

Procesy dynamicznego formowania rzek (River Formation Dynamics, RFD)

- Heurystyczna metoda optymalizacji podobna do algorytmów mrówkowych (ACO)
- RFD może być postrzegana jako gradientowa wersja algorytmu ACO
- RFD oparta na mechanizmach formowania rzek w naturze:
 - procesy erozji ziemi
 - nanoszenie osadów

Procesy RFD

- Metoda RFD zastosowana do różnego rodzaju problemów NP-trudnych:
 - szukanie minimalnej ścieżki w drzewie
 - szukanie minimalnego drzewa rozpinającego o zmiennym kosztowo grafie
 - problemy polegające na tworzeniu pewnego rodzaju powłoki drzew

Procesy RFD w naturze

- Rozważmy swobodny spadek mas wody z pewnej wysokości
- Grawitacja powoduje przemieszczanie się pewną ścieżką w dół do miejsca przeznaczenia
- Padający deszcz w górach powoduje, że woda stara się znaleźć własną drogę w dół do morza
- Przemieszczająca się woda powoduje erozję ziemi przekształcając ją w koryta rzek

Procesy RFD w naturze

- Silne nachylenia ścieżek przepływającej wody powodują usuwanie gleby z koryta rzeki
- Gleba jest nanoszona na ścieżkach o małym nachyleniu
- Rzeki wpływają na środowisko naturalne w zależności od różnicy wysokości:
 - redukując (np. erozję)
 - zwiększając (np. nanoszenie)

Procesy RFD w naturze

- Istnieje wiele punktów (źródeł) od których odbywa się przepływ mas wody
- Najkrótsza ścieżka jest tworzona z grupy krótszych ścieżek
- Konsekwencją jest tworzenie dopływów i meandrów rzeki
- Jeśli woda przepływa tylko z jednego punktu (brak innych źródeł wody), to trasa mas wody ma tendencję do znalezienia najkrótszej trasy (skuteczny sposób zmniejszenia wysokości)

Algorytm RFD

- Zamiast pozostawiania feromonu na gałęziach grafu, w algorytmie RFD węzłom grafu przypisana jest wysokość
- Przepływające sztuczne krople wody powodują:
 - erozję ziemi (zmniejszając wysokość węzłów)
 - nanoszenie osadów (zwiększając wysokość węzła)

Algorytm RFD

- Prawdopodobieństwo tego że kropla wybierze daną krawędź jest proporcjonalne do gradientu krawędzi (nachylenia stoku)
- Gradient krawędzi zależy od różnicy wysokości pomiędzy dwoma wybranymi węzłami i ich odległości (koszt krawędzi)
- Na początku algorytmu, wszystkie węzły mają przypisaną tą samą wysokość (środowisko jest płaskie)

Algorytm RFD

- Węzeł docelowy jest wyjątkiem i reprezentuje 'dziurę'
- Krople w swoich węzłach źródłowych są uwalniane, (rozsiane po całym grafie - środowisku) by w kolejnych krokach jedna z nich osiągnęła węzeł docelowy
- Erozja węzłów tworzy nowe zbocza, które propagują ten proces na sąsiednie węzły

Algorytm RFD

- Nowe krople są wstawiane w węzłach źródłowych w celu przekształcenia ścieżek oraz wzmocnienia erozji obiecujących dróg
- Po kilku krokach algorytmu można uzyskać obiecujące (krótsze) ścieżki z węzła źródłowego do docelowego
- Ścieżki te są dane w postaci sekwencji zmniejszających się gałęzi od źródła do miejsca docelowego

Algorytm RFD

- Zalety RFD w porównaniu do ACO:
 - lokalne cykle nie są tworzone i wzmacniane
 - mrówki zwykle biorą pod uwagę poprzednie ścieżki aby uniknąć powtarzające się trasy
 - wysokość węzła nie prowadzi kropel do takich sytuacji jak w ACO
 - po znalezieniu krótszej trasy przez RFD, kolejne wzmocnienie ścieżki jest szybsze

Algorytm RFD

- Zalety RFD w porównaniu do ACO:
 - krawędzie krótsze mają większe nachylenie zbocza przez co natychmiast są preferowane przez inne krople
 - zastosowanie mechanizmu erozji powoduje unikanie nieskutecznych rozwiązań
 - jeśli ścieżka prowadzi do węzła, który jest niższy od sąsiednich węzłów (ślepy zaułek) wówczas krople nanoszą osad co powoduje zwiększenie wysokości węzła

