

以調變式記憶規劃改善植基於斜率以及 ϵ -constraint

decomposition 之 MOEA/D

尹邦嚴

國立暨南國際大學

pyyin@ncnu.edu.tw

陳柏諭

國立暨南國際大學

f22418@gmail.com

摘要

MOEA/D 演算法利用目標空間拆解(decomposition)觀念在演化式多目標演算法(MOEA)的改良上，將多目標問題利用拆解的技術分解成數個單一子問題。MOEA/D 利用一組均勻間隔的權重向量，演化一群候選解，再將拆解後的子問題最佳化，以同時達到良好的收斂率(convergence)及分散率(diversity)。本研究提出兩個不同於以往的 MOEA/D 拆解策略的改進方法，slope update 演算法利用斜率概念來判斷更適合的更新候選解對象。projection update 演算法在目標空間中加入 ϵ -constraint 的概念，將其他目標維度以 ϵ 條件限制在最小的可接受的範圍，來求解某單一目標維度的最佳解。另外使用了路徑重劃(Path-relinking)來改進候選解彼此分佈不均勻的情況。實驗結果顯示，我們所提出的方法在 7 個標竿測試問題的表現上皆較傳統的 MOEA/D 有更佳的表現。

關鍵字：拆解、演化式多目標演算法、斜率、 ϵ -constraint、路徑重劃

壹、簡介

現實生活中常會遇到許多兩難的抉擇，有時候選擇其中一個就必須放棄另一個，沒有所謂魚與熊掌兼得如此的美事，因此在兩者之間如何去做抉擇這就是多目標最佳化的問題。而在實務上，我們只能找到一些特定性質的解集合，稱為 Pareto-optimal Set。這些解之間彼此存在著不被支配(Non-dominated)的關係，並在目標空間中會形成一個鋒面，稱之為 Pareto Front，而多目標最佳化的目的則是要找到能產生近似 Pareto Front 的解集合。而一般在評估多目標最佳化的方法時，主要針對兩個面向來衡量。一個為解的鋒面是否收斂(Convergence)接近至 Pareto Front；另一個則為解的分散度(Diversity)，追求 Front 上面的解是否夠均勻分佈，若越均勻分散則代表越具有鋒面的代表性。

而過去提出來解決多目標最佳化問題的著名方法有 MOPSO(Coello, Pulido, and Lechuga 2004)、NSGA-II(Srinivas and Deb 1994)、SPEA-II(Zitzler and Thiele. 1999)等。MOPSO 是以 PSO 為基礎加入外部存放空間(External Repository)、網格(Grid)以及突變的機制來求解多目標問題，外部存放空間存放目前所求得的所有 non-dominated 解，若新產生的解欲加入外部存放空間中，會利用 Grid 機制來尋找外部存放空間中解過於密集的網格，並將網格中的解取代，而突變則是用來彌補 PSO 集中收斂至特定區域而加入的擾亂機制。NSGA-II 則是利用 GA 加入了非支配排序(non-dominated sorting)以及擁擠距離(crowding distance)的概念來提昇解的收斂度及分散度。非支配排序是將親代與所產生的子代群體做合併排序來挑選下一代母體，並給不同支配能力程度的解有差異的適合度(fitness)，以達到良好收斂的品質；擁擠距離則是在當母體有過多候選解時作為取捨的評估方式，並確保 Front 上的解能夠被均勻的分散。SPEA-II 則是使用外部母體(External Population)存放目前最佳的 non-dominated 解，並利用 k-th nearest distance 及 fitness sharing 技術做有效的搜索。

MOEA/D 是由 Qingfu Zhang(2007)發表，利用拆解策略將多目標問題轉化為多個單一目標子問題(subproblem)，每個子問題在初始時即會賦予一組均勻產生的權重向量作為該子問題評估解品質的基礎。標準 MOEA/D 經由 Reproduction 以及 Improvement 所產生的子問題會利用拆解策略作為是否更新的依據，整個演化過程中各個子問題會按照所帶有的權重向量往特定的 Pareto Front 方向做趨近。此外 MOEA/D 加入了鄰居的概念，利用計算每個子問題所帶有之權重之間的歐幾里得距離，藉此定義出每個子問題候選解的鄰居。在演化的過程中，當子問題利用演化策略產生新的一個較佳解時，連帶會利用 decomposition 計算方式判斷其鄰居是否能一併更新。

本研究嘗試改進標準 MOEA/D 演算法，提出 MOEA/D slop update 以及 projection update 兩種不同的演算法變形，在收斂性與分散性同時兼顧之下尋找近似 Pareto Front。MOEA/D slop update 基於在目標空間中加入斜率的概念，以從座標原點到目標空間中各個候選解所形成的導引射線作為演化的引導方向，MOEA/D projection update 則是使用了 ϵ -constraint programming 作為演化更新的基礎。另外兩者皆使用路徑重劃(Path-relinking)來改進候選解彼此分佈不均勻的情況，提高了在目標空間中尋找到未被支配解的機會。

貳、文獻探討

本節探討多目標最佳化問題中的 MOEA/D 演算法及其變形，首先定義多目標最佳化問題的數學模式及相關概念，然後再比較本研究以及目前發表的各種 MOEA/D 演算法。

一、多目標最佳化問題

在現實生活中有許多問題同時要最佳化多個相互衝突的目標，此類問題我們稱之為多目標最佳化問題。而這類問題中多個目標之間各有所取捨，因此很難去找到一個解同時最佳化這些目標式。一般多目標最佳化問題的數學模型如下(Zitzler and Thiele 1999):

$$\text{Min } (f_1(x), f_2(x), \dots, f_n(x)) \quad (1)$$

$$\text{Subject to } x = (x_1, x_2, \dots, x_m) \in X \quad (2)$$

其中，有 n 個目標式 $f_1(x), f_2(x), \dots, f_n(x)$ ， m 個決策變數(decision Variable)構成決策向量 x (decision vector)， X 為參數空間(parameter space)。目的是要找到所有不被支配解(Non-dominated Solutions)，這些不被支配解稱之為 Pareto-optimal Set，而它們的目標式值會在目標空間中行成一個鋒面，稱之為 Pareto Front。不被支配解以數學式觀念描述如下[20]，在不失一般性的情況下，以一個最小化問題為例，假設目前有兩個解 $a, b \in X$ ，若滿足

$$\forall i \in \{1, 2, \dots, n\}: f_i(a) \leq f_i(b) \quad \wedge \exists j \in \{1, 2, \dots, n\}: f_j(a) < f_j(b) \quad (3)$$

