

應用模糊邏輯於基因演算法之網路服務品質的組合

施舜凱¹與范姜永益²

¹ 天主教輔仁大學資訊工程系, 403226053@mail.fju.edu.tw

² 天主教輔仁大學資訊工程系, yyfanj2@gmail.com

Abstract

本篇論文使用了模糊理論的模型來改進基因演算法的適應函數 [7]，在使用者的需求處理上，我們分為兩種情況，功能性與非功能性，這裡我們主要是著重於非功能性的部分，我們引進了模糊理論來幫助我們進行非功能性需求的處理，包括 QoS 屬性與非明確形式的需求，透過使用模糊理論與基因演算法，將使用者需求的滿足程度作為演化的參數加入到演化的過程之中，尋找出對使用者需求的最佳結果與滿足程度。

1 緒論

近年來隨著網路的興起，網路上也出現許多的服務應用來滿足使用者的需求，但是隨著越來越複雜的要求，單一的服務已經無法滿足使用者，其中，基因演算法也被廣泛的應用來處理服務組合的問題，基因演算法的演化機制能讓程式根據不同的使用者需求進行挑選與進化，讓程式能自行尋找能夠符合使用者需求的解答，然而，不同的使用者在要求方面會有各種不同的變化，而這些需求如果使用傳統的二元邏輯方法是沒有辦法處理得，而模糊理論的

使用，在面對這些非明確形式類型的需求時，在處理上較二元邏輯來的容易且在結果方面也比較好。

接下來的章節中，在第三章會介紹 Service 性質、執行架構與合成方式，第四章是基因演算法的介紹與本篇論文所提出的方法，第五章是實驗的部分，最後第六章是結論。

2 相關研究

WSC 問題的核心是，如何在面對不同使用者要求的情況下，根據現有的服務找出能夠與使用者要求相對應與最好的服務組合，近年來，各式各樣的方法也相繼的被提出來解決 WSC 的問題，[1] 提到目前現實中的 Service 有許多都是無法被直接使用的，服務的提供者會給予這些服務特殊的限制來限制他們的使用情況，而這會對服務組合的結果產生巨大的影響，因此除了 Service 的輸入與輸出的參數比對外，[1] 也額外考慮了這些限制，[1] 使用了兩個 pre-processing 來處理擁有相同功能性屬性與不同限制情況的 Service，在服務組合的部分則使用了 Graph Search Algorithm 來找出能符合使用者需求的組合服務。

為了能更好的處理一些特別的使用者要求 (e.g.

Table 1: Related approach

author	approach	QoS or Semantic	Fuzzy model
CHAO, Kuo-Ming, et al.[2]	Fuzzy model	Both	True
CHENG, JiuJun, et al.[3]	Fuzzy Predicate Petri net (FPPN)	Both	True
LIN, Manshan, et al.[4]	deep-first branch and bound method	QoS	True
WANG, Ping, et al.[5]	Linguistic Entropy Method (LEM)	QoS	True
DI PENTA, et al.[6]	Genetic Algorithm and Fuzzy model	QoS	True

語義式的需求)，近年來模糊理論也大量的應用在 Service 組合的研究領域中，使用明確形式難以處理的部分，模糊模型都能夠進行有效的處置，[2] 提到，在表示一個 Service 時往往需要許多的資料，軟體的數位簽證、服務的位置、服務的語義以及服務的基礎數據，有些屬性能使用明確形式來表示，而有些屬性使用模糊的形式才能表達較為完整的意義，而需要模糊形式的原因也在於使用者在進行服務搜尋時，往往會使用帶有模糊的語句 (e.g. 尋找價格便宜的飯店)，[2] 則提出一個方法將使用者的搜尋情境與 Service 的屬性進行模糊化，當使用者進行服務搜尋時也能使用模糊的語句來加速整體服務搜尋的過程，進而找出符合自己需求的服務。[3] 則是將使用者需求、可用的 Service 與模糊的內容透過 Horn Clause 的形式來進行轉換，再依據轉換的 Horn Clause 結果繪製成相關的 Fuzzy Petri Net 來進行服務組合計算，最後的結果再經由 T-invariant、使用者的需求限制、QoS 以及使用者偏好來確認結果是否為使用需要且為最佳的組合服務。[4] 把重點放在非功能性屬性的處理上，大部分的網路服務在功能性的屬性上常會有所重複，而使用者往往也根據非功能性的屬性來做為最後組合服務的挑選基

