Nelder-Mead (NM) Simplex Search Method and NM+PSO

Outlines

- 1. NM單體搜尋法簡介 (Nelder-Mead simplex Search Method)
- 2. 全域/區域 演算法的缺點 (Problems of local and global search methods)
- 3. 參數的選擇之參考文獻 (Parameter Selection)
- 4. 混合NM-PSO的方法(Hybrid NM-PSO)
- 5. 計算結果 (Results)
- 6. 結論 (Conclusions)

- The NM simplex search method, first proposed by Spendley, Hext, and Himsworth (1962) and later refined by Nelder and Mead (1965), a local search method particularly designed for unconstrained optimization problems without using gradient information.
- The operations of this method are to rescale the simplex at each iteration based on the landscape and local behavior of the function by using four basic procedures: **reflection**, **expansion**, **contraction**, and **shrinkage**. Through these procedures, the simplex can successfully improve itself and get closer to the optimum as desired if an initial point is properly chosen.

The Nelder-Mead Simplex Search Method

- NM單體搜尋法於1965年由Nelder和Mead所提出。
  (proposed by Nelder and Mead in 1965)
- 這是一種不需要用到梯度資訊(without using gradient)的無限制式(unconstrained)區域(local)搜尋最佳化法
- 雖然它求解的速度較傳統的直接搜尋法還快(efficient)
- 但是對起始點的選擇非常敏感(sensitive to initial points)
- 也無法保證可以獲得全域最佳解(no guarantee of global optimum)。
- It is a direct search method based on function comparison.
- often applied to nonlinear optimization problems for which derivatives may not be known.

The Nelder-Mead Simplex Search Method

- NM單體搜尋方法運用在化學、生物、神經學、統計、工程學、質量管理、醫療保健和融合技術等領域上。Many applications, for example, chemistry, biology, neuroscience, statistics, engineering, management, medicine and health care, technology fusion, etc.
- 有些文獻針對NM進行研究，例如巴頓和Ivey ，Nazareth和Tzeng 等人，針對NM單體搜尋方法進行各種各樣的修改。Various enhanced versions are available.

The Nelder-Mead Simplex Search Method

- NM單體搜尋法作法主要係在一*N*維空間中，由*N*+1個頂點(vertex)，形成一單體（simplex)，藉由四個程序之運作，利用迭代方式，**將最差(worst)的頂點替換掉**，透過這樣的迭代更新，以改變單體之形狀，使得最後能逼近至最佳解。The method uses the concept of a simplex, which is a special polytope of ***n* + 1 vertices** in *n* dimensions. The simplest approach is to replace the worst point with a point reflected through the centroid of the remaining ***n*** points.
- The operation of the Nelder-Mead search method is based on the local behavior characteristics of the objective function. It utilizes four procedures:反射(Reflection)、擴張(Expansion)、收縮(Contraction)、and 縮小(Shrinkage) to rescale the shape of the simplex。

The Nelder-Mead Simplex Search Method

執行方法簡單說明如下(以N=2為例):

The original NM simplex procedure is outlined below and the pivot (主要) steps of the NM algorithm are illustrated through a two-dimensional case (N = 2).

1. 初始化(**Initialization**)：

若目標函數有N個變數，則產生N+1組頂點，以形成一單體。將所有頂點代入目標函數中，得到目標函數值。

For the minimization of a function of N variables, create N + 1 vertex points to form an initial N-dimensional simplex. Evaluate the function value at each extreme point (or vertex) of the simplex.

The Nelder-Mead Simplex Search Method

2. Reflection（反射）：

- Determine $X_{high}, X_{\sec hi}, X_{low}$, so that their corresponding target values are $f_{high}, f_{\sec hi}, f_{low}$, respectively.
- Then, find the centroid X_{cent} of the simplex excluding X_{high}, and calculate the following equation to obtain a new vertex through reflection of the worst vertex X_{high}

$$X_{refl} = (1+\alpha)X_{cent} - \alpha X_{high}$$

where α is the reflection coefficient ($\alpha > 0$). Nelder and Mead recommend taking $\alpha = 1$.

- If $f_{low} \le f_{refl} \le f_{\sec hi}$, then replace X_{high} with X_{refl}.

The Nelder-Mead Simplex Search Method

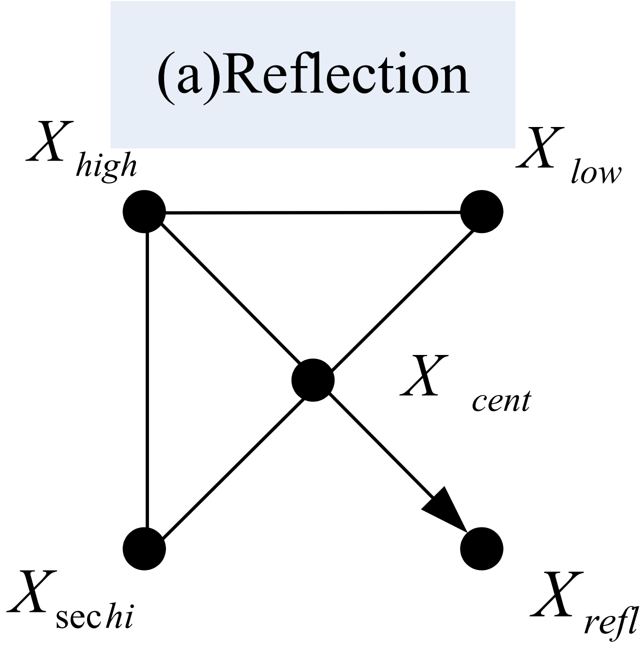
2. 反射(**Reflection**)：

決定 $X_{high}, X_{\sec hi}, X_{low}$，使其分別對應目標函數值 $f_{high}, f_{\sec hi}, f_{low}$，接著求出該單體(扣除 X_{high} 後)之重心 X_{cent}，根據下列方程式，經由最差的點反射經得到一新頂點:

$$X_{refl} = (1+\alpha)X_{cent} - \alpha X_{high}$$

其中，α 是反射係數(α >0)。 Nelder and Mead 建議取1。如果 $f_{low} \le f_{refl} \le f_{\sec hi}$，則以 X_{refl} 取代 X_{high} 。

The Nelder-Mead Simplex Search Method



$$X_{refl} = (1+\alpha)X_{cent} - \alpha X_{high}$$

- **2. Reflection.** In each iteration, determine X_{high}, $X_{\sec hi}$, X_{low} vertices, indicating the highest, the second highest, and the lowest function values that occur, respectively. Let f_{high}, $f_{\sec hi}$, f_{low} represent the corresponding observed function values.

