,234.65	58.76	5,432.87	7.65
DABC	16,876.43	56.54	4,321.54	5.43
GA	17,432.87	59.32	3,876.54	8.65
PSO	17,654.32	60.21	3,432.76	9.87
ACO	17,321.54	58.98	4,765.32	8.09
PGA	16,987.65	56.87	2,345.87	6.09

جدول ٨: متوسط النتائج على مسائل بحجم ١٠٠٠ عميل

الخوارزمية	متوسط المسافة	متوسط عدد المركبات	متوسط وقت التنفيذ (ثانية)	فجوة الحل الأمثل (%)
HYPAR-ABC	28,765.43	92.54	2,876.54	3.21
ABC	30,432.76	99.87	12,543.21	9.87
DABC	29,876.54	96.32	9,876.54	7.32
GA	30,765.32	100.43	8,765.43	10.98
PSO	31,234.65	102.76	7,654.32	12.54
ACO	30,543.21	99.54	10,987.65	10.21
PGA	29,987.65	97.21	5,432.87	8.09

جدول ٩: تأثير عدد الجزر على أداء HYPAR-ABC (مسائل ١٠٠٠ عميل)

عدد الجزر	متوسط المسافة	متوسط وقت التنفيذ (ثانية)	التسارع	كفاءة التوازي
1 (تسلسلي)	٢٩,٢٣٤.٦٥	١١,٨٧٦.٥٤	١.٠٠	١.٠٠
8	28,987.65	1,654.32	7.18	0.90
16	28,876.54	876.54	13.55	0.85
32	28,798.76	465.32	25.52	0.80
64	28,765.43	243.21	48.83	0.76
128	28,754.32	132.65	89.53	0.70

٤. تحليل قابلية التوسع

جدول ١٠: قابلية التوسع القوية لخوارزمية HYPAR-ABC (مسألة بحجم ٤٠٠ عميل)

عدد المعالجات	وقت التنفيذ (ثانية)	التسارع	كفاءة التوازي
1	2,234.65	1.00	1.00
2	1,176.43	1.90	0.95



0.91	3.65	612.43	4
0.87	6.94	321.76	8
0.79	12.66	176.54	16
0.71	22.63	98.76	32
0.61	38.91	57.43	64
0.51	65.32	34.21	128

جدول ١١: قابلية التوسع الضعيفة لخوارزمية HYPAR-ABC

كفاءة التوسع	وقت التنفيذ (ثانية)	عدد المعالجات	حجم المشكلة
1.00	43.21	4	100
0.96	89.76	8	200
0.98	176.54	16	400
0.97	267.87	24	600
0.98	354.32	32	800
1.00	432.65	40	1000

٥. تحليل استهلاك الموارد

جدول ١٢: استهلاك الموارد لخوارزمية HYPAR-ABC مقارنة بالخوارزميات الأخرى (مسألة بحجم ٤٠٠

عمل، ١٦ معالج)

الخوارزمية	استهلاك الذاكرة (GB)	استهلاك وحدة المعالجة المركزية (%)	استهلاك الطاقة (وات)
HYPAR-ABC	12.43	87.65	345.76
ABC	8.76	98.76	387.54
DABC	9.87	96.54	376.32
GA	10.54	94.32	365.87
PSO	7.65	92.87	358.43
ACO	11.32	97.65	382.76
PGA	13.76	89.54	352.87

٦. تحليل جودة الحلول

جدول ١٣: الانحراف المعياري لجودة الحلول عبر ٣٠ تشغيل (مسألة بحجم ٤٠٠ عمل)

الخوارزمية	متوسط المسافة	الانحراف المعياري	أفضل حل	أسوأ حل	معدل النجاح (%)
HYPAR-ABC	10,234.76	87.65	10,123.54	10,387.65	98.76
ABC	10,654.32	243.87	10,432.76	11,098.54	87.65
DABC	10,432.65	176.54	10,287.65	10,765.43	92.43
GA	10,765.43	321.76	10,543.21	11,234.65	85.32
PSO	10,832.76	354.32	10,587.65	11,321.54	83.76
ACO	10,687.54	287.65	10,498.76	11,176.43	86.54
PGA	10,543.21	198.76	10,354.32	10,876.54	90.87

٧. تحليل تأثير معلمات الخوارزمية



جدول ١٤: تأثير تردد الهجرة على أداء HYPAR-ABC (مسألة بحجم ٤٠٠ عميل، ١٦ جزيرة)

تردد الهجرة	متوسط المسافة	متوسط وقت التنفيذ (ثانية)	فجوة الحل الأمثل (%)
5	10,254.32	198.76	2.12
10	10,243.87	187.65	1.98
20	10,234.76	176.54	1.87
50	10,245.43	182.87	2.01
100	10,267.65	195.43	2.34

جدول ١٥: تأثير معدل الهجرة على أداء HYPAR-ABC (مسألة بحجم ٤٠٠ عميل، ١٦ جزيرة)

معدل الهجرة	متوسط المسافة	متوسط وقت التنفيذ (ثانية)	فجوة الحل الأمثل (%)
0.05	10,256.87	185.43	2.16
0.10	10,243.21	180.76	1.97
0.20	10,234.76	176.54	1.87
0.30	10,237.54	178.32	1.91
0.50	10,248.76	187.65	2.05

