

광마이크로셀 이동통신망에서의 채널관리를 위한 동적 섹터결정

이채영 · 윤정훈

한국과학기술원(KAIST) 산업공학과

Microcell Sectorization for Channel Management in a PCS Network by Tabu Search

Cha-Young Lee · Jung-Hoon Yoon

Recently *Fiber-optic Micro-cellular Wireless Network* is considered to solve frequent handoffs and local traffic unbalance in microcellular systems. In this system, central station which is connected to several microcells by optical fiber manages the channels. We propose an efficient sectorization algorithm which dynamically clusters the microcells to minimize the blocked and handoff calls and to balance the traffic loads in each cell. The problem is formulated as an integer linear programming. The objective is to minimize the blocked and handoff calls. To solve this real time sectorization problem the Tabu Search is considered. In the tabu search intensification by Swap and Delete-then-Add (DTA) moves is implemented by short-term memory embodied by two tabu lists. Diversification is considered to investigate proper microcells to change their sectors. Computational results show that the proposed algorithm is highly effective. The solution is almost near the optimal solution and the computation time of the search is considerably reduced compared to the optimal procedure.

1. 서론

최근 서비스가 제공되고 있는 개인휴대통신(PCS) 망은 보다 저렴한 비용으로 높은 수준의 통화품질을 유지하고 서비스 용량 및 영역을 확장하는 것을 목표로 한다. 이에 따라 전파 스펙트럼을 효율적으로 사용할 수 있는 마이크로셀 기반의 PCS 망이 사용자나 사업자 모두에게 이익을 주는 방법으로 제시되고 있다. 마이크로셀룰러 시스템은 시스템 용량의 증가, 저전력 송신과 장비의 소형화로 비용을 절감하고, 보다 유연한 셀 배치를 할 수 있으며, 운영자로 하여금 보다 혁신적인 서비스를 제공할 수 있도록 한다.

마이크로셀룰러 시스템을 적용시키기 위하여 다양한 방법들이 제시되어 왔다. 그 중 중요한 방법은 중앙 기지국(Central Base Station)에서 광섬유, 동축 케이블, 밀리미터 웨이브 라디오 등을 통하여 마이크로셀에 광대역 아날로그 전송을 하는 방법이다. 그 중에서도 특히 광섬유 기술을 활용한 광대역 전송 기술이 연구되고 실험되어 왔다. Ohmoto(1993)는 spectrum delivery

switch를 사용하여 각 subcarrier가 광섬유를 통해 TDMA와 FDMA방식을 사용한 중앙제어방식의 마이크로셀 전파 시스템을 제안하였다. Cheong(1999)이 제안한 시스템은 광마이크로셀룰러 CDMA 시스템으로 접속망 토폴로지가 subcarrier multiplexing 기술에 바탕을 둔 multi-drop bus 방식으로 원격 안테나 제어를 가능하게 하는 group simulcasting을 하고 있다. 광마이크로셀룰러 CDMA 시스템에서의 group simulcasting 방식은 잦은 핸드오프에 따른 통화 품질의 하락과 시그널 트래픽의 혼잡을 완화한다(Shanankarannarayanan, 1995; Ariyavisitakul, 1998). 또한 높은 throughput을 위해 파장분할다중화(WDM)를 이와 같은 접속 네트워크에 구현할 수 있다. 따라서 미래의 광대역 ISDN 비동기 전송 방식(ATM) 기반의 지능망은 이러한 접속망과 쉽게 인터페이스를 이룰 수 있도록 하며, 또한 이러한 hybrid fiber-radio(HFR) 접속망을 통해서 다양한 서비스가 제공될 수 있도록 한다.

용량의 증가를 위한 셀 크기의 축소는 해결해야 할 몇 가지 문제를 야기한다. 설치 및 유지의 문제를 제외하면 해결해야 할 문제는 핸드오프와 셀 계획이라는 두 부분으로 나눌 수 있

다. 핸드오프는 통화 중에 새로운 셀에 진입하는 경우 통화회선의 적절한 교환을 통해 중단없이 통화를 계속할 수 있도록 하는 처리과정을 의미한다. 마이크로셀룰러 시스템에서 첫 번째 문제는 셀 크기가 작아짐으로 해서 짧은 기간 내에 셀을 통과하는 트래픽이 많아져서 잦은 핸드오프가 발생한다는 것이다. 두 번째 문제는 보다 어려운 문제로서, 각각의 기지국의 전파 영역에 대한 것이다. Kuek (1993)은 마이크로셀 시스템에서의 트래픽 변동에 따른 자원 이용이 매크로셀 시스템보다 비효율적임을 보였다. 이 문제를 해결하기 위해 TDMA와 FDMA에서 여러 종류의 동적 채널 할당 기법 (Dynamic Channel Allocation)들과 동적 자원 운영 기법 (Dynamic Resource Management Scheme; DRMS)들이 제시되었다 (Shanankarannarayanan, 1995; Kazrela, 1996). CDMA에서의 DRMS는 (Cheong, VTC 게재 예정)에서 제시되었다.

