

Shuffled Frog Leaping Algorithm

R08546010 吳珍瑋

蛙跳演算法

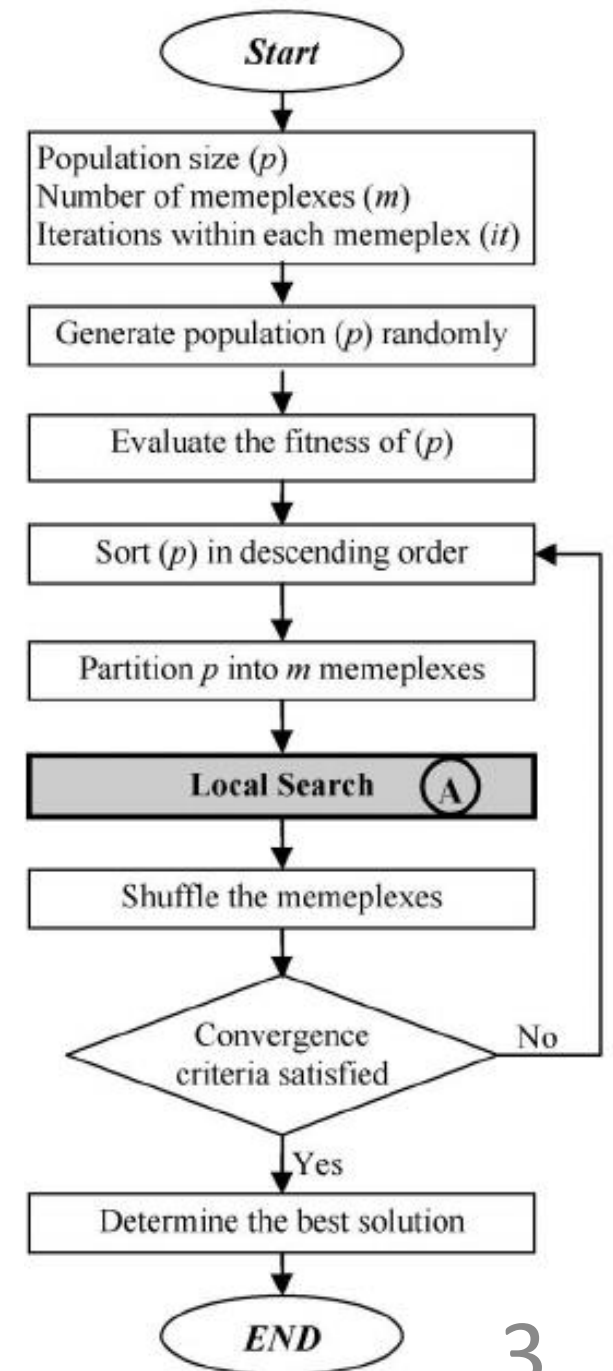
蛙跳演算法(Shuffled Frog Leaping Algorithm)最早於2006年提出

起初的想法是在一片溼地裡布滿了石頭, 有很多隻青蛙在這片棲地裡生活, 他們透過在石頭間跳躍來尋找食物較多的地方。

同時每隻青蛙都有自己不同的文化, 每個文化代表不同的解, 他們藉由文化的高低也就是解的好壞來進行分組, 文化相同的青蛙會群聚在一起, 並做訊息的交流, 在交流的過程中, 會產生綜合組內文化最好和最壞的青蛙的新青蛙, 來取代最壞的青蛙進行迭代更新。

