

Abstract geometric lines in the top left corner, consisting of several overlapping, irregular polygons and lines in a light beige color.

超啟發式演算法

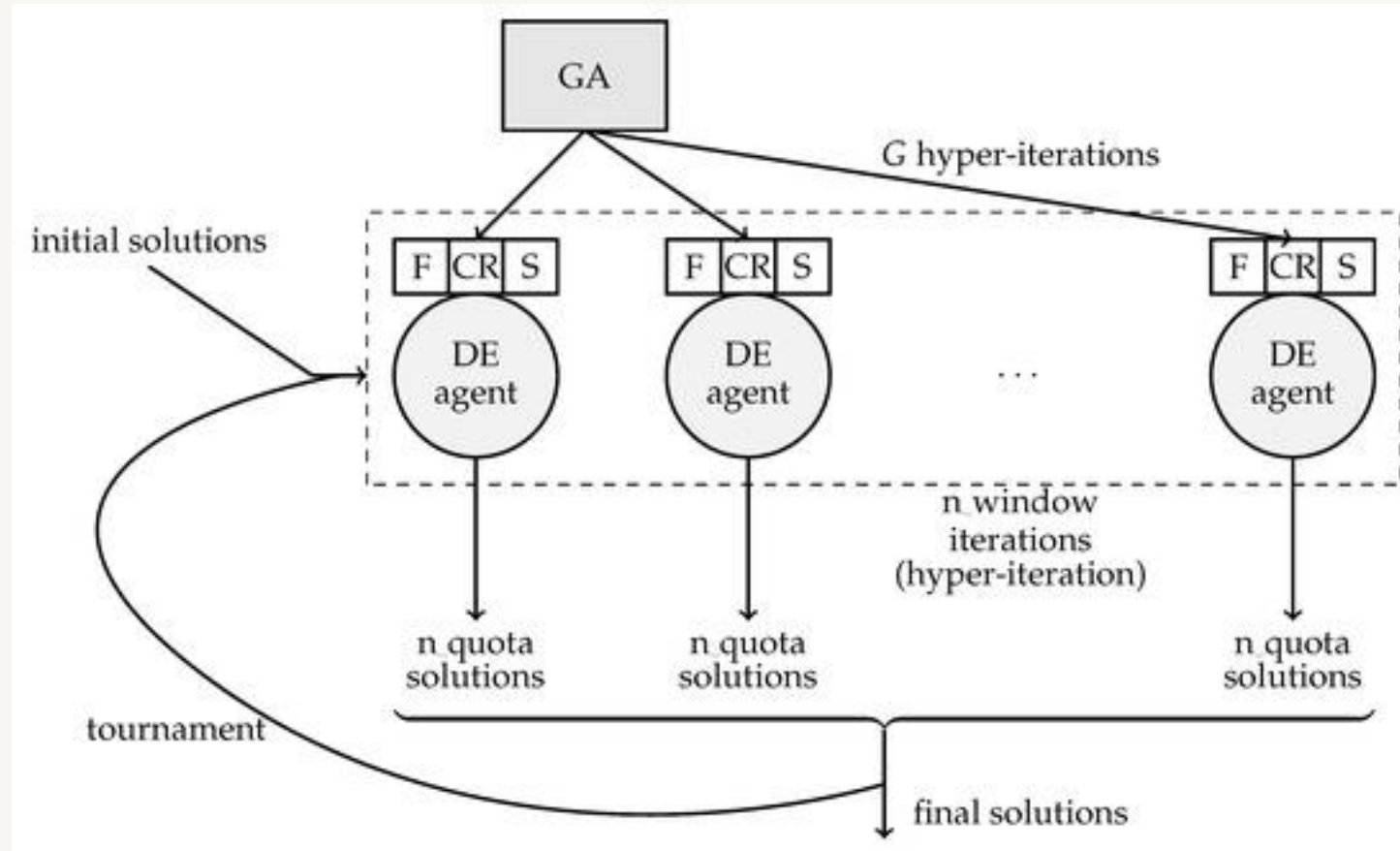
軟創三乙 511172176 李則霖

簡述

利用在超啟發式演算法課程所學，瞭解超啟發演算法的演算概念。利用現有的學術文章了解現有超啟發演算在全域搜尋上的解決方案，並且嘗試使用生成式AI工具理解論文中解決方案的執行方法。

1. ————— **HyperDE超啟發式演算法**
2. ————— **演算機制**
3. ————— **實驗方法與產出**
4. ————— **論文規劃**

超差分演化演算法 (Hyper Differential Evolution)



Manescu, A.-R., & Dumitrescu, B. (2023). HyperDE: An adaptive hyper-heuristic for global optimization. *Algorithms*, 16(9), 451. <https://doi.org/10.3390/a16090451>

演算機制

HyperDE的演算法機制主要包括以下幾個關鍵組件和步驟，這些組件共同作用以實現超參數調整和優化性能的提升：

1. 多層結構

HyperDE採用兩層結構：

- **高層算法**：使用穩態遺傳算法（GA）作為高層算法，負責管理和調整低層的DE實例的超參數。
- **低層算法**：多個DE實例作為低層算法，每個實例使用不同的超參數進行優化。

2. 超參數的定義

HyperDE的主要超參數包括：

- **F**：權重因子，範圍在 $[0, 1]$ 之間，影響變異步驟的強度。
- **CR**：交叉率，範圍在 $[0, 1]$ 之間，決定了父代和子代之間的基因交換概率。
- **S**：變異策略索引，通常是一個整數，指示使用的變異策略類型。