準，[4] 將服務選擇的問題正規化為模糊限制滿足的問題 (fuzzy constraint satisfaction problem)，組合的方法則是透過 statechart 的方式來進行 deep-first branch and bound 的搜尋以找出最佳的組合服務。[5] 認為雖然 QoS 是評定一個 Service 重要的指標，在其中包含了功能與非功能性的 QoS 屬性，目前許多的方法都無法客觀地來使用這些屬性，造成在服務的選擇上可能無法找出適合的服務，[5] 利用模糊理論將服務的 QoS 功能與非功能性兩者屬性進行轉換，再透過 LEM (linguistic entropy method) 的計算為服務進行 QoS criteria 的評分以幫助使用者來找出最符合他們需求的服務。

如同 WSC 需要自動化與搜尋最佳解的問題的領域中，基因演算法近年來也越來越被廣泛的使用，基因演算法在使用需要有一些限制來避免演化的結果太過於發散而無法收斂到好的後代或是演化的後代無法滿足使用者，雖然基因演算法有許多的優點但是還是要依靠設計好的適應函數與參數設計來幫助進行基因 (Service) 篩選才能得到優良的後代 (CWS)，[6] 使用模糊邏輯針對不精確的 QoS 需求限制來計算 QoS 值並表示服務整體的滿足水準，在計算時，[6] 使用 QoS 需求限制與服務 QoS 屬性的模

糊化函數進行函數的重合取得組合服務 QoS 的函數面積，並使用積分的運算來算出最後的結果滿足程度，Table 1 列出了相關研究各方法所使用的演算法與方法模型，並記錄各方法在進行服務組合處理時所使用到的屬性。

3 服務組合屬性與架構

進行服務組合時，我們需要對服務的屬性和其組合的方式與架構進行定義，完成了服務屬性與架構的定義後，就能進行服務的組合。在服務屬性的部分我們使用了 7 個屬性來表示一個服務，分別為，ID, I, O, P, E, NF 與 ON，各個屬性的細節於 [7]，Section 3.1 有完整的定義，在組合服務運行時需要使用一些結構屬性才能完整的敘述組合服務的執行過程，[7]，Section 3.3 中有組合服務架構的詳細定義。CWS 本身是由多個 Service 所組成的，透過組合不同的服務來完成一個完整的組合服務，服務的組合方式表示於 Figure. 1 與 Table 2，組合方式的細節則於 [7]，Section 4.1 與 4.2 中說明。