則之稱為 a 支配 b ($a < b$: a dominates b)，因此， a 在此多目標最佳化問題上優於 b 。

多目標最佳化問題的目的是要找到一組最佳解，同時具備優良的收斂率(convergence)及分散率(diversity)，收斂率是指這些解的目標式值盡可能的接近真實的 Pareto Front；分散率則是指這些解的目標式值盡可能彼此均勻分散，以期有效表達 Pareto Front。

二、拆解策略

拆解策略(decomposition)是一個過去常被應用在多目標最佳化上，使一個多目標問題拆解成數個單一目標子問題的方法，如:Weighted Sum Approach 及 Tchebycheff Approach (Zhang Q. and Li H.)。針對一個多目標問題解給予一組權重向量 λ ，此權重向量代表每個目標的重要程度，將每個目標函數值依權重轉換後所得的值即為多目標以 λ 單一化後所得到的子問題結果，因此，使用多個均勻分佈的 λ 向量，即可將原多目標問題拆解成數個單一目標子問題，而分別求解，降低原問題的困難度。以 Weighted Sum Approach 為例，其拆解公式如下:

$$\text{minimize } g^{we}(x | \lambda) = \sum_{i=1}^m \mu_i f_i(x) \quad (4)$$

也就是將每個目標函數值乘上該目標對應的權重並加總所得的值做為以 $\lambda = (\mu_1, \dots, \mu_m)^T$ 拆解後所得到的子問題結果。若以 Tchebycheff Approach 作為 decomposition 的方法，需先選定一個參考點(reference point) $z^* = (z_1^*, \dots, z_m^*)^T$ ，而 z_i^* 為當前所有曾經看過在第

i 個目標的最大值。對於每個子問題會給予一組權重向量 λ ，利用不同的權重向量計算的結果來達到使多目標問題單一化的目的，其轉換公式如下：

$$\text{minimize } g^{te}(x|\lambda, z^*) = \max_{1 \leq i \leq m} \{\lambda | f_i(x) - z_i^* | \} \quad (5)$$

ε -constraint programming 方法則是 Chankong 與 Haimes 於 1983 年提出，主要概念在於多目標問題中將其他目標限制在一個最小的可接受範圍內，針對某單一目標而求解得到一組最佳值。而其數學定義如下：

$$\text{Min } f(x) = (f_1(x), f_2(x), \dots, f_n(x)) \quad (6)$$

$$\text{Subject to } x = (x_1, x_2, \dots, x_m) \in X \quad (7)$$

$$f_k(x) \geq \varepsilon_k \quad k = 1, 2, \dots, n \quad (8)$$

當其他目標被限制在一可行解範圍的同時，我們能對特定單一目標維度做最佳化的動作，達到 decomposition 的目的，同時尋找最佳 Pareto Front。

三、 MOEA/D

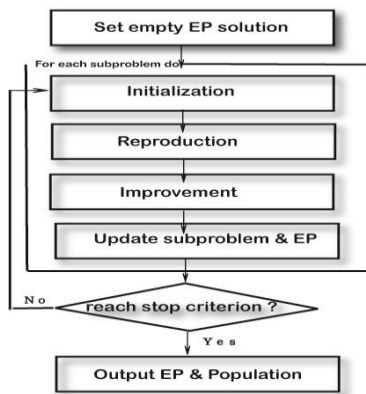


圖 1. MOEA/D 之演算法流程圖

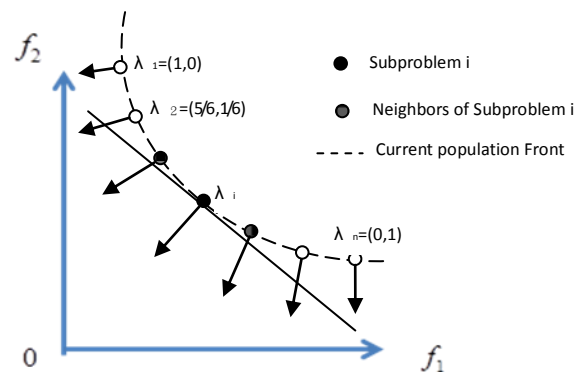


圖 2. MOEA/D 之主要概念圖

MOEA/D 演算法在近年來已有許多不同的改進版本，不論是在改變演化的方法或是加入不同的策略應用。傳統 MOEA/D 在流程上主要分成四個主要部分，如圖 1 所示，分別為 initialization、reproduction、improvement 以及 update。首先會產生與母體數量相等的權重向量，並在 initialization 時計算每組權重向量兩兩之間的歐幾里得距離，定義距離較近者則成為鄰居，再將權重向量均勻分配給隨機初始的母體。Reproduction 部分則利用演化策略產生出新的子代，並在 improvement 給予突變的機制。最後 update 部分判斷子代是否優於親代，若是則將子代取代親代，並將新產生的解存入外部存放母體 (External Population) 之中，判斷外部存放母體內的解彼此之間是否皆為 non-dominated 關係，若發現有被 dominated 的解則予以刪除。最後達到停止條件則輸出外部存放母體與最終母體。

MOEA/D 將 decomposition 的概念導入演化式多目標演算法之中，利用子問題所賦予的權重來計算 decomposition 值作為更新人口的基準。圖 2 為 MOEA/D 的主要概念圖，

在初始時即會給予均勻分佈的 $\lambda = (\mu_1, \dots, \mu_m)^T$ 權重向量，演化時會根據子問題各自的權重向量 λ_i 的法向量方向做改進，來達到擴展出均勻且收斂的 Solution Front 效果。