- Find X_{cent}, the center of the simplex excluding X_{high} (in the minimization case). Generate a new vertex X_{refl} by reflecting the worst point according to the following equation:

$$X_{refl} = (1+\alpha)X_{cent} - \alpha X_{high}$$

- where α is the reflection coefficient ($\alpha > 0$).
  If $f_{low} \le f_{refl} \le f_{sechi}$, accept the reflection by replacing X_{high} with X_{refl}.
- Nelder and Mead recommend taking $\alpha = 1$.

- **3. Expansion.** Should reflection produce a function value smaller than f_{low} (*i.e.* $f_{refl} < f_{low}$),the reflection is expanded in order to extend the search space in the same direction for further function improvement and the expansion point is calculated by the following equation:

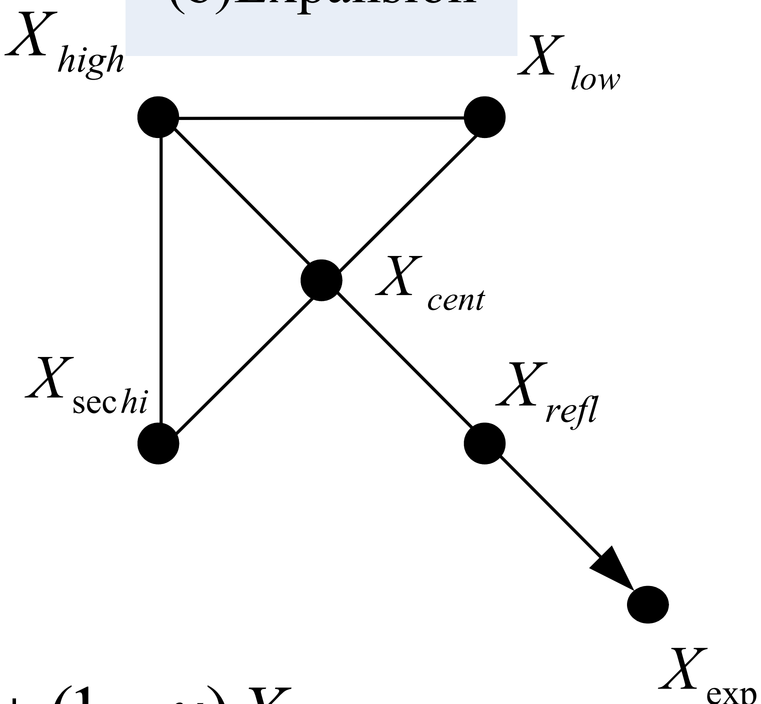
$$X_{\exp} = \gamma X_{refl} + (1-\gamma) X_{cent}$$

- where γ is the expansion coefficient (γ > 1). If $f_{\exp} < f_{refl}$, the expansion is accepted by replacing X_{high} with $X_{\exp}$; otherwise, X_{refl} replaces X_{high} .

- Nelder and Mead recommend taking $\gamma = 2$.

The Nelder-Mead Simplex Search Method

(b)Expansion



$$X_{\exp} = \gamma X_{refl} + (1-\gamma)X_{cent}$$

The Nelder-Mead Simplex Search Method

3. 擴張(Expansion)：

當 f_{refl} 比 f_{low} 還要好時，則做擴張動作，計算：

$$X_{\exp} = \gamma X_{refl} + (1-\gamma) X_{cent}$$

其中，γ 是擴張係數($\gamma > 1$)。Nelder and Mead 建議取2。如果 $f_{\exp} < f_{low}$，則以 $X_{\exp}$ 取代 X_{high}，否則以 X_{refl} 取代 X_{high}。

- **4. Contraction.** When $f_{refl} > f_{sechi}$ and $f_{refl} \le f_{high}$, then X_{refl} replaces X_{high} and contraction is tried (see (c) of Fig. 1). If $f_{refl} > f_{high}$, then direct contraction is performed. The contraction vertex is calculated by the following equation:

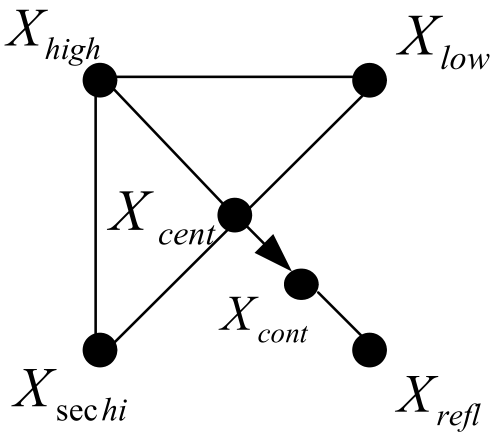
$$X_{cont} = \beta X_{high} + (1 - \beta) X_{cent}$$

- where β is the contraction coefficient $(0 < \beta < 1)$. If $f_{cont} \le f_{high}$, the contraction is accepted by replacing X_{high} with X_{cont} .

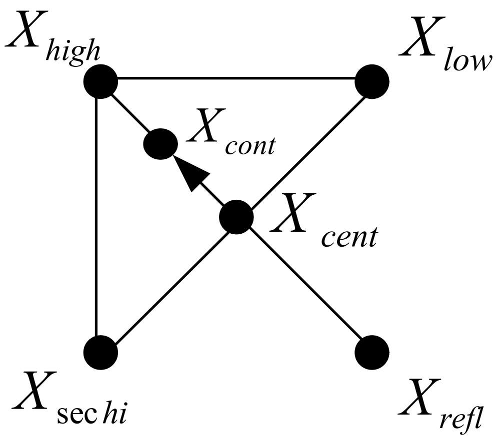
- Nelder and Mead recommend taking $\beta = 0.5$.

The Nelder-Mead Simplex Search Method

(c) Contraction when X_{refl} is better than X_{high}



(d) Contraction when X_{high} is better than X_{refl}



The Nelder-Mead Simplex Search Method

4. 收縮(Contraction)：

當 f_{refl} 介於 f_{high} 跟 $f_{\sec hi}$ 之間時，則 X_{refl} 取代 X_{high}，並且嘗試在 X_{cent} 跟 X_{high} 之間尋找 X_{cont}，此為向外收縮；當 f_{refl} 比 f_{high} 還要大時，則 X_{refl} 不取代 X_{high}，並且嘗試在 X_{cent} 跟 X_{high} 之間尋找 X_{cont}，此為向內收縮，計算：

$$X_{cont} = \beta X_{high} + (1-\beta) X_{cent}$$

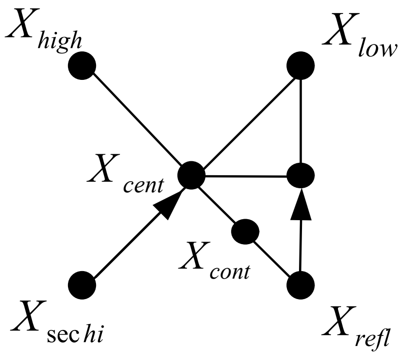
其中，β 是收縮係數($0<\beta<1$)。Nelder and Mead 建議取0.5。如果 $f_{cont} \leq f_{high}$，則以 X_{cont} 取代 X_{high}。

