

BPSO를 이용한 리포팅 셀 위치관리시스템 최적 설계

변지환* · †김성수*

Optimal Design of Reporting Cell Location Management System Using BPSO

Ji-Hwan Byeon* · †Sung-Soo Kim*

■ Abstract ■

The objective of this paper is to propose a Binary Particle Swarm Optimization(BPSO) for design of reporting cell management system. The assignment of cells to reporting or non-reporting cells is an NP-complete problem having an exponential complexity in the Reporting Cell Location Management(RCLM) system. The number of reporting cells and which cell must be reporting cell should be determined to balance the registration(location update) and search(paging) operations to minimize the cost of RCLM system. Experimental results demonstrate that BPSO is an effective and competitive approach in fairly satisfactory results with respect to solution quality and execution time for the optimal design of location management system.

Keywords : Binary Particle Swarm Optimization(BPSO), Reporting Cell Location Management System

1. 서 론

1.1 연구의 배경 및 목적

무선환경에서는 단말기가 이동할 수 있기 때문

에 동적인 면이 고려되어야 하고, 원활한 통화연결을 위해서는 실시간으로 움직이는 고객의 위치정보를 효율적으로 관리해야 한다. Hac[7]의 논문에서 따르면 위치관리시스템 방법인 리포팅 셀 위치관리(Reporting Cell Location Management, RCLM)

논문접수일 : 2010년 09월 17일 논문수정일 : 2010년 12월 09일 논문게재확정일 : 2011년 02월 23일

* 강원대학교 산업공학과

† 교신저자

은 네트워크의 셀을 리포팅셀과 리포팅셀이 아닌 셀로 구별해서 관리하는 방법이다. RCLM 방법의 업데이트 과정은 사용자가 리포팅셀로 결정된 셀로 이동하여 들어오는 경우에 발생하고, 페이징 과정은 마지막에 업데이트 과정을 수행한 리포팅셀로부터 다른 리포팅셀을 지나지 않고 이동할 수 있는 리포팅셀이 아닌 셀의 집합을 탐색할 때 발생한다. 위치관리는 이동통신 네트워크에서 발생하는 두 가지 중요한 정보처리 과정인 위치등록 업데이트(Location update)와 페이징(paging) 과정을 효율적으로 관리하는 것이 주된 목적이다. 즉, 위치관리의 목적은 업데이트와 페이징 과정에서 발생하는 비용요소를 최소화하는 것이다.

이 문제는 Bar and Kessler[5]가 NP-complete로 규정 하였다. Subrata[10, 11]의 논문에 따르면 위치관리의 RCLM 방법에서 업데이트는 사용자가 리포팅셀로 이동해 들어오는 경우에 발생하므로 업데이트 비용은 리포팅셀에서 발생하고, 리포팅셀로 이동해 들어오는 사용자의 수가 증가할수록 비용도 증가한다. 그리고 페이징은 탐색하는 셀의 수와 통화 연결이 발생하는 횟수가 증가할수록 페이징 비용이 증가한다. 결국 업데이트 비용과 페이징 비용은 서로 트레이드오프(trade-off) 관계에 있고, 리포팅셀 방법의 총 비용을 최소화할 수 있는 효율적인 시스템 설계를 위해 최적의 리포팅셀과 리포팅셀이 아닌 셀의 수와 위치를 결정해야 한다. Subrata[10]는 타부서치, 유전자알고리즘, 개미군알고리즘 등의 휴리스틱 알고리즘을 사용하여 최적 설계를 시도하였다.

김성수[1, 2] 논문에서는 리포팅셀 위치관리 시스템의 최적설계를 위하여 랭킹개미시스템(Ranking Ant Colony System) 방법을 제안하고 분석하였다. 세대가 진행됨에 따라 적합도 값이 우수한 가능해들을 선발하여 페로몬 업데이트 과정을 수행함으로써 최적해의 탐색 성능을 향상시켰다. 또한, 리포팅셀 위치관리시스템을 대상으로 하여 개미시스템(ant system), 랭킹개미시스템(rank-based ant system) 방법, 최대-최소개미시스템(Max-Min ant system)

방법, 랭킹개미시스템과 최대-최소개미시스템방법을 혼합한 방법을 비교 분석하여 효율성과 강건성(robustness)이 가장 뛰어난 최적 설계 방법을 제시하고 실험하였다.