이 논문은 다음과 같이 구성된다. 2장에서는 광마이크로셀룰러 시스템과 동적 섹터화에 대해 다룬다. 마이크로셀의 섹터화 문제에 대한 수리 모형은 3장에서 다루도록 한다. 4장에서는 이 문제를 해결하기 위한 타부 서치 기법을 소개하고 5장에서는 이에 대한 실험 결과를 보이고, 6장에서 결론을 맺도록 한다.

2. 광마이크로셀 시스템의 개요

2.1 시스템 개요

광마이크로셀룰러 시스템에서는 모든 마이크로 기지국 (micro-base-station; mBS)이 중앙국 (central station; CS)에 연결되어 있다 (<그림 1>). CS는 mBS들을 운영, 제어하며 각 mBS들을 PSTN (public switched telephone network)이나 MSC (mobile switching center)에 연결한다. 모든 mBS들은 섹터로 묶이게 되

며 같은 섹터에 속한 mBS들에게는 같은 신호를 동시에 방송하게 된다. 이러한 것을 simulcasting이라 한다. simulcasting을 위해 CS와 mBS간의 접속망은 multi-drop-bus 토폴로지가 되어야 한다. CDMA에서의 전파 자원 (radio frequency; RF)은 통화 채널 원소 (channel element; CE)의 집합이다. 통화량에 따라 동적으로 서비스할 수 있도록 보통 3개의 섹터가 고정되어 사용된다. 일반적으로 하나의 채널 카드 (channel card)는 24개의 CE를 가지며 3개의 섹터에 4개의 채널 카드가 할당된다. 이렇게 96개의 CE를 가지는 집합을 VBS (virtual base station)이라고 하고 하나의 CS는 여러 개의 VBS를 가질 수 있다. 즉, 하나의 OMS (Operation & Management System)는 적어도 3개의 섹터를 운영, 관리할 수 있다. 모든 RF 자원은 CS에 위치하며 OMS가 관리한다. 모든 변/복조기들 또한 CS에 위치하게 되며 OMS 역시 CS에 설치되어 각 mBS의 트래픽 요구에 따라 자원을 할당할 수 있도록 한다.

CS에서 이동국 (mobile station; MS)으로의 순방향에서, 각 섹터에 해당하는 전파 신호 (radio signal)는 서로 다른 IF (intermediate frequency)로 변환되고 순방향 제어 채널과 reference tone과 결합되어 순방향 광신호로 E/O (electronic-to-optic) 변환된다. 순방향 광신호는 HFR 접속망을 통해 전송된다. 그리고 나서 mBS에서는 순방향 광신호가 O/E (optic-to-electronic) 변환되고, 자신이 할당된 섹터의 IF 신호를 골라내기 위해 bandpass filtering 한 후에 원래의 RF band로 복원되며 증폭된 후에 공중으로 전송하게 된다.

이동국에서 CS로의 역방향에서, 안테나에 수신된 각 전파 신호는 bandpass filtering된 후에 서로 다른 IF로 변환되고, 역방향 제어 채널과 결합하여 E/O 변환되어 역방향 광신호를 생성한다. 역방향 광신호는 HFR 접속망을 통해 CS에게 전송된다. 그리고 나서, CS에서는 역방향 광신호가 O/E 변환되고, 각 섹터에 해당하는 IF를 골라내기 위해 bandpass filtering된 후에 원래의 RF band로 복원된다.