四、MOEA/D 變形

近年來有許多新的 MOEA/D 改進版本，它們大多著墨於 reproduction 策略的應用結合，以及 update 策略的改進。在最初 MOEA/D 中所使用的演化策略為採取二元交配以及多項式突變，之後發展出結合差分進化法(Li H. and Zhang Q 2009)、粒子群最佳化演算法(Noura et.al 2010)、模擬退火演算法(Li and Dario Landa-Silva 2010)等 reproduction 策略，目的在於演化中能夠期望找到更好的解。而在 update 策略的部分，MOEA/D 的概念中主要是利用拆解策略將多目標轉化成單一目標值，比較單一目標值優劣作為決定是否更新的基準。而文獻中主要的拆解方法有 Tchebycheff Approach、Normal boundary intersection 以及 Weighted sum approach。除了拆解策略外，有些文獻中還結合了其他策略，如 Decomposition-based Memetic Algorithm(Xin Yao et.al 2010)加入 NSGA-II 中的 Fast non-dominated sorting 以及 crowding distance 的概念，讓母體做批次的更新。另外還有 MOEA/D-EGO(Zhang 2010)結合高斯分配模型，進而定義每個子問題的改進期望值(Expected Improvement)作為是否更新的比較基礎。而本研究將加入新的斜率概念改善舊有的引導方式與資源利用率，以及提出加入 ε -constraint programming 拆解方法的新概念演算法，兩者並加入路徑重劃法(Path-relinking)進一步擴大搜尋範圍，增加尋找到不被支配解的機會。

表 1. MOEA/D 近年文獻之改進策略

Initialization	Reproduction	Improvement	Update
Random	1. Simulate Binary Crossover (SBX) 2. DE-Operator 3. Guided Mutation 4. MOEA/D + PSO 5. MOEA/D + SA	1. Polynomial mutation	1. Tchebycheff Approach 2. Normal boundary intersection 3. weight sum 4. Fast non-dominated sorting and crowding distance 5. Expected Improvement

五、調變式記憶規劃(Adaptive Memory Programming)

調變式記憶規劃是以記憶過去資訊為基礎的次經驗演算法，例如禁忌搜尋演算法、擴散式搜尋演算法等。禁忌搜尋演算法記憶過去曾經搜索過的路徑資訊，以避免重複搜索相同的方向來探索目標空間。擴散式搜尋演算法則是以母體為基礎，保留較佳解作為演化的參考集合，此外利用路徑重劃法(Path-relinking)增加解的多樣性。而在本研究針對所求目標在 Pareto Front 上多樣性不足的部分使用路徑重劃的策略。

Path-relinking 方法論(Glover and Laguna 2000)的精神在於從一些品質不錯的解到另一些品質不錯的解之間的路徑上，有相當大的機會存在著更佳的解。在建構一條 relinked path 的過程中，必須先選出一個初始解再以另一個解作為引導。在 relinked path 上，由初始解向引導解做鄰居移動，經過幾次的鄰居移動後，原本初始解內所含的屬性將會隨著鄰居移動而產生改變，轉變的過程中會越來越接近引導解的屬性，如此所建構

出來的 relinked path 上有可能獲得更多樣性(diversity)的解，在品質(convergence)上也有可能超越引導解而延伸出更好的搜尋軌道空間(trajjectory space)。以圖 3 為例，在初始解引導解之間做五等分的切割，每個等分我們以隨機線性組合的方式產生一組嘗試解，切割五等分即會有五組新的解。目的在於期望能夠在初始解與引導解之間找到更多的可能性。

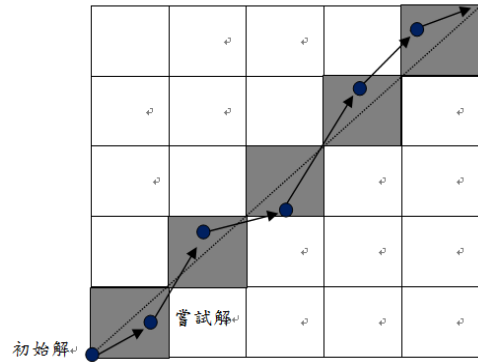


圖 3. Path-relinking 概念圖

參、研究方法

本節將敘述我們提出的幾個改進策略應用在 slop update 與 projection update 兩個演算法上，分別在 MOEA/D 的原始架構上所做的改良策略及貢獻。

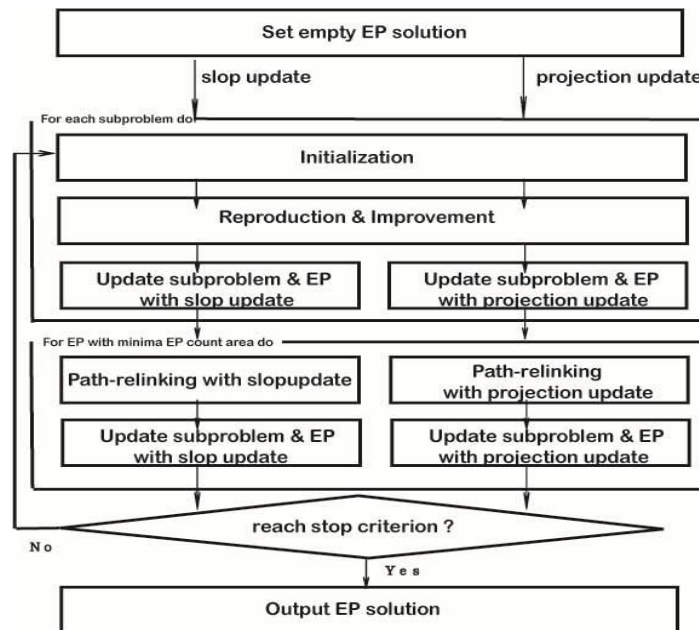


圖 4. MOEA/D slop update 與 projection update 之演算法流程圖

slop update 與 projection update 為本研究所提出的改良方法，如圖 4 所示，除了原本 MOEA/D 的架構外，我們加入路徑重劃的方法，在所屬區塊內 EP 最少的參考點與鄰近參考點之間做更多的搜尋，以改進 diversity 提高能夠找到不被支配解的機會。在輸出的部分我們以 EP 作為實驗結果的比較對象。

一、參考部分更新(Reference Section Update)

參考部分更新(Reference Section Update)主要包含有三種不同得更新概念，包含切線斜率更新、原點輻射斜率、以及投影區更新。利用這三種策略以達到演化更新以及判斷何者為目標候選解的目的。