Kennedy[8] 논문에 따르면, 위 최적화 방법들과 비교하여 Particle Swarm Optimization(PSO)은 파라미터의 조정을 통해 개체의 전역탐색과 지역탐색 능력이 균형을 유지할 수 있다. 일반적으로 PSO는 이론의 간결성, 구현의 용이성, 연산의 효율성과 같은 특징을 나타내고 짧은 계산 시간 내에 최적해를 생성할 수 있으며 다른 확률적 방법보다 안정적인 수렴특징을 나타내고 있다. 이상현[3] 논문에 따르면 일반적인 PSO 알고리즘은 모집단의 각 particle의 위치가 랜덤하게 생성되는 것으로 시작된다. 각 particle은 velocity update rule과 position update rule에 의해 전역적인 해 탐색을 시작한다. 탐색과정에서 발견한 해 중에 최적해인 $pbest$ (particle best)를 기억하고 있으며, 집단에서 발견한 해 중에 최적해인 $gbest$ (global best)와 그 해의 위치벡터 x_{gbest} 의 정보를 공유한다. 각 particle은 현재 위치벡터와 속도벡터, 그리고 x_{pbest} , x_{gbest} 를 이용해서 식 (1)에 의해 이동하게 되며, 위치벡터의 수정된 속도를 이용해서 식 (2)와 같이 행해진다. 이와 같은 과정을 반복함으로써 최적해를 찾아가게 된다.

$$v_i(t+1) = w \times v_i(t) + c_1 \times r_1 \times [pbest_i - x_i(t)] + c_2 \times r_2 \times [gbest - x_i(t)] \quad (1)$$

조재훈[4] 논문에 따르면 $v_i(t+1)$ 는 현재 particle i 의 속도벡터, $v_i(t)$ 은 수정된 particle i 의 속도벡터, $v_i(t)$ 는 현재 particle i 의 위치벡터, $pbest_i$ 는 particle i 가 현재까지 탐색한 최선의 위치벡터, $gbest$ 는 전체 particle들이 현재까지 탐색한 최선의 위치벡터이다. c_1 과 c_2 는 각 particle이 $pbest$ 와 $gbest$ 로 향하는 확률적 가속의 가중치이다. r_1 과 r_2 는 $[0, 1]$ 내의 랜덤 상수이다. w 는 inertia weight라 하는데 현재의 속도에 대한 기존 속도의 영향을 조절하기

위한 제어 파라미터로써 탐색과정 초기에는 전역 탐색의 강화를 위해 w 값을 크게 설정하고, 탐색 후 반부에는 지역탐색을 위해 w 를 감소시킨다.

위치벡터의 수정은 현재의 위치와 수정된 속도를 이용해서 다음과 같은 식으로 행해진다.

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

이와 같은 과정을 반복함으로써 최적해를 찾아가게 된다.

Kennedy[9]는 discrete PSO(DPSO)를 제안하였고 이상헌[3] 논문에서는 이 DPSO를 이용하여 차량경로문제를 최적화 하였다. 본 논문에서는 위치관리시스템의 효율적인 관리를 위해서 RCLM 방법을 대상으로 하여 연산이 효율적이고 짧은 계산 시간 내에 최적해를 생성할 수 있고 다른 확률적 방법보다 안정적인 수렴특징을 갖고 있는 Binary PSO(BPSO) 방법을 적용하여 시스템을 최적설계 하고자 한다.

2. 본론

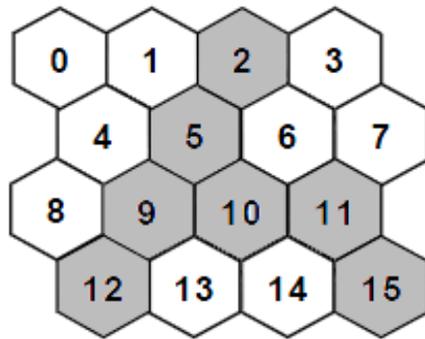
2.1 위치관리시스템의 평가함수

RCLM 시스템에서 사용자에게 통화요구가 발생하면 통화연결을 위해 사용자의 위치를 탐색해야 하는데 그에 앞서 사용자가 최종으로 업데이트 한 리포팅셀로부터 페이징 영역을 계산해야 한다.

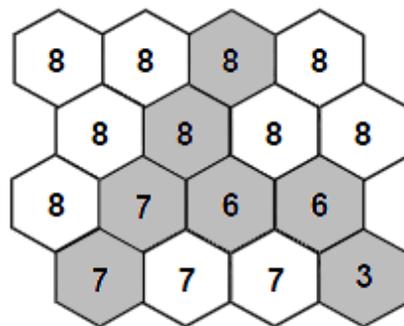
<그림 1>(a)에서 2, 5, 9, 10, 11, 12, 15번 셀이 리포팅셀로 결정된 네트워크의 예라고 가정하자. 만일 사용자가 최종으로 업데이트한 리포팅셀이 2번 셀이라면 페이징 영역은 리포팅셀인 2번 셀을 포함하여 0, 1, 2, 3, 4, 6, 7, 8번 셀이다. 즉, 2번 리포팅셀에서 업데이트한 사용자를 찾기 위한 페이징 대상 영역은 8개의 셀이다.