- **5. Shrink.** If $f_{cont} > f_{high}$ in Step 4, contraction has failed and shrinkage will be the next attempt. This is done by shrinking the entire simplex (except X_{low}) by:

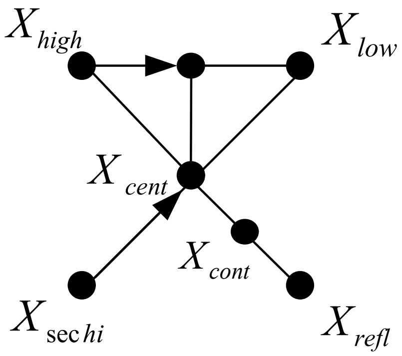
$$X_i \leftarrow \delta X_i + (1-\delta) X_{low}$$

  - Nelder and Mead recommend taking $\delta = 0.5$

The Nelder-Mead Simplex Search Method



(e) Shrink after failed contraction for the case where X_{refl} is better than X_{high}



(f) Shrink after failed contraction for the case where X_{high} is better than X_{refl}

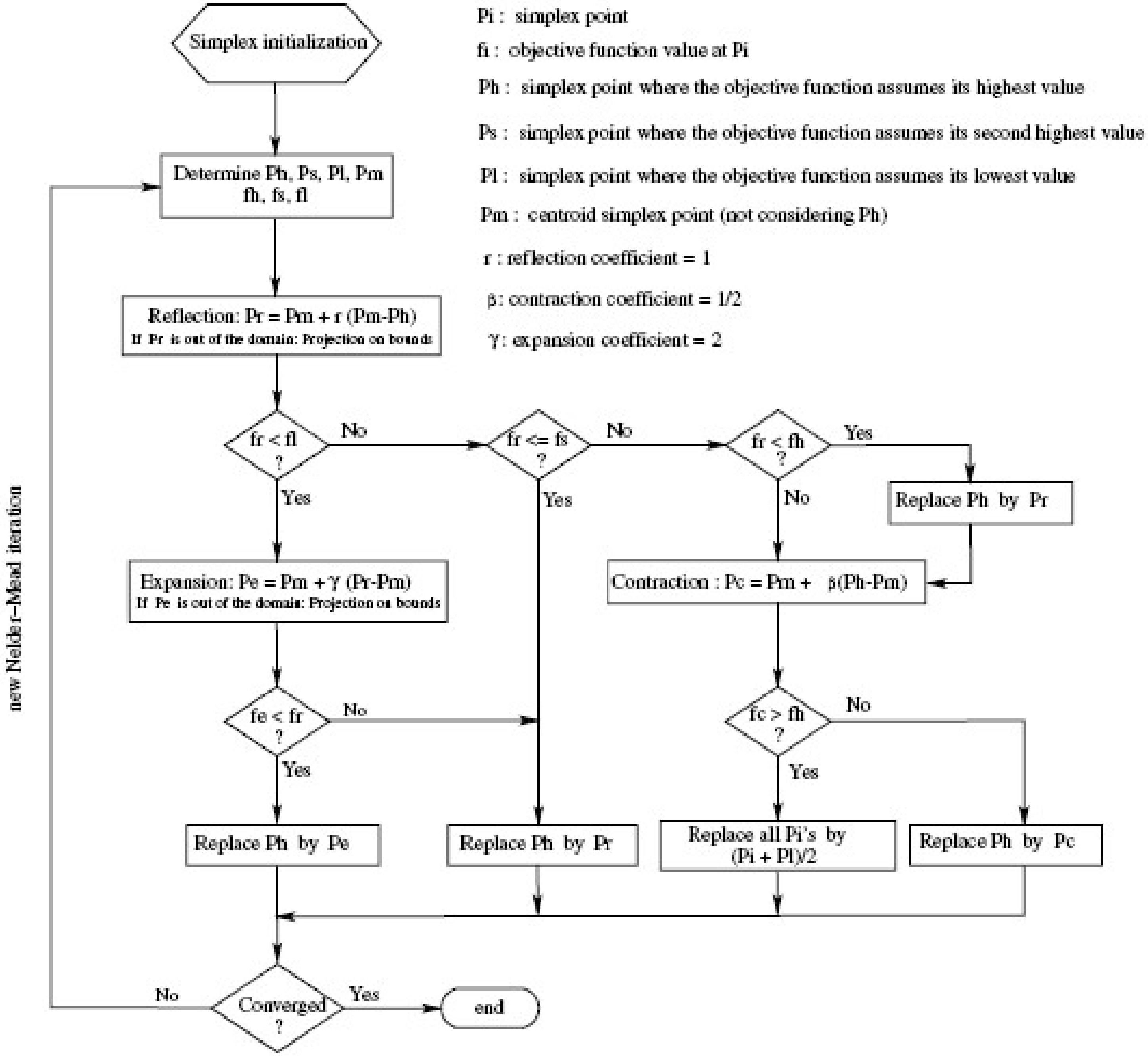
The Nelder-Mead Simplex Search Method

5. 縮小(Shrinkage)：

在步驟4時，當 $f_{cont} > f_{high}$ ，表示收縮失敗，這時候就得使用縮小的方式。計算：

$$X_i \leftarrow \delta X_i + (1-\delta) X_{low}$$

其中，δ是擴張係數(0<δ<1)。Nelder and Mead 建議取0.5，以縮小整個單體(shrink every points except X_{low})。