(一) 切線斜率更新

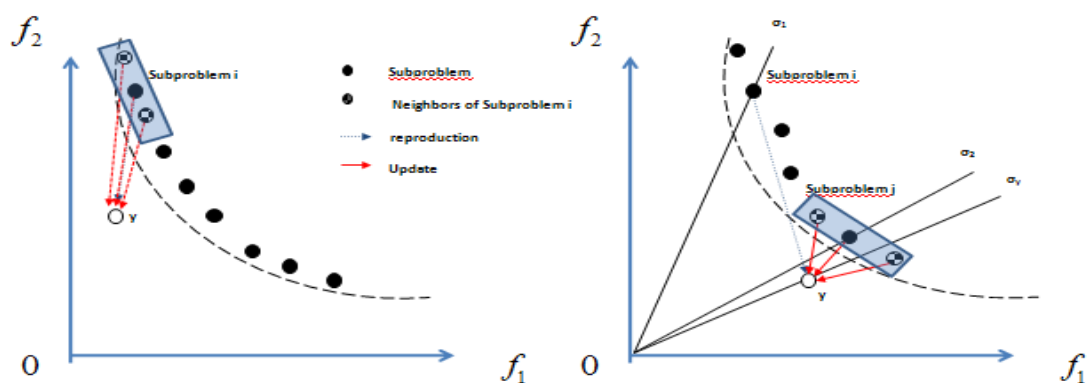


圖 5. (左)傳統 MOEA/D 與(右)MOEA/D slop update 之更新概念

過去 MOEA/D 在演化過程中產生新的解只能用於鄰居的改進，如圖 5(左)所示，subproblem i 產生子代 y 並判斷是否能夠更新親代鄰居，但在實際上新的解產生的方式是利用演化方法做變數的改變，如圖 5(右)中所示，可以看到新產生的子代 y 在目標空間中不一定適合改進當前的 subproblem i 與其鄰居，因此本研究提出利用斜率概念判斷此一新的解所適合更新的對象，改進斜率差值 θ 最小的 T 個子問題增加整體利用率減少資源浪費。 σ_i 為 subproblem i 與原點相連的斜率； σ_y 為子代 y 與原點相連的斜率，而 θ 計算公式定義如下：

$$\text{Min } \theta = |\sigma_i - \sigma_y| \quad (9)$$

(二) 原點輻線斜率

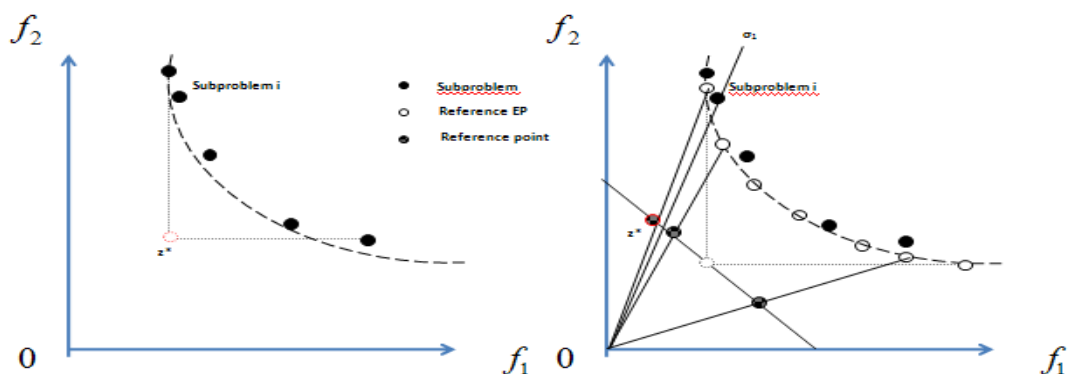


圖 6. (左)傳統 MOEA/D 單一參考點與(右)斜率射線區分多參考點概念

如圖 6(左)所示，傳統 MOEA/D 是以個別目標維度目前找到的最佳值作為單一參考點，為兼顧解的多樣性，如圖 6(右)我們利用均分的原點輻射斜率射線將目標空間做等

分，使 subproblem 不再是以單一參考點(reference point)作為引導的方向，而是選擇所屬的射線區塊中的 reference point 來引導，使 Pareto Set 能夠均勻的分布在整個 Front 上。

(三) 投影區更新

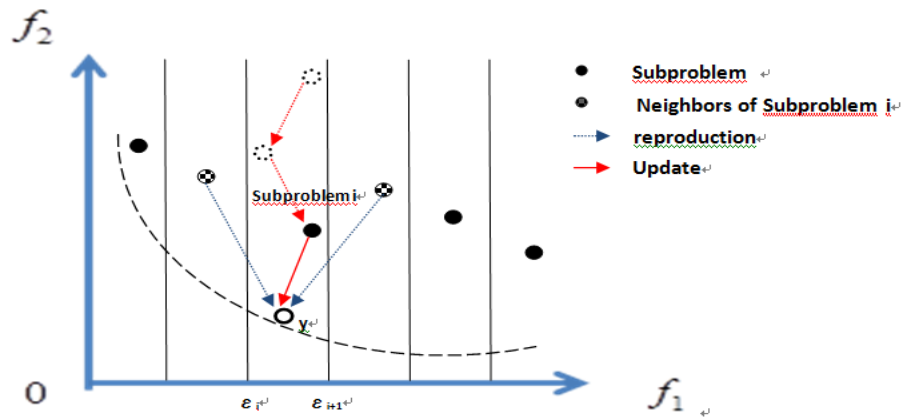


圖 7. Projection update 概念圖

在 projection update 演算法中，利用單一目標維度之法向量將解空間做等分切割，因此在目標空間中的解在當下皆有所屬的分割區塊，而這些區塊我們稱之為投影區。如圖 7 所示，當 subproblem_i 所屬的 ϵ -constraint programming 演化分割區塊中產生經由其鄰居 subproblem 演化出的新的子代 y ，在屬於相同的 f_1 目標維度投影區之中是否更新僅需要針對另一個目標維度 f_2 的目標值的優劣作為比較更新的依據。以圖 7 所示之最小化問題為例，若子代 y 的 f_2 值小於 subproblem_i 的 f_2 值時，代表子代 y 優於 subproblem_i，並予以更新。在 projection update 演算法中，產生的子代會判斷其所屬的投影區並且更新所屬相同投影區的 subproblem。

二、外部存放母體(External Population, EP)與參考解集合(reference set)

外部存放母體是用來存放演化過程中找到的不被支配解，在本研究是以 EP 所形成的 Front 作為最後結果的評斷。在每個 subproblem 演化的過程中皆會判斷是否有產生新的不被支配解來更新 EP 以及 EP 中是否有部分的解被新產生的解所支配，而應予以刪除及更新。