Algorytm RFD

- Zalety RFD w porównaniu do ACO:
 - wysokość węzłów z ‘ślepych zaułkami’ staje się równa sąsiadującym i inne nie będą tam wpływać
 - więcej kropli może być łączona w węzle, aż masa wody sięga sąsiednich węzłów (powstawanie jezior)
 - powstanie jeziora powoduje, że inne krople mogą przekroczyć ten węzeł i przejść do drugiego obok
 - ścieżka nie zostanie przerwana dopóki osad nie zasypie dziury (ukryty mechanizm unikania nieskutecznych ścieżek)

Schemat algorytmu RFD

Procedure RFD

begin

generuj krople

zainicjuj węzły

while (nie wszystkie krople przepływają tą samą ścieżką)

przemieść krople

przeprowadź erozję ścieżki

nanieś osad

analizuj ścieżki

end while

end

Opis algorytmu RFD

- ***Generowanie kropeł*** - wszystkie krople są umieszczane w początkowych węzłach
- ***Inicjalizacja węzłów*** – złożona z dwóch operacji:
 - wysokość węzła docelowego ustala się równą 0 (reprezentuje morze – ostateczny cel wszystkich kropeł)
 - wysokość pozostałych węzłów jest ustalana na pewną ustaloną wartość

Opis algorytmu RFD

- Pętla **while** w algorytmie jest wykonywana dopóki nie wszystkie krople przebędą tą samą sekwencję węzłów lub też inny alternatywny warunek zostanie spełniony
- Warunek taki stosowany do w celów ograniczenia liczby iteracji lub czasu realizacji
- Inna możliwość zakończenia pętli - jeśli najlepsze rozwiązanie znalezione do tej pory nie poprawiane jest w ciągu ostatnich n iteracji

Opis algorytmu RFD

- Zasada przejścia – prawdopodobieństwo, że k -ta kropla drop i -tym węźle przesunie się do węzła j

$$p_k(i, j) = \begin{cases} \frac{\nabla(i, j)}{\sum_{l \in V_k(i)} \nabla(i, l)}, & \text{if } j \in V_k(i) \\ 0, & \text{if } j \notin V_k(i) \end{cases}$$

gdzie

$V_k(i)$ – zbiór sąsiednich węzłów dla i -tego, które mogą być wybrane przez krople

$\nabla(i, j)$ – ujemny gradient pomiędzy węzłami i oraz j

Opis algorytmu RFD

- Ujemny gradient pomiędzy dwoma węzłami i & j

$$\nabla(i, j) = \frac{a(i) - a(j)}{d(i, j)}$$

gdzie

$a(x)$ – wysokość węzła x

$d(i, j)$ – długość krawędzi łącząca węzły i -ty z j -tym

Opis algorytmu RFD

- Początek algorytmu, wysokość wszystkich węzłów jest taka sama - $\sum_{l \in V_k(i)} \nabla(i, l) = 0$
- Nadanie płaskiemu gradientowi specjalnego traktowania – zmodyfikowana zasada przejścia:
 - prawdopodobieństwo przejścia kropli po krawędzi z zerowym gradientem ustawiane na pewną niezerową wartość
- Pozwala to kroplom na rozprzestrzenianie się po płaskiej powierzchni w początkowych krokach algorytmu

Opis algorytmu RFD

- ***Erozja ścieżek*** – ścieżki podlegają erozji w zależności od przemieszczeń w poprzednich krokach
- Jeżeli kropla przemieszcza się z węzła A do B to erozji podlega węzeł A
- Wysokość węzła A jest redukowana w zależności od aktualnego gradientu pomiędzy węzłami A i B

Opis algorytmu RFD

- Erozja jest tym większa im większe nachylenie pomiędzy węzłami A i B
- Dla krawędzi płaskich erozja jest mała
- Wysokość docelowego węzła (np. morza) nie zmienia się i pozostaje równa 0 podczas wszystkich kroków algorytmu

Opis algorytmu RFD

- Proces erozji zapobiega w praktyce, wykonywania przez krople zamkniętych cykli, ponieważ trasa musi zawierać co najmniej jedno wzniesienie
- Zgodnie z podstawowymi zasadami krople nie mogą wspinać się
- W ACO trasy feromonów obejmujące niezależne ścieżki mogą sobie wzajemnie przeszkadzać w taki sposób, że mrówki żywią się cyklem powrotnym i podążają dalej