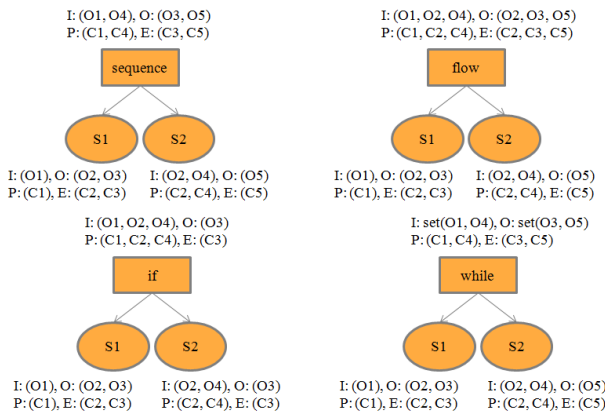


Figure 1: Examples to synthesis rules

Table 2: Aggregation functions for four activities

QoS Attr\Structures	Sequence	If	Flow	While
Response Time()	$\sum_{i=1}^m T(S_i)$	$\sum_{i=1}^n \frac{1}{n} * T(S_i)$	$\text{Max}\{T(S_i)_{i \in \{1..p\}}\}$	$\sum_{i=1}^k 2 * T(S_i)$
Cost()	$\sum_{i=1}^m C(S_i)$	$\sum_{i=1}^n \frac{1}{n} * C(S_i)$	$\sum_{i=1}^p C(S_i)$	$\sum_{i=1}^k 2 * C(S_i)$
Availability()	$\prod_{i=1}^m A(S_i)$	$\sum_{i=1}^n \frac{1}{n} * A(S_i)$	$\prod_{i=1}^p A(S_i)$	$\prod_{i=1}^k A(S_i)^2$

4 基因演算法與模糊模型

Figure. 2 是基因演算法運作的流程，基因演算法原先是用在生物的演化上，用來計算生物的演化對於自然環境生存適度，近年來常用於解決搜尋最佳解與優化搜尋問題上，在使用上，基因演算法會產生出多數個的個體，透過適應函數給予個體不同的適應度，做為演算法挑選的基準，優先選擇高適應度的個體重複進行交配、突變的運算來搜尋問題的最佳解。

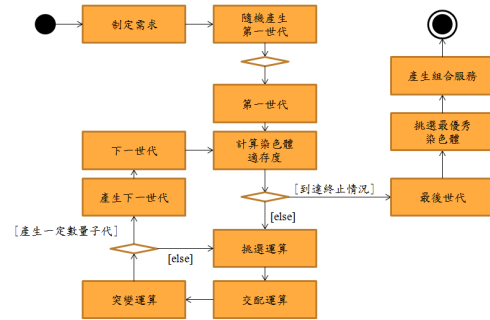


Figure 2: The process of genetic algorithm

4.1 演算法運算元

生物演化的過程中會使用三種運算，挑選、交配與突變，在挑選運算中，演算法會從當代的子代中隨機挑選出兩組染色體，這兩組染色體會根據適存度來決定是誰是贏家，贏家將會成為親代其中之一，Figure. 3 是挑選運算的全部流程，親代的挑選總共會經過二次，選出來的兩個親代會繼續進行後面的交配與突變運算。

在交配運算中，演算法會隨機在 level-2 選出一個結點，會讓選中的結點進行交換，即生物染色體的基因交換，Figure. 4 是交配運算元的例子。一般而言，為了確保基因演算法的正常運作，突變運算的機率不會太高，突變演算進行時，會從 level-2 中選出兩個結點來做運算，Figure. 6 則是四種突變運算情況的示意圖。