在 slop update 與 projection update 演算法中我們引進 Scatter Search(Glover and Laguna 2000)中有關參考解集合(reference set)的概念，所謂 reference set 是指存放高品質且富有多樣性解答的集合，這些參考解在演化過程中是以動態方式持續被更新。而傳統 MOEA/D 對所有目前人口的更新都是參考同一個 referent point: z^* ，由於 z^* 不見得是每個個別解在其子問題改進方向上的參考解，所以降低了個別解更新的機會，而減緩了演化的速度。

因此，為了改善此一缺點，本研究中 slop update 演算法利用在目標空間中均勻分割的射線將目標空間等分成 N 個區塊，通常 N 為 10%~20%母體大小。如圖 3 所示每個區塊計算與射線斜率最相近的 EP，將這些 EP 稱之為參考 EP(Reference EP)，而這些 Reference EP 的集合即為 reference set。為了拉大引導的差距，我們將 Reference EP 與原

點連線，以其相交於通過 Z^* 斜率為 -1 之直線的点作為此區塊的 referent point。而 reference set 也會隨著 EP 的變動做改變。

此外在 projection update 演算法中則是將限制的目標維度做動態的 M 等分切割，切割大小則會隨著所有 subproblem 在目前此目標維度的最大值與最小值有所更動， M 的大小則與母體大小相同。而我們將個別維度中目前尋找到的最佳解作為 Reference EP。而在 projection update 演算法中 Reference EP 主要是在路徑重劃的流程中扮演引導点的腳色。

在意義上 Reference EP 即最具代表性的最佳解，再者利用等分區塊的方式使其具有多樣性的概念。因 Reference EP 仍存有各維度的變數值而不是僅有多目標的目標值，除了應用在尋找 Pareto Front 的方向引導上，也能作為子問題路徑重劃的導引點。

三、 within-section slop update 與 projection update

Slop update 與 projection update 為基於 MOEA/D 作為改進的新構想。兩者在更新策略的部分皆包含判斷與尋找最適更新候選解的概念，即 slop update 中所屬的相似斜率區塊以及 projection update 的投影區，我們將此概念稱之為 within-section 的更新，從變數空間中演化產生的子代，在解空間中不一定能夠有效更新其親代，經過實驗證明利用相似斜率以及投影區尋找最適合更新的候選解，能夠減少大量資源的浪費。並且能夠有效的達到多目標改進收斂程度(convergence)的目的。

四、 inter-section slop update 與 projection update

在尋找最佳 Pareto front 的過程中，當目標問題不易求解時常常會造成鋒面的不均勻問題。針對此一問題，我們利用 Path-relinking 方法各自將 slop update 的斜率區塊以及 projection update 的投影區中的 reference EP 做為 PR 的引導點。為節省資源我們在所屬區塊內 EP 最少的參考點與鄰近參考點之間做更多的搜尋，以改進 diversity 提高能夠找到不被支配解的機會。

五、 虛擬碼

圖 8 為 slop update 結合 PR 之詳細流程架構。在初始步驟中，會先給予體一組均勻散佈的權重向量，根據這組權重向量計算彼此兩兩之間的歐幾里得距離，也就是說最接近的一組權重向量即是自己本身，而最接近的 T 個權重向量則定義為此候選解的鄰居。

在 step 2 會針對每個 subproblem 進行不同的演化策略產生新的子代，例如使用模擬二元交配法與多項式突變法、差分進化(Li and Zhang 2009)等等。本研究則是從鄰居中隨機選出兩個候選解進行線性交配產生子代 y ，另外使用多項式突變法進行子代的突變產生 y' 。而 y' 會利用斜率以及是否被支配的關係來更新參考點，而在判斷 y' 是否能夠取代舊有的親代則是利用 decomposition 的方法將此多目標集合計算為單一值，本研究採用 Tchebycheff Approach 作為 decomposition 的方法，若子代 y' 的 Tchebycheff 值 g^{te} 較小則代表相較於比較對象更接近近似 Pareto Front，應予以取代。

在 step 2.4 中，過去 MOEA/D 的策略是產生新的子代時，連帶會比較親代的鄰居，對於每個鄰居比較 g^{te} 值，若新的子代 g^{te} 值較小則也會將此鄰居取代。但由於產生子代的操作是在變數空間(variable space)而非在目標空間(objective space)，產生的子代可能離

親代目標值偏移很遠以至於無法有效的更新以及利用，或者可能會偏離 Front 的均勻導引方向，造成 Solution Set 在 Front 上面的不均勻。MOEA/D-slop update 則是利用斜率概念將目標空間利用均勻的射線區隔出不同的區塊，當產生 y' 後不再是用來更新該候選解與其鄰居，而是更新與斜率最接近的 R 個候選解。此一方法能更有效的運用演化出來的子代，而不會造成演化資源的浪費，並可同時引導候選解均勻的往 Pareto Front 移動。在 step 2.5 則將 y' 加入 EP 中，若 EP 中沒有支配 $F(y')$ 的解，將 $F(y')$ 加入 EP 並刪除 EP 中被 $F(y')$ 所支配的解。

在 step3 本研究利用均勻切割的各個區塊所屬的 reference point 對應之 reference EP 做為 PR 的引導點，但為了節省資源我們選擇區塊中 EP 數量最少者作為 PR 策略的改進對象。從 reference EP 到其最近的 reference EP 各個變數維度之間做 S 等分的切割，即 PR 所移動的步數。每個區塊再給予一個 0 到 1 的隨機的變化量，使其更能夠多樣性的變化。

在 Step 3.2 中，PR 所產生的 S 個 relinked solution 皆會根據與其斜率最接近的候選解計算 Tchebycheff 值判斷是否更新。Step 3.3 則判斷 relinked solution 是否能更新以及取代 EP。

