ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO MÔN HỌC MÁY HỌC

**Đề tài :**

**NGHIÊN CỨU VÀ SO SÁNH CÁC THUẬT TOÁN ĐOM ĐÓM LAI TRONG BÀI TOÁN PHÂN CỤM DỮ LIỆU TỰ ĐỘNG**

**Dựa trên bài báo :**

A Comparative Performance Study of Hybrid Firefly  Algoritms for Automatic Data Clustering

(Absalom el-shamir Ezugwu, Moyinoluwa b. Agbaje1, Nahla Aljojo,Rosanne Els,Haruna Chiroma, and Mohamed Abd Elaziz)

**GIẢNG VIÊN HƯỚNG DẪN :** TS. Trần Thái Sơn

**HỌC VIÊN THỰC HIỆN :**

20C12007 - Trần Đình Lâm

20C11040 - Đặng Nhật Minh

20C11035 - Trương Thế Kiệt

**KHÓA :** 30

**NGÀNH : Khoa học máy tính**

TP.HCM, 04-2021

**Mục Lục**

[**Mục Lục** 1](#_Toc70935336)

[**Bảng danh sách các hình** 2](#_Toc70935337)

[**Bảng thông tin chi tiết nhóm** 3](#_Toc70935338)

[**1.** **Giới thiệu tổng quan** 4](#_Toc70935339)

[**1.1.** **Bài toán Clustering** 4](#_Toc70935340)

[**1.2.** **Bài toán Optimization** 6](#_Toc70935341)

[**2.** **Cơ sở lý thuyết** 7](#_Toc70935342)

[**2.1.** **Thuật toán đom đóm FA** 7](#_Toc70935343)

[**2.2.** **Thuật toán PSO** 10](#_Toc70935344)

[**2.3.** **Thuật toán ABC** 13](#_Toc70935345)

[**2.4.** **Thuật toán IWO** 15](#_Toc70935346)

[**2.5.** **Thuật toán TLBO** 17](#_Toc70935347)

[**2.6.** **Lịch sử cải tiến, lai tạo giữa các thuật toán** 18](#_Toc70935348)

[**3.** **So sánh các phương pháp lai được bài báo đề xuất** 18](#_Toc70935349)

[**3.1.** **Các hàm mục tiêu được sử dụng** 18](#_Toc70935350)

[**a.** **Compact-Separated** [2] 18](#_Toc70935351)

[**b.** **Davis-Bouldin** [2] 19](#_Toc70935352)

[**3.2.** **4 Thuật toán lai được đề xuất** 19](#_Toc70935353)

[**4.** **Thực nghiệm và so sánh** 22](#_Toc70935354)

[**4.1.** **Dataset** 22](#_Toc70935355)

[**4.2.** **Kết quả thực nghiệm và phân tích** 22](#_Toc70935356)

[**4.3.1.** **Kết quả so sánh tổng quan** 22](#_Toc70935357)

[**4.3.2.** **So sánh thời gian chạy thuật toán** 23](#_Toc70935358)

[**5.** **Kết luận** 23](#_Toc70935359)

[References 25](#_Toc70935360)

# **Bảng danh sách các hình**

[Hình A Phân loại các thuật toán clustering [3] 5](file:///C:\Users\LAP11777-local\Documents\GitHub\document\MayHoc\ReportFinal.docx#_Toc70935052)

[Hình B Mã giả của thuật toán đom đóm [1] 9](#_Toc70935053)

[Hình C Thuật toán đom đóm cải tiến [1] 10](#_Toc70935054)

[Hình D Công thức cập nhật vị trí, vận tốc của cá thể 11](#_Toc70935055)

[Hình E Mô phỏng vector vận tốc 12](file:///C:\Users\LAP11777-local\Documents\GitHub\document\MayHoc\ReportFinal.docx#_Toc70935056)

[Hình F Mã giả thuật toán PSO 13](file:///C:\Users\LAP11777-local\Documents\GitHub\document\MayHoc\ReportFinal.docx#_Toc70935057)

[Hình G Mã giả thuật toán ABC 15](#_Toc70935058)

[Hình H Mã giả thuật toán IWO 17](#_Toc70935059)

[Hình I Lưu đồ tổng quát thuật toán lai [1] 20](file:///C:\Users\LAP11777-local\Documents\GitHub\document\MayHoc\ReportFinal.docx#_Toc70935060)

[Hình J Lưu đồ chi tiết thuật toán FAPSO [4] 21](file:///C:\Users\LAP11777-local\Documents\GitHub\document\MayHoc\ReportFinal.docx#_Toc70935061)

[Hình K Thông số các bộ dữ liệu được sử dụng để chạy thuật toán lai 22](#_Toc70935062)

[Hình L Kết quả chạy thực nghiệm dựa vào 2 chỉ số đánh giá [1] 23](file:///C:\Users\LAP11777-local\Documents\GitHub\document\MayHoc\ReportFinal.docx#_Toc70935063)

[Hình M So sánh thời gian chạy thuật toán trên các tập dữ liệu [1] 23](file:///C:\Users\LAP11777-local\Documents\GitHub\document\MayHoc\ReportFinal.docx#_Toc70935064)

# **Bảng thông tin chi tiết nhóm**

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã nhóm:** | **CLA-05** | | |
| **Tên nhóm:** | **K2014** | | |
| **Số lượng:** | **3** | | |
| **MSSV** | **Họ tên** | **Email** | **Điện thoại** |
| 20C12007 | Trần Đình Lâm | tdlam123@gmail.com | 0383522356 |
| 20C11035 | Trương Thế Kiệt | truongthekiet709@gmail.com |  |
| 20C11040 | Đặng Nhật Minh | minhdangnhat685@gmail.com |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **BẢNG PHÂN CÔNG & ĐÁNH GIÁ HOÀN THÀNH CÔNG VIỆC** | | | |
| **Người thực hiện** | **Công việc thực hiện** | **Mức độ hoàn thành** | **Đánh giá của nhóm** |
| 20C12007  Trần Đình Lâm | Dịch bài báo sang tiếng Việt và giải thích các thuật ngữ | 90% | 9/10 |
| Chỉnh sửa slide |
| Thuật toán lai FAPSO và thực nghiệm |
| 20C11035  Trương Thế Kiệt | Thuật toán ABC | 80% | 8/10 |
| Bài toàn Optimization |
| Soạn slide cho buổi serminar |
| 20C11040  Đặng Nhật Minh | Thuật toán đom đóm, PSO | 80% | 8/10 |
| Chuẩn bị form báo cáo |
| Soạn slide cho buổi serminar |