演算機制

3. 初始化

- **群體初始化**：在演算法開始時，隨機生成一組DE實例，每個實例都有一組隨機選擇的超參數（F, CR, S）。
- **適應度評估**：每個DE實例在給定的優化問題上進行評估，計算其適應度值。

6. 終止條件

演算法在達到預定的迭代次數或當適應度值不再顯著改善時終止，並返回最佳解。

7. 性能評估

在演算法結束後，HyperDE會對其性能進行評估，通常會與其他優化算法進行比較，以驗證其有效性和優越性。

演算運作

1.初始化：

設定超參數範圍（ F, CR, S ）。

初始化一個種群，包含多個DE實例，每個實例具有不同的超參數組合。

2.主迴圈：

對於每一代：

對於每個DE實例：

- 生成候選解。
- 計算候選解的適應度。
- 根據適應度選擇最好的解。
- 更新超參數（ F, CR, S ）以改進搜索性能。

3.選擇最佳解：

在所有DE實例中選擇適應度最好的解作為當前代的最佳解。

4.終止條件：

檢查是否滿足終止條件（如達到最大代數或適應度不再改善）。

實驗方法

在HyperDE演算法的研究中，實驗方法主要包括以下幾個步驟：

1. 基準函數選擇

研究中使用了12個基準函數，這些函數來自CEC 2022競賽，這些基準函數的選擇目的在評估演算法在不同類型的優化問題上的性能。

2. 演算法設置

HyperDE演算法的設置包括初始化超參數（F, CR, S）和解的初始種群。這些超參數的範圍分別為：

- $F \in [0, 1]$ （權重因子）
- $CR \in [0, 1]$ （交叉率）
- $S \in [0, 5]$ （整數，變異策略的索引）。

實驗方法

3. 實驗設計

- 演算法的性能評估是通過多次獨立運行來進行的，每個基準函數的測試都重複進行60次，以獲得穩定的平均結果。
- 每次運行中，HyperDE會生成一組解，並使用固定的超參數進行搜索，以獲得穩定的平均結果。

4. 性能評估

演算法的性能通過以下指標進行評估：

- **平均性能**：計算每個基準函數的平均最小值。
- **Friedman排名**：對所有參與的演算法進行排名，以比較其相對性能。
- **收斂速度**：通過繪製收斂曲線來分析不同演算法在解空間中的表現。

5. 結果分析

- 實驗結果包括HyperDE與其他算法（如SHADE、LSHADE等）的比較，並分析其在不同基準函數上的表現差異。
- 研究還考慮了執行時間，並對比了不同算法在相同條件下的效率。

實驗產出

1.性能比較：

HyperDE算法在多個基準函數上進行了測試，並與其他優化算法（如SHADE、LSHADE、DE等）進行了比較。

根據Friedman排名，HyperDE的排名為1.875，顯示其在所有評估的算法中表現最佳。其他算法的排名依次為：

1. SHADE: 2.680
2. LSHADE: 3.097
3. Slime Mold Algorithm (SMA): 4.222
4. HyperBES: 4.916
5. Differential Evolution (DE): 5.458

2.統計顯著性：

使用Wilcoxon簽名秩檢驗來評估HyperDE與其他算法之間的統計顯著性。結果顯示：

1. 在F1、F2、F4、F6和F7上，HyperDE與其他算法之間存在顯著差異，p值非常低（例如F1的p值約為 1.63×10^{-11} ）。
2. 在F3上，HyperDE與SHADE/LSHADE之間沒有顯著差異（p值為0.16），但DE的表現明顯不如HyperDE。

實驗產出

3.收斂性能：

HyperDE在大多數基準函數上顯示出良好的收斂性能，通常能夠更快地找到更優的解。

4.執行時間：

雖然HyperDE的性能優越，但其執行時間比一些簡單的算法（如SHADE和LSHADE）慢60-80%。這是因為HyperDE需要進行額外的計算和操作。

5.基準函數的表現：

在12個基準函數上進行的實驗顯示，HyperDE在大多數情況下都能提供優於其他算法的解，特別是在F1、F2、F4、F6和F7等函數上。

實驗結論

1.提出的算法：

本文提出了一種基於差分演化（DE）算法的自適應超啟發式方法HyperDE，並與其他類似方法進行了比較。

2.性能優越性：

HyperDE在CEC競賽的基準測試中表現出色，獲得了最佳的相對誤差和Friedman排名，顯示出其在全局優化中的優越性。

3.低超參數需求：

所提出的方法具有較低的超參數需求，無需進行額外的調整，這使得其在多種問題上的適用性更強。

4.未來的研究方向：

研究計劃進一步提高算法的效率，探索其內在的並行性，並將該技術擴展到其他高效的啟發式算法中。

5.總結：

HyperDE及其變體（HyperSSA和HyperBES）顯示出相對於其對應算法的顯著改進，為未來的優化研究提供了新的思路和方法。

論文規劃

1. 先了解IEEE論文格式規範。
2. 嘗試利用生成式AI產出更好的解決方案。
3. 嘗試利用生成式AI產出符合規範的TEX程式碼產出論文。

IEEE *Xplore*[®]
Digital Library

TEX

圖片來源:網路抓取

報告結束，感謝聆聽。

TSE,LIN LE

511172176@m365.fju.edu.com

[511172176 \(李則霖 LI, TSE-LIN\) · GitHub](#)