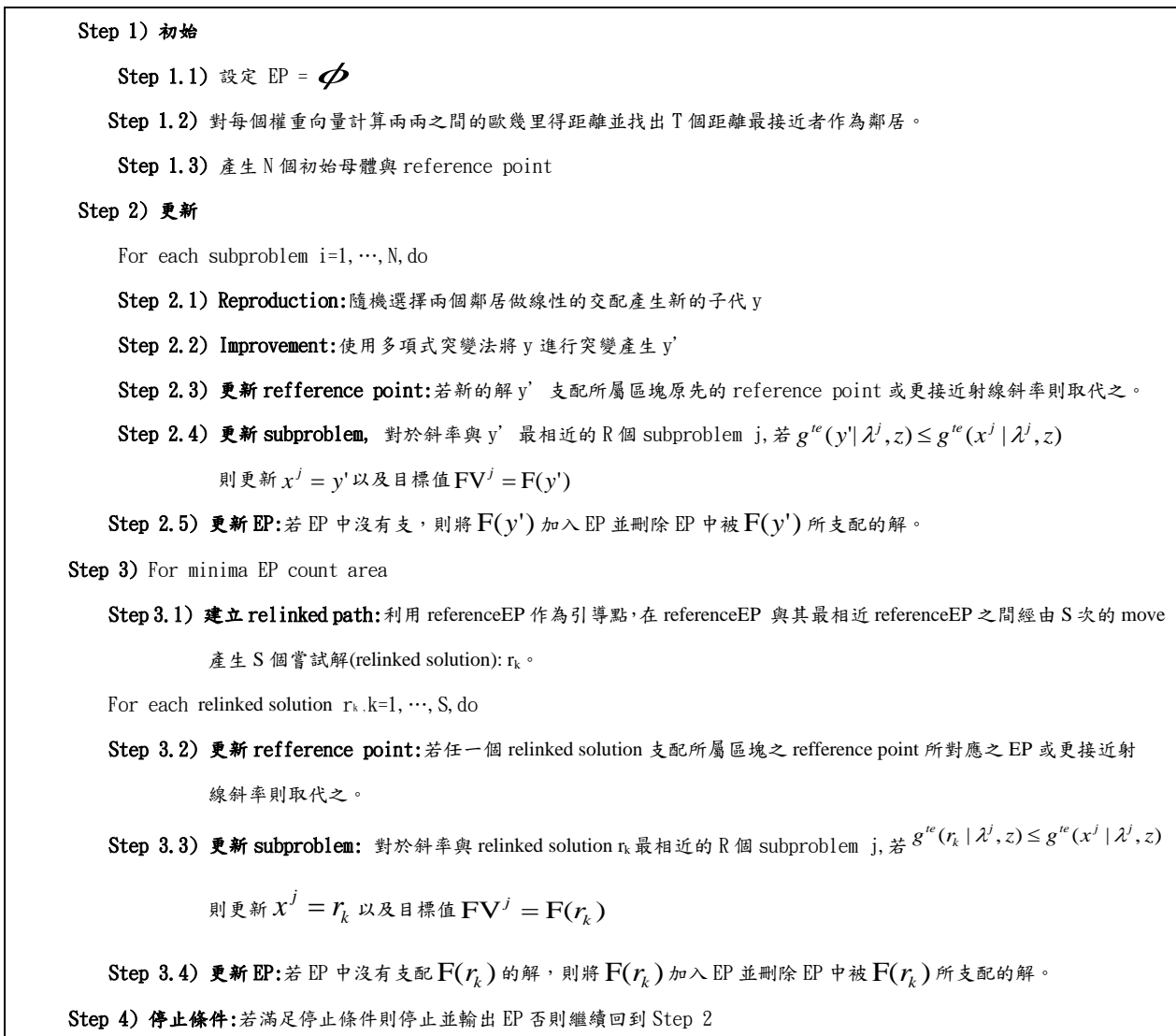


圖 8. slop update & PR 之流程架構

Projection update 則是結合 MOEA/D 與 ε -constraint programming 方法所創新的多目標最佳化演算法，對應於 slop update & PR 演算法之流程架構，projection update 結合 PR 在初始步驟中，除了預期要最佳化的單一維度之外，針對其他各個目標維度 $F_i(x)$ 做 N 等份的切割，切割大小以 ε 作為限制條件，而本實驗將各切割大小設定為動態切割，在目前目標維度內所有 subproblem 的目標值中找出其最大與最小值做均等分割。

在 step 2 產生新的子代的方法同於 slop update 演算法。而在判斷 y' 是否能夠取代舊有的親代則是利用 ε -constraint 概念判斷所屬的限制區塊以及在預期最佳化的目標維度中，目標值是否優於親代的方法來判定。

在 step 2.4 中，projection update 則是利用 ε -constraint 概念將其他目標空間區隔出不同的限制區塊，當產生 y' 後判斷所屬的限制區塊去更新同屬於此限制區塊內的候選解。此一方法能有效的運用演化出來的子代，而不會造成演化資源的浪費，並可同時引導候選解均勻的往 Pareto Front 移動。在 step 2.5 則將 y' 加入 EP 中，若 EP 中沒有支配 $F(y')$ 的解，將 $F(y')$ 加入 EP 並刪除 EP 中被 $F(y')$ 所支配的解。

在 step3 利用均勻切割的各個區塊所屬的 reference EP 作為 PR 的引導點，做法同 slop update 演算法，PR 所產生的 S 個 relinked solution 則皆會根據 ε -constraint 概念判斷屬於相同限制區塊候選解是否能更新。Step 3.3 則判斷 relinked solution 是否能更新以及取代 EP。

肆、實驗結果

為了評估本研究所提出之演算法效能，本研究使用七種測試問題並以文獻常利用的 Invert Generational Distance (IGD) 以及 Hypervolume 評估指標來比較原始 MOEA/D 做法與我們提出的 slop update & PR 和 projection update & PR 策略做效能分析。我們使用的實驗平台是 Intel core i5 個人電腦 2.4GHz CPU 1.92GB RAM, 以 C# 實作所有程式。

一、評估指標(Performance Metrics)

本研究使用 Invert Generational Distance (IGD) 與 Hypervolume 作為多目標問題的評估指標，IGD 公式定義如下：

$$IGD(P^*, P) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|} \quad (10)$$

其中 P^* 為真實的 Pareto Front, v 為 Pareto Front 上分佈的解，而 $d(v, P)$ 則為 Pareto Front 與演算法所求得的 Solution Front P 之間的最短距離，若 $|P^*|$ 數量越大則越能表現完整的 Pareto Front, IGD 在某種意義上能夠代表多樣性以及收斂性，IGD 值越小則代表 P 越接近 P^* 且均勻分佈。

另外 Hypervolume 指標則是計算演算法求得之 P 到某一參考點之間所覆蓋的體積，通常參考點會定義為比 P 所有成員更差的解，因此，若所求得體積越大，則代表 P 的品質越好。