Opis algorytmu RFD

- Załóżmy, że nowa droga, która jest lepsza niż inne ścieżki do tej pory została znaleziona
- W algorytmie RFD, erozja szybko faworyzuje taką ścieżkę w odróżnieniu od innych
- Gradienty takiej nowej ścieżki są średnio lepsze do poprzednich ścieżek

Opis algorytmu RFD

- ***Nanoszenie osadu*** – po zakończeniu procesu erozji, wysokość wszystkich węzłów w grafie nieznacznie wzrasta
- Celem jest zapobieżenie erozji prowadzącej do sytuacji, w której wszystkie wysokości $\rightarrow 0$
- Gradienty nieistotne, mogłoby zniszczyć wszystkie utworzone ścieżki
- W ostatnim etapie algorytmu zapisuje się najlepsze znalezione do tej pory rozwiązanie

Udoskonalenia algorytmu RFD

- Pozwalamy kroplom na poruszanie się po ścieżkach, dla których gradient jest zerowy
- Zastosowany mechanizm:
 - krople mogą wspinać się po zboczach z małym prawdopodobieństwem
- To prawdopodobieństwo jest odwrotnie proporcjonalne do wzrostu gradientu
- Mechanizm taki poprawia poszukiwania optymalnej ścieżki

Udoskonalenia algorytmu RFD

- Prawdopodobieństwo wspinaczki po zboczach jest redukowane w kolejnych iteracjach algorytmu
- Idea zapożyczona z algorytmu symulowanego wyżarzania
- Zastosowana technika pozwala uniknąć zablokowania w lokalnych optimach

Udoskonalenia algorytmu RFD

- Gromadzenie kropli w celu zmniejszenia liczby ruchów
- Krople w tym samym węźle są łączone w jedną kroplę o większym rozmiarze
- Na początku algorytmu, kropla ma pewien założony rozmiar N
- Kiedy kropla zostaje przeniesiona do innych węzłów, wówczas zostaje podzielona na krople o mniejszych rozmiarach, każda z nich po innej ścieżce

Udoskonalenia algorytmu RFD

- Gromada kropli pozwala zmniejszyć liczbę decyzji podejmowanych przez algorytm na każdym kroku
- Mniej kropel do rozważenia (mniejszy wysiłek obliczeniowy dotyczący ruchu każdej kropli)
- Jedna decyzja dla każdej grupy
- Zwiększenie skuteczności algorytmu RFD

Udoskonalenia algorytmu RFD

- Zasady mechanizmu przemieszczania dużej kropli wody:
 - operator $\lfloor X \rfloor$ zwraca całkowitą część X rozmiaru kropli wody
 - prawdopodobieństwo przyjęcia wyboru kierunku przemieszczenia dotyczy tak jakby to była kropla o wielkość 1
 - deterministyczny podział kropli na części na przenoszonych do odpowiednich miejsc docelowych

Udoskonalenia algorytmu RFD

Przykład

Kropla o rozmiarze 204 w węźle A posiada dostępne kierunki przemieszczenia B, C i D , z prawdopodobieństwami: 0.35, 0.22 oraz 0.43

$$\lfloor 204 \cdot 0.35 \rfloor = \lfloor 71.40 \rfloor = 71 \quad \text{wpływa do } B$$

$$\lfloor 204 \cdot 0.22 \rfloor = \lfloor 44.88 \rfloor = 44 \quad \text{wpływa do } C$$

$$\lfloor 204 \cdot 0.43 \rfloor = \lfloor 87.72 \rfloor = 87 \quad \text{wpływa do } D$$

$204 - (71 + 44 + 87) = 2$ krople losowo przesuwane jako osobne krople wg reguły przemieszczenia

Poszukiwania stochastyczną dyfuzją (Stochastic Diffusion Search, SDS)

- SDS – populacyjny algorytm oparty ma najlepszym dopasowaniu wzorca (Bishop 1989)
- SDS – forma bezpośredniej komunikacji (jeden-do-jednego) pomiędzy agentami
- SDS komunikacja - podobna do mechanizmu łączenia par zatrudnionych przez jeden gatunek mrówek (*Leptothorax acervorum*)

Poszukiwania stochastyczną dyfuzją (Stochastic Diffusion Search, SDS)

- Agenci w SDS:
 - wykonują małym kosztem częściowe oceny hipotez (kandydatów rozwiązania problemu poszukiwań)
 - przekazują informację o hipotezach (mechanizm dyfuzji informacji) poprzez bezpośrednią komunikację jeden-do-jeden