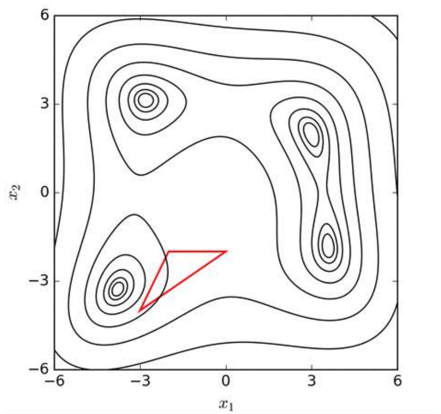

Pi : simplex point
fi : objective function value at Pi
Ph : simplex point where the objective function assumes its highest value
Ps : simplex point where the objective function assumes its second highest value
Pl : simplex point where the objective function assumes its lowest value
Pm : centroid simplex point (not considering Ph)
r : reflection coefficient = 1
β: contraction coefficient = 1/2
γ: expansion coefficient = 2
Simplex initialization
Determine Ph, Ps, Pl, Pm
fh, fs, fl
Reflection: Pr = Pm + r (Pm-Ph)
If Pr is out of the domain: Projection on bounds
fr < fl ?
No
Yes
fr <= fs ?
No
Yes
fr < fh ?
Yes
No
Replace Ph by Pr
Expansion: Pe = Pm + γ (Pr-Pm)
If Pe is out of the domain: Projection on bounds
Contraction : Pc = Pm + β(Ph-Pm)
fe < fr ?
No
Yes
fc > fh ?
No
Yes
Replace Ph by Pe
Replace Ph by Pr
Replace all Pi's by
(Pi + Pl)/2
Replace Ph by Pc
Converged ?
No
Yes
end
new Nelder–Mead iteration

$$X_{refl} = (1+\alpha)X_{cent} - \alpha X_{high}, \quad \alpha \leftarrow 1$$

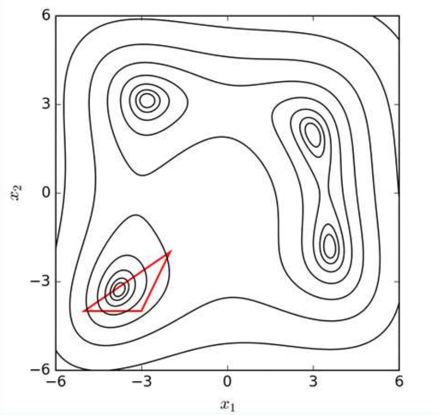
$$X_{\exp} = \gamma X_{refl} + (1-\gamma)X_{cent}, \quad \gamma \leftarrow 2$$

$$X_{cont} = \beta X_{high} + (1-\beta)X_{cent}, \quad \beta \leftarrow 0.5$$

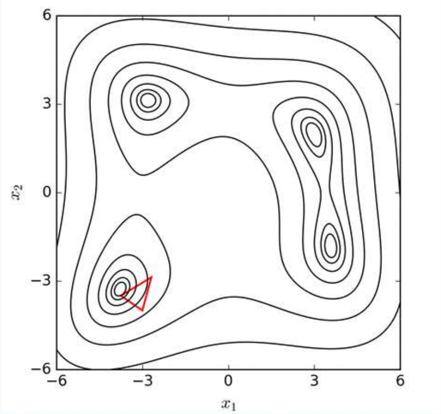
$$X_i \leftarrow \delta X_i + (1-\delta)X_{low}, \quad \delta \leftarrow 0.5$$



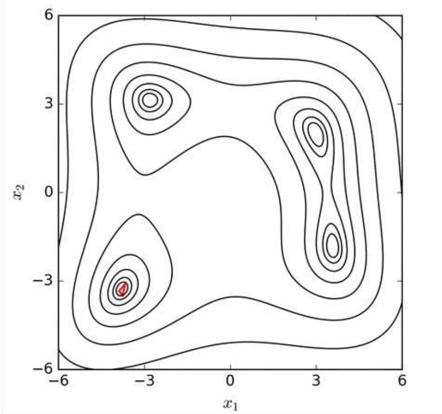
Nelder–Mead simplex search over the Rosenbrock banana function (above) and Himmelblau's function (below)



Nelder–Mead simplex search over the Rosenbrock banana function (above) and Himmelblau's function (below)



Nelder–Mead simplex search over the Rosenbrock banana function (above) and Himmelblau's function (below)



Nelder–Mead simplex search over the Rosenbrock banana function (above) and Himmelblau's function (below)

18. the Colville function:

$$100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 +$$
$$+10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1),$$

where $-10 \leq x_i \leq 10$.

The function has a global minimum value of 0 at $(x_1, x_2, x_3, x_4) = (1, 1, 1, 1)$.

全域/區域 搜尋法的缺點

Problems of local and global search methods

Limitations or problems of PSO：

- **Slow Convergence in Complex Problems**: Slow convergence at later stage, especially for complex problems.
- **Scalability:** PSO can struggle with high-dimensional optimization problems, as its performance deteriorates with an increase in the number of dimensions.
- **Sensitivity to Parameter Settings:** The performance of PSO is highly sensitive to its parameters, such as inertia weight, cognitive and social coefficients. Improper tuning of these parameters can lead to poor performance.
- **Poor Fine-Tuning**: While PSO performs well in the early stages of optimization, it often struggles with fine-tuning solutions in later stages, leading to suboptimal results.
- **Premature Convergence**: PSO can get trapped in local optima, especially in complex, multimodal optimization problems where the global optimum is hard to find.

Limitations or problems of NM Simplex Search：

- **Inefficiency in High-Dimensional Spaces**: tends to perform poorly when applied to high-dimensional optimization problems.
- **Convergence to Non-Optimal Points:** The algorithm may converge to a local minimum rather than the global minimum, especially for multimodal or nonconvex problems.
- **Dependence on Initial Simplex:** The performance of the algorithm is heavily dependent on the initial starting points (the initial simplex)
- **No Guarantee of Convergence**: does not guarantee convergence to a solution for non-smooth functions. It may continue indefinitely without reaching a minimum if the function's surface is irregular or discontinuous.
- **Limited for Non-Convex Problems**: While NM Simplex can be effective for convex problems, it struggles significantly with non-convex functions, where the algorithm can easily be trapped in local minima.

Parameter selection for NM

參數的選擇之參考文獻

[1] Shu-Kai S. Fan*, Erwie Zahara., "A hybrid simplex search and particle swarm optimization for unconstrained optimization," *European Journal of Operational Research*, Vol. 181, pp. 527-548, 2007.

各種演算法之參數的選擇 (Sensitivity of Control Parameters) 1/5

粒子群聚最佳化法（PSO）

$$v_i' = w \times v_i + c_1 \times rand \times (Pbest_i - x_i) + c_2 \times rand \times (Gbest - x_i)$$

Sensitivity analysis of inertia weight

Inertia weight w	Rate of successful minimization (%)				
	Powell	Beale	Hellical	Box	Wood
$0.5 + (rand(\ )/2.0)^{*}$	100	98	100	100	99
$rand(\ )$	100	100	91	100	100

各種演算法之參數的選擇 2/5

粒子群聚最佳化法

$$v_i' = w \times v_i + c_1 \times rand \times (Pbest_i - x_i) + c_2 \times rand \times (Gbest - x_i)$$