Hac[7]의 논문에서는 위의 8이라는 값을 2번 리포팅셀의 vicinity value라 하고 이를 사용하여 페이징 비용을 계산했다. 위의 예에서 4번 리포팅셀

이 아닌 셀은 리포팅셀 2, 5, 9, 12번 셀의 페이징 대상이 된다. 즉, 4번 셀이 가질 수 있는 vicinity value는 8, 8, 7, 7이고, 이 중 가장 큰 값인 8이 4번 셀의 vicinity value가 된다. 이런 계산 과정을 통하여 각 셀에 대한 vicinity value를 구한 결과가 <그림 1>(b)에 나타나 있다.



(a) 셀 번호 0~15



(b) 각 셀의 Vicinity value

<그림 1> 4X4 RCLM 네트워크

위의 내용으로부터 업데이트와 페이징 비용으로 구성된 총비용은 Subrata와 Zomaya가 제시한 것처럼 식 (3)과 같이 표현할 수 있다. Wm_k 는 셀 k 로 이동하는 트래픽 양, W_k 는 셀 k 의 통화 요구량, 그리고 C 는 위치등록 업데이트 상수를 나타낸다. C 는 일반적으로 업데이트 비용이 페이징 비용보다 상대적으로 크기 때문에 페이징 비용의 가중치를 1로 두었을 때 업데이트 비용의 가중치로 사용되는 상수이다. Godim[6]은 위치등록 비용의 가

중치는 페이징 비용의 가중치를 1로 했을 때 10으로 산정하는 것이 적절하다고 설명하고 있다. 평가 함수(비용함수)는 다음과 같이 정의할 수 있고 본 논문에서는 이를 최소화하고자 한다.

$$\text{Minimize } C \cdot \sum_{k=1}^N Wm_k \cdot Z_k + \sum_{k=1}^N Wc_k \cdot V_k \quad (3)$$

위의 식에서 만약 셀 k 가 리포팅셀일 경우는 Z_k 값은 1이고, 리포팅셀이 아닐 경우는 0값을 갖게 된다. V_k 는 k 번 셀의 vicinity value이다.

〈표 1〉 4×4 네트워크 실험데이터

Cell	Wci	Wmi	Cell	Wci	Wmi
0	517	518	8	251	445
1	573	774	9	224	2149
2	155	153	10	841	1658
3	307	1696	11	600	952
4	642	1617	12	25	307
5	951	472	13	540	385
6	526	650	14	695	1346
7	509	269	15	225	572

〈그림 1〉에 대한 각 셀의 통화요구량 Wc_k , 셀 k 이동하는 트래픽 Wm_k 이 〈표 2〉와 같다고 하자. 위 데이터를 사용하여 식 (3)에 따라 업데이트 비용, 페이징 비용, 총 비용을 다음과 같이 계산할 수 있다. 업데이트 비용은 리포팅셀로 결정된 2, 5, 9, 10, 11, 12, 15셀에서 발생되고, $10 \times (153 + 472 + 2149 + 1658 + 952 + 307 + 572)$ 로 계산된다. 페이징 비용은 각 셀에 대한 vicinity value와 각 셀의 통화 요구량을 곱하여 $(8 \times 517 + 8 \times 573 + 8 \times 155 + 8 \times 307 + 8 \times 642 + 8 \times 951 + 8 \times 526 + 8 \times 509 + 8 \times 251 + 7 \times 224 + 6 \times 841 + 6 \times 600 + 7 \times 25 + 7 \times 540 + 7 \times 695 + 3 \times 225)$ 로 산출하였다.

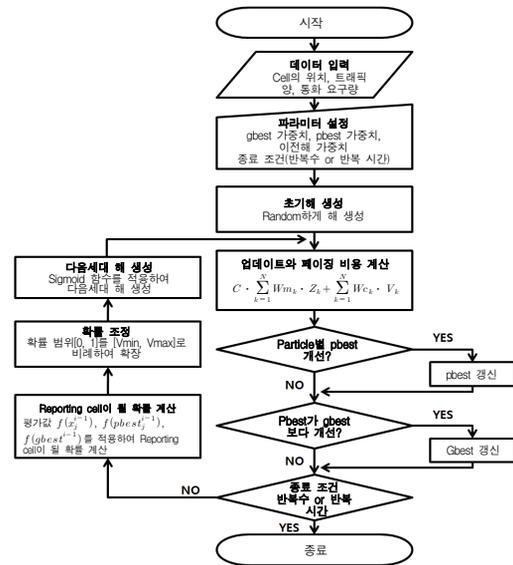
즉, 각 셀의 통화요구량만큼 탐색 영역 내에 있는 셀 모두에게 페이징해야 한다. 평가값(15.54)은 총 비용(117787)을 모든 셀에서 요구하는 통화요구량의

합으로 나누어 통화요구량 1통화에 대한 비용으로 계산할 수 있다.

2.2 PSO를 적용한 RCLM 시스템 최적화 방법

〈그림 2〉는 RCLM 설계 문제에 BPSO 방법을 적용했을 때의 흐름도를 나타낸 것이다. RCLM 설계의 목적은 리포팅셀과 리포팅셀이 아닌 셀의 수와 위치를 결정하는 것이다.