Poszukiwania stochastyczną dyfuzją (Stochastic Diffusion Search, SDS)

- Mechanizm dyfuzji generuje wysokiej jakości rozwiązania reprezentowane przez grupę agentów tej samej hipotezy
- Działanie SDS - analogia do gry restauracyjnej

SDS - Gra restauracyjna

- Grupa agentów uczestniczy w długiej konferencji w obcym mieście
- Co noc każdy agent musi zjeść kolację w jakiejś restauracji
- W mieście znajduje się bardzo duży wybór restauracji, a każda z nich oferuje duży wybór dań

SDS - Gra restauracyjna

- Problemów agentów – znaleźć najlepszą restaurację:
 - maksymalna liczba agentów zadowolonych z posiłków
- Każdy agent zachowuje hipotezę o najlepszej restauracji
- Co noc każdy agent testuje hipotezę poprzez spożycie posiłku losowo wybranego dania

SDS - Gra restauracyjna

- Następnego ranka przy śniadaniu każdy agent, który nie był zadowolony ze swojego posiłku pyta się losowo innego agenta aby podzielił się informacją o wrażeniach ze spożywania kolacji
- Jeżeli wrażenie było dobre przyjmuje wówczas ten fakt jako swoją ocenę
- W przeciwnym przypadku wybiera losowo inną restaurację z listy jeszcze nie odwiedzonych do przetestowania kolejnego posiłku

Algorytm SDS

- SDS może być przekształcony dla problemów optymalizacji

$$\max_{\mathbf{x}} f(\mathbf{x})$$

gdzie

$f(\mathbf{x})$ – funkcja celu

\mathbf{x} – hipoteza o najlepszym położeniu rozwiązania (podobieństwo pomiędzy wzorcem a pozycją \mathbf{x} w przestrzeni poszukiwań)

Algorytm SDS

- SDS może być łatwo zaimplementowany do problemów optymalizacji:
 - funkcja celu jest dekomponowana na rozłożone części (oceniana niezależnie)
- SDS stosuje rój n agentów, każdy z nich zachowuje hipotezę x_i o lokalnym optimum

Schemat algorytmu SDS

Procedure SDS

begin

Inicjalizacja agentów

repeat

Ocena agentów

Dyfuzja agentów

until kryterium zatrzymania

end

Schemat algorytmu SDS

Inicjalizacja

- Zwykle początkowe hipotezy każdego agenta są wybierane losowo z równomiernym prawdopodobieństwem
- Każda informacja o możliwych dostępnych rozwiązaniach a priori może być użyta jako inicjacja początkowego wyboru hipotezy

Algorytm SDS

Testowanie

- Każdy agent zachowuje hipotezę \mathbf{x}_i
- Boolowska 'funkcja-test' zwraca prawdę:
 - jeżeli losowo wybrane częściowe rozwiązanie (hipoteza \mathbf{x}_i) - j -ta podcecha jest obecną pozycją \mathbf{x}_{ij} w przestrzeni poszukiwań

Algorytm SDS

Testowanie

- Funkcja testowa zwraca:
 - *prawdę* (agent staje się aktywnym członkiem)
 - *fałsz* (agent staje się aktywnym członkiem)
- Wyniki testu dla danej hipotezy x_i są prawdopodobieństwem, że funkcja testująca będzie zwracać prawdę

Algorytm SDS

Dyfuzja

- Hipotezy są przekazywane za pomocą między-agentowej komunikacji
- Pasywni agenci komunikują się losowo z wybranym innym członkiem roju
- Aktywny agent komunikuje się z hipotezą pasywnego agenta
- Reinicjalizacja nowej hipotezy wybranej losowo z przestrzeni poszukiwań

Algorytm SDS

Kryterium zatrzymania algorytmu

- W kolejnych krokach SDS szuka skupisk agentów tej samej hipotezy
- Dwie możliwości zatrzymania algorytmu
 - *Silne kryterium zakończenia obliczeń*
 - *Słabe kryterium zakończenia obliczeń*

Algorytm SDS

- *Silne kryterium zakończenia*
 - po ustaleniu, że największe skupisko agentów przekracza minimalny próg, sprawdza się, czy rozmiar klastra jest [stochastycznie] trwały w danej liczby powtórzeń
- *Słabe kryterium zakończenia*
 - sprawdza stabilność i minimalny rozmiar ogólnej liczby czynnych agentów, (jako całkowita działalność jest silnie uzależniona od najlepszego rozwiązania znalezionego do tej pory)