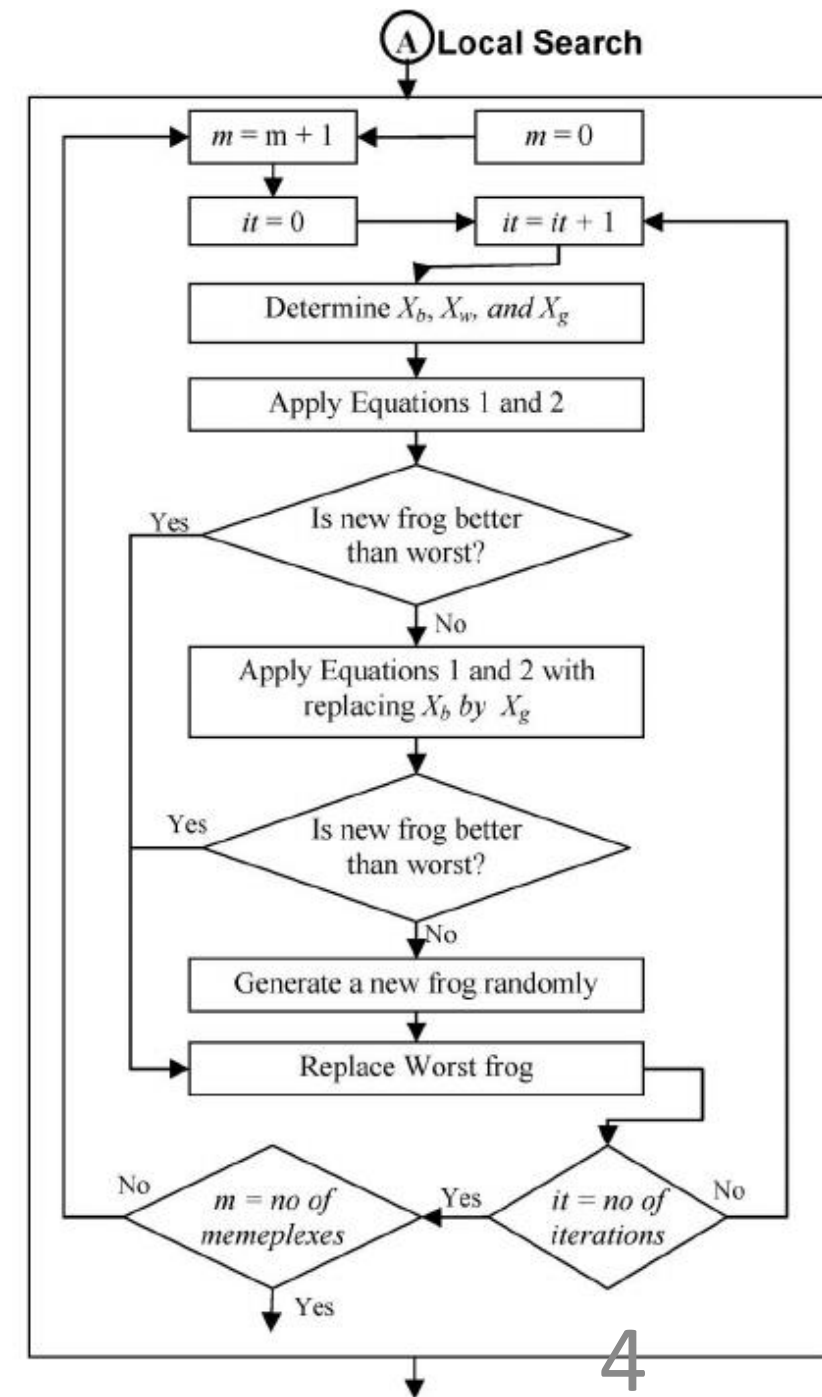
演算法架構 - 全域搜索

- (1) 生成散佈在各處的隨機解
- (2) 計算每個解的 Fitness 值
- (3) 將每個解依照 Fitness 排列
- (4) 將排列完的分成 m 組
- (5) 對每個分組內進行區域搜索
- (6) 將更新完的解重新排列
- (7) 檢視是否滿足停止條件，若是則計算最佳解值，若否則回到(3)