초기 가능해군을 생성하기 위하여 한 Swarm당 Setting된 수만큼의 particle을 Random하게 생성하고 식 (3)으로 각 particle의 해를 평가한다. 다음 세대는 이전의 해와, $pbest$, $gbest$ 의 해를 적절한 비율로 반영하고, 지역해에 빠짐을 방지하고 해 탐색의 다양성을 높이기 위해 일정 확률로 재할당한다. 세대를 거듭하면서 우수한 해가 $pbest$, $gbest$ 로 갱신되고 다음 세대에 반영됨으로써 최적해에 수렴하게 된다.



〈그림 2〉 PSO 방법의 수행 절차

2.2.1 해의 표현

RCLM 방법을 위한 PSO의 해의 표현은 네트워크의 전체 셀의 수를 길이로 하는 1차원 행렬의 형태

이고, 해를 구성하는 모든 셀에 1 또는 0의 값, 즉, 리포팅셀과 리포팅셀이 아닌 셀을 결정한다. <표 2>는 <그림 1>의 16개 셀 4×4 네트워크를 해로 나타낸 것이다.

<표 2> 4×4 네트워크 가능해

Cell 0	Cell 1	Cell 2	Cell 3	Cell 4	Cell 5	Cell 6	Cell 7
0	0	1	0	0	1	0	0
Cell 8	Cell 9	Cell 10	Cell 11	Cell 12	Cell 13	Cell 14	Cell 15
0	1	1	1	1	0	0	1

전술한 것처럼 모든 셀은 이전의 해와 $pbest$, $gbest$ 의 해로부터 계산될 확률의 영향을 받아 리포팅셀 또는 리포팅셀이 아닌 셀로 결정된다

2.2.2 Binary PSO의 설계

<표 2>와 같이 i 세대에 만들어진 j 번째 해의 평가값을 $f(x_j^i)$ 라 하면, <그림 3>(a)와 같이 순차적으로 세대가 진행될 때마다 <그림 3>(b)와 같이 Particle번호 별로 가장 좋은 평가값을 가지는 결과가 $pbest_j^i$ 에 저장된다. 마찬가지로 (c)와 같이 $pbest_j^i$ 중 가장 좋은 평가값을 가지는 결과가 $gbest^i$ 에 저장 된다.

$f(x_j^i) = x_j^i$ 의 평가값

$x_j^i = i$ 세대에 만들어진 j 번째 해

$i =$ 세대 번호($0 < i < \text{세대수}$)

$j =$ 파티클 번호($0 < j < \text{파티클의 수}$)

x_1^2 는 x_1^1 , $pbest_1^1$ 과 $gbest^1$ 을 식 (1)와 식 (2)에 적용하여 생성한다. 평가함수 $f(x_1^1)$ 과 $f(x_1^2)$ 를 비교하여 더 좋은 값을 $pbest_1^2$ 에 저장한다. 이와 같이 $pbest_2^2$, $pbest_3^2$, ..., $pbest_j^2$ 를 구하고 이 값들 중에서 가장 좋은 값을 $gbest^2$ 에 저장한다. 다시 x_3^3 은 x_2^2 , $pbest_1^2$ 와 $gbest^2$ 를 식 (1)와 식 (2)에 적용하여, 생성한다. 이와 같이 세대를 진행하여 최적해를 구한다.

$$1\text{세대} \begin{pmatrix} \text{Particle 1} = f(x_1^1) \\ \text{Particle 2} = f(x_2^1) \\ \text{Particle 3} = f(x_3^1) \\ \vdots \\ \text{Particle } j = f(x_j^1) \end{pmatrix}$$

$$2\text{세대} \begin{pmatrix} \text{Particle 1} = f(x_1^2) \\ \text{Particle 2} = f(x_2^2) \\ \text{Particle 3} = f(x_3^2) \\ \vdots \\ \text{Particle } j = f(x_j^2) \end{pmatrix}$$

⋮

$$i\text{세대} \begin{pmatrix} \text{Particle 1} = f(x_1^i) \\ \text{Particle 2} = f(x_2^i) \\ \text{Particle 3} = f(x_3^i) \\ \vdots \\ \text{Particle } j = f(x_j^i) \end{pmatrix}$$

(a) 순차적 세대진행

$$pbest_j^i \begin{pmatrix} \text{Particle 1} = \min[f(x_1^1), f(x_1^2), \dots, f(x_1^i)] \\ \text{Particle 2} = \min[f(x_2^1), f(x_2^2), \dots, f(x_2^i)] \\ \text{Particle 3} = \min[f(x_3^1), f(x_3^2), \dots, f(x_3^i)] \\ \vdots \\ \text{Particle } j = \min[f(x_j^1), f(x_j^2), \dots, f(x_j^i)] \end{pmatrix}$$

(b) $pbest$ 업데이트

$$gbest^i \begin{pmatrix} \min \begin{pmatrix} pbest(\text{Particle 1}) \\ pbest(\text{Particle 2}) \\ pbest(\text{Particle 3}) \\ \vdots \\ pbest(\text{Particle } j) \end{pmatrix} \end{pmatrix}$$

(c) $gbest$ 업데이트

<그림 3> PSO 세대 진행과 $pbest$, $gbest$ 업데이트

위에서 설명한 x_1^1 , $pbest_1^1$ 과 $gbest^1$ 을 <그림 4>와 같다고 할 때 BPSO를 식 (1)와 식 (2)에 적용하여 생성하는 과정을 다음과 같은 예제로 설명할 수 있다.