Strategie rekrutacji SDS

- Strategie rekrutacji mają wpływ tylko w fazie dyfuzji SDS
- Trzy strategie rekrutacji:
 - *Pasywna rekrutacja* (standardowy mechanizm)
 - *Aktywna rekrutacja*
 - *Dualna rekrutacja*

Strategie rekrutacji SDS

Pasywna rekrutacja

- Pasywny agent A wybiera losowo innego agenta B do komunikacji z nim
- Jeżeli B jest aktywnym agentem wówczas hipoteza agenta B jest przekazywana agentowi A
- Optymalna grupa agentów może się zwiększać w pojedynczej iteracji

Strategie rekrutacji SDS

Aktywna rekrutacja

- Każdy aktywny agent A wybrany losowo kontaktuje się z innym agentem B
- Jeżeli B jest pasywnym agentem, wówczas B jest rekrutem (hipoteza A komunikuje się B)
- Rozmiar grupy agentów może tylko maksymalnie zwiększyć się dwukrotnie w iteracji (jeżeli każdy aktywny agent wybiera pasywnego agenta)

Algorithms SDS

Dualna rekrutacja

- Aktywna i pasywna rekrutacja działa równolegle
- Aktywni jak i pasywni agenci transferują hipotezy z aktywnych do pasywnych członków
- Najbardziej zbliżona do komunikacji w systemach biologicznych

Algorytm SDS

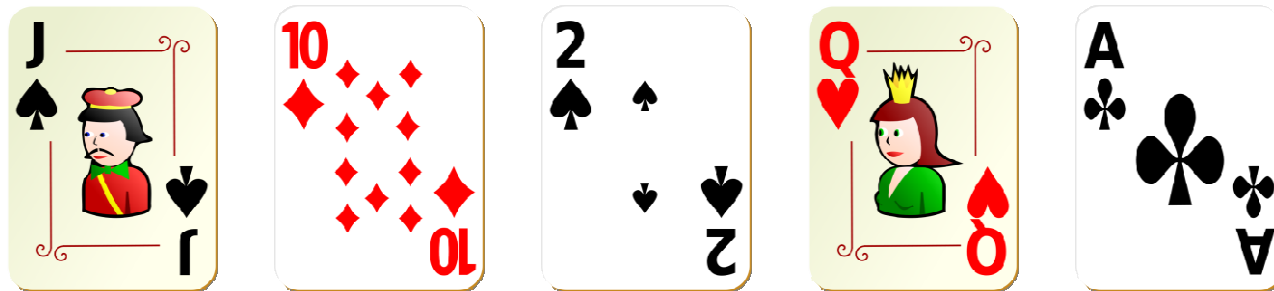
Dualna rekrutacja

- W praktyce, może powodować konflikt
- Priorytetowość musi być określona zarówno:
 - przypisanie hipotezy agenta aktywnego do pasywnego (priorytet aktywności) lub
 - pasywny agent kopiuje hipotezę z aktywnego agenta (priorytet bierności)

Przykład SDS

Królowa kier

- Zadanie znaleźć wzorzec w przestrzeni parametrów
- Znalezienie wymaganego obiektu spośród zbioru obiektów
- Rozważmy znalezienie konkretnej karty



Przykład SDS

Królowa kier

- Każda karta gry jest jednoznacznie opisana za pomocą dwóch właściwości ocenianych oddzielnie: **typ** i **wartość**
- Dziedzina wartości **typu**:
 - *kier, karo, pik, trefl*
- Możliwe **wartości**:
 - *as, dwa, trzy... do 10, walet, królowa i król*

Przykład SDS

Królowa kier

- Przestrzeń rozwiązań składa się ze zbioru {1; 2; 3; 4; 5}, pozycja każdej z pięciu kart opisana kolejnym numerem licząc od lewej
- Zadanie znaleźć królową kier, karta numer 4
- Wszyscy agencje są nieaktywni na początku algorytmu inicjowani hipotezami z zakresu od 1 do 5

Przykład SDS

Królowa kier

- W fazie testowej, każdy agent losowo ocenia albo rodzaj albo wartość karty
- Jeśli typ karty jest 'kier' lub jego wartość jest 'królową' to agent będzie w stanie aktywnym w kolejnej fazie dyfuzji
- W pozostałych przypadkach wszyscy agenci są nieaktywni