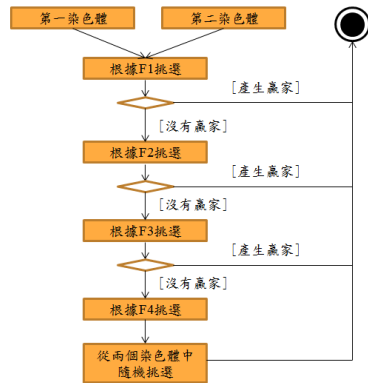


Figure 3: The selection mechanism

4.2 模糊模型

模糊理論的系統在處理的過程總共會分為三個部分執行，Fuzzification, Rule evaluation, Defuzzifi-

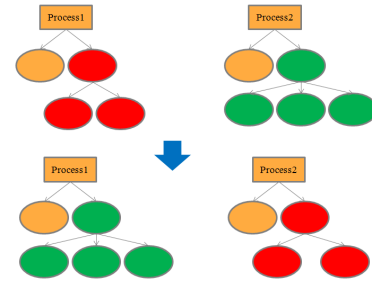


Figure 4: The crossover operators

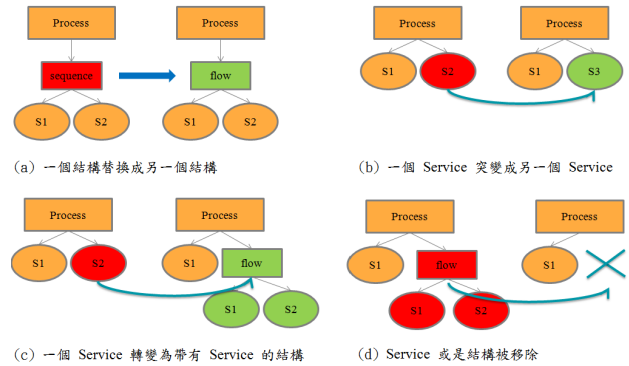


Figure 5: The mutation operators

cation, Fuzzification, 使用了 membership function 轉換明確集合與模糊集合用來描述目前系統碰到的狀況，Rule evaluation, 根據狀況來判定目前系統應該採取的動作，Defuzzification, 將模糊系統的輸出結果從模糊集合轉換成明確集合，本篇方法所使用的 membership function 與解模糊的計算方法，我們會在下一節進行介紹。

4.3 適應性函數

適存函數是基因演算法中用來評論一個個體是否能在環境中適合生存的重要指標，這邊所使用的適存函數則是用來評論一個 CWS 是否有合乎使用者

的要求，本篇所使用的適應函數 F1, F2, F3 定義與 [7] 相同，再來我們會介紹本篇所改進的適應函數 F4。

Definition. (Fitness Function 4(F4))

Fitness Function4 中我們會根據 Service 的 QoS 數值計算使用者對於結果 CWS 的整體滿足程度，使用在前面章節提到的非功能性的合成方法，來算出正確的 CWS QoS 數值，我們將使用模糊模型來進行 F4 的計算。

整個滿足程度的模糊理論的計算可分為三個部分，Fuzzification, Rule Evaluation, Defuzzification，在 Fuzzification 的部分，在我們的方法中所使用到的 QoS 屬性有三個，這三個屬性中 Time 與 Cost 的性質較為接近，因此我們決定使用兩個歸屬函數，分別給 Time, Cost 與 Availability 使用，兩個歸屬函數我們表示在 Figure. 6 中，在歸屬函數 (a) 中，我們分為三種程度，分別是 S(符合), N(有點不符合), NN(高度不符合)，在 Time 與 Cost 的計算之中，provided CWS 進行比較的對象為 required CWS，當 provided 的 QoS 數值與 required QoS 數值越相近時也表示 provided CWS 的表現越差，因此在歸屬函數 (a) 中 x 軸的正向表示負向的程度，在歸屬函數 (b) 中分為兩種程度 S(滿足) 與 P(高度滿足)，provided CWS 越接近 required CWS 代表 provided CWS 有達到使用者要求的值。

在規則的選擇上，我們將依據不同的歸屬函數規則來進行不同的滿意程度計算，根據歸屬函數的程度表示，總共會有 18 種規則，系統則根據這些規則來進行解模糊化的計算。由於計算滿意程度時缺少特定的值可做為使用者需求滿意程度的參考值，在解模糊化方式的選擇上也相對較少，最後我們選

擇的解模糊方法為權重平均法，解模糊化的公式表示在 Figure. 7 (a)，操作變量則依不同歸屬度給予不同的參考值，在實際的情況中，不同的使用者可能會對 QoS 屬性會有不同權重偏好，因此我們引用原 F4 中的權重概念將公式改為 Figure. 7 (b)，將最後計算出來的滿足程度作為 F4 的值來進行基因演算法的演化，以下是 F4 範例的計算。

Fitness Function 4 Example:

User Request:

Cost = 7, Response Time= 3.1, Availability = 0.89,

Composition Service Provide:

1. Cost = 11, Response Time= 7.1, Availability = 0.97,

2. Cost = 3, Response Time= 1.7, Availability = 0.73,

User Preference Weight:

1. 1.0, 0.7, 0.3 2. 1.0, 1.0, 1.0

Result 1:

Time membership: 1.0, 0

Cost membership: 0.214, 0.462

Availability membership: 0.727, 0.932

Satisfaction level:

$$1. \frac{(1.0*1.0*0)+(0.7*0.214*0.462)+(0.3*0.727*0.932)}{(1.0+0.214+0.727)} = 0.142 = 14.2\%(final\ satisfaction\ level)$$

$$2. \frac{(1.0*1.0*0)+(1.0*0.214*0.462)+(1.0*0.727*0.932)}{(1.0+0.214+0.727)} = 0.406 = 40.6\%(final\ satisfaction\ level)$$

Result 2:

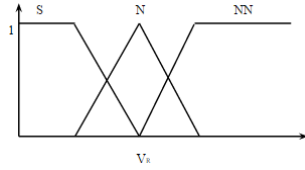
Time membership: 0.548, 0.813

Cost membership: 0.429, 0.843

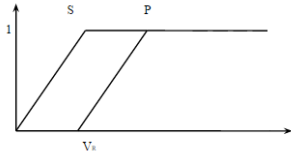
Availability membership: 0.82, 0.615

Satisfaction level:

$$\begin{aligned}
& 1. \frac{(0.3*0.548*0.813)+(0.7*0.429*0.843)+(1.0*0.82*0.615)}{(0.548+0.429+0.82)} \\
& = 0.496 = 49.6\%(final\ satisfaction\ level) \\
& 2. \frac{(1.0*0.548*0.813)+(1.0*0.429*0.843)+(1.0*0.82*0.615)}{(0.548+0.429+0.82)} \\
& = 0.723 = 72.3\%(final\ satisfaction\ level)
\end{aligned}$$



(a) function for Cost, ResponseTime



(b) function for Availability

Figure 6: non-functional membership function

$$\frac{\sum_i \mu(u_i) * u_i}{\sum_i \mu(u_i)} = \frac{p1*m1 + p2*m2 + p3*m3}{m1 + m2 + m3}$$

(a) Defuzzy Function

$$\frac{w1*(p1*m1) + w2*(p2*m2) + w3*(p3*m3)}{m1 + m2 + m3}$$

(b) Extended Defuzzy Function

Figure 7: Function of User Satisfaction Computation

5 實驗與結果

在 Table 3 中我們列出實驗中所使用的環境，而實驗中使用的功能性需求、QoS 需求、Service 屬

性以及基因演算法的設定我們表示在 Table 4 之中。

Hardware	Intel Core i5-4200 2.8GHz CPU 4GB RAM
Operating System	Windows 8.1
Programming Language	Java SE 8.0
JAVA APIs	JAXP DOM and OWL-API
Service Repository	110 Services

Table 3: Experiment Environment

Requirement1	
Input	O1, O3, O5
Output	O12
Precondition	C1
Effect	C2
Transaction Property	Atomic
QoS Request	(20,20,0.5)(40,40,0.01)(1, 1,1)
The Number of Chromosomes	2000
The Number of Generations	200
Crossover Probability	0.87
Mutation Probability	0.05

Table 4: Experiment Setting

本篇論文主要的方法是提出一個新模式來計算基因演算法中的 Fitness Value⁴，在原公式的方法中，只有針對 Repository 中的 Service QoS 數值來進行計算，在計算上，原本的公式使得計算 F4 較具彈性，但是過程中使用者需求並沒有參與演化過程，新公式在計算上結合了模糊理論的應用在使用者需求的符合與滿意程度方面的結果上更具參考價值，我們在 Figure. 8 中列出三個使用者需求與滿意程度結果的範例，三個範例從嚴格至寬鬆分為三種程度，分別為寬鬆 (b), 嚴格 (c), 中庸 (a), Figure. 8 圖形左邊 y 軸為原 F4，右邊 y 軸為新 F4，x 軸為計算的 QoS 屬性，根據三個範例圖形的結果我們可以觀察到，在計算值比較小的部分原 F4 與新 F4 函數