Sensitivity analysis of c_1 coefficient

c_1 coefficient	Rate of successful minimization (%)				
	Powell	Beale	Hellical	Box	Wood
0.20	99	96	96	100	99
0.40	99	96	95	100	100
0.60*	100	100	100	100	100
0.80	100	89	90	99	100
1.00	100	98	90	99	100
1.20	100	89	93	99	98
1.40	100	97	94	99	100
1.60	100	97	97	97	100
1.80	100	98	100	100	100
2.00	100	82	93	95	99

Sensitivity analysis of c_2 coefficient

c_2 coefficient	Rate of successful minimization (%)				
	Powell	Beale	Hellical	Box	Wood
0.20	100	97	96	100	100
0.40	100	94	89	100	99
0.60	100	90	85	100	100
0.80	100	97	71	100	100
1.00	100	91	92	99	100
1.20	100	99	100	100	99
1.40	100	90	100	100	99
1.60*	100	100	100	100	100
1.80	100	97	84	99	100
2.00	100	92	94	98	99

各種演算法之參數的選擇 3/5

Sensitivity analysis of mutation coefficient

Mutation coefficient (λ)	Rate of successful minimization (%)				
	Powell	Beale	Hellical	Box	Wood
0.25	100	93	93	100	99
0.40	99	100	88	100	99
0.55	100	100	98	99	99
0.70	99	96	99	100	100
0.85*	100	98	100	100	98

$$\mathbf{x}_i^{\text{new\_gbest}} = \mathbf{x}^{\text{old\_gbest}} + \boldsymbol{\varepsilon}, \quad i = 1, 2, 3, 4, 5, \tag{12}$$

where $\boldsymbol{\varepsilon}^{\mathrm{T}} = [\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_N]$, $\varepsilon_j \sim N(0, \sigma), j = 1, 2, \ldots, N,$

if mutation success rate = 2/5, then $\sigma^{new} = \sigma^{old}$.

if mutation success rate > 2/5, then $\sigma^{new} = (1/\lambda)\sigma^{old}$

if mutation success rate < 2/5, then $\sigma^{new} = \lambda\sigma^{old}$

Introductory example: mutation mechanism

- z values drawn from normal distribution $N(\mu,\sigma)$
  - mean μ is set to 0
  - Standard deviation (variation) σ is called mutation step size
- σ is varied across generations by the "1/5 success rule":
- This rule resets σ after every *k* iterations by
  - $\sigma = \sigma / c$, if $p_s > 1/5$ (wider search step, exploration)
  - $\sigma = \sigma \cdot c$, if $p_s < 1/5$ (search around the current solution, exploitation)
  - $\sigma = \sigma$, if $p_s = 1/5$
- where p_s is the % of successful mutations over a number of trials, $0.8 \leq c \leq 1$

NM單體搜尋法參數的選擇

$$X_{refl} = (1+\alpha)X_{cent} - \alpha X_{high}, \quad \alpha \leftarrow 1$$

$$X_{\exp} = \gamma X_{refl} + (1-\gamma)X_{cent}, \quad \gamma \leftarrow 2$$

$$X_{cont} = \beta X_{high} + (1-\beta)X_{cent}, \quad \beta \leftarrow 0.5$$

$$X_i \leftarrow \delta X_i + (1-\delta)X_{low}, \quad \delta \leftarrow 0.5$$

NM單體搜尋法

$\alpha \leftarrow 1$

$\gamma \leftarrow 2$

$\beta \leftarrow 0.5$

$\delta \leftarrow 0.5$

Table 2
Sensitivity analysis of reflection coefficient

Reflection coefficient (α)	Rate of successful minimization (%)				
	Powell	Beale	Hellical	Box	Wood
0.50	100	84	95	94	99
0.75	100	92	95	96	99
1.00	100	92	100	95	98
1.25	100	94	73	99	98
1.50*	100	100	100	100	100
1.75	98	90	98	100	98
2.00	97	82	95	98	99

Table 3
Sensitivity analysis of expansion coefficient

Expansion coefficient (γ)	Rate of successful minimization (%)				
	Powell	Beale	Hellical	Box	Wood
1.50	100	82	96	100	99
1.75	100	93	94	100	100
2.00	100	94	97	100	99
2.25	100	96	84	94	99
2.50	100	88	99	100	100
2.75*	100	100	100	100	99
3.00	100	96	98	99	99

各種演算法之參數的選擇 5/5

NM單體搜尋法

$$\alpha \leftarrow 1$$

$$\gamma \leftarrow 2$$

$$\beta \leftarrow 0.5$$

$$\delta \leftarrow 0.5$$

Table 5
Sensitivity analysis of contraction coefficient

Contraction coefficient (β)	Rate of successful minimization (%)				
	Powell	Beale	Hellical	Box	Wood
0.25	100	86	91	97	89
0.50	100	95	69	97	100
0.75*	100	100	99	100	99

Table 6
Sensitivity analysis of shrinking coefficient

Shrinking coefficient (δ)	Rate of successful minimization (%)				
	Powell	Beale	Hellical	Box	Wood
0.25	100	71	100	100	97
0.50*	100	100	99	100	100
0.75	99	97	99	98	99

混合NM-PSO的方法
Hybrid NM-PSO

Mememic algorithm (混合型演算法)
Hybrid NM-PSO的方法 1/12

- 雖然粒子群聚最佳化法已經被廣為應用在許多工程最佳化的問題上，其所達到的效果大致上也令人滿意，但是在實際的工程最佳化問題通常非常複雜，搜尋範圍也可能很廣泛，使得最佳化演算法之收斂率及精確性的降低，針對這些問題，因此必須改善原始粒子群聚最佳化法。
- 事實上，如何在鑽探搜尋及探索搜尋間取得平衡(balance between exploitation and exploration)，至今仍是一個重要且還未完全定論(not yet solved)的問題。