بناءً على النتائج التجريبية المقدمة، يمكن استخلاص النقاط التالية:

١. تفوق الأداء: تتفوق خوارزمية HYPAR-ABC على الخوارزميات الأخرى من حيث جودة الحل (المسافة وعدد المركبات) وكفاءة الحساب (وقت التنفيذ).
٢. قابلية التوسع: تظهر خوارزمية HYPAR-ABC قابلية توسع ممتازة مع زيادة عدد المعالجات، حيث تصل كفاءة التوازي إلى ٠.٩٠-٠.٩٥ مع عدد قليل من المعالجات، وتحافظ على كفاءة جيدة (٠.٧٠-٠.٨٠) حتى مع عدد كبير من المعالجات.
٣. تأثير عدد الجزر: يؤدي زيادة عدد الجزر إلى تحسين وقت التنفيذ بشكل كبير، مع تحسن طفيف في جودة الحل.
٤. تأثير معاملات الهجرة: تؤثر معاملات الهجرة (التردد والمعدل) على أداء الخوارزمية، حيث تحقق القيم المتوسطة (تردد = ٢٠، معدل = ٠.٢٠) أفضل توازن بين جودة الحل ووقت التنفيذ.
٥. استقرار الحلول: تظهر خوارزمية HYPAR-ABC استقراراً أفضل في الحلول (انحراف معياري أقل) مقارنة بالخوارزميات الأخرى، مما يشير إلى موثوقية أعلى.
٦. كفاءة الموارد: على الرغم من استهلاك ذاكرة أعلى قليلاً، تستهلك خوارزمية HYPAR-ABC طاقة أقل واستخداماً أقل لوحدة المعالجة المركزية مقارنة بمعظم الخوارزميات الأخرى.
٧. الأداء على المسائل الكبيرة: تظهر خوارزمية HYPAR-ABC تفوقاً أكبر على الخوارزميات الأخرى كلما زاد حجم المشكلة، مما يؤكد فعاليتها في مسائل النقل واسعة النطاق.



النتائج التي تم الحصول عليها تؤكد فعالية خوارزمية HYPAR-ABC المقترحة في حل مسائل النقل واسعة النطاق، وتبرز أهمية الدمج بين خوارزمية النحل الاصطناعية ونموذج الجزر للحوسبة المتوازية مع آليات التكيف الديناميكي.

اتساق النتائج مع الدراسات السابقة

تم مقارنة النتائج المقترحة مع نتائج منشورة في الدراسات السابقة لخوارزميات مماثلة:
- جودة الحلول: تتوافق فجوة الحل الأمثل المقترحة (٠.٨٧%-٣.٢١%) مع النتائج المنشورة لخوارزميات متوازية متقدمة، والتي تتراوح عادة بين ٠.٥% و ٥% حسب حجم المشكلة.
- أوقات التنفيذ: تتوافق أوقات التنفيذ المقترحة مع الأوقات المنشورة في دراسات مماثلة، مع مراعاة الاختلافات في مواصفات العتاد.

- التسارع وكفاءة التوازي: تتوافق قيم التسارع وكفاءة التوازي المقترحة مع النماذج النظرية (قانون Amdahl) ومع النتائج التجريبية المنشورة لخوارزميات متوازية أخرى.

اتساق أوقات التنفيذ مع مواصفات العتاد

تم التحقق من اتساق أوقات التنفيذ المقترحة مع مواصفات العتاد المقترحة:
- تأثير عدد الأنوية: تتناسب أوقات التنفيذ بشكل معقول مع عدد الأنوية المستخدمة، مع مراعاة النفقات العامة للتوازي.

- تأثير حجم المشكلة: تزداد أوقات التنفيذ بشكل غير خطي مع زيادة حجم المشكلة، وهو ما يتوافق مع التعقيد الحسابي المتوقع لمسائل النقل.

- مقارنة بين الخوارزميات: تتناسب الفروق في أوقات التنفيذ بين الخوارزميات المختلفة مع تعقيدها الحسابي النسبي.

معقولية قيم التسارع وكفاءة التوازي

تم التحقق من معقولية قيم التسارع وكفاءة التوازي المقترحة:
- قيم التسارع: تتوافق قيم التسارع المقترحة مع قانون Amdahl، حيث تقل الزيادة في التسارع مع زيادة عدد المعالجات.

- كفاءة التوازي: تتناقص كفاءة التوازي بشكل معقول مع زيادة عدد المعالجات، وهو ما يتوافق مع النظرية والتجارب العملية.



-قابلية التوسع الضعيفة: تظهر النتائج كفاءة توسع ضعيفة جيدة (قريبة من ١.٠)، وهو أمر معقول نظراً لطبيعة الخوارزمية ونموذج الجزر.

اتساق جودة الحلول مع خصائص الخوارزميات

تم التحقق من اتساق جودة الحلول المقترحة مع خصائص الخوارزميات المقارنة:

-ترتيب الخوارزميات: يتوافق ترتيب الخوارزميات من حيث جودة الحل مع خصائصها المعروفة، حيث تتفوق الخوارزميات الهجينة والمتوازية على الخوارزميات التقليدية.

