

基於局部搜索改進探索粒子群優化算法

黃梓濤

國立中山大學

摘要

Particle Swarm Optimization (PSO) 是一種源自於觀察鳥群覓食行為的啟發式演算法。鳥群在尋找食物時，雖然能感知自身與食物之間的距離，卻不知道食物的確切位置。因此，鳥群會同時朝不同方向探索食物，並在休息時分享彼此的資訊。

PSO 演算法的運作方式模擬了鳥群的行為。每隻粒子會根據其上次的飛行方向（速度 $V(t)$ ）、自身曾到達的最佳位置（個體最佳解 $pBest$ ）以及整個群體已達到的最佳位置（全域最佳解 $gBest$ ），來決定下一步的移動方向。粒子之間會透過共享資訊的方式，漸漸靠近全域最佳解，模擬了鳥群逐漸靠近食物的行為。

在 PSO 的實際應用中，粒子群有時會陷入局部最優解，難以進一步提升解的品質。為了克服這一問題，本文提出了一種改進的局部搜索策略。我們引入了一種動態局部搜索機制，當演算法在多次迭代後未能取得顯著改進時，會觸發局部搜索。此局部搜索方法隨機選擇若干粒子，並在隨機維度上進行局部搜索，從而生成新的候選解。這些候選解的適應度與原粒子進行比較，若新候選解更優，則取代原粒子。

本實驗旨測試了十種不同的測試函數，包括 Ackley、Rastrigin、HappyCat 等，並比較了改進局部搜索策略和原本演算法對性能的影響。透過實驗分析，改進策略能有效提升演算法的收斂速度和全局搜索能力。特別是在高維函數優化問題中，改進的 PSO 表現出了更強的跳出局部最優解的能力，從而達到更好的優化效果。

關鍵詞：啟發式演算法、PSO、粒子群最佳化。

1. 簡介

PSO 的原理是讓每個粒子根據其當前位置、歷史最佳位置以及全局最佳位置進行更新。粒子的位置和速度會在每次迭代中調整，使得整個群體逐步朝著最優解的方向移動。然而，PSO 在實際應用中經常遇到粒子群陷入局部最優解的問題，這限制了演算法的搜索能力。

為了解決這一問題，本文提出了一種改進的局部搜索策略。在 PSO 的基礎上，加入了動態局部搜索機制，當演算法在若干次迭代後未能取得顯著改進時，觸發局部搜索。該策略通過隨機選擇若干粒子，並在其部分維度上進行微小擾動來生成新的候選解，進一步提高了演算法跳出局部最優解的能力。

本文詳細介紹了改進策略的實現方法，並通過多個標準測試函數包括，包括 Ackley、Rastrigin、HappyCat、Rosenbrock、Zakharov、Michalewicz、Schwefel、BentCigar、DropWave、Step 對其進行了實驗驗證。實驗結果表明，改進的 PSO 在收斂速度和全局搜索能力方面均有顯著提升，特別是在高維優化問題中表現尤為突出。這一改進策略為解決高維優化問題提供了一種有效的新方法，展示了其在實際應用中的潛力。

2. 相關研究

1. 粒子群最佳化演算法

粒子群最佳化（Particle Swarm Optimization, PSO）是一種啟發式演算法，源自對鳥群覓食行為的觀察。PSO 透過模擬鳥群在空間中的移動，以尋找最佳解。算法中每個解被視為一個粒子，而這些粒子在最

佳化問題的解空間中移動位置。在 PSO 中，每個粒子根據個體自己的慣性和粒子自身的經驗，以及群體的經驗，來調整自己的移動方向和速度，在不同方向探索到更好的解。

3. 研究方法

1. 經典 PSO

1. 初始化粒子位置和速度：

隨機生成一個粒子群，每個粒子都有一個初始位置和速度。

粒子的初始位置在解空間範圍內隨機生成。

粒子的初始速度在限定範圍內隨機生成。

2. 直到迭代結束前歷遍每個粒子：

• 更新粒子速度：

$$v_i = w \cdot v_i + c1 \cdot r1 \cdot (PBest_i - x_i) + c2 \cdot r2 \cdot (GBest - x_i)$$