		(a)	(b)	(c)															
x_1^1	0	0	1	0	1	0	0	1	0	0	1	0	0	0	1	0	0	1	0
$pbest_1^1$	0	1	0	0	1	0	1	0	0	0	0	1	0	0	1	0	0	1	0
$gbest^1$	0	0	0	0	1	0	1	0	0	1	1	0	0	0	0	0	1	0	1
x_1^2																			

<그림 4> x_1^1 , $pbest_1^1$, $gbest^1$ 의 해 표현과 BPSO 방법 적용

<그림 4>의 x_1^1 , $pbest_1^1$, $gbest_1^1$ 의 평가값을 $f(x_1^1) = 5$, $f(pbest_1^1) = 3$, $f(gbest_1^1) = 2$ 라 하고 w , $c_1 \times r_1$, $c_2 \times r_2$ 를 모두 1이라고 하면, 새로운 x_1^2 를 생성할 때 식 (1)와 식 (2)을 적용하여 x_1^1 , $pbest_1^1$, $gbest_1^1$ 의 유전인자를 받아와서 x_1^2 해를 생성할 때 x_1^1 , $pbest_1^1$, $gbest_1^1$ 의 유전자를 받아오는 확률은 각각의 평가 값에 따라 다음과 같이 각각 계산된다.

$$\frac{\frac{1}{f(x_1^1)}}{\frac{1}{f(x_1^1)} + \frac{1}{f(pbest_1^1)} + \frac{1}{f(gbest_1^1)}} = 0.2$$

$$\frac{\frac{1}{f(pbest_1^1)}}{\frac{1}{f(x_1^1)} + \frac{1}{f(pbest_1^1)} + \frac{1}{f(gbest_1^1)}} = 0.32$$

$$\frac{\frac{1}{f(gbest_1^1)}}{\frac{1}{f(x_1^1)} + \frac{1}{f(pbest_1^1)} + \frac{1}{f(gbest_1^1)}} = 0.48$$

<그림 4>(a)는 x_1^1 , $pbest_1^1$, $gbest_1^1$ 의 유전자가 모두 1이고 각 유전자를 받아들 확률이 0.2, 0.32, 0.48이므로 x_1^2 의 <그림 4>(a) 유전자를 1로 받아들 확률은 0.2, 0.32, 0.48의 합인 1이 되는 것이다. <그림 4>(b)는 x_1^1 , $pbest_1^1$, $gbest_1^1$ 의 유전자가 모두 0이므로 x_1^2 의 유전자를 1로 받아들 확률은 0이다. <그림 4>(c)는 x_1^1 은 0, $pbest_1^1$ 은 1, $gbest_1^1$ 은 1이므로 x_1^2 의 유전자를 1로 받아들 확률은 0.32, 0.48의 합인 0.8이다. 즉, 각 해의 유전 인자를 확률적으로 받아들이는 것을 확률의 합으로 계산되고 식 (1)의 $v_i(t+1)$ 가 계산된다.

확률의 합은 0과 1사이의 값이고, 0과 1사이의 값을 $v_{min} = -4$, $v_{max} = 4$ 사이의 값으로 변환하여 식 (4)에 Kennedy and Eberhart[9]가 제안한 Binary particle swarm optimization의 Sigmoid 함수를 다음과 같이 적용할 수 있다.

$$S(v_i(t+1)) = \frac{1}{1 + e^{-v_i(t+1)}} \quad (4)$$

$$x_1^2 = \begin{cases} 0 & \text{if } r \geq S(v_i(t+1)) \\ 1 & \text{if } r < S(v_i(t+1)) \end{cases}$$

0과 1사이의 랜덤 값 r 이 Sigmoid 함수 값보다 크면 리포팅 셀이 아니고 0으로 표시되고 작으면 리포팅 셀이 되어 1로 표시된다. $v_i(t+1)$ 의 값이 클수록 리포팅셀 즉, '1'이 될 확률이 높아지고 $v_i(t+1)$ 의 값이 작을수록 리포팅셀이 아닌 '0'이 될 확률이 높아진다. 일반적으로 $v_i(t+1)$ 의 범위는 $[-4, 4]$ 내로 제한한다.