1. **Giới thiệu tổng quan**
2. **Bài toán Clustering**
   * 1. **Khái niệm:**

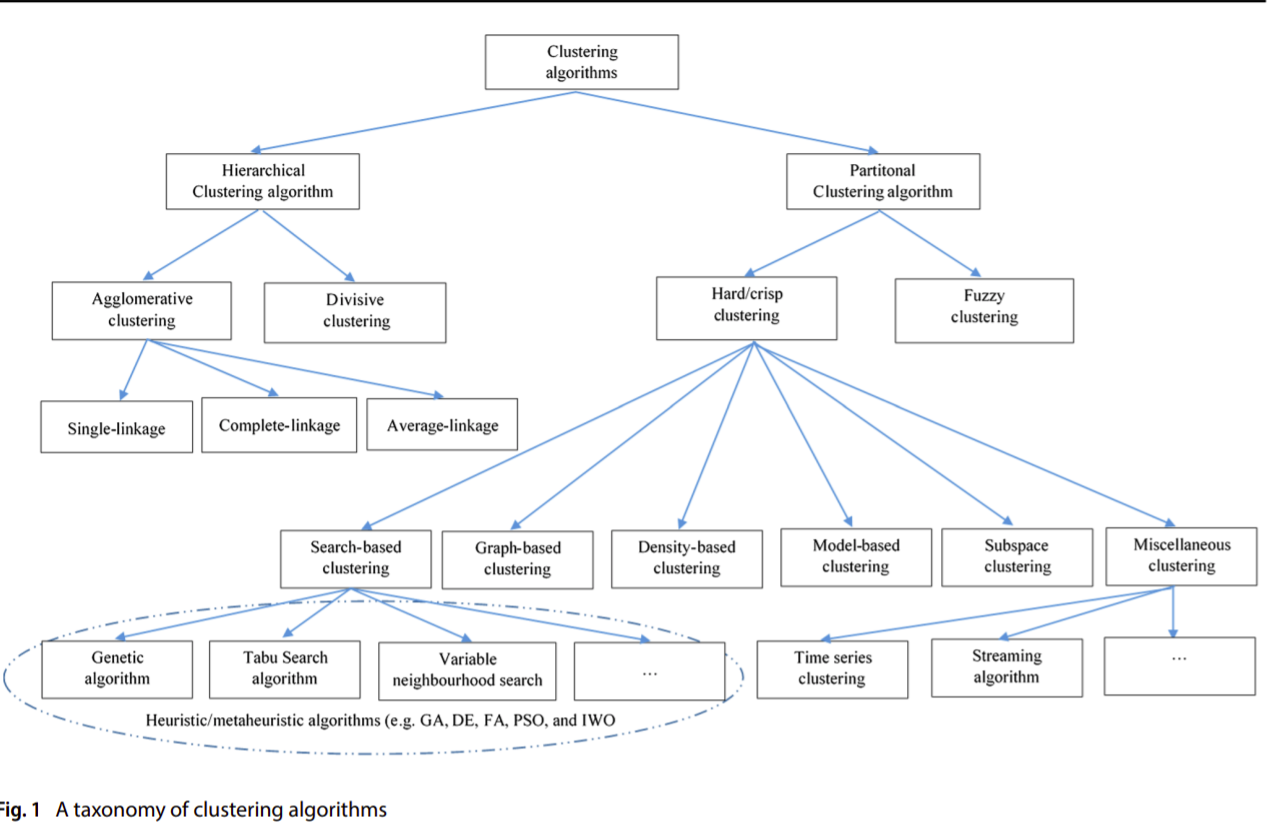
Bài toán clustering là một bài toán phân nhóm toàn bộ dữ liệu X thành các nhóm nhỏ dựa trên sự liên quan giữa các dữ liệu trong mỗi nhóm. Ví dụ: phân nhóm khách hàng dựa trên hành vi mua hàng. Điều này cũng giống như việc ta đưa cho một đứa trẻ rất nhiều mảnh ghép với các hình thù và màu sắc khác nhau, ví dụ tam giác, vuông, tròn với màu xanh và đỏ, sau đó yêu cầu trẻ phân chúng thành từng nhóm. Mặc dù không cho trẻ biết mảnh nào tương ứng với hình nào hoặc màu nào, nhiều khả năng chúng vẫn có thể phân loại các mảnh ghép theo màu hoặc hình dạng.

* + 1. **Đặc điểm của bài toán clustering:**
* Số cụm dữ liệu không được biết trước
* Có nhiều các tiếp cận, mối cách lại có vài kỹ thuật
* Các kỹ thuật khác nhau thường mang lại kết quả khác nhau.

Trong bài toán clustering, một số độ đo khoảng cách phổ biến thường được sử dụng để tính khoảng cách giữa hai điểm hoặc hai cụm, chẳng hạn như:

* Độ đo Euclid r1
* Độ đo Euclid chuẩn r2
* Độ đo Manhattan
* Độ đo Cosine
* Độ đo Jaccard
* Độ đo Hamming
* Độ đo Kullback-Leibler (KL)
  + 1. **Phân loại các thuật toán Clustering**

Hình 1-A dưới đây mô tả khái quát các loại thuật toán clustering hiện có:



Hình A Phân loại các thuật toán clustering [4]

**Hierarchical clustering:**

Agglomerative (đi từ dưới lên): Ban đầu chọn K là một số lớn gần bằng số điểm dữ liệu. Sau khi thực hiện phân cụm K-meas lần đầu, các cụm gần nhau được ghép lại thành một cụm. Lúc này khoảng cách giữa các cụm có thể được xác định bằng khoảng cách giữa các tâm cụm. Tiếp tục K-means ở các bước tiếp theo để giảm dần số lượng cụm. Các tiêu chí chọn hai cụm để sát nhập có thể bao gồm:

* Centroid-linkage: Sát nhập hai cụm có khoảng cách giữa hai tâm của hai cụm này là nhỏ nhất.
* Single-linkage: khoảng cách giữa hai điểm gần nhau nhất thuộc hai cụm. Sát nhập hai cụm có khoảng cách này nhỏ nhất.
* Average-linkage: trung bình các khoảng cách giữa hai cặp điểm bất kì thuộc hai cụm. Sát nhập hai cụm có khoảng cách này nhỏ nhất.
* Complete-linkage: khoảng cách giữa hai điểm xa nhau nhất của hai cụm, sát nhập hai cụm có khoảng cách này là nhỏ nhất.
* Radius: bán kính của một cụm là khoảng cách từ tâm tới điểm xa nhất của cụm, sát nhập hai cụm nếu hai cụm tạo ra một cụm có bán kính nhỏ nhất.
* Diameter: đường kính của một cụm là khoảng cách của hai điểm xa nhau nhất trong cụm, sát nhập hai cụm nếu chúng tạo nên một cụm có đường kính nhỏ nhất.

Divisive: "đi từ trên xuống": Ban đầu k-means với K nhỏ để thu được các cụm lớn. Sau đó tiếp tục K-means trên mỗi cụm đến khi kết quả chấp nhận được.

Ưu điểm của phân cụm phân lớp là không phải xác định trước số lượng cụm điều này khá vượt trội so với K-Means. Tuy nhiên, nó không hoạt động tốt với lượng dữ liệu khổng lồ.

Thuật toán phân cụm phân lớp có thể được sử dụng để xác định, dự đoán số cụm trước khi thực hiện thuật toán K-Means.

**Partitioning clustering:**

Là phương pháp non-hierarchical clustering phổ biến nhất, là phương pháp phân cụm one-level, các cluster không thể hiện dưới dạng cấp bậc. Hướng tiếp cận cơ bản nhất trong Partition clustering là tách cụm độc quyền, giống như Hard-clustering, mỗi object chỉ thuộc một cluster duy nhất.

Thuật toán phân cụm theo phương pháp này, phổ biến và đơn giản nhất đó chính là K-Means. Nổi trội hơn, nâng cao hơn trong cụm phương pháp clustering này là các thuật toán Search-base lấy cảm hứng từ tự nhiên như GA, PSO, FA,ABC,…

1. **Bài toán Optimization**

Optimization là quá trình tìm giá trị tốt nhất(maximum/minimum) trong không gian dữ liệu lớn của một hàm mục tiêu bằng cách sử dụng các thuật toán. Trong đó việc tìm kiếm được giá tối ưu hóa được xem là vấn đề NP-Hard.

Trong bài toán Optimization, hàm mục tiêu là hàm cần tối ưu hóa trong bài toán. Hàm này được chọn, hoặc suy ra từ yêu cầu của bài toán. ta giải bài toán bằng cách tìm ra cực tiểu toàn cục (global optima) của hàm này, bằng các kỹ thuật, thuật toán hay phương pháp tối ưu hóa.

Những kỹ thuật tối ưu hóa thường được dùng đề tìm kiếm các giải pháp cho bài toán tối ưu hóa toàn cục. Trong đó chia thành 2 loại kỹ thuật khác nhau: Thuật toán chính xác và thuật toán gần đúng. Thuật toán chính xác là các mô hình toán học đã được dùng để tìm kiếm giải pháp tối ưu. Thuật toán gần đúng (metaheuristic) là giải pháp gần tối ưu trong một thời gian giải thuật ngắn, phù hợp cho những công trình phức tạp.

Thuật toán metaheuristic thường có thể tìm được lời giải gần tối ưu trong một khoảng thời gian ít hơn hơn, tốn ít công sức hơn. Thành phần chính của những phương pháp metaheuristic là thăm dò(exploration)- quá trình khám phá không gian tìm kiếm và tránh bẫy trong các cực trị địa phương(local optimal), khai thắc(exploiation)-Quá trình cải thiện bất kỳ giải pháp hứa hẹn nào đã thu được cho đến thời điểm hiện tại.

1. **Cơ sở lý thuyết**
2. **Thuật toán đom đóm FA**

Thuật toán đom đóm (Firefly algorithm - FA) là một thuật toán tối ưu hóa lấy cảm hứng từ thiên nhiên, dựa trên trí thông minh bầy đàn được phát triển bởi Xin-She Yang (2009) [1] bắt chước cách bầy đom đóm giao tiếp với nhau.