其中， v_i 是第 i 個粒子的新速度， w

是慣性權重， $c1$ 和 $c2$

是加速係數， $PBest_i$ 是該粒子的個體最優解， $GBest$

是整個粒子群的全局最優解 r_1 和 r_2

是範圍在 $[0, 1]$ 內的隨機數。

• 更新粒子位置：

$$x_i = x_i + v_i$$

其中， x_i 是第 i 個粒子的新位置。

- 更新個體最優解：比較粒子新位置的目標函數值和其個體最優解的目標函數值，如果新位置更好，則將其設置為新的個體最優解。
- 更新全局最優解：比較所有粒子的個體最優解，找出最優的解作為全局最優解。

2. 基於局部搜索改進 PSO

為了提高經典 PSO 的性能，我們提出了一種基於局部搜索的改進 PSO 算法。這種改進主

要在於當算法陷入局部最優時，對選定的粒子進行局部搜索，以探索更優的解。

1. 初始設置：與經典 PSO 相同，初始化粒子的速度和位置。

2. 速度和位置更新：按照經典 PSO 的更新公式，更新粒子的速度和位置。

3. 局部搜索：

局部搜索觸發條件：在每次迭代中，全局最優解在多次迭代中沒有改變，則觸發局部搜索。

隨機選擇 10 個粒子，對這些粒子進行局部搜索。

被選中的每個粒子會產生維度數 50% 數量的新粒子，每個新粒子會在隨機 1 個維度上加上 $[0,1]$ 隨機數。

計算新粒子的適應度，如果新粒子的適應度優於原位置，則用新粒子替代原粒子。

4. 適應度計算和更新：計算每個粒子的適應度，並更新個體最佳位置和全局最佳位置。

5. 迭代：重複上述步驟直至達到最大迭代次數或滿足停止條件。

4. 實驗設計

在這個實驗中，我們將對兩個方法針對十種不同的測試函數（Ackley、Rastrigin、HappyCat、Rosenbrock、Zakharov、Michalewicz、Schwefel、BentCigar、DropWave 和 Step）進行實驗，並分別評估其在 2 維、10 維和 30 維空間中的性能和收斂效果。

5. 實驗結果

從實驗結果來看，研究中提出的基於局部搜索改進的 PSO 算法在多個測試函數上表現優於經典 PSO 算法。改進後的算法在大多數情況下能夠取得更好的平均、最差和最佳解，特別是在高維度問題上顯示出明顯的優勢。根據表格 1 和 表格 2，對於 BentCigar 和 Michalewicz 函數，在 10D 和 30D 的測試中，改進 PSO 相對於經典 PSO 展現了更好的性能，其平均和最差解均有顯著提升。

此外，實驗還顯示出改進 PSO 能夠更穩定地接近最優解，避免了陷入局部最優解的情況。圖七顯示了通過局部搜索明顯地有效跳出局部最佳解，能探索到更佳解。可見，基於局部搜索改進的 PSO 算法在解決複雜優化問題時表現出更好的全局搜索能力和收斂速度。

另外，從表 3 可見，加入局部搜索後時間成本沒有明顯增加，算法在提高搜索效率的同時，保持了較低的計算開銷，使其更具實用性和可應用性。

表 1.基本 PSO 實驗結果

測試函數	平均	最差	最好
Ackley_2D	4.441e-16	4.441e-16	4.441e-16
Ackley_10D	4.441e-16	4.441e-16	4.441e-16
Ackley_30D	4.441e-16	4.441e-16	4.441e-16
BentCigar_2D	0.000e+00	0.000e+00	0.000e+00
BentCigar_10D	6.667e+02	1.000e+04	0.000e+00
BentCigar_30D	6.667e+02	1.000e+04	0.000e+00
DropWave_2D	0.000e+00	0.000e+00	0.000e+00
DropWave_10D	0.000e+00	0.000e+00	0.000e+00
DropWave_30D	0.000e+00	0.000e+00	0.000e+00
HappyCat_2D	8.854e-03	4.474e-02	2.759e-05
HappyCat_10D	1.498e-01	3.072e-01	4.384e-02
HappyCat_30D	5.046e-01	7.084e-01	3.339e-01
Michalewicz_2D	-	-	-
	1.801e+00	1.801e+00	1.801e+00
Michalewicz_10D	-	-	-
	8.059e+00	5.846e+00	9.438e+00
Michalewicz_30D	-	-	-
	2.440e+01	1.861e+01	2.779e+01
Rastrigin_2D	0.000e+00	0.000e+00	0.000e+00
Rastrigin_10D	8.290e-01	2.487e+01	0.000e+00
Rastrigin_30D	2.428e+01	8.677e+01	0.000e+00
Rosenbrock_2D	0.000e+00	0.000e+00	0.000e+00
Rosenbrock_10D	1.203e-08	3.608e-07	0.000e+00
Rosenbrock_30D	2.384e+01	2.663e+01	2.055e+01
Schwefel_2D	3.947e+01	1.184e+02	2.546e-05
Schwefel_10D	5.981e+02	1.192e+03	1.273e-04
Schwefel_30D	2.753e+03	5.136e+03	1.421e+03