$$v_{max} = 4, v_{min} = -4, -4 \leq v_i(t+1) \leq 4$$

i) <그림 4>(a)와 같이 x_1^1 , $pbest_1^1$, $gbest_1^1$ 의 특정 셀이 리포팅 셀이면(즉, 유전인자가 모두 '1'이면) 유전자 1로 받아들 확률은 0.2, 0.32, 0.48의 합인 1이 되고 -4와 4사이의 값으로 변환하면 4가 되어 $S(4) = 0.982$ 즉, 특정 셀이 리포팅셀이 될 확률은 98.2%가 된다.

$$S(4) = \frac{1}{1 + e^{-4}} = 0.982$$

ii) <그림 4>(b)와 같이 x_1^1 , $pbest_1^1$, $gbest_1^1$ 의 특정 셀이 리포팅셀이 아니면(즉, 유전인자가 모두 '0'이면) 유전자를 1로 받아들 확률은 0이 되고 -4와 4사이의 값으로 변환하면 -4가 되어 $S(-4) = 0.017986$ 이 된다. 다시 말해서 특정 셀이 리포팅셀이 될 확률은 1.7986%가 되는 것이다.

$$S(-4) = \frac{1}{1 + e^{-(-4)}} = 0.017986$$

iii) <그림 4>(c)와 같이 x_1^1 의 7번째 셀이 리포팅 셀이 아니고 $pbest_1^1$, $gbest_1^1$ 의 7번째 셀이 리포팅 셀이면(즉, x_1^1 의 유전인자는 '0'이고, $pbest_1^1$, $gbest_1^1$ 의 유전인자는 '1'이면) x_1^2 의 7번째 셀의 유전자를 1로 받아들 확률은 0.32, 0.48의 합인 0.8이 되고 -4와 4사

이의 값으로 변환하면 식 (4)의 $v_i(t+1)$ 이 2.4가 되어 $S(2.4) = 0.916827$ 이 된다. 다시 말해서 x_1^2 의 7번째 셀이 리포팅 셀이 될 확률은 91.6872%가 된다.

$$S(2.4) = \frac{1}{1 + e^{-2.4}} = 0.916827$$

3. 실험결과 및 분석

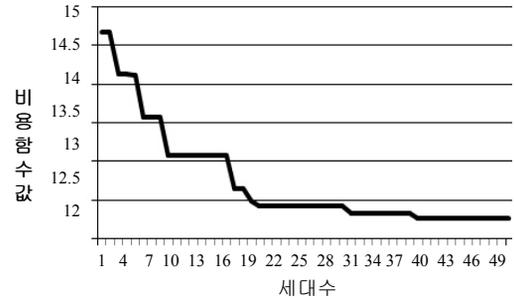
본 절에서는 RCLM 문제에 BPSO 방법을 적용시킨 실험 결과에 대해 분석하였다. 먼저 제 2.1절에서 사용한 16개의 셀로 이루어진 4x4 네트워크를 입력데이터 <표 1>을 이용하여 Intel(R) Core (TM)2 Quad CPU 2.33GHz, 4.00GB RAM, 윈도우7 32비트 운영체제, Visual C++ 환경에서 실험하였다.

BPSO 방법에서 사용된 초기 파라미터 값은 다음과 같다. 4x4 네트워크의 한 세대에 생성되는 particle의 수는 10개, 식 (1)의 *inertio weight*는 1.5로 시작해서 세대가 지나감에 따라 점차 감소시켜 마지막 세대에는 1이 되도록 하였다. 이는 앞서 말했듯이 탐색과정 초기에는 전역탐색의 강화를 위해 w 값을 크게 설정하고, 탐색 후반부에는 지역탐색을 위해 w 를 감소시킨 것이다. $pbest$ 의 *weight* 1, $gbest$ 의 *weight* 0.7을 적용하였다. 이와 같은 파라미터 값은 여러 번의 실험 결과를 통하여 얻을 수 있었다.

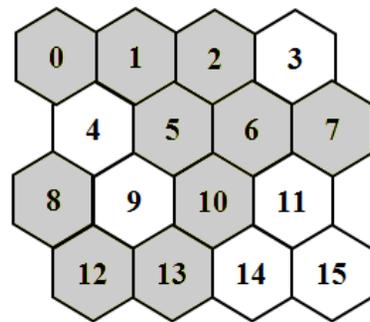
<그림 5>(a)는 4x4 문제에 BPSO를 적용했을 때 세대수가 증가함에 따라 비용함수값이 수렴하는 과정을 나타낸 것이다. <그림 5>(b)는 4x4 네트워크의 16개 셀 최적해 비용함수값(통화요구량 1콜 당 비용), 12.25와 최적설계 결과를 나타낸 것이다. 16개 셀 중에서 10개의 셀이 리포팅셀(회색으로 표시된 셀 0, 1, 2, 5, 6, 7, 8, 10, 12, 13)로 나머지 6개 셀이 리포팅셀이 아닌 셀(흰색으로 표시된 셀)로 선택해야만 최소 비용의 최적 위치관리시스템을 설계할 수 있다.