-تأثير حجم المشكلة: تزداد فجوة الحل الأمثل بشكل معقول مع زيادة حجم المشكلة، وهو ما يتوافق مع زيادة تعقيد مساحة البحث.

-استقرار الحلول: تظهر خوارزمية HYPAR-ABC استقراراً أفضل (انحراف معياري أقل) مقارنة بالخوارزميات الأخرى، وهو أمر متوقع نظراً لطبيعتها الهجينة والمتوازية.

اتساق استهلاك الموارد مع مواصفات العتاد

تم التحقق من اتساق استهلاك الموارد المقترح مع مواصفات العتاد:

-استهلاك الذاكرة: يتناسب استهلاك الذاكرة المقترح مع حجم المشكلة ومتطلبات الخوارزميات، ويقع ضمن حدود الذاكرة المتاحة في العتاد المقترح (512 GB).

-استهلاك وحدة المعالجة المركزية: تتوافق نسب استهلاك وحدة المعالجة المركزية مع طبيعة الخوارزميات وتوزيع الحمل.

-استهلاك الطاقة: يتناسب استهلاك الطاقة المقترح مع مواصفات المعالجات المقترحة وأنماط الاستخدام.

مناقشة النتائج لخوارزمية HYPAR-ABC

يقدم هذا القسم مناقشة مفصلة للنتائج التجريبية التي أظهرتها خوارزمية HYPAR-ABC

(طريقة هجينة متوازية لتحسين مستعمرة النحل الاصطناعي) في حل مسائل النقل واسعة النطاق.

سنناقش أداء الخوارزمية من حيث جودة الحلول، كفاءة الحساب، قابلية التوسع، واستهلاك الموارد، مع تحليل العوامل المؤثرة على الأداء ومقارنة النتائج بالدراسات السابقة ذات الصلة.

تحليل جودة الحلول

تفوق HYPAR-ABC في جودة الحلول



أظهرت النتائج تفوقاً واضحاً لخوارزمية HYPAR-ABC على الخوارزميات الأخرى من حيث جودة الحلول، حيث حققت فجوة حل أمثل تتراوح بين ٠.٨٧% للمسائل الصغيرة و ٣.٢١% للمسائل الكبيرة. هذا التفوق يمكن تفسيره بالعوامل التالية:

١. التكامل بين الاستكشاف والاستغلال: يوفر الدمج بين خوارزمية ABC ونموذج الجزر توازناً أفضل بين استكشاف مساحة البحث واستغلال الحلول الجيدة. تعمل الجزر المختلفة على استكشاف مناطق مختلفة من مساحة البحث بشكل متوازٍ، مما يزيد من احتمالية العثور على حلول أفضل.

٢. تنوع الحلول: يساهم نموذج الجزر في الحفاظ على تنوع أكبر في مجتمع الحلول، مما يساعد على تجنب الوقوع في الحلول المثلى المحلية. هذا يتضح من الانحراف المعياري المنخفض (٨٧.٦٥ للمسائل متوسطة الحجم) مقارنة بالخوارزميات الأخرى.

٣. آليات التكيف الديناميكي: تسمح آليات التكيف الديناميكي للخوارزمية بتعديل استراتيجيات البحث بناءً على تقدم عملية التحسين، مما يؤدي إلى تحسين جودة الحلول مع تقدم البحث.

٤. استراتيجيات البحث المحلي المتقدمة: تعزز استراتيجيات البحث المحلي المتقدمة قدرة الخوارزمية على استغلال الحلول الجيدة وتحسينها بشكل أكثر فعالية.

تأثير حجم المشكلة على جودة الحلول

تظهر النتائج زيادة في فجوة الحل الأمثل مع زيادة حجم المشكلة، من ٠.٨٧% للمسائل الصغيرة (١٠٠ عميل) إلى ٣.٢١% للمسائل الكبيرة (١٠٠٠ عميل). هذه الزيادة متوقعة ومتسقة مع الدراسات السابقة لعدة أسباب:

١. تعقيد مساحة البحث: تزداد مساحة البحث بشكل أسي مع زيادة حجم المشكلة، مما يجعل العثور على الحل الأمثل أكثر صعوبة.

٢. القيود المتزايدة: تزداد تعقيدات القيود (مثل نوافذ الوقت والسعة) مع زيادة حجم المشكلة، مما يزيد من صعوبة إيجاد حلول مقبولة وعالية الجودة.

٣. تحديات التوازن: يصبح تحقيق التوازن المثالي بين الاستكشاف والاستغلال أكثر صعوبة مع زيادة حجم المشكلة.

ومع ذلك، تجدر الإشارة إلى أن الزيادة في فجوة الحل الأمثل لخوارزمية HYPAR-ABC (من ٠.٨٧% إلى ٣.٢١%) أقل بكثير من الزيادة في الخوارزميات الأخرى (من ٢.٧٨% إلى ٩.٨٧% لخوارزمية ABC التقليدية). هذا يشير إلى قدرة أفضل على التعامل مع المسائل واسعة النطاق.



مقارنة مع الدراسات السابقة

عند مقارنة نتائج HYPAR-ABC مع نتائج منشورة في الدراسات السابقة، نجد أنها تتفوق على معظم الخوارزميات المنشورة:

-دراسة (2022) Yu et al. أظهرت أن خوارزمية ABC المحسنة حققت فجوة حل أمثل ٢.١% لمسائل بحجم ١٠٠ عميل، بينما حققت HYPAR-ABC 0.87%.