Przykład SDS

Królowa kier

- W fazie dyfuzji, każdy nieaktywny agent wybiera losowo innego agenta do kontaktu przedstawiciel
- Jeżeli wybrany agent do komunikacji jest aktywny to kopiujemy jego hipotezę
- W przeciwnym przypadku agent generuje losowo nowa hipotezę
- Kryterium stopu - wszyscy agenci są aktywni

Przykład SDS

- Początkowa grupa losowych 3 agentów

Agent	Karta	Własność	Stan
1	3	typ	nieaktywny
2	1	typ	nieaktywny
3	5	wartość	nieaktywny

- Agenci po pierwszej fazie testowania
- Żaden agent nie ma typu=*kier* i wartości=*królowa* (wszyscy są nieaktywni)

Przykład SDS

- Agenci po pierwszej fazie dyfuzji
- Każdy agent skontaktował się innym nieaktywnego agentem, zatem generowane są losowo nowe hipotezy

Agent	Stan	Kontakt	Nowa Karta
1	nieaktywny	2	4
2	nieaktywny	1	2
3	nieaktywny	1	1

Przykład SDS

- Agenci po drugiej fazie testowania
- Agent 1 - wartość=*królowa* (aktywny)
- Agent 2 - typ=*karo* (nieaktywny)
- Agent 3 - wartość=*walet* (nieaktywny)

Agent	Karta	Własność	Stan
1	4	wartość	aktywny
2	2	typ	nieaktywny
3	1	wartość	nieaktywny

Przykład SDS

- Agenci po drugiej fazie dyfuzji
- Agent 1 był aktywny i nie kontaktował się
- Agent 2 skontaktował się z agentem 1 i skopiował jego hipotezę: dyfuzja informacji
- Agent 3 skontaktował się z agentem 2 i wygenerował nową losową hipotezę

Agent	Stan	Kontakt	Nowa Karta
1	aktywny	-	-
2	nieaktywny	1	4
3	nieaktywny	2	5

Przykład SDS

- Agenci po trzeciej fazie testowej
- Agent 1 typ=*kier* i agent 2 wartość=*królowa* (aktywni)
- Agent 3 typ=*pik* (nieaktywny)

Agent	Karta	Własność	Stan
1	4	typ	aktywny
2	4	wartość	aktywny
3	5	typ	nieaktywny

Przykład SDS

- Agenci po trzeciej fazie dyfuzji
- Agenci 1 i 2 są aktywni
- Agent 3 skontaktował się z agentem 2 i skopiował jego hipotezę

Agent	Stan	Kontakt	Nowa Karta
1	active	-	-
2	active	-	-
3	inactive	2	4

Przykład SDS

- Agenci po czwartej fazie testowej
- Wszyscy agencji posiadają wartość=*królowa* lub typ=*kier* i są aktywni

Agent	Karta	Własność	Stan
1	4	wartość	aktywny
2	4	wartość	aktywny
3	4	typ	aktywny

Zastosowania SDS

- SDS zastosowany do wielu problemów:
 - wybór lokalizacji dla sieci bezprzewodowych
 - detekcja sekwencji w bio-informatyce
 - samo-lokalizacja robota mobilnego
 - rozpoznawanie obiektów
 - śledzenie ruchów oka
 - śledzenie ruchu warg
 - szukanie w tekście

Bibliografia

1. Bishop, J.M., (1989), Stochastic searching networks, in 1st IEE Conf. on ANNs, pp. 329-331, London UK.
2. Bishop, J.M. & Torr, P., (1992), The Stochastic Search Network, in Lingard, R., Myers, D.J. & Nightingale, C. (eds), Neural Networks for Vision, Speech & Natural Language, pp. 370-388, Chapman Hall.
3. De Meyer, K., Bishop, J.M., Nasuto, S.J., (2002), Small world effects in Lattice Stochastic Diffusion Search, Proc. ICANN 2002, in Lecture Notes in Computer Science, 2415, pp. 147-152, Madrid, Spain.
4. De Meyer, K., Nasuto, S.J. & Bishop, J.M., (2006), Stochastic Diffusion Optimisation: the application of partial function evaluation and stochastic recruitment in Swarm Intelligence optimisation, Volume 2, Chapter 12 in Abraham, A., Grosam, C., & Ramos, V. (eds), (2006), Swarm intelligence and data mining, Springer Verlag.