위와 같은 방법으로 64개 셀로 이루어진 8X8 네트워크로 <표 3>의 입력데이터를 사용해서 실험

하였다.



(a) 비용함수값의 수렴경향



(b) 비용함수값 최선해 (evaluation value 12.25)

<그림 5> 4x4 RCLM 네트워크

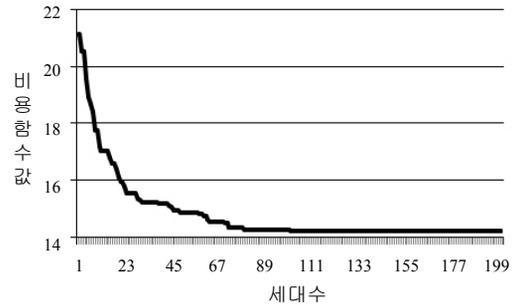
<표 3> 8x8 네트워크 실험데이터

Cell	Wci	Wmi	Cell	Wci	Wmi
0	968	533	32	952	121
1	745	907	33	367	1410
2	827	515	34	132	1011
3	705	1965	35	439	1298
4	902	1336	36	134	1634
5	498	1318	37	153	1750
6	807	1292	38	612	1948
7	62	1789	39	216	662
8	339	541	40	878	700
9	212	1071	41	957	765
10	787	1759	42	363	756
11	664	1416	43	820	436
12	938	1413	44	362	672

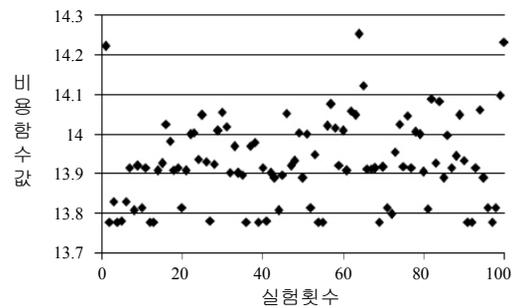
13	719	1224	45	356	822
14	794	484	46	637	1912
15	543	1892	47	626	1402
16	184	626	48	345	524
17	787	104	49	135	1400
18	319	1408	50	175	393
19	25	1256	51	596	1272
20	934	1637	52	677	1197
21	414	1950	53	283	462
22	104	101	54	139	548
23	881	539	55	307	500
24	694	655	56	272	113
25	793	131	57	931	47
26	955	1227	58	38	1676
27	126	450	59	896	1017
28	268	470	60	164	1307
29	96	1081	61	78	499
30	285	1714	62	303	1451
31	368	308	63	578	1606

초기 파라미터 값으로 8×8 네트워크의 한 세대에 생성되는 particle의 수는 25개, 나머지 파라미터는 위 4×4 네트워크의 예시와 같은 값을 사용하여 실험하였다.

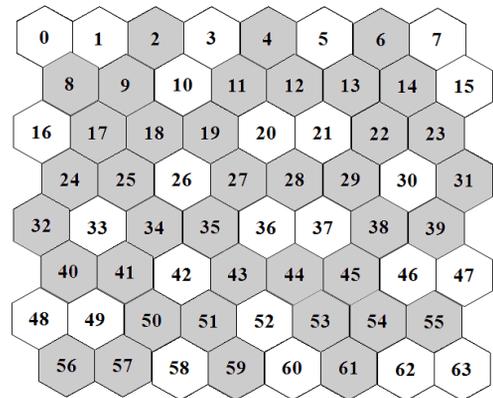
<그림 6>(a)는 8×8 네트워크 문제에 BPSO를 적용했을 때 세대수가 증가함에 따라 비용함수 값이 수렴하는 과정을 나타낸 것이다. <그림 6>(b)는 8×8 네트워크 문제의 100회 실험 결과를 나타낸 것이다. <그림 6>(c)는 8×8 네트워크의 100회 실험 결과 중에서 64개 셀 최적해 비용함수값(통화요구량 1콜 당 비용), 13.78과 최적설계 결과를 나타낸 것이다. 64개 셀 중에서 39개 셀이 리포팅셀(회색으로 표시된 셀 2, 4, 6, 8, 9, 11, 12, 13, 14, 17, 18, 19, 22, 23, 24, 25, 27, 28, 29, 31, 32, 34, 35, 38, 39, 40, 41, 43, 44, 45, 50, 51, 53, 54, 55, 56, 59, 59, 61)로 나머지 25개 셀이 리포팅셀이 아닌 셀(흰색으로 표시된 셀)로 선택해야만 최소 비용의 최적 위치관리시스템을 설계할 수 있다.