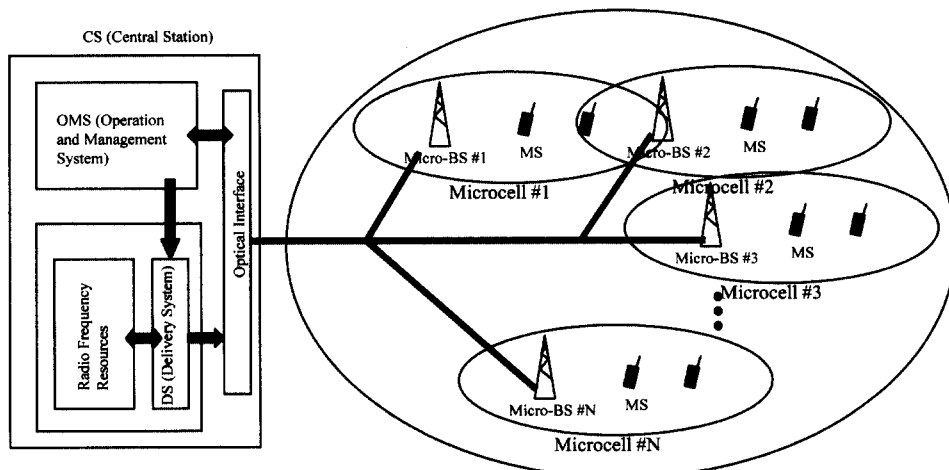


그림 1. 광마이크로셀룰러시스템의 구조.

그림 2. 트래픽 분포와 섹터화.

2.2 마이크로셀의 동적 섹터화

통신 서비스에서 신규 통화시도가 실패하는 호 차단과 통신 시스템 상의 이유로 진행 중이던 통화가 중단되는 호 절단 확률은 품질 수준(QoS)의 중요한 측정치이다. CDMA 시스템에서 이것들은 hard capacity (HC)와 soft capacity (SC), 트래픽 부하 등에 좌우된다. HC는 하나의 VBS의 트래픽 채널 원소(channel element; CE)의 수를 말하고, SC는 한 섹터에서 일정한 QoS 수준을 유지하면서 제공할 수 있는 최대 통화 수를 의미한다. 비록 충분한 양의 CE가 있다 할지라도 통화량이 특정 섹터의 SC를 초과하게 된다면 호 차단이 발생된다.

동적 섹터화는 트래픽 부하에 따라 동적으로 섹터를 형성하는 기법이다. <그림 2>는 하나의 VBS에서 3개의 섹터가 있고, 9개의 마이크로셀이 있는 경우의 simulcasting 그룹의 예를 보이고 있다. 이 예에서 VBS의 HC는 96 CE이고 각 섹터의 SC는 40 CE이다. HC만 고려한다면 호 차단이 전혀 없지만, 섹터의 SC를 초과함으로써 30개의 통화가 차단됨을 알 수 있다. 이처럼 특정 섹터의 트래픽 부하가 높은 경우에 일부 마이크로셀을 트래픽 부하가 낮은 섹터에 할당함으로써 전체적인 트래픽 부하의 균형을 맞추게 된다.

2.3 비용 요소

동적 섹터화 방법은 트래픽 부하에 따라 동적으로 섹터를 재구성한다. t 시점에서의 구성이 $t+1$ 시점에서 재구성되었다고 가정하면 다섯 가지의 비용이 발생하게 된다. 이러한 비용들의 합을 최소화하는 섹터 구성을 찾는 것이 이 논문의 목표이다. 다섯 가지 비용은 다음과 같다.

- (1) HC에 의한 호 차단 비용: 2.1에서의 경우와 같이 하나의 VBS에 3개의 섹터가 있다고 가정하자. 마이크로셀 내의 모든 호는 VBS에 할당된 자원을 사용해야 한다. 따라서 $t+1$ 시점에서 호의 합이 VBS의 HC를 초과하게 되면 CE의 부족으로 인한 호 차단이 발생한다. 이동통신시장에서 호 차단은 매우 중대한 문제이기 때문에 트래픽의 균형을 맞추므로써 이를 최소화할 필요가 있다.
- (2) SC에 의한 호 차단 비용: 한 섹터에 속한 mBS들은 똑같은 전파 신호를 simulcast하게 되고 똑같은 간섭을 일으

킨다. 이러한 간섭이 증가하면 섹터 내의 호에 대한 간섭이 증가하게 된다. 간섭은 CDMA의 수행도에 크게 영향을 끼치기 때문에 간섭을 일정 수준 이하로 유지하는 것이 중요하다. 따라서 $t+1$ 시점에서 한 섹터의 호의 합이 SC를 초과하게 되면 시스템에서는 일정 수준의 간섭을 유지하기 위해 새로운 호에 대해 차단을 하게 된다.