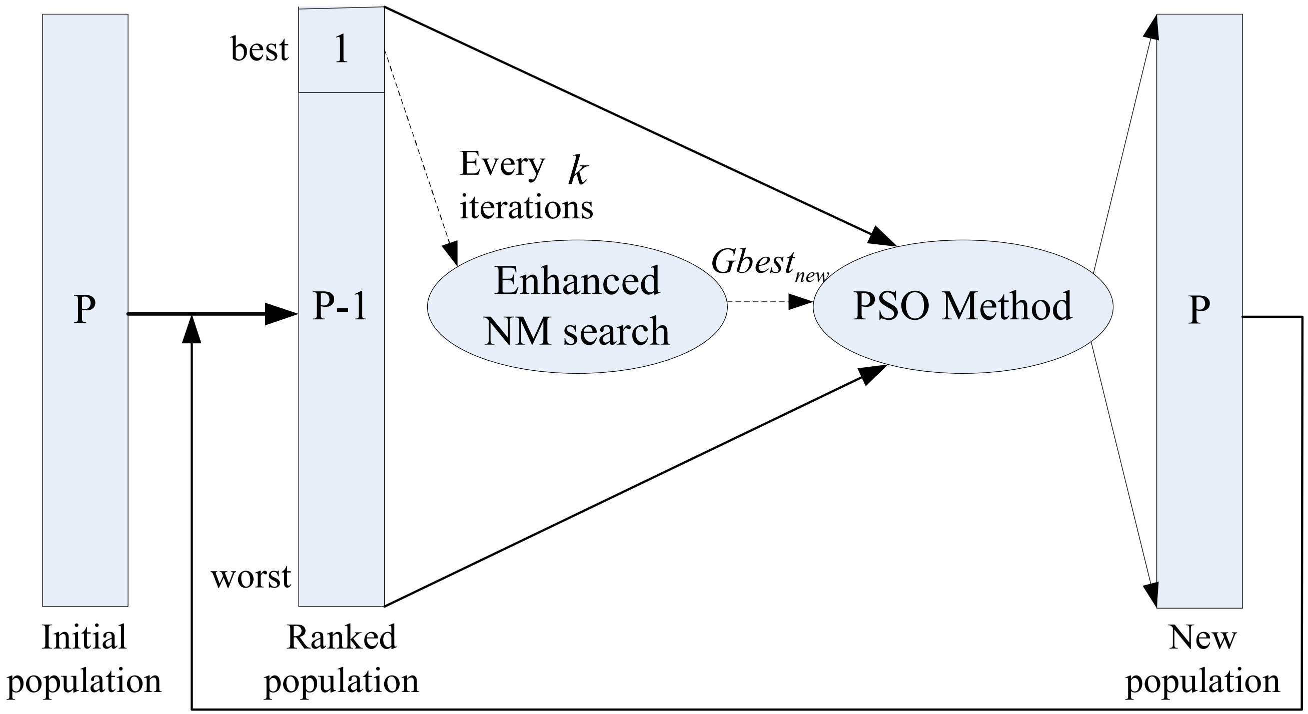
混合NM-PSO的方法 2/12

- 嵌入改良式**NM搜尋法**之粒子群聚最佳化演算法NM-PSO 其具體作法如下:

- Initial population:以隨機的方式產生P個粒子。
- Ranked population: 計算每個粒子的適應值，並且依照適應值的好壞排序。
- Exploitation search:每當*k*次迭代，取出族群中最好的粒子 *Gbest*，並在其附近產生N個點做改良式NM simplex 搜尋，經迭代*g*次後產生出新的粒子 $Gbest_{new}$ ，如果 $Gbest_{new}$ 比 *Gbest* 還要好的話，則 $Gbest_{new}$取代 *Gbest*，以提供給PSO進行演化。
- Exploration search:利用前一步驟所得之 *Gbest* ，將原族群做粒子群聚最佳化，並更新至新的族群中。

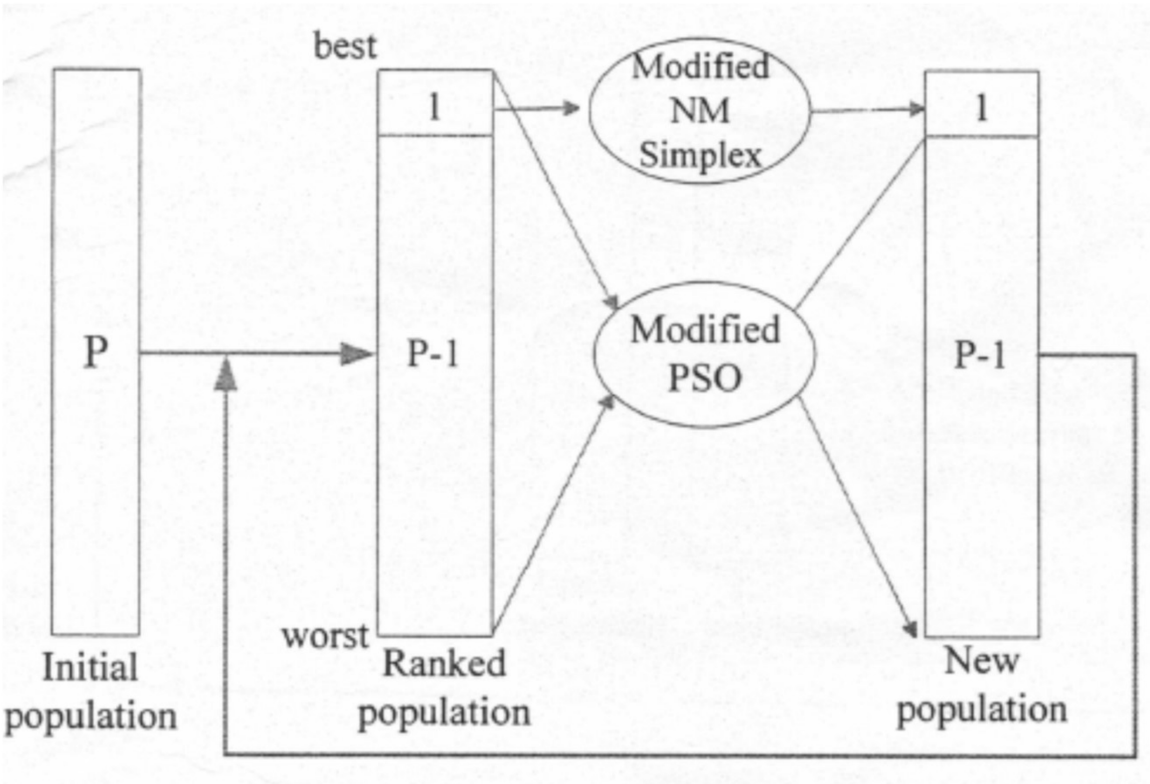
混合NM-PSO的方法 3/12

- 嵌入改良式NM搜尋法之粒子群聚最佳化演算法NM-PSO



混合NM-PSO的方法 4/12

- 其他文獻混合的架構(一)



混合NM-PSO的方法 5/12

■ 其他文獻混合的方法(一)