Đom đóm là một loại côn trùng có cánh và có thể phát ra ánh sáng nhấp nháy về đêm. Ánh sáng này dùng để thu hút các con đom đóm khác di chuyển về phía nó, và cảnh báo về các mối nguy hiểm hoặc nguồn thức ăn

**Ý tưởng thuật toán**

Ở thời điểm ban đầu mỗi con đom đóm trong bầy sẽ ở một vị trí ngẫu nhiên trong không gian tìm kiếm và di chuyển theo một hướng ngẫu nhiên. Khi một con đom đóm tìm thấy thức ăn, nó sẽ phát ra ánh sáng với cường độ tỉ lệ với lượng thức ăn tìm được. Mỗi con đóm đóm sẽ dựa vào ánh sáng để di chuyển về nơi có nhiều thức ăn nhất

**Nội dung thuật toán**

Cường độ ánh sáng trên đường đi sẽ giảm dần do bị hấp thụ một phần bởi môi trường, cho cường độ ánh sáng I tại điểm cách O một khoảng r, I tỉ lệ nghịch với bình phương khoảng cách:

với r là khoảng cách, I là cường độ ánh sáng tại điểm cách một khoảng r

Cường độ ánh sáng của mỗi con đom đóm phát ra sẽ tỉ lệ với hàm cần tối ưu của bài toán.

Thuật toán đom đóm được phát triển dựa trên giả thuyết rằng:

* Khi một con đom đóm (A) bị thu hút bởi một con con đom đóm khác (B) thì nó chỉ dựa trên độ sáng (mà nó cảm nhận được) của con đom đóm B và không bị ảnh hưởng bởi các yếu tố khác (giới tính đom đóm, mùi hương, mùa, ...)
* Độ thu hút tỉ lệ thuận với cường độ ánh sáng phát ra, một con đom đóm ít phát sáng hơn sẽ bị thu hút bởi một con đom đóm phát sáng sáng hơn, độ thu hút sẽ giảm dần theo khoảng cách giữa hai con đom đóm
* Nếu hai con đom đóm có cùng độ sáng (và là độ sáng lớn nhất trong quần thể) thì chúng sẽ di chuyển ngẫu nhiên

Độ sáng cảm nhận được tại 1 điểm và độ thu hút là 2 yếu tố quan trọng trong thuật toán đom đóm. Cường độ ánh sáng do con đom đóm phát ra tỉ lệ nghịch với giá trị của hàm mục tiêu tại điểm mà con đom đóm đang đứng, và trong quá trình truyền đi, độ sáng sẽ thay đổi theo độ lớn khoảng cách và lượng ánh sáng bị hấp thụ bởi môi trường, kết hợp lại, ta có phương trình:

với I0 là cường độ ánh sáng tại điểm cớ khoảng cách r = 0, gamma là hệ số hấp thụ ánh sáng của môi trường, r là khoảng cách

Vì độ thu hút tỉ lệ với cường độ ánh sáng cảm nhận được, ta có phương trình:

với β0­ là độ thu hút tại điểm r = 0

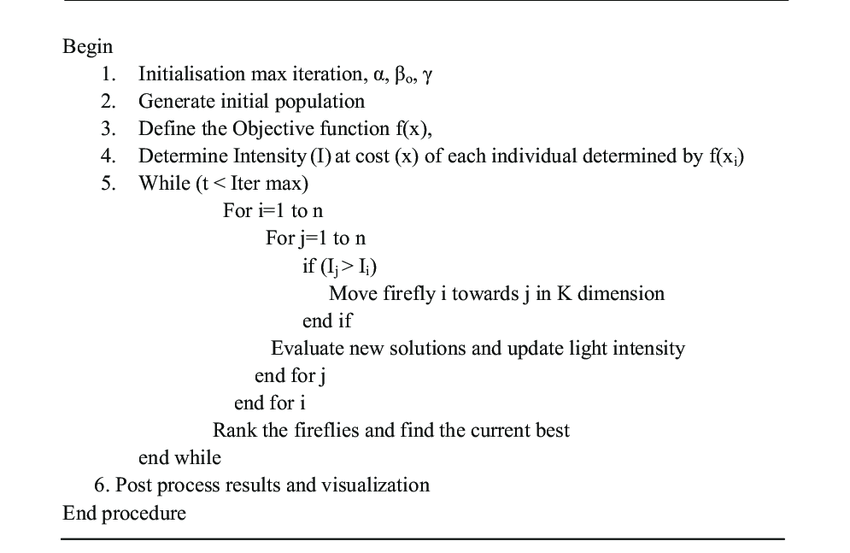
Khoảng cách Euclid giữa 2 con đom đóm xi và xj được tính bởi công thức:

với d là số chiều của dữ liệu

Sự chuyển động của 1 con đom đóm được xác định bởi công thức:

với , , là một số ngẫu nhiên từ phân phối Gauss

**Mã giả của thuật toán đom đóm**



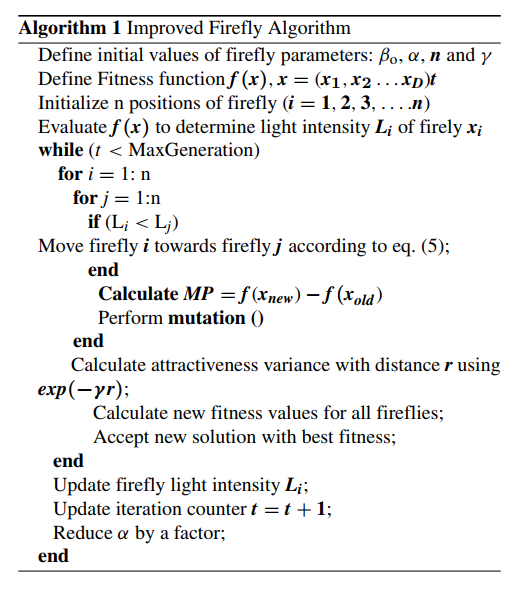
Hình B Mã giả của thuật toán đom đóm [2]

*Nhận xét: trong thuật toán ta thấy mỗi con đom đóm chỉ đi theo những con đom đóm sáng hơn nó mà không quan tâm tới personal best của mình, cách làm này sẽ giúp thuật toán hội tụ nhanh hơn, nhưng sẽ giảm khả năng tìm ra điểm cực trị tốt nhất*

Trong bài báo, để tăng khả năng khai thác và khám phá của thuật toán đom đóm, để có thể áp dụng vào những bài toán nhiều chiều một cách hiệu quả hơn, nhóm tác giả đã cải tiến thuật toán đom đóm bằng cách giới thiệu thuật ngữ mutation strategy (MP) để làm tăng độ thu hút của những con đom đóm có cường độ ánh sáng thấp hơn.

*Ý tưởng cơ bản là sau mỗi lần lặp, những con đom đóm bị đột biến sẽ di chuyển ngẫu nhiên thêm một khoảng nữa (để tăng khả năng tìm kiếm)*

Thuật toán đom đóm sau khi được cải tiến sẽ có dạng như sau:



Hình C Thuật toán đom đóm cải tiến [2]

1. **Thuật toán PSO**

Particle Swarm Optimazation (PSO) là một kĩ thuật tối ưu ngẫu nhiên dựa trên trí thông minh bầy đàn (population based stochastic optimization technique) được giới thiệu bởi Eberhart và Kennedy vào năm 1995 lấy cảm hứng từ cách di chuyển và trao đổi thông tin khi kiếm ăn của các thành viên trong quần thể bầy chim và đàn cá.

**Ý tưởng thuật toán**

Tại thời điểm ban đầu, mỗi thành viên trong quần thể ở một vị trí ngẫu nhiên và bay theo một hướng ngẫu nhiên. Sau một thời gian, một số thành viên tìm được thức ăn, và gửi tín hiệu tới các thành viên khác trong bầy tùy theo lượng thức ăn tìm thấy được. *Dựa vào thông tin đó, những thành viên khác sẽ di chuyển về nơi có nhiều thức ăn nhất (mà nó biết được).*

Ta có thể kết hợp PSO với một số thuật toán khác để giải quyết các bài toán về tối ưu.