二、相對效能分析(Comparative Performance)

本實驗以 ZDT1、ZDT2、ZDT3、ZDT4、ZDT6、LZ5、LZ8 七種標竿測試函數 (Benchmark Function) 來評估我們提出的 slop update & PR 和 projection update & PR 策略與原始 MOEA/D 之效能差異。ZDT 與 LZ 為多目標演化式演算法中廣為被使用的測試對象。在實驗參數的部分我們設定母體數 $N=100$ 、鄰居數 $T=20$ 、母體更新數 $R=5$ 、PR 步數 $S=4$ 、終止條件為 3,000,000 次的 Function Evaluation 並進行 30 次獨立測試 (Independent Run) 取其平均。

表 2. IGD 效能比較

<i>IGD</i>	<i>ZDT1</i>	<i>ZDT2</i>	<i>ZDT3</i>	<i>ZDT4</i>	<i>ZDT6</i>	<i>LZ5</i>	<i>LZ8</i>	<i>normalize</i>
MOEA/D	0.001803	0.003387	0.034543	0.19644	0.041237	0.005367	0.02320009	1
Slop update	0.00079	0.002045	0.03653	0.244599	0.002479	0.00538	0.0264342	0.7923096
Slop update & PR	0.0007	0.00154	0.037374	0.004298	0.001724	0.005069	0.04425176	0.691488
Projection update & PR	0.000675	0.000679	0.035317	0.099034	0.000272	0.004052	0.0720604	0.8527371

表 3. Hypervolume 效能比較比較

<i>Hypervolume</i>	<i>ZDT1</i>	<i>ZDT2</i>	<i>ZDT3</i>	<i>ZDT4</i>	<i>ZDT6</i>	<i>LZ5</i>	<i>LZ8</i>	<i>normalize</i>
MOEA/D	120.3536	119.8717	127.8636	125.6144	114.7496	117.1212	111.656401	1
Slop update	120.5619	120.1355	128.3431	126.3539	117.1204	117.1289	109.04612	1.0015594
Slop update & PR	120.5415	120.0782	128.4673	126.3625	117.2965	117.3005	111.588468	1.005297
Projection update & PR	120.6593	120.3021	128.7207	128.3356	117.6199	117.7931	108.18951	1.0048855

表 2 根據 IGD 指標來評估未加入任何策略之 MOEA/D、加入 slop 更新策略之 slop update、slop update 與 PR 交互使用以及 projection update & PR 之四種策略。我們可以看到 slop update & PR 為三者中表現最佳者，相較於未使用本研究方法策略之 MOEA/D 在 IGD 以及 Hypervolume 中 7 個測試函數各有 5 個趨於領先，normalize 則代表 7 個測試函數以 MOEA/D 為基準，將其指標值除以 MOEA/D 指標值並加總之後取平均所得到的整體比較標準，IGD normalize 值越小代表整體效能表現越佳，Hypervolume 則是越大代表 Front 品質越好。從表 2 可以看到 slop update & PR 在 IGD 指標相較於 MOEA/D 有著超過 30% 降幅的顯著的效能呈現。在表 3 中也能看到本實驗演算法在 Hypervolume 指標上整體表現也是相對較好的。而 projection update & PR 則是本研究所提出的一個創新的方法，在數據上雖然稍劣於 slop update & PR 所求得的解，但相較於傳統 MOEA/D 在效能以及求解品質上還是有一定程度的領先。

圖 9 是 slop update & PR 與 MOEA/D 在經過 3,000,000 次 function evaluation 後所得到的實驗結果，slop update & PR 為本研究求解效能最佳的演算法，從圖中可以看到淺色點 slop update & PR 在解的數量、品質以及均勻度均有較佳的表現。