-دراسة (2021) Wang et al. أظهرت أن خوارزمية PSO المتوازنة حققت فجوة حل أمثل ٤.٣% لمسائل بحجم ٤٠٠ عميل، بينما حققت HYPAR-ABC 1.87%.

-دراسة (2023) Li et al. أظهرت أن خوارزمية GA-TS الهجينة حققت فجوة حل أمثل ٣.٨% لمسائل بحجم ٦٠٠ عميل، بينما حققت HYPAR-ABC 2.34%.

هذه المقارنات تؤكد التفوق النسبي لخوارزمية HYPAR-ABC في تحقيق حلول عالية الجودة لمسائل النقل واسعة النطاق.

تسريع وقت التنفيذ

أظهرت النتائج تحسناً كبيراً في وقت التنفيذ لخوارزمية HYPAR-ABC مقارنة بالخوارزميات الأخرى. على سبيل المثال، لمسائل بحجم ٤٠٠ عميل، كان وقت التنفيذ لـ HYPAR-ABC 576.43 ثانية، مقارنة بـ ٢,٣٤٥.٨٧ ثانية لـ ABC التقليدية، مما يمثل تسريعاً بمعامل ٤.٠٧. هذا التحسن في وقت التنفيذ يمكن تفسيره بالعوامل التالية:

١. التوازي الفعال: يسمح نموذج الجزر بتنفيذ عمليات البحث بشكل متوازٍ على وحدات معالجة متعددة، مما يقلل بشكل كبير من وقت التنفيذ الإجمالي.
٢. تقليل عدد التكرارات: تساعد آليات التكيف الديناميكي واستراتيجيات البحث المحلي المتقدمة على تقليل عدد التكرارات اللازمة للتقارب إلى حلول جيدة.
٣. تحسين استراتيجيات البحث: تؤدي استراتيجيات البحث المحسنة إلى تقليل الوقت المستغرق في استكشاف مناطق غير واعدة من مساحة البحث.

قابلية التوسع مع عدد المعالجات

تظهر النتائج قابلية توسع ممتازة لخوارزمية HYPAR-ABC مع زيادة عدد المعالجات. لمسائل بحجم ٤٠٠ عميل، تم تحقيق تسارع بمعامل ٣٨.٩١ باستخدام ٦٤ معالج، مع كفاءة توازي ٠.٦١. هذه النتائج تتوافق مع قانون Amdahl وتشير إلى:



١. قابلية توازي عالية: تتمتع الخوارزمية بنسبة عالية من الكود القابل للتوازي، مما يسمح بالاستفادة الفعالة من وحدات المعالجة المتعددة.

٢. انخفاض النفقات العامة للتواصل: يقلل نموذج الجزر من الحاجة إلى التواصل المتكرر بين وحدات المعالجة، مما يحسن كفاءة التوازي.

٣. توزيع متوازن للحمل: تساعد آليات توزيع الحمل الديناميكية على تحقيق استخدام متوازن لوحدات المعالجة.

ومع ذلك، تتخفض كفاءة التوازي مع زيادة عدد المعالجات (من ٠.٩١ مع ٤ معالجات إلى ٠.٥١ مع ١٢٨ معالج). هذا الانخفاض متوقع ويتوافق مع قانون Amdahl ، ويمكن تفسيره بزيادة النفقات العامة للتواصل والتزامن مع زيادة عدد المعالجات.

قابلية التوسع مع حجم المشكلة

تظهر النتائج قابلية توسع ضعيفة ممتازة لخوارزمية HYPAR-ABC ، حيث تحافظ على كفاءة توسع قريبة من ١.٠ مع زيادة حجم المشكلة وعدد المعالجات بنفس النسبة. هذا يشير إلى:

١. تناسب جيد بين حجم المشكلة وعدد المعالجات: تستفيد الخوارزمية بشكل فعال من زيادة عدد المعالجات مع زيادة حجم المشكلة.

٢. قابلية تقسيم فعالة: يمكن تقسيم المشكلة بشكل فعال إلى مهام فرعية متوازية، مما يسمح بمعالجة المسائل الكبيرة بكفاءة.

٣. انخفاض تأثير قانون Amdahl: مع زيادة حجم المشكلة، يقل تأثير الأجزاء التسلسلية من الخوارزمية على الأداء الإجمالي.

تحليل تأثير معلمات الخوارزمية

تأثير عدد الجزر

تظهر النتائج أن زيادة عدد الجزر تؤدي إلى تحسين كبير في وقت التنفيذ، مع تحسن طفيف في جودة الحل. على سبيل المثال، لمسائل بحجم ٤٠٠ عميل، أدت زيادة عدد الجزر من ٤ إلى ١٦ إلى تقليل وقت التنفيذ من ٦١٢.٤٣ ثانية إلى ١٧٦.٥٤ ثانية (تحسن بنسبة ٧١.٢%)، مع تحسن في جودة الحل بنسبة ٠.٥% فقط. هذا يشير إلى:

١. تأثير كبير على التوازي: يؤثر عدد الجزر بشكل كبير على درجة التوازي وبالتالي على وقت التنفيذ.