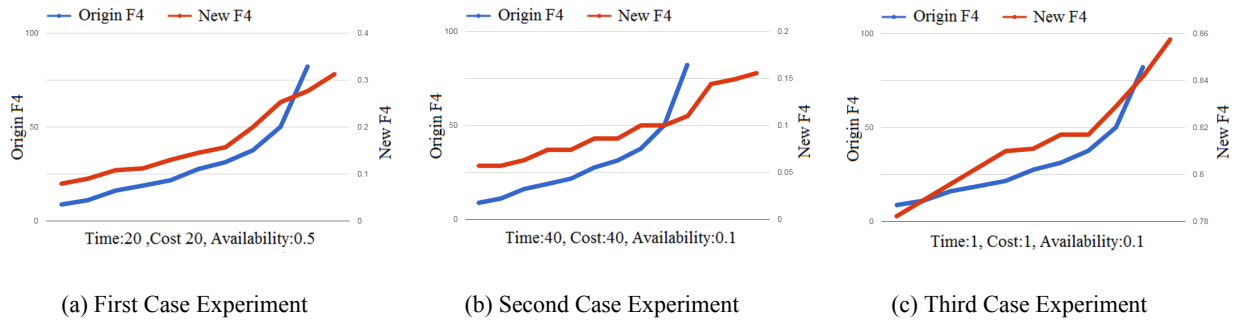


Figure 8: Three Case Experiment Result

的差距相當的微小，然而隨著計算值的增加，原 F4 與新 F4 函數末端也出現了顯著的變化，讓原 F4 函數走向呈現了指數函數成長，新 F4 的函數仍保持著一次方函數成長走向，且隨著計算值的繼續增長，函數的斜率變化的更加地激烈。

6 結論

本篇論文提出了一個新的基因演算法適應函數，我們在方法中使用了模糊理論，讓原本的模型在遇到非功能性的使用者需求時能以模糊的形式來進行處理，當系統接受使用者需求的輸入，先以歸屬函數將使用者進行需求轉換為模糊形式，再進行模糊規則判斷決定滿足程度的參考值，再進行解模糊化的動作，透過權重平均法得到使用者對演化結果的滿足程度，最後將結果作為適應函數加入至演化的過程中，讓最後的結果能夠更加地客觀。

References

- [1] WANG, PengWei, et al, "Constraint-aware approach to web service composition," *Systems, Man, and Cybernetics: Systems, IEEE Transactions*, 44.6: pp. 770-784, 2014.
- [2] CHAO, Kuo-Ming, et al, "Fuzzy matchmaking for web services," *Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference*, IEEE, pp. 721-726, 2005.
- [3] CHENG, Jiujun, et al, "Automatic composition of Semantic Web services based on fuzzy predicate Petri nets," *Automation Science and Engineering, IEEE Transactions*, 12.2: pp. 680-689, 2015.
- [4] LIN, Manshan, et al, "Solving QoS-driven web service dynamic composition as fuzzy constraint satisfaction," *e-Technology, e-Commerce and e-Service, 2005. EEE'05. Proceedings. The 2005 IEEE International Conference*, IEEE, pp. 9-14, 2005.

- [5] WANG, Ping, et al, "A fuzzy model for selection of QoS-aware web services," *e-Business Engineering*, 2006. *ICEBE'06. IEEE International Conference*, IEEE, pp. 585-593, 2006.
- [6] DI PENTA, Massimiliano, and TROIANO, Luigi, "Using fuzzy logic to relax constraints in GA-based service composition," *Late breaking paper presented at the Genetic and Computation Conference*, 2005.
- [7] FANJIANG, Yong-Yi, and SYU, Yang, "Semantic-based automatic service composition with functional and non-functional requirements in design time: A genetic algorithm approach," *A genetic algorithm approach. Information and Software Technology*, 56.3: 352-373, 2014.