演算法架構 – 區域搜索

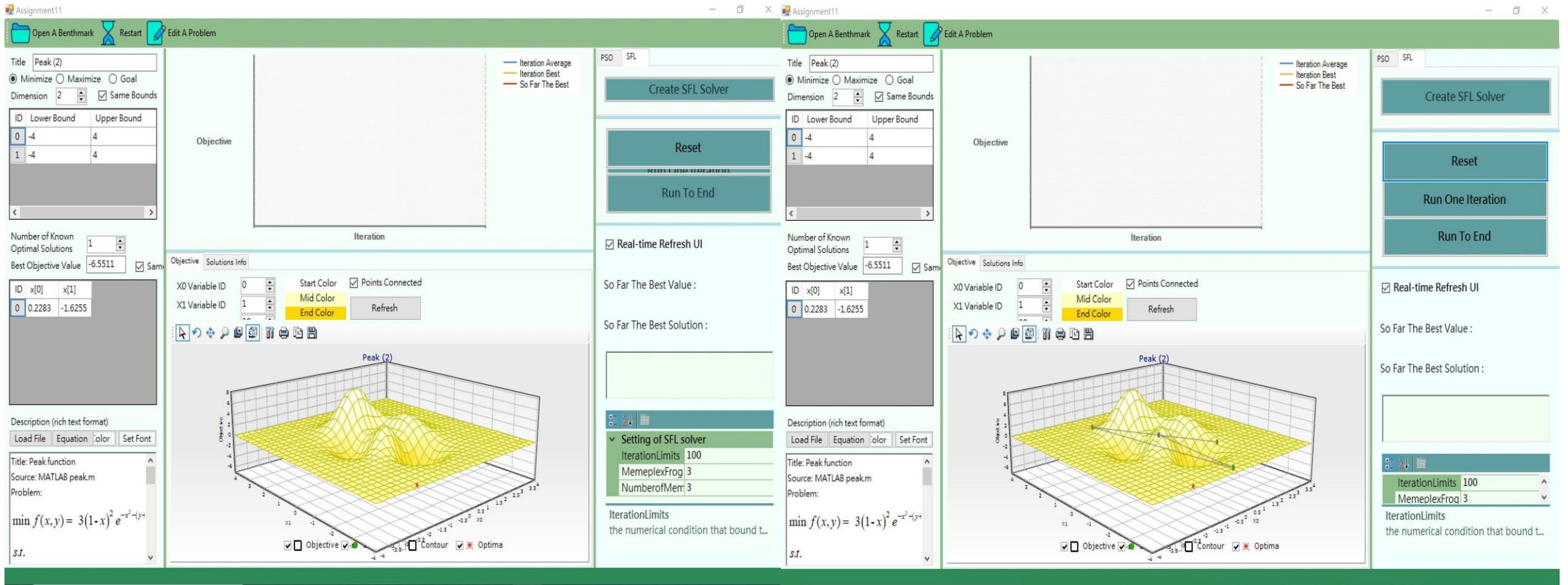
- (1) 找出全域最佳解、每個族群中的最佳解及每個族群中的最差解
- (2) 透過族群最佳解及族群最差解建構一個新的解
- (3) 檢查新的解是否優於原始的族群最差解，若是則跳至(7)，否則跳至(4)
- (4) 重新以全域最佳解及族群最差解建構一個新的解
- (5) 檢查新的解是否優於原始的族群最差解，若是則跳至(7)，否則跳至(6)
- (6) 隨機生成一組新的解
- (7) 以新的解取代原本的族群最差解



問題與結論

- 原本想針對SFL與PSO求解做比較，但由於撰寫的程式可能中間有發生錯誤，導致只能成功create solver以及reset，而無法進一步進行運算，以至於沒有產出預期的比較結果。

問題與結論



Reference

Eusuff, M., Lansey, K., & Pasha, F. (2006). Shuffled frog-leaping algorithm: A memetic meta-heuristic for discrete optimization. *Engineering Optimization*, 38(2), 129–154.

G. G. Samuel, C. C. A. Rajan, "A modified shuffled frog leaping algorithm for long-term generation maintenance scheduling", *Proc. 3rd Int. Conf. Soft Comput. Problem Solv.*, pp. 11-24, 2014.