1.初始族群 (Initial population)：以隨機亂數的方式產生 P 個粒子。

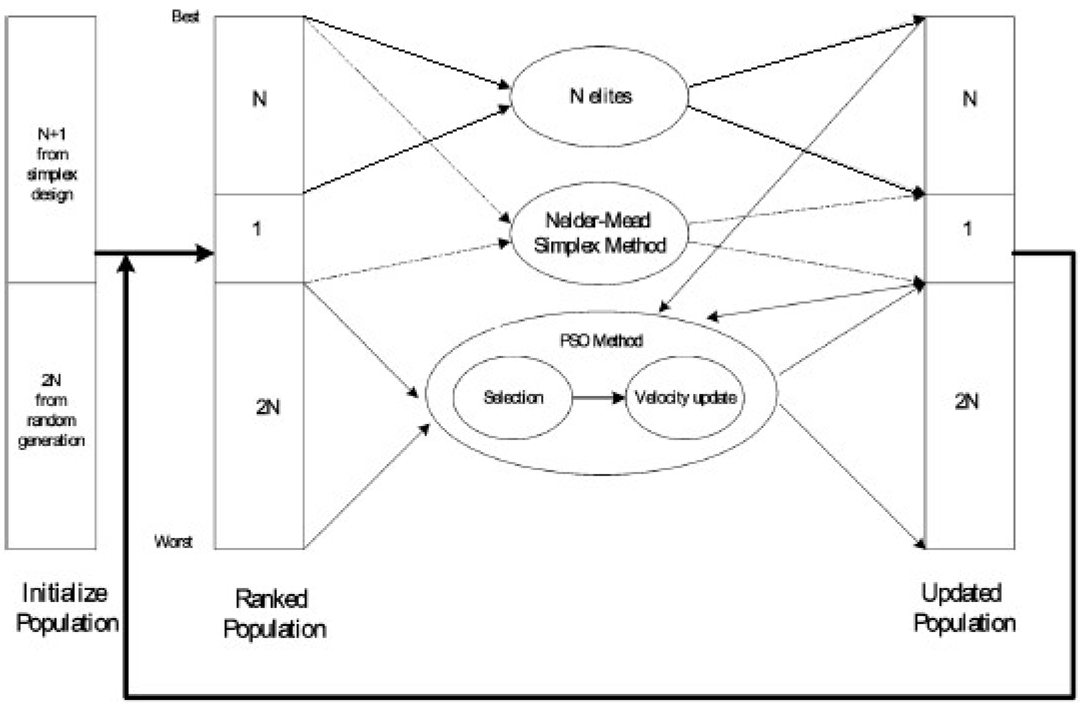
2.排序族群 (Ranked population)：依照適應值的好壞排序。

3.鑽探搜尋 (Exploitation search)：取最好的 1 個粒子，在其附近產生 N 個點做改良式 NM Simplex 搜尋，最後將最好的 1 點更新至新的族群 (population) 中。

4.探索搜尋 (Exploration search)：將原族群 (population) 做粒子群聚最佳化，做完後先扣除最差的粒子，才將 P-1 個粒子更新至新的族群 (population) 中。

混合NM-PSO的方法

- 其他文獻混合的架構(二)



混合NM-PSO的方法 7/12

- 其他文獻混合的方法(二)

1. **Initialization**. Generate a population of size $3N+1$.

Repeat

2. **Evaluation & Ranking**. Evaluate the objective function value of each particle. Rank them based on the objective function value.

3. **N Elites**. Save the top N elites.

4. **NM Simplex Search**. Apply a NM operator to the top $N+1$ particles and replace the $(N+1)^{th}$ particle with the update.

5. **PSO Method**. Apply PSO operator for updating $2N$ particles with worst objective function value.

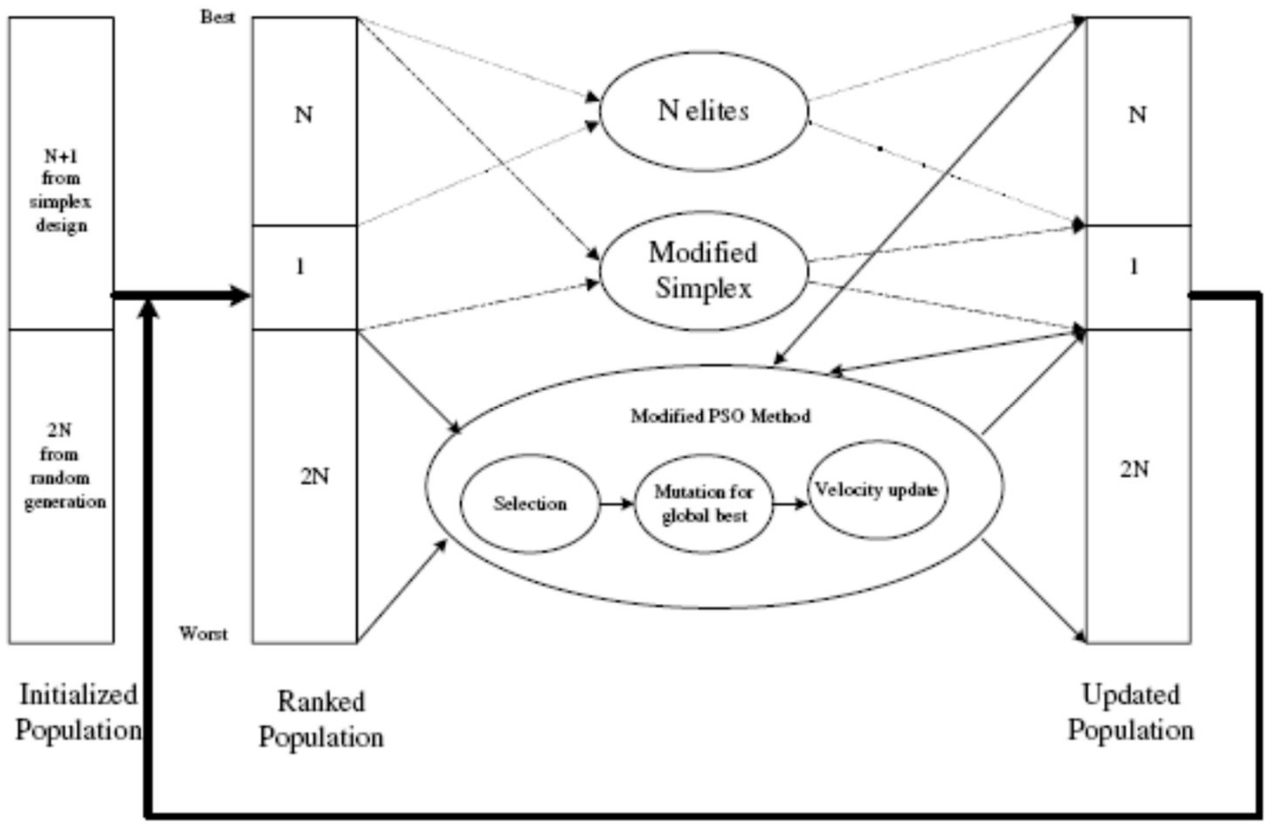
   Selection. From the population select the global best particle and the neighborhood best particles.

   Velocity Update. Apply velocity update to the 2N particles with worst objective function value according equations (5) and (6).

Until a termination criterion is met.