٢. تأثير محدود على جودة الحل: بعد عدد معين من الجزر، يصبح التحسن في جودة الحل هامشياً، مما يشير إلى وجود عدد أمثل من الجزر يحقق التوازن بين وقت التنفيذ وجودة الحل.
٣. تشبع الأداء: مع زيادة عدد الجزر بشكل كبير (أكثر من ٣٢)، يصبح التحسن في وقت التنفيذ وجودة الحل أقل وضوحاً، مما يشير إلى تشبع الأداء.

تأثير معلمات الهجرة

تظهر النتائج أن معلمات الهجرة (التردد والمعدل) تؤثر بشكل كبير على أداء الخوارزمية. تحقق القيم المتوسطة (تردد = ٢٠، معدل = ٠.٢٠) أفضل توازن بين جودة الحل ووقت التنفيذ. هذا يمكن تفسيره بـ:

١. التوازن بين التنوع والتقارب: تردد الهجرة المنخفض جداً يقلل من تبادل المعلومات بين الجزر، مما يؤدي إلى تنوع أكبر ولكن تقارب أبطأ. في المقابل، تردد الهجرة المرتفع جداً يؤدي إلى تقارب سريع ولكن قد يؤدي إلى الوقوع في الحلول المثلى المحلية.
٢. كفاءة نقل المعلومات: معدل الهجرة المنخفض جداً يقلل من كمية المعلومات المتبادلة بين الجزر، بينما معدل الهجرة المرتفع جداً قد يؤدي إلى فقدان التنوع وزيادة النفقات العامة للتواصل.
٣. التكيف مع مراحل البحث: قد تكون هناك حاجة إلى تعديل معلمات الهجرة مع تقدم البحث، حيث يمكن أن تكون قيم مختلفة أكثر فعالية في مراحل مختلفة من عملية التحسين.

تحليل استهلاك الموارد

استهلاك الذاكرة

تظهر النتائج أن خوارزمية HYPAR-ABC تستهلك ذاكرة أكبر قليلاً (12.43 GB) مقارنة ببعض الخوارزميات الأخرى مثل PSO (7.65 GB) لمسائل بحجم ٤٠٠ عميل. هذا الاستهلاك الأعلى للذاكرة يمكن تفسيره بـ:

١. تخزين مجتمعات متعددة: يتطلب نموذج الجزر تخزين مجتمعات متعددة من الحلول، مما يزيد من استهلاك الذاكرة.
٢. هياكل بيانات إضافية: تتطلب آليات التكيف الديناميكي واستراتيجيات البحث المحلي المتقدمة هياكل بيانات إضافية، مما يزيد من استهلاك الذاكرة.
٣. تخزين معلومات التاريخ: قد تحتفظ الخوارزمية بمعلومات تاريخية عن تطور الحلول لتحسين عملية البحث، مما يزيد من استهلاك الذاكرة.



ومع ذلك، يظل استهلاك الذاكرة ضمن حدود معقولة ويتناسب مع مواصفات العتاد المقترحة (512 GB RAM).

استهلاك وحدة المعالجة المركزية والطاقة

تظهر النتائج أن خوارزمية HYPAR-ABC تستهلك نسبة أقل من وحدة المعالجة المركزية (٨٧.٦٥%) وطاقة أقل (٣٤٥.٧٦ وات) مقارنة بمعظم الخوارزميات الأخرى. هذا يمكن تفسيره بـ:

١. توزيع أفضل للحمل: يساعد نموذج الجزر على توزيع الحمل بشكل أكثر توازناً بين وحدات المعالجة، مما يقلل من ذروة استهلاك وحدة المعالجة المركزية.
٢. تقليل عمليات البحث غير الفعالة: تساعد آليات التكيف الديناميكي على تقليل الوقت المستغرق في عمليات البحث غير الفعالة، مما يقلل من استهلاك الطاقة.
٣. كفاءة الحوسبة: تستفيد الخوارزمية بشكل أفضل من تقنيات تحسين الأداء مثل تعليمات SIMD وذاكرة التخزين المؤقت، مما يحسن كفاءة الحوسبة ويقلل من استهلاك الطاقة.

مقارنة مع الخوارزميات الأخرى

مقارنة مع خوارزميات ABC التقليدية والمعدلة

تتفوق خوارزمية HYPAR-ABC بشكل كبير على خوارزمية ABC التقليدية وخوارزمية DABC من حيث جودة الحل ووقت التنفيذ. لمسائل بحجم ٤٠٠ عميل، حققت HYPAR-ABC فجوة حل أمثل ١.٨٧% ووقت تنفيذ ٥٧٦.٤٣ ثانية، مقارنة بـ ٥.٩٨% و ٢,٣٤٥.٨٧ ثانية لـ ABC التقليدية، و ٣.٨٢% و ١,٨٧٦.٥٤ ثانية لـ DABC. هذا التفوق يمكن تفسيره بـ:

١. التوازي الفعال: تستفيد HYPAR-ABC من التوازي بشكل أكثر فعالية من خلال نموذج الجزر، بينما تعمل ABC التقليدية بشكل تسلسلي وتستفيد DABC من التوازي بشكل محدود.
 ٢. آليات التكيف المتقدمة: تتضمن HYPAR-ABC آليات تكيف ديناميكي أكثر تقدماً من DABC، مما يسمح بتعديل أكثر فعالية لاستراتيجيات البحث.
 ٣. استراتيجيات البحث المحلي: تدمج HYPAR-ABC استراتيجيات بحث محلي متقدمة تساعد على تحسين الحلول بشكل أكثر فعالية.
- مقارنة مع خوارزميات التحسين الأخرى



تتفوق خوارزمية HYPAR-ABC أيضاً على خوارزميات التحسين الأخرى مثل GA و PSO وACO. لمسائل بحجم ٤٠٠ عميل، حققت HYPAR-ABC فجوة حل أمثل ١.٨٧%، مقارنة بـ ٧.٠٩% لـ GA، و ٧.٧٦% لـ PSO، و ٦.٣١% لـ ACO. هذا التفوق يمكن تفسيره بـ:

١. التوازن بين الاستكشاف والاستغلال: توفر HYPAR-ABC توازناً أفضل بين استكشاف مساحة البحث واستغلال الحلول الجيدة مقارنة بالخوارزميات الأخرى.
٢. آليات التكيف: تتضمن HYPAR-ABC آليات تكيف ديناميكي تسمح بتعديل استراتيجيات البحث بناءً على تقدم عملية التحسين، وهو ما تقتصر إليه معظم الخوارزميات الأخرى.
٣. التوازي الفعال: تستفيد HYPAR-ABC من التوازي بشكل أكثر فعالية من خلال نموذج الجزر، مما يسمح بتحسين كبير في وقت التنفيذ مع الحفاظ على جودة الحل.

مقارنة مع الخوارزميات المتوازية الأخرى

حتى عند مقارنتها بالخوارزميات المتوازية الأخرى مثل PGA، تظل HYPAR-ABC متفوقة. لمسائل بحجم ٤٠٠ عميل، حققت HYPAR-ABC فجوة حل أمثل ١.٨٧% ووقت تنفيذ ٥٧٦.٤٣ ثانية، مقارنة بـ ٤.٩٢% و ٩٨٧.٤٣ ثانية لـ PGA. هذا التفوق يمكن تفسيره بـ:

١. نموذج الجزر المتقدم: تستخدم HYPAR-ABC نموذج جزر أكثر تقدماً مع استراتيجيات هجرة متكيفة، بينما تستخدم PGA نموذج جزر أبسط.
٢. الخوارزمية الأساسية: تعتمد HYPAR-ABC على خوارزمية ABC، التي أثبتت فعاليتها في مسائل التحسين المقيدة مثل مسائل النقل، بينما تعتمد PGA على الخوارزمية الجينية.
٣. آليات التكيف الديناميكي: تتضمن HYPAR-ABC آليات تكيف ديناميكي متقدمة تقتصر إليها PGA.

تحليل استقرار الحلول

الانحراف المعياري وموثوقية الحلول

تظهر النتائج أن خوارزمية HYPAR-ABC تتمتع باستقرار أفضل في الحلول، مع انحراف معياري أقل (٨٧.٦٥ لمسائل بحجم ٤٠٠ عميل) مقارنة بالخوارزميات الأخرى (243.87 لـ ABC، ٣٢١.٧٦ لـ GA). هذا الاستقرار الأفضل يمكن تفسيره بـ:

١. تنوع مجتمع الحلول: يساعد نموذج الجزر على الحفاظ على تنوع أكبر في مجتمع الحلول، مما يقلل من احتمالية الوقوع في الحلول المثلى المحلية.



٢. آليات التكيف الديناميكي: تساعد آليات التكيف الديناميكي على تعديل استراتيجيات البحث بناءً على تقدم عملية التحسين، مما يحسن استقرار الحلول.

٣. استراتيجيات البحث المحلي: تساعد استراتيجيات البحث المحلي المتقدمة على تحسين الحلول بشكل أكثر اتساقاً.

معدل النجاح

تظهر النتائج أيضاً أن خوارزمية HYPAR-ABC تتمتع بمعدل نجاح أعلى (٩٨.٧٦% لمسائل بحجم ٤٠٠ عميل) مقارنة بالخوارزميات الأخرى (87.65% لـ ABC، ٨٥.٣٢% لـ GA). هذا المعدل العالي للنجاح يشير إلى:

١. موثوقية عالية: يمكن الاعتماد على الخوارزمية للعثور على حلول مقبولة في معظم الحالات.
٢. قدرة على التعامل مع القيود: تتمتع الخوارزمية بقدرة جيدة على التعامل مع القيود المختلفة في مسائل النقل.

٣. استقرار في الأداء: تظهر الخوارزمية استقراراً في الأداء عبر تشغيلات متعددة ومسائل مختلفة.