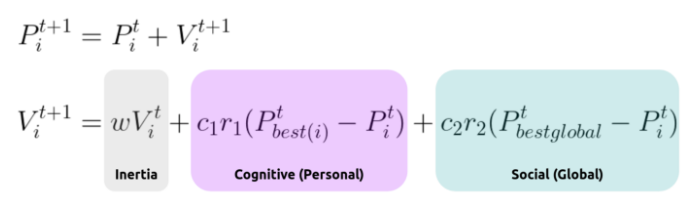
**Nội dung thuật toán**

**Cá thể**: mỗi cá thể được xem như một giải pháp tiềm năng để đạt được global minimum trong không gian tìm kiếm *(dựa trên vị trí của nó)*. *Không con nào biết được chính xác global minimum ở đâu, nhưng dựa vào vị trí của mỗi con ta có thể tính được giá trị của hàm mục tiêu cần tối ưu*

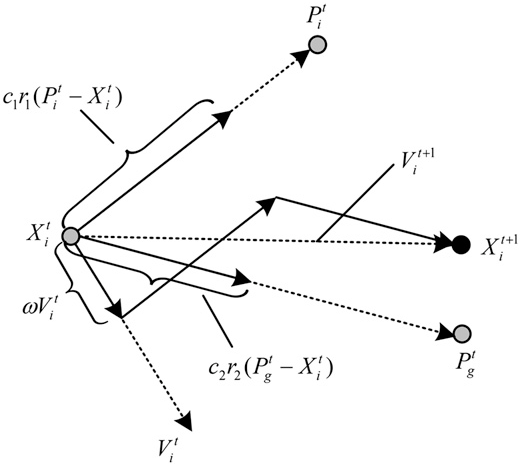
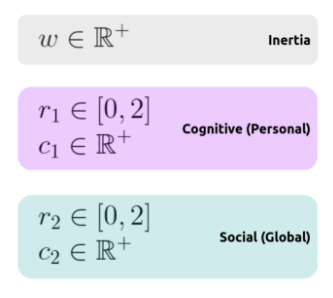
Ở lúc ban đầu, mỗi cá thể sẽ được khởi tạo vị trí và vận tốc ngẫu nhiên.

với t là số thứ tự của lần lặp, i là số thứ tự của cá thể trong quần thể

**Quần thể**: qua mỗi lần lặp, vector vận tốc của mỗi cá thể được thay đổi một cách ngẫu nhiên dựa theo hướng của vị trí nhiều thức ăn nhất của cá thể đó tìm được (personal best), và hướng của vị trí có nhiều thức ăn nhất của quần thể (global best)



Hình D Công thức cập nhật vị trí, vận tốc của cá thể



Hình E Mô phỏng vector vận tốc

trong đó, các trọng số có ý nghĩa

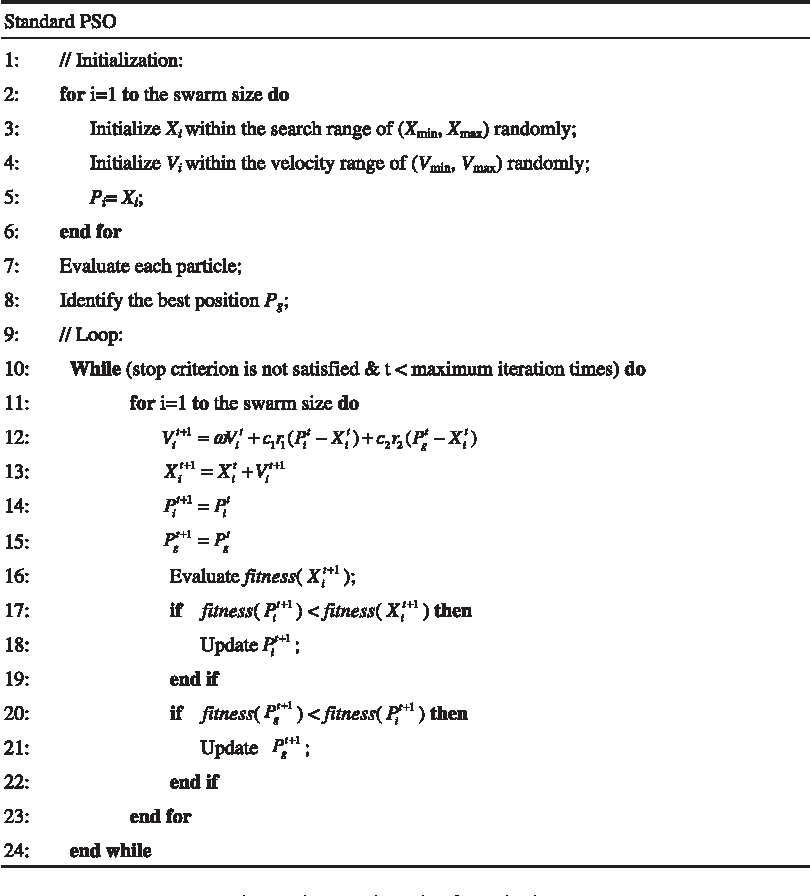
r1 và r2 là duy nhất cho mỗi cá thể và mỗi lần lặp  
c1 và c2 là hệ số cho biết độ phụ thuộc của vận tốc mới vào personal best và global best  
w là hệ số ứng với vận tốc hiện tại  
 là điểm personal best của cá thể thứ i tại lần lặp thứ t, là điểm global best tại lần lặp thứ t

*Bộ trọng số đó sẽ quyết định vận tốc (và vị trí) mới của cá thể, tạo ra sự cân bằng giữa khai phá và tìm kiếm (exploitaion and exploration) để tăng tốc hội tụ và tìm ra điểm cực trị tốt hơn*

*Nhận xét các trọng số:*

* *Với w lớn và c1, c2 nhỏ: quần thể khó hội tụ do mỗi cá thể chỉ đi theo hướng hiện tại của mình.*
* *Với c1 lớn và c2 nhỏ: quần thể hội tụ chậm do mỗi cá thể tập trung phần lớn vào hướng personal best của nó.*
* *Với c1 nhỏ và c2 lớn: quần thể hội tụ nhanh (do các cá thể đều hướng tới điểm global best của quần thể) nhưng điểm global minimum tìm được có thể không phải là điểm tốt nhất (do khả năng tìm kiếm bị hạn chế)*

**Mã giả thuật toán PSO**



Hình F Mã giả thuật toán PSO

1. **Thuật toán ABC**

Artificial Bee Colony (ABC) algorithm là thuật toán meta-heuristic dựa trên bầy đàn được giới thiệu năm 2005 bởi Karaboga để tối ưu các bài toán số.

**Ý tưởng thuật toán**

ACB lấy cảm hứng từ hành vi tìm kiếm thức ăn thông minh của ông mật.Trong đó bầy ông nhân tạo được chia thành ba nhóm: ong thợ (employed bees), ong giám sát(onlooker bees), ong trinh sát (scout bees).

* Ong thợ: đi đến nguồn thức ăn mà nó khám phá được trước đó.
* Ong quan sát: quan sát điệu nhảy của ong thợ để quyết định.
* Ong trinh sát: thực hiện tìm kiếm ngẫu nhiên để tìm kiếm nguồn thức ăn mới.

**Nội dung**

Thuật toán được chia làm bốn giai đoạn :

***Initialization Phase***

Khởi tạo ngẫu nhiên vị trí các food source, số lượng food source, số lượng ong thợ, ông giám sát, số lần chạy. Trong đó số lượng ong thợ, ông giám sát và food source là bằng nhau.

Tính toán giá trị của hàm mục tiêu, hàm fit(công thức tính giá trị hàm fit sẽ được giải thích ở phase Employed Bee).

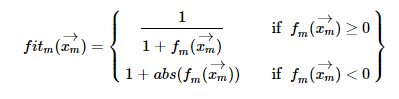
***Employed Bees Phase***

Tìm kiếm vị trị nguồn thức ăn lân cận dựa vào thông tin nguồn thức ăn trong kí ức trước đó. Vị trí nguồn thức ăn lân cận được tính theo công thức:

F:\SĐH\Kì_1\MAY_HOC\document\MayHoc\photos\abc_newfoodsourceposition.PNG

Sau khi tìm vị trí của nguồn thức ăn lân cần, các giá trị của hàm mục tiêu, hàm fit được tính toán lại. Nếu giá trị giá trị của hàm fit tại vị trí mới tốt hơn vị trí cũ trước đó, giá trị vị trí tốt nhất trong kí ức sẽ được cập nhật. Với mỗi lần mà vị trí nguồn thức ăn lân cận mới không tốt hơn vị trí cũ, số lần trial sẽ được tăng lên, và được reset khi giá trị tốt hơn.