混合NM-PSO的方法 8/12

- 其他文獻混合的架構(三)



混合NM-PSO的方法 9/12

- 其他文獻混合的方法(三)

1. **Initialization**. Generate a population of size 3N+1.

Repeat

2. **Evaluation & Ranking**. Evaluate the fitness of each particle.

   Rank them based on the fitness results.

3. **N Elites**. Save the top N elites.

4. **Modified Simplex**. Apply a simplex operator to the top N+1 particles and

   replace the N+1th particle with the update.

5. **Modified PSO**. Apply modified PSO operator for updating 2N particles with

   worst fitness.

Until a termination criterion is reached.

Part I Selection. From the population select the global best particle and the clustering best particles.

Part II Mutation heuristic for the global best particle.

1. Define x^{old_gbest} as the global best particle, set the mutation success rate to 2/5 and the mutation coefficient (λ) to 0.85.
2. Generate 5 particles based on the position of the global best particle with variance σ according to the following equation

$$x_i^{new\_gbest} = x^{old\_gbest} + N(0,\sigma), \qquad i = 1, 2, 3, 4, 5.$$

3. For each mutation

{

if mutation success rate = 2/5, then $\sigma^{new} = \sigma^{old}$;

if mutation success rate > 2/5, then $\sigma^{new} = (1/\lambda)\sigma^{old}$;

if mutation success ra

混合NM-PS 10/12

- 其
- 他 修
- 文 改
- 獻 P
- 混 S
- 合 O
- 的 的
- 方 作
- 法 法
- 三

Part I: Selection. From the population select the global best particle and the neighborhood best particles.

Part II: Mutation heuristic for the global best particle. The algorithm is as below.

1. Define $x^{old-gbest}$ as the global best particle, set the mutation success rate to 2/5 and the mutation coefficient (λ) to 0.85.
2. Generate 5 particles based on the position of the global best particle with variance σ according to equation 11.
3. For each mutation

{

if mutation success rate = 2/5, then $\sigma^{new} = \sigma^{old}$.

if mutation success rate > 2/5, then $\sigma^{new} = (1/\lambda)\sigma^{old}$

if mutation success rate < 2/5, then $\sigma^{new} = \lambda\sigma^{old}$

}

4. Replace the old global best particle ($x^{old-gbest}$) with the new global best particle found among the $x_i^{new-gbest}$.

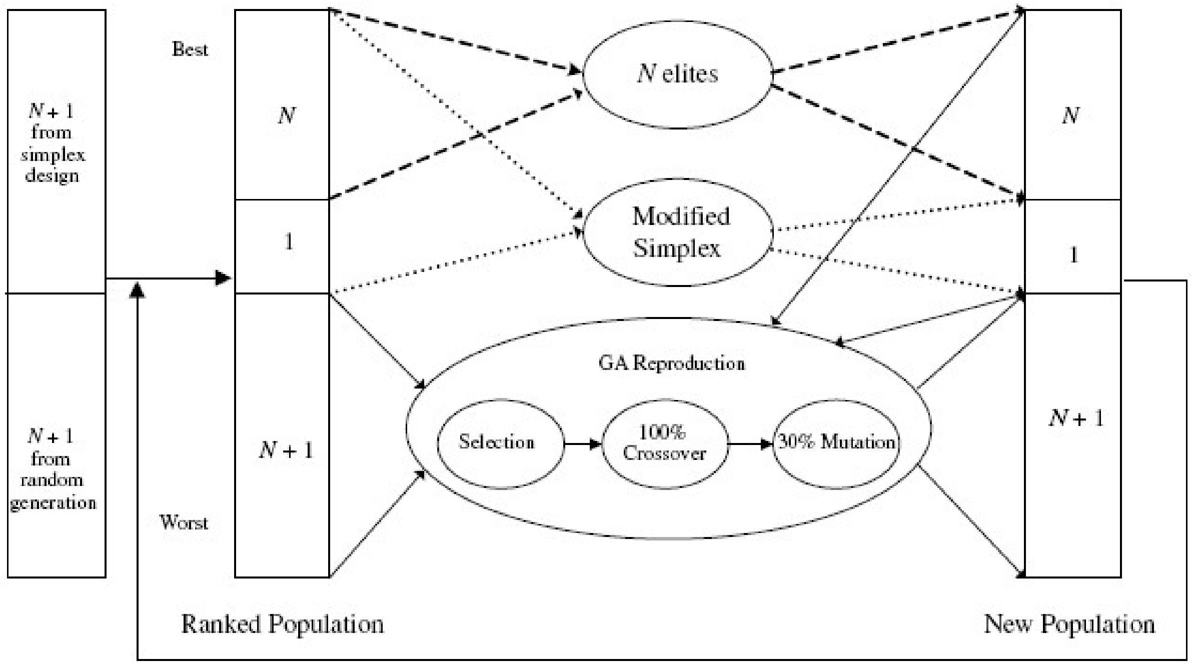
Part III: Velocity update. Apply velocity update to the 2N particles with worst fitness according to:

$$V_{id}^{New}(t+1) = w \times V_{id}^{old}(t) + c_1 \times rand() \times (p_{id}(t) - x_{id}^{old}(t)) + c_2 \times rand() \times (p_{gd}(t) - x_{id}^{old}(t))$$

$$x_{id}^{New}(t+1) = x_{id}^{old}(t) + V_{id}^{New}(t+1)$$

混合NM-PSO的方法 11/12

- 其他文獻混合的架構(四)



混合NM 12/12

- 其
- 他
- 文
- 獻
- 混
- 合
- 的
- 方
- 法
- 四

1. **Initialization**. Generate a population of size $2(N+1)$.

Repeat

2. **Evaluation & Ranking**. Evaluate the fitness of each chromosome. Rank them based on the fitness.
3. **N Elites**. Save the top N elites.
4. **Modified Simplex**. Apply a simplex operator to the top $N+1$ chromosomes and replace the $(N+1)^{th}$ chromosome with the update.
5. **GA Reproduction**. Apply GA operator for updating $N+1$ chromosomes with worst fitness.

   5.1 (**Selection**). From the population select the $N+1$ best chromosomes based on fitness.

   5.2 (**100% Crossover**). Using the $N+1$ best particles, apply two parents crossover to update the worst $N+1$ chromosomes by the following equation.

$$x_i' = x_i + 0.25x_{i+1} \qquad i = 1, 2, \ldots, N;$$

$$x_i' = x_i + 0.25x_1 \qquad i = N+1$$

   5.3 (**30% Mutation**). Apply mutation with the 30% mutation probability to the worst $N+1$ updated chromosomes according to the equations described below.

$$x_k' = x_k + rand \times N(0,1)$$

Until a termination condition is met.