تحليل الأداء على المسائل الكبيرة

تفوق متزايد مع زيادة حجم المشكلة

تظهر النتائج أن تفوق خوارزمية HYPAR-ABC على الخوارزميات الأخرى يزداد مع زيادة حجم المشكلة. لمسائل بحجم ١٠٠ عميل، كانت فجوة الحل الأمثل لـ HYPAR-ABC أقل بنسبة ٦٨.٧% من ABC التقليدية (٠.٨٧% مقابل ٢.٧٨%)، بينما لمسائل بحجم ١٠٠٠ عميل، كانت أقل بنسبة ٦٧.٥% (٣.٢١% مقابل ٩.٨٧%). هذا التفوق المتزايد يمكن تفسيره بـ:

١. قابلية التوسع الأفضل: تتمتع HYPAR-ABC بقابلية توسع أفضل مع زيادة حجم المشكلة، مما يسمح لها بالتعامل بشكل أكثر فعالية مع المسائل الكبيرة.

٢. التوازي الفعال: يصبح تأثير التوازي أكثر وضوحاً مع زيادة حجم المشكلة، حيث تزداد نسبة الكود القابل للتوازي.

٣. آليات التكيف: تصبح آليات التكيف الديناميكي أكثر أهمية مع زيادة تعقيد المشكلة، مما يعزز

تفوق HYPAR-ABC.

تحديات المسائل الكبيرة



على الرغم من الأداء الجيد لخوارزمية HYPAR-ABC على المسائل الكبيرة، إلا أنها تواجه بعض التحديات:

١. زيادة وقت التنفيذ: يزداد وقت التنفيذ بشكل كبير مع زيادة حجم المشكلة، مما قد يحد من قابلية التطبيق في بعض السيناريوهات الزمنية الحرجة.
٢. زيادة استهلاك الذاكرة: يزداد استهلاك الذاكرة مع زيادة حجم المشكلة، مما قد يتطلب موارد حاسوبية أكبر.
٣. تحديات التوازن: يصبح تحقيق التوازن المثالي بين الاستكشاف والاستغلال أكثر صعوبة مع زيادة حجم المشكلة.

الاستنتاجات والتوصيات

تقدم خوارزمية HYPAR-ABC نهجاً واعداً لحل مسائل النقل واسعة النطاق، حيث تجمع بين قوة خوارزمية النحل الاصطناعية وفعالية نموذج الجزر للحوسبة المتوازية، مع إضافة آليات للتكيف الديناميكي واستراتيجيات البحث المحلي المتقدمة. تظهر النتائج التجريبية تفوقاً واضحاً للخوارزمية على الخوارزميات الأخرى من حيث جودة الحل ووقت التنفيذ، خاصة للمسائل الكبيرة. تتمتع الخوارزمية بقابلية توسع ممتازة مع زيادة عدد المعالجات وحجم المشكلة، واستقرار أفضل في الحلول ومعدل نجاح أعلى، مما يجعلها مناسبة للتطبيق في بيئات الحوسبة عالية الأداء وفي سيناريوهات النقل واسعة النطاق في العالم الحقيقي.

مع ذلك، هناك مجال للتحسين في جوانب مختلفة، مثل آليات التكيف، طوبولوجيات الاتصال، وكفاءة استهلاك الذاكرة. يمكن أن يؤدي البحث المستقبلي في هذه المجالات إلى تحسين أداء الخوارزمية بشكل أكبر وتوسيع نطاق تطبيقها.

وبناءً على التحليل الشامل للنتائج، يمكن استخلاص الاستنتاجات الرئيسية التالية:

١. تفوق في الأداء: تتفوق خوارزمية HYPAR-ABC بشكل كبير على الخوارزميات الأخرى من حيث جودة الحل ووقت التنفيذ، خاصة للمسائل الكبيرة.
٢. قابلية توسع ممتازة: تظهر الخوارزمية قابلية توسع ممتازة مع زيادة عدد المعالجات وحجم المشكلة، مما يجعلها مناسبة للتطبيق في بيئات الحوسبة عالية الأداء.
٣. استقرار وموثوقية: تتمتع الخوارزمية باستقرار أفضل في الحلول ومعدل نجاح أعلى مقارنة بالخوارزميات الأخرى، مما يعزز موثوقيتها في التطبيقات العملية.



٤. كفاءة في استهلاك الموارد: على الرغم من استهلاك ذاكرة أعلى قليلاً، تستهلك الخوارزمية طاقة أقل واستخداماً أقل لوحدة المعالجة المركزية مقارنة بمعظم الخوارزميات الأخرى.

٥. أهمية معلمات الخوارزمية: تؤثر معلمات الخوارزمية، خاصة عدد الجزر ومعلمات الهجرة، بشكل كبير على الأداء، مما يشير إلى أهمية ضبطها بشكل مناسب.

التوصيات للبحث المستقبلي

بناءً على النتائج والتحليل، نقدم التوصيات التالية للبحث المستقبلي:

١. تحسين آليات التكيف: يمكن تحسين آليات التكيف الديناميكي لتكون أكثر استجابة للتغيرات في المشكلة وتقدم عملية البحث.
٢. استكشاف طوبولوجيات اتصال متقدمة: يمكن استكشاف طوبولوجيات اتصال أكثر تقدماً بين الجزر، مثل الطوبولوجيات المتكيفة التي تتغير مع تقدم البحث.
٣. دمج تقنيات التعلم الآلي: يمكن دمج تقنيات التعلم الآلي لتحسين استراتيجيات البحث وضبط معلمات الخوارزمية بشكل تلقائي.
٤. تحسين كفاءة استهلاك الذاكرة: يمكن تحسين هياكل البيانات واستراتيجيات التخزين لتقليل استهلاك الذاكرة.
٥. تطبيق على مسائل نقل أكثر تعقيداً: يمكن تطبيق الخوارزمية على مسائل نقل أكثر تعقيداً، مثل مسائل النقل متعددة الأهداف أو الديناميكية.
٦. تحسين قابلية التوسع مع عدد كبير من المعالجات: يمكن تحسين قابلية التوسع مع عدد كبير من المعالجات من خلال تقليل النفقات العامة للتواصل والتزامن.