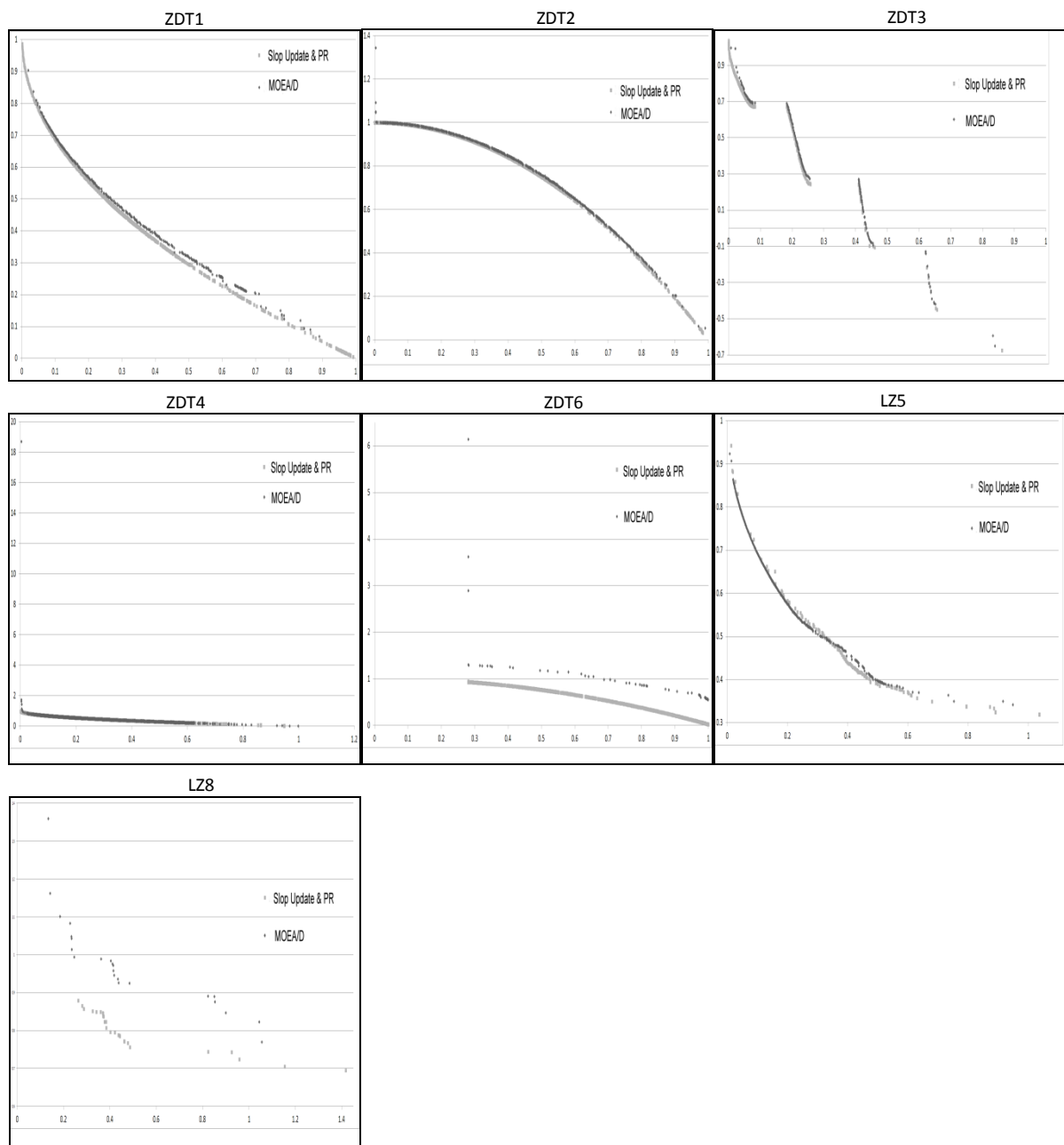


圖 9. slop update & PR 與 MOEA/D 之實驗結果比較

伍、結論

本研究提出兩個針對傳統 MOEA/D 做為改進基礎的演算法，slop update 演算法以 objective space 斜率做為改進策略，當母體經由演化策略產生新的子代，更新的對象不再侷限為其候選解及其鄰居，而是利用近似斜率的概念找到與子代斜率最相近之候選解作為評估更新的對象。projection update 演算法則是創新結合 MOEA/D 以及 ϵ -constraint 的方法，充分運用先前提出的相似概念進行演化，利用 ϵ 範圍限制子代所要更新的候選解對象。再者我們加入了 Path-relinking 的策略，選擇當下 EP 數量最少的區塊，利用屬於此區塊的 reference EP 作為引導點，提高在 relinked path 上尋找更多樣性以及高品質

的解的機會。並且在實驗結果中我們可以得到驗證，說明本實驗所提出的策略能確實有效改善演算法的整體效能。

陸、致謝

本研究部份成果由國科會專題研究計畫(NSC 98-2410-H-260-018-MY3)所贊助。

柒、參考文獻

1. Chen, C.-M., Chen, Y.-p., and Zhang, Q. "Enhancing MOEA/D with Guided Mutation and Priority Update for Multi-objective Optimization."
2. Chiang, T.-C., and Lai, Y.-P. "MOEA/D-AMS: Improving MOEA/D by an Adaptive Mating Selection Mechanism," in: IEEE, 2011.
3. Coello, Pulido, and Lechuga "Handling multiple objectives with particle swarm optimization," IEEE Transaction on Evolutionary Computation (vol 8) 2004, pp pp. 256-279.
4. Glover, F., and Laguna, M. "Fundamentals of scatter search and path relinking.," Contr. Cybernetic (29:3) 2000, pp pp. 653– 684.
5. Li, H., and Landa-Silva, D. "An Adaptive Evolutionary Multi-objective Approach Based on Simulated Annealing," Evolutionary Computation Journal) 2010.
6. Li, H., and Zhang Q "Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II," IEEE Transactions on Evolutionary Computation (vol13:no2) 2009, pp pp. 284-302.
7. Li, H., and Zhang, Q. "The Performance of a New Version of MOEA/D on CEC09 Unconstrained MOP Test Instances," in: IEEE, 2009.
8. Liu, H.-L., Gut, F.-q., and Cheun, Y.-m. "T-MOEA/D: MOEA/D with Objective Transform in Multi-objective Problems," in: IEEE, 2010.
9. Mei, Y., Tang, K., and Yao, X. "Decomposition-Based Memetic Algorithm for Multiobjective Capacitated Arc Routing Problem," in: IEEE, 2010.
10. Moubayed, N. A., Petrovski, A., and McCall, J. "A Novel Multi-Objective Particle Swarm Optimisation based on Decomposition," 2010.
11. Srinivas, N., and Deb, K. "Multiobjective optimization using non-dominated sorting in genetic algorithm," Evolutionary Computation Journal (vol. 28:No. 3) 1994, pp pp.221-248.
12. Zhang, Q., and Li, H. "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," IEEE Transactions on Evolutionary Computation (vol11:no6) 2007, pp pp. 712–731.
13. Zhang, Q., Li, H., Maringer, D., and Tsang, E. "MOEA/D with NBI-style Tchebycheff approach for Portfolio Management," in: IEEE, 2010.
14. Zhang, Q., Liu, W., Tsang, E., and Virginas, B. "Expensive Multiobjective Optimization by MOEA/D with Gaussian Process Model," IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION (vol14:no3) 2010.
15. Zitzler, E., and Thiele, L. "Multiobjective evolutionary algorithms: a comparative case study and strengthen Pareto approach, " IEEE Transaction on Evolutionary Computation (vol13:no4) 1999, pp pp. 257-271.

Adaptive Memory Programming for Improving MOEA/D based on slop and ϵ -constraint decomposition

Peng-Yeng Yin

Department of Information Management, Chi Nan University
pyyin@ncnu.edu.tw

Bo-Yu Chen

Department of Information Management, Chi Nan University
f22418@gmail.com

Abstract

MOEA/D is a new trend of multi-objective optimization problem research in recent years. Using the concept of decomposition in object space to improve the multi-objective optimization evolution algorithm (MOEA) and decomposes multi-objective optimization problem into a number of single sub-problems. MOEA/D develop a group of candidate solution by using uniform spread weight vector, and optimizes the sub-problems simultaneously to attain well convergence and diversity. In this paper we propose two improvable strategies based on the concept of slop and projection. Slop update algorithm use the guide lines which are formed by origin and reference points in object space to be guide direction of evolution. Projection update algorithm add the concept of ϵ -constraint programming which limit the other object dimensions in minimum range possible and explore optimal solution in single object dimension. Furthermore, we combined the Path-relinking algorithm in order to improve the uneven candidate solution with each other. In experiments and result, the method we proposed also perform better than the traditional MOEA/D in the seven benchmark problem instances.

Keywords : decomposition 、 multi-objective evolution algorithm 、 slop 、 ϵ -constraint 、 path relinking