- (3) 소프트 핸드오프 비용: 이동국이 통화 중에 서로 다른 VBS에 속한 두 개의 마이크로셀의 경계선을 지나게 되면 이 이동국에 대해서는 서로 다른 CE에 대한 핸드오프가 필요하게 된다. 이러한 이동국을 서비스하기 위해 시스템은 최소한 두 개의 CE를 소비하게 된다. 만약 특정 마이크로셀 쌍에서의 이동 비율이 다른 쌍에 비해 높다면 이러한 마이크로셀들은 같은 VBS에 속하게 하는 것이 유리하다.
- (4) 소프트 핸드오프 비용: 이동국이 통화 중에 같은 VBS내의 서로 다른 섹터에 속한 두 개의 마이크로셀의 경계선을 지나는 경우에는 같은 CE를 사용하는 핸드오프가 발생한다.
- (5) 재구성에 의한 강제적 핸드오프 비용: t 시점에서 $t+1$ 시점으로 섹터가 재구성이 되었을 때, 일부 mBS들은 그들이 속한 섹터가 바뀌게 된다. 이러한 mBS들은 자신의 파일럿 PN 오프셋(offset)을 동시에 바꾸면서 호를 지속시킨다. 이러한 과정에서 모든 이동국들은 호를 지속시키기 위해 순간적으로 오프셋을 바꾸게 되면서 마치 핸드오프를 하는 것처럼 느끼게 된다.

3. 수리 모형

이동 사용자는 N 개의 6각형 마이크로셀들로 구성된 서비스 영역에 분포되어 있다고 가정한다. 이때 각 마이크로셀에서의 트래픽 수요, $i = 1, \dots, N$ 가 주어진다. P_{ij} 를 이동국이 마이크로셀 i 에서 j 로 이동할 확률이라 할 때 마이크로셀 i 에서 j 로의 핸드오프 호의 수는 $h_{ij} = p_{ij} TD$ 가 된다.

CS가 M 개의 VBS들을 가지고 있다고 가정한다. HC_m 과 SEC_m 은 각각 VBS m , $m = 1, \dots, M$ 의 HC와 그에 속한 섹터들의 집합을 나타낸다. 마찬가지로 SC_k 는 섹터 k 의 SC를 나타낸다.

수식화에 필요한 각 변수를 살펴보면, 이항상수 $x_{ik} = 1$ 은 마이크로셀 i 가 섹터 k 에 속해 있음을 나타내고, 이항변수 $y_{im} = 1$ 은 마이크로셀 i 가 VBS m 에 속함을 나타낸다. 또한 $z_{ijm} = 1$ 은 마이크로셀 i 와 j 가 VBS m 에 속함을 나타낸다. 소프트핸드오프 비용은 변수 $z_{ij} = 1 - \sum_m z_{ijm}$ 을 사용한다. 소프트핸드오프는 두 셀 i 와 j 가 서로 다른 VBS에 속할 때, 즉 $z_{ijm} = 0$ 일 때에 발생한다. 또한 이항변수 $w_{ijk} = 1$ 은 마이크로셀 i 와 j 가 섹터 k 에 속해 있음을 나타내며, 소프트핸드오프 비용은 $w_{ij} - z_{ij}$ 에 의해 계산되며 $w_{ij} = 1 - \sum_k w_{ijk}$ 이다. 소프트핸드오프는 마이크로셀 i 와 j 가 같은 VBS에 속한 서로 다른 섹터에 있는 경우에

발생한다. 이항 변수인 a_{ik} 는 현재의 섹터화에서 마이크로셀 i 가 속한 섹터가 이전 섹터화에서 속한 섹터와 다르면 1값을 부여하고 그렇지 않으면 0값을 부여한다.

변수 hc_m 은 $hc_m = \sum_i TD_i y_{im} - HC_m$ 으로 정의되며 VBS m 의 현재 트래픽 수요와 HC와의 차이를 나타낸다. 또한 $sc_k = \sum_i TD_i x_{ik} - SC_k$ 는 섹터 k 의 현재 트래픽 수요와 SC와의 차이를 나타낸다. $c_i, i = 1, \dots, 5$ 는 2.3절에서 설명하고 있는 다섯 가지의 비용 요소들 각각에 대한 벌금 (penalty)을 나타낸다. 이 중에서 HC, SC와 관련된 비용에서의 벌금은 호가 차단될 때에만 발생하기 때문에 목적함수에 hc_m^+ 와 sc_k^+ 를 적용하며, $hc_m = hc_m^+ - hc_m^-$, $sc_k = sc_k^+ - sc_k^-$ 이 되며, hc_m^+ , hc_m^- , sc_k^+ , sc_k^- 는 비음 (nonnegative) 실수 변수이다. 또한 소프트 핸드오프와 소프트 핸드오프와 관련된 비용에서는 각각의 경우에 대해 핸드오프가 발생한 트래픽에 대해 적용하도록 한다. 마지막으로 재구성에 의한 경제적 핸드오프 비용은 속한 섹터가 바뀐 모든 마이크로셀 ($a_{ik} = 1$)에 대해서