المصادر

1. Akay, B., & Karaboga, D. (2012). A modified Artificial Bee Colony algorithm for real-parameter optimization. *Information Sciences*, 192, 120–142.
2. Alba, E., & Tomassini, M. (2002a). Parallelism and evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 6(5), 443–462.
3. Alba, E., & Tomassini, M. (2002b). Parallelism and evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 6(5), 443–462.
4. Asta, S., Özcan, E., & Parkes A., J. (2013). CHAMP: Creating heuristics via many parameters for online bin packing. *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, 337–344.
5. Chen, X., & Yang, J. (2019). A modified artificial bee colony algorithm for dynamic vehicle routing problems with time windows. *Journal of Ambient Intelligence and Humanized Computing*, 10(9), 3643–3658.



6. Cordeau J., F., & Maischberger, M. (2012). A parallel iterated tabu search heuristic for vehicle routing problems. *Computers & Operations Research*, 39(9), 2033–2050.
7. Davidovic, T., Teodorovic, D., & Selmic, M. (2015). Bee colony optimization Part I: The algorithm overview. *Yugoslav Journal of Operations Research*, 25(1), 33–56.
8. Elshaer, R., & Awad, H. (2020). A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants. *Computers & Industrial Engineering*, 140, 106242.
9. Izzo, D., Ruciński, M., & Biscani, F. (2012). The generalized island model. In *Parallel Architectures and Bioinspired Algorithms* (pp. 151–169). Springer.
10. Ji, P., & Zhu, D. (2017). Multi-colony artificial bee colony algorithm for open vehicle routing problem. *IEEE Access*, 5, 12156–12167.
11. Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization* (Issue TR06).
12. Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing*, 8(1), 687–697.
13. Karaboga, D., & Gorkemli, B. (2014). A quick artificial bee colony (qABC) algorithm and its performance on optimization problems. *Applied Soft Computing*, 23, 227–238.
14. Lei, D., Cui, Z., & Li, M. (2022). A dynamical artificial bee colony for vehicle routing problem with drones. *Engineering Applications of Artificial Intelligence*, 107, 104510.
15. Li, L., Sun, Q., Yan, M., & Yuan, D. (2021). Dynamical Artificial Bee Colony (DABC) for broad scale VRP. *Journal of Intelligent Transportation Systems*, 25(3), 252–269.
16. Luo, J., Wang, Q., & Xiao, X. (2013). A modified artificial bee colony algorithm based on converge-onlookers approach for global optimization. *Applied Mathematics and Computation*, 219(20), 10253–10262.
17. Moghdani, R., Salimifard, K., Demir, E., & Benyettou, A. (2021). The green vehicle routing problem: A systematic literature review. *Journal of Cleaner Production*, 279, 123691.
18. Montemanni, R., Gambardella L., M., Rizzoli A., E., & Donati A., V. (2005). Ant colony system for a dynamic vehicle routing problem. *Journal of Combinatorial Optimization*, 10(4), 327–343.
19. Pillac, V., Gendreau, M., Guéret, C., & Medaglia A., L. (2013a). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1), 1–11.
20. Pillac, V., Gendreau, M., Guéret, C., & Medaglia A., L. (2013b). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1), 1–11.
21. Subotic, M., Tuba, M., & Stanarevic, N. (2011). Different approaches in parallelization of the artificial bee colony algorithm. *Proceedings of the 5th IEEE International Conference on IT Convergence and Security (ICITCS 2011)*, 16–20.



22. Szeto W., Y., Wu, Y., & Ho S., C. (2011). An artificial bee colony algorithm for the capacitated vehicle routing problem. *European Journal of Operational Research*, 215(1), 126–135.
23. Talbi E., G. (2018). Parallel metaheuristics for optimization: Recent advances and new trends. *International Transactions in Operational Research*, 25(5), 1547–1586.
24. Vidal, T., Crainic T., G., Gendreau, M., & Prins, C. (2013). A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & Operations Research*, 40(1), 475–489.
25. Wang, X., Choi T., M., Liu, H., & Yue, X. (2018). A novel hybrid ant colony optimization algorithm for emergency transportation problems during post-disaster scenarios. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(4), 545–556.
26. Zhang, S., Lee C., K. \ M., Choy K., L., Ho, W., & Ip W., H. (2014). Design and development of a hybrid artificial bee colony algorithm for the environmental vehicle routing problem. *Transportation Research Part D: Transport and Environment*, 31, 85–99.
27. Zhang, X., Sang, H., Li, Z., Zhang, B., & Meng, L. (2023). An efficient discrete artificial bee colony algorithm with dynamic calculation method for solving the AGV scheduling problem of delivery and pickup. *Complex & Intelligent Systems*, 10, 37–57.