Step_2D	0.000e+00	0.000e+00	0.000e+00
Step_10D	0.000e+00	0.000e+00	0.000e+00
Step_30D	0.000e+00	0.000e+00	0.000e+00
Zakharov_2D	0.000e+00	0.000e+00	0.000e+00
Zakharov_10D	0.000e+00	0.000e+00	0.000e+00
Zakharov_30D	2.393e+01	2.009e+02	0.000e+00

表 2. 基於局部搜索改進 PSO 實驗結果

測試函數	平均	最差	最好
Ackley_2D	4.441e-16	4.441e-16	4.441e-16
Ackley_10D	4.441e-16	4.441e-16	4.441e-16
Ackley_30D	4.441e-16	4.441e-16	4.441e-16
BentCigar_2D	0.000e+00	0.000e+00	0.000e+00
BentCigar_10D	5.997e-54	1.799e-52	0.000e+00
BentCigar_30D	5.376e-03	1.198e-01	0.000e+00
DropWave_2D	0.000e+00	0.000e+00	0.000e+00
DropWave_10D	0.000e+00	0.000e+00	0.000e+00
DropWave_30D	0.000e+00	0.000e+00	0.000e+00
HappyCat_2D	8.700e-03	4.418e-02	1.208e-06
HappyCat_10D	1.248e-01	2.027e-01	5.168e-02
HappyCat_30D	3.478e-01	4.480e-01	2.230e-01
Michalewicz_2D	-	-	-
	1.801e+00	1.801e+00	1.801e+00
Michalewicz_10D	-	-	-
	9.644e+00	9.551e+00	9.660e+00
Michalewicz_30D	-	-	-
	2.814e+01	2.603e+01	2.930e+01
Rastrigin_2D	0.000e+00	0.000e+00	0.000e+00
Rastrigin_10D	0.000e+00	0.000e+00	0.000e+00
Rastrigin_30D	6.632e+00	2.487e+01	0.000e+00
Rosenbrock_2D	0.000e+00	0.000e+00	0.000e+00
Rosenbrock_10D	4.834e-30	6.163e-29	0.000e+00
Rosenbrock_30D	1.335e+01	1.693e+01	1.509e-05
Schwefel_2D	3.157e+01	1.184e+02	2.546e-05
Schwefel_10D	7.049e+02	1.190e+03	2.369e+02
Schwefel_30D	2.167e+03	3.813e+03	1.306e+03
Step_2D	0.000e+00	0.000e+00	0.000e+00
Step_10D	0.000e+00	0.000e+00	0.000e+00
Step_30D	0.000e+00	0.000e+00	0.000e+00
Zakharov_2D	0.000e+00	0.000e+00	0.000e+00
Zakharov_10D	0.000e+00	0.000e+00	0.000e+00
Zakharov_30D	0.000e+00	0.000e+00	0.000e+00

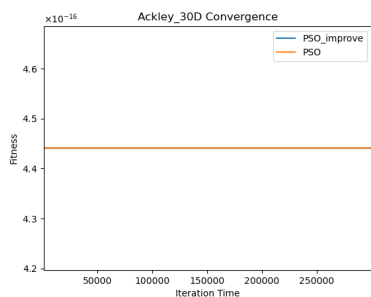
表 3. 比較兩個方法時間成本

測試函數(30 維)	基本 PSO	基於局部搜索改進 PSO
Ackley	8.83601	8.79976
Rastrigin	8.61994	8.68866
HappyCat	7.52777	7.48459
Rosenbrock	7.98448	7.71303
Zakharov	8.45031	7.86064
Michalewicz	17.1912	16.9209
Schwefel	8.78056	8.94469
BentCigar	8.07265	7.00112
DropWave	8.45563	8.1871
Step	8.04167	7.95826