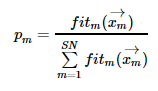
Giá trị của hàm fit sẽ được tính theo công thức sau:



Trong đó f(xm) là giá trị hàm mục tiêu tại vị trí xm

***Onlooker Bees Phase***

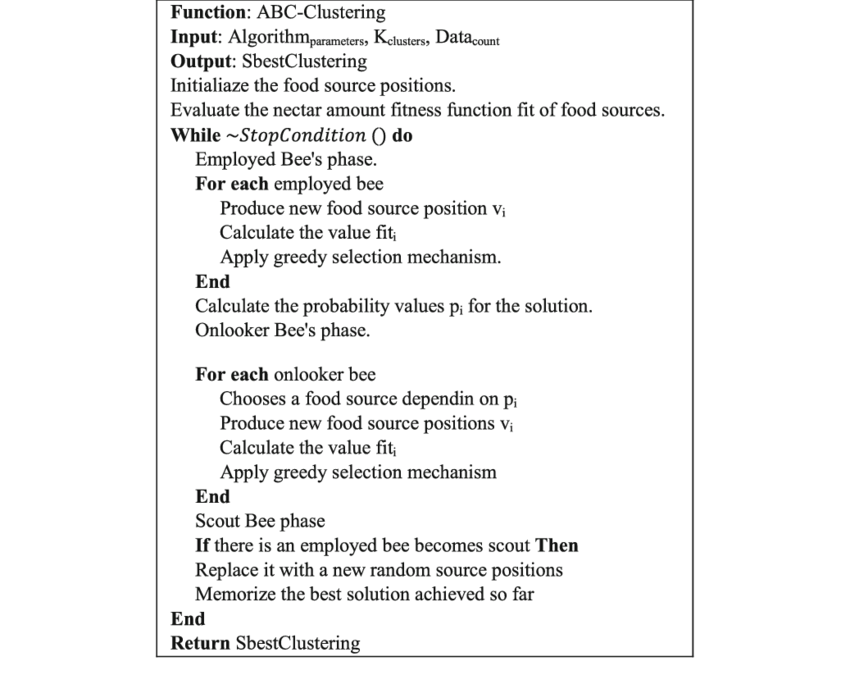
Quyết định lựa chọn nguồn thức ăn dựa vào giá trị xác suất được tính toán thông qua công thức:



***Scout Bees Phase***

Những con ông thợ khi giá trị trial đạt ngưỡng giới hạn (tức sau một số lần thức nghiệm tìm nguồn thức ăn lân cận nó không tìm được giải pháp tốt hơn) sẽ phải tìm lại vị trí nguồn thức ăn một cách ngẫu nhiên.

**Mã giã thuật toán ABC**



Hình G Mã giả thuật toán ABC

*Nhận xét:* *Nhìn chung thuật toán ABC có thể giúp hạn chế được các bẫy local optimal do sau một số lần giới hạn nếu không có giá trị khả thi hơn thì vị trí thức ăn tức giá trị của hàm mục tiêu sẽ được random lại và tiếp tục tìm kiếm.*

1. **Thuật toán IWO**

Invasive weed optimization(IWO): thuật toán tối ưu cỏ dại xâm lấn, lấy cảm hứng từ các quần thể cỏ dại ngoài tự nhiên dựa vào hành vi sống, sự thích nghi tốt với môi trường được đề xuất bởi Alireza Mehrabian và Caro Lucas năm 2006.

**Nội dung**

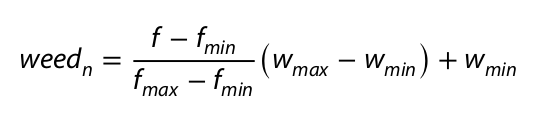
Thuật toán chia làm 4 giai đoạn:

* Initialization: Khởi tạo quần thể ngẫu nhiên các hạt trên không gian tìm kiếm
* Reproduction: Các hạt phát triển thành cây, và sinh ra nhiều hạt, tùy thuộc vào giá trị hiện tại
* Spatial dispersal: Các hạt sinh ra được tiếp tục phân tán, rồi phát triển thành cây mang giá trị
* Competitive exclusion: Khi đạt được số lượng cây quy định, tiến hành chọn lọc ra các cây có giá trị cao (low fitness), bỏ đi các cây khác kém hơn.

Quá trình loại trừ theo cơ chế sau:

* Mỗi cá thể đều sinh ra hạt dựa vào giá trị hiện tại của chính nó
* Các hạt sinh ra được phân tán ngẫu nhiên
* Giá trị của mỗi cá thể được so sánh với cha của nó, và chỉ giữ lại các cây có giá trị cao.

Công thức sinh hạt như sau:



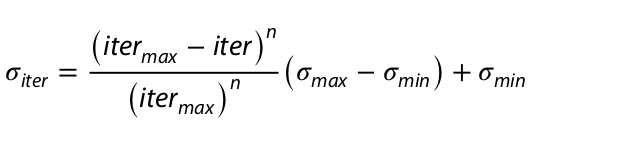
Trong đó:

f: giá trị sức sống hiện tại của cá thể

fmax và f\_min là 2 ngưỡng sức sống của quần thể hiện tại

wmax và wmin là 2 ngưỡng số lượng cá thể đã đặt ra

Quá trình phân tán hạt giống mô tả trong công thức:



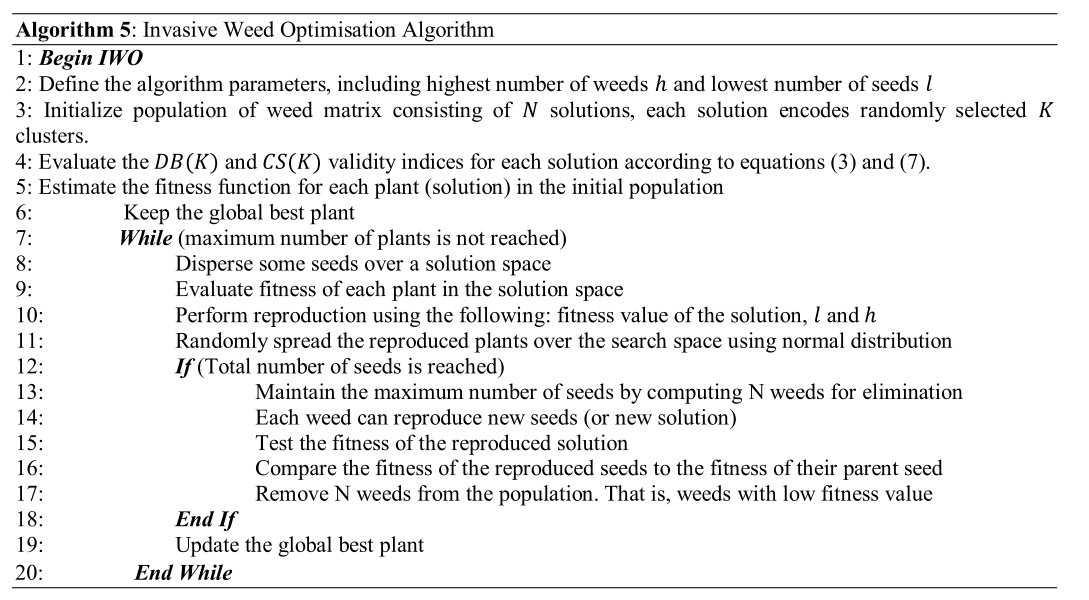
Trong đó:

Itermax là số vòng lặp tối đa

σiter là độ lệch chuẩn hiện tại

n là hệ số điều chỉnh

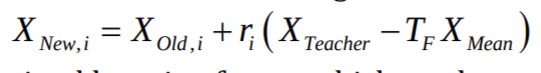
**Mã giả thuật toán**



Hình H Mã giả thuật toán IWO

1. **Thuật toán TLBO**

Teaching learning-based optimization: Tối ưu hóa dạy và học, là một thuật toán heuristic phổ biến, ra đời năm 2011. Thuật toán lấy cảm hứng từ hiện tượng người giáo viên tác động lên nhóm học sinh. Theo đó thuật toán chia thành 2 pha riêng biệt:

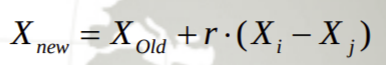
* Pha dạy: Giáo viên truyền đạt kiến thức như nhau đối với các học sinh. Khả năng lĩnh hội kiến thức sẽ phụ thuộc vào lực học của mỗi học sinh. 

là hệ số học (nhận giá trị 1 hoặc 2)

là người giỏi nhất

là độ giỏi trung bình của cả lớp

random trong khoảng

* Pha học: Học sinh tự tăng độ giỏi của mình thông qua việc giao tiếp với bạn bè trong lớp. Ở đó, bạn học sẽ được chọn ngẫu nhiên, trong trường hợp bạn học giỏi hơn, học sinh đó sẽ tăng được kiến thức, và ngược lại.