(a) 비용함수값의 수렴경향



(b) 100회 실험결과



(c) 비용함수값 최선해

(evaluation value 13.78)

<그림 6> 8×8 RCLM 네트워크

기존 연구 결과와 비교 분석하기 위하여 <표 4>에서 기존 4가지 실험 결과[1, 2]와 본 논문에서 제안하는 BPSO 방법을 비교 분석하였다. 실험 1은 일반적인 개미시스템이며, 실험 2는 개미시스템에 최대-최소 개미시스템 전략을 추가한 것이고, 실험 3

은 개미시스템에 랭크개미시스템 전략을 추가한 것이고, 실험 4는 실험 2와 실험 3을 혼합 적용한 것이다. 실험 5는 본 논문에서 제안하는 BPSO 방법을 적용하여 <그림 6>(b)와 같이 100회 실험 결과를 평균값, 최소값, 최대값, 편차, 탐색률을 정리한 것이다. 이 결과에 따르면 본 논문에서 제안하는 BPSO 방법이 기존에 제안한 다양한 개미시스템 방법 중에서 가장 좋은 결과 실험 4와 비교했을 때 해들의 평균값과 편차는 약간 큰 반면 최적해 탐색 확률은 상대적으로 우수한 것으로 분석되었다. 따라서 추후 BPSO 방법도 실험 4와 같이 더 안정적이고 효율화된 방법으로 발전시킬 수 있도록 수렴성과 다양성의 적절한 밸런스를 맞출 수 있는 연구가 진행되어야 할 것이다.

<표 4> 8×8 네트워크 결과 비교

방법	평균값	최소값	최대값	편차	탐색률
실험 1	14.141	13.911	14.503	0.103	0%
실험 2	14.203	13.943	14.496	0.109	0%
실험 3	13.992	13.782	14.583	0.125	2%
실험 4	13.877	13.782	14.077	0.071	10%
실험 5	13.927	13.782	14.25	0.107	12%

4. 결 론

본 논문에서는 무선통신 네트워크의 최소 서비스 단위인 셀을 대상으로 사용자의 위치등록 과정과 페이징 과정을 효율적으로 수행하기 위한 RCLM 시스템 설계에 어떻게 BPSO 방법을 적용할 수 있는지 상세히 설명하였다. PSO를 개발한 Kennedy[8]가 주장하는 것처럼 리포팅셀 위치관리시스템 최적설계 문제의 $pbest$ 와 $gbest$ 를 업데이트하고 다양성과 수렴성의 밸런스를 조절하기 때문에 구현이 용이하고 연산이 효율적이다. 또한, 본 논문에서 제시한 BPSO 방법이 기존연구의 개미군 최적화 방법보다 최적해 탐색률이 우수한 것으로 분석되었다. 추후, 본 논문에서 제안한 BPSO 방법을 더 개선 발

전시켜서 더 큰 사이즈의 무선 네트워크 문제에 적용하여 성능을 분석 검증할 것이다. 또한, 다른 위치관리시스템인 Location Area planning 방법과의 성능 비교분석도 추가적인 연구가 필요할 것으로 분석 되었다.

참 고 문 헌

- [1] 김성수, 김근배 “랭크개미군전략을 이용한 리포팅셀 위치관리시스템 최적 설계”, 『IE interfaces』, 제19권, 제2호(2006), pp.168-173.
- [2] 김성수, 김형준, 안준식, 김일환 “무선통신 네트워크에서 위치관리 최적설계를 위한 최대-최소개미시스템과 랭크개미시스템의 혼합 방법”, 『전기학회논문지』, 제56권, 제7호(2007).
- [3] 이상헌, 황선호 “Hybrid-PSO 해법을 이용한 수요지 제한이 있는 다용량 차량경로문제”, 『대한산업공학회지』, 제35권, 제2호(2009), pp.150-159.
- [4] 조재훈, 이대중, 송창규, 전명근 “상호정보량과 Binary particle swarm optimization을 이용한 속성 선택 기법”, 『한국지능시스템학회논문지』, 제19권, 제2호(2009), pp.191-196.
- [5] Bar and Kessler, “Tracking mobile users in wireless communications networks,” *IEEE Trans. Information Theory*, Vol.39(1993), pp. 1877-1886.
- [6] Godim, “genetic algorithms and the location area partitioning problem in cellular networks,” Proc. IEEE 46th Vehicular Technology conf. Mobile Technology for the Human Race, 1996.
- [7] Hac, A. and S. Zhou, “Locating strategies for personal communication networks : A novel tracking strategy,” *IEEE J. Selected Areas in Comm.*, Vol.15(1997), pp.1425-1436.
- [8] Kennedy and Eberhart, “Particle Swarm Optimization,” IEEE Inter. Conf. on Neural Net-

- work, 1995.
- [9] Kennedy and Eberhart, "A Discrete Binary Version of the Particle Swarm Algorithm," IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation, 1997.
- [10] Subrata, R. and A.Y. Zomaya, "A comparison of three artificial life techniques for re-
- porting cell planning in Mobile Computing," *IEEE Trans. Parallel And Distributed Systems*, Vol.14, No.2(2003), pp.142-153.
- [11] Subrata, R. and A.Y. Zomaya, "Evolving cellular automata for location management in mobile computing networks," *IEEE Trans. Parallel And Distributed Systems*, Vol.14, No.1(2003), pp.13-26.