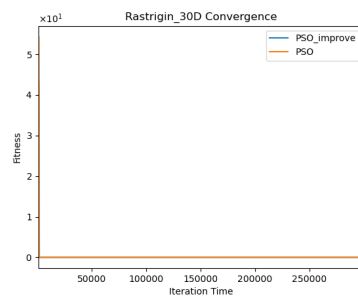
6. 結論

通過對該算法在多個標準測試函數上的實驗驗證，發現改進後的 PSO 算法相對於經典 PSO 算法具有更好的全局搜索能力和收斂速度。改進後的算法能夠在大多數情況下取得更好的平均、最差和最佳解，特別是在高維度問題上顯示出明顯的優勢。

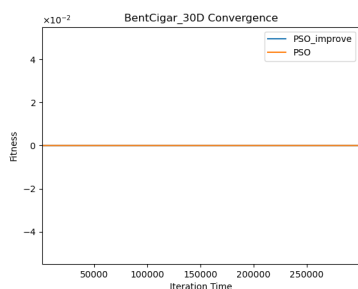
此外，實驗結果還表明，改進後的 PSO 算法能夠更穩定地接近最優解，避免了陷入局部最優解的情況。總的來說，基於局部搜索改進的 PSO 算法在解決複雜優化問題時表現出顯著的優越性，為相關領域的研究和應用提供了有價值的參考和指導。



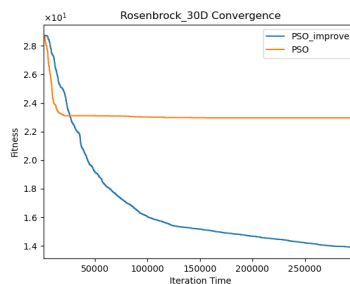
圖一 .Ackley 30 維收斂比較



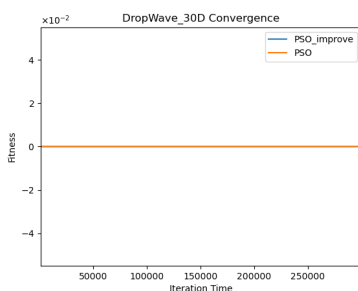
圖六 .Rastrign 30 維收斂比較



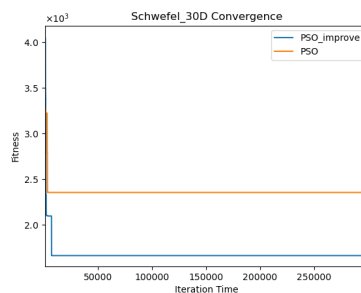
圖二 .BentCigar 30 維收斂比較



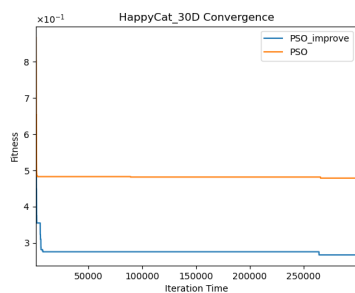
圖七 .Rosenbrock30 維收斂比較



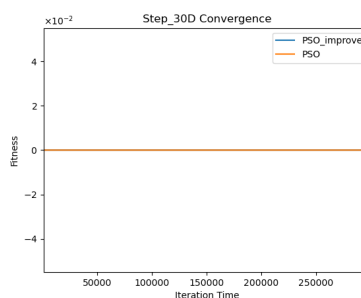
圖三 .DropWave 30 維收斂比較



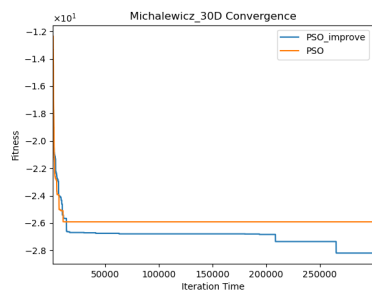
圖八 .Schwefel 30 維收斂比較



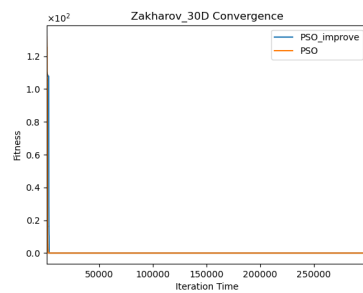
圖四 .Happy Cat 30 維收斂比較



圖九 .Step 30 維收斂比較



圖五 .Michalewicz 30 維收斂比較



圖十.Zakharov 30 維收斂比較