Theo đó, Chất lượng trung bình của cả lớp học phụ thuộc vào chất lượng của người dạy.

1. **Lịch sử cải tiến, lai tạo giữa các thuật toán**

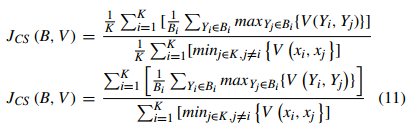
* Hassanzadeh và Meybodi đã trình bày một phương pháp kết hợp dựa trên FA và k-means [2]
* Zabihi và Nasiri đề xuất việc sử dụng thuật toán ABC theo history-driven để giải quyết phân cụm dữ liệu, với cơ chế ghi nhớ dựa trên phân vùng không gian nhị phân đã được kết hợp vào thuật toán ABC, để cải thiện hiệu suất phân cụm.
* Merwe và Engelbrecht đã đề xuất việc sử dụng PSO để giải quyết vấn đề phân cụm
* Zhao đã làm về việc cải thiện hiệu suất của thuật toán k-mean bằng cách lai nó với PSO để tránh bị ảnh hưởng trực tiếp bởi các trung tâm cụm ban đầu.
* Nikkam và cộng sự đã đề xuật một thuật toán cải tiến hiệu quả kết hợp ACO và các thuật toán simulated annealing (SA) để giải quyết. Kết quả mô phỏng của ACO-SA cho thấy thuật toán kết hợp hoạt động hiệu quả hơn SA cơ bản, ACO và k-mean riêng lẻ.
* Satapathy và Naik đã phát triển một thuật toán TLBO đã được sử dụng để tìm các tâm của một số lượng cụm do người dùng chỉ định.

1. **So sánh các phương pháp lai được bài báo đề xuất**
2. **Các hàm mục tiêu được sử dụng**
   1. **Compact-Separated** [3]

Là chỉ số ước lượng chất lượng phân cụm dựa vào tỉ lệ của 2 yếu tố:

* Điểm biểu thị chất lượng phân tán trong cụm (within-cluster scatter): Khoảng cách xa nhất có thể giữa 2 điểm trong cụm
* Điểm phân tách giữa các cụm (between-cluster separation): Khoảng cách giữa 2 tâm cụm

Công thức được giới thiệu trong bài báo như sau:

Trong đó:

lần lượt là các tập dữ liệu đã phân thành K cụm, và độ đo khoảng cách được chọn

là khoảng cách giữa 2 điểm bất kì được xét trong cùng một cụm

là khoảng cách giữa 2 tâm cụm đang xét

càng nhỏ thì kết quả phân cụm càng tốt và ngược lại.

* 1. **Davis-Bouldin** [3]

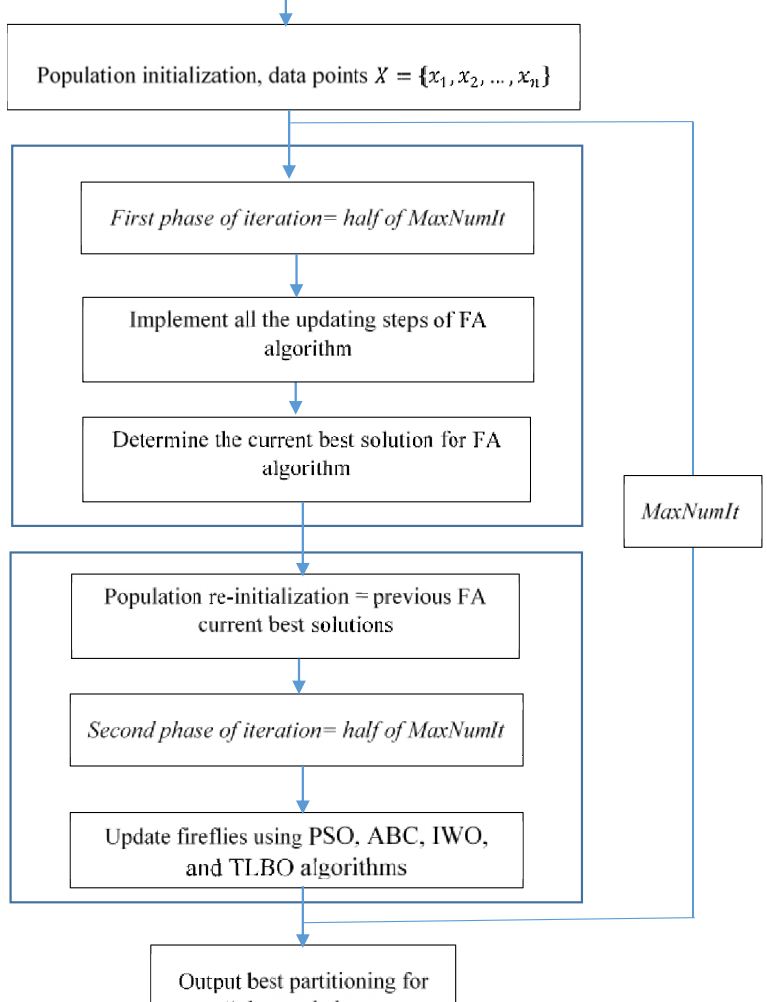
Là chỉ số ước lượng chất lượng phân cụm rất phổ biến, đề xuất bởi Davis và Bouldin. Phép đánh giá này dựa trên 2 tiêu chí:

* intra-cluster (khoảng cách trung bình giữa tất cả các điểm trong cụm tính từ tâm cụm)
* inter-cluster (Khoảng cách giữa 2 tâm cụm)

Nhìn chung 2 chỉ số đánh giá này có cách vận hành tương đối giống nhau, đều dựa vào 2 yếu tố nội suy từ tập dữ liệu đầu vào và tập các tâm ứng viên.

1. **4 Thuật toán lai được đề xuất**

Lưu đồ thuật toán tổng quát của 4 phương pháp lai được mô tả trong bài báo [4] như hình sau:

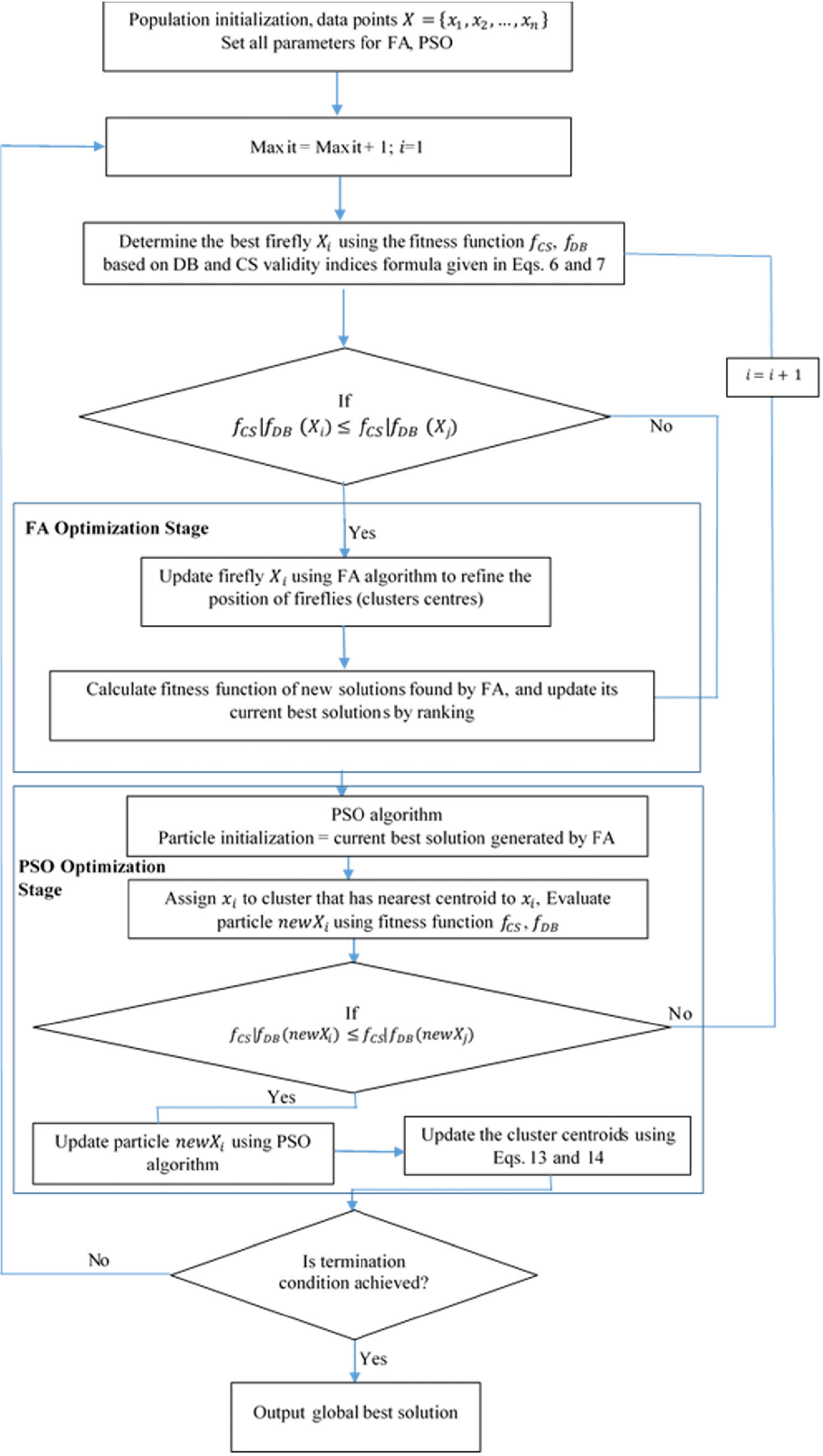
Theo đó, trong thuật toán này, mỗi lần duyệt qua một cá thể đom đóm đều được chia thành 2 pha rõ rệt:

Hình I Lưu đồ tổng quát thuật toán lai [2]

* Pha 1 vẫn thực hiện cập nhật cá thể theo thuật toán FA. Bước này là dễ hiểu vì thuật toán gốc là FA, có ưu điểm là tận dụng được khả năng tìm kiếm ngẫu nhiên của thuật toán FA để nhanh chóng tìm ra global optima.
* Pha 2 dựa trên bước đi của pha 1, sử dụng cách di chuyển của PSO để cập nhật vị trí của cá thể ngay sau đó. Bước này sẽ giúp quần thể đom đóm tăng cơ hội rơi vào local optima có thể có xung quanh cá thể đang duyệt.

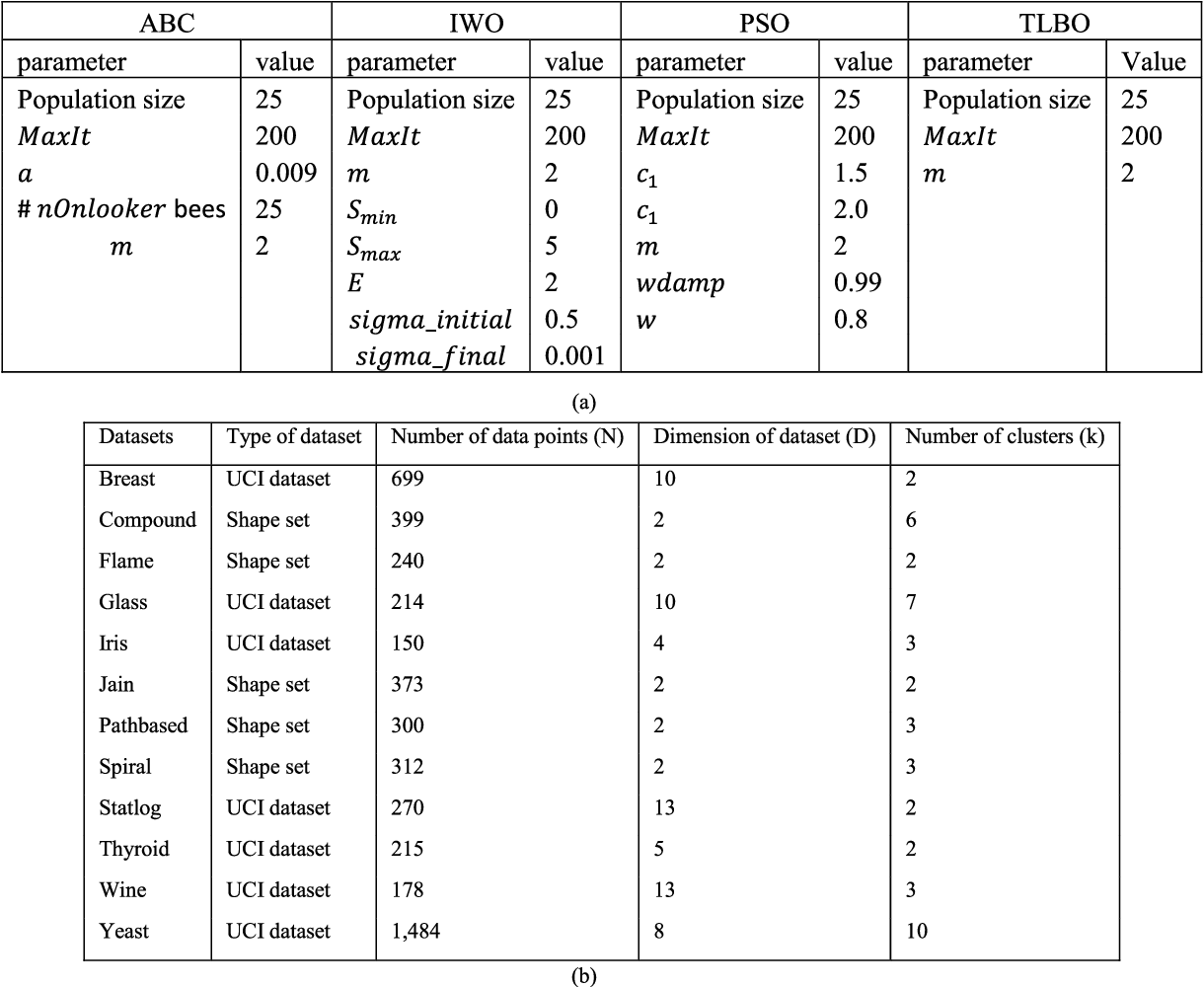
Kết hợp cả 2 yếu tố trên có thể giúp bài toán hội tụ nhanh hơn, khai thác được thế mạnh của 2 chiến lược tìm kiếm độc lập nhau.

Mô tả chi tiết một phép lai cụ thể là FA-PSO được thể hiện trong lưu đồ thuật toán sau [5]:



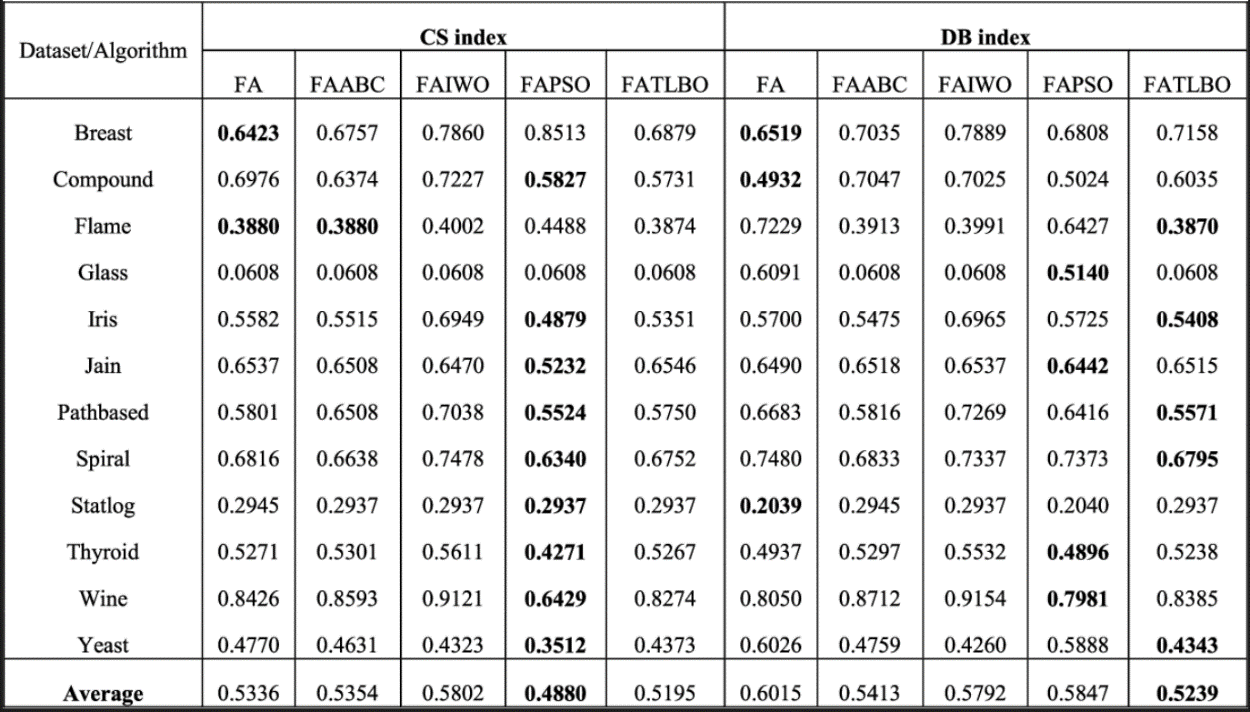
Hình J Lưu đồ chi tiết thuật toán FAPSO [5]

1. **Thực nghiệm và so sánh**
2. **Dataset**

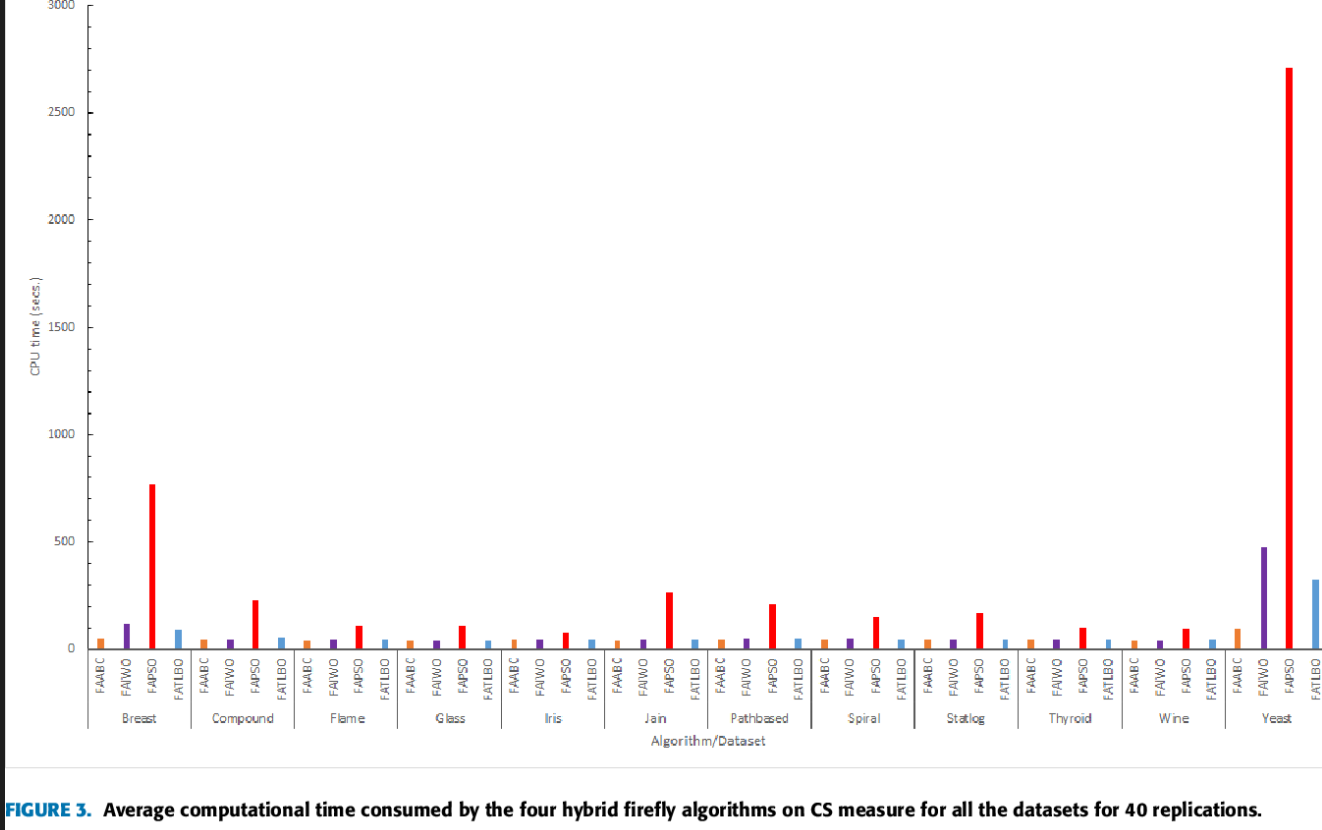
Để thực hiện chạy thực nghiệm các thuật toán, nhóm tác giả đã sử dụng 12 bộ dữ liệu khá phổ biến trên UCI để làm cơ sở so sánh, được liệt kê trong bảng sau:

Hình K Thông số các bộ dữ liệu được sử dụng để chạy thuật toán lai

1. **Kết quả thực nghiệm và phân tích**
2. **Kết quả so sánh tổng quan**



Hình L Kết quả chạy thực nghiệm dựa vào 2 chỉ số đánh giá [2]

1. **So sánh thời gian chạy thuật toán**

Hình M So sánh thời gian chạy thuật toán trên các tập dữ liệu [2]

1. **Kết luận**

Trong nghiên cứu này, nhóm tác giả đã kết hợp thuật toán đom đóm (FA) với 4 thuật toán khác bao gồm PSO, ABC, IWO và TLBO.

Kết quả thực nghiệm cho thấy: thuật toán lai giữa FA và PSO (FAPSO) cho kết quả vượt trội so với 3 thuật toán lại còn lại về chất lượng phân cụm và tốc độ chạy. Mặt khác thì FATLBO đạt được kết quả tốt thứ hai (sau FAPSO) nhưng có thời gian chạy nhanh hơn, FAIWO cho kết quả tệ nhất cả về mặt kết quả và thời gian chạy.

Nhóm tác giả định hướng tương lai sẽ kết hợp thuật toán đom đóm với các thuật toán khác để giải những bài toán phân cụm phức tạp hơn.

*Nhận xét: Các phương pháp, phép đánh giá trong bài báo được chọn là thích hợp, tập dữ liệu được sử dụng cũng rất phổ biến trong giới nghiên cứu. Những đóng góp này của nhóm tác giả đã cho thấy tiềm năng cải tiến vẫn còn trong bài toán clustering. Tuy nhiên cá nhân nhóm đánh giá các bộ dữ liệu chưa đủ lớn, các đặc trưng chưa đủ mạnh và chưa sát với thực tế, nên khả năng ứng dụng vào các dự án thực tế còn chưa nổi bật rõ ràng so với các thành tựu trước đó.*

# References

|  |  |
| --- | --- |
| [1] | X.-S. Yang, "Firefly Algorithms for Multimodal Optimization," 2009. |
| [2] | (. I. M. B. A. N. A. R. E. H. C. (. I. A. M. A. E. Absalom El-shamir Ezugwu 1, "A Comparative Performance Study of Hybrid Firefly Algorithms for Automatic Data Clustering," *Digital Object Identifier 10.1109/ACCESS.2020.3006173,* vol. 8, 2020. |
| [3] | Y. B. M. V. Maria Halkidi, "Clustering Validity Checking Methods: Part II". |
| [4] | A. E. Ezugwu, "Nature‑inspired metaheuristic techniques for automatic clustering: a survey and performance study," 2020. |
| [5] | A. e. E. ,. a. R. E. Moyinoluwa b. Agbaje, "Automatic Data Clustering Using Hybrid Firefly," vol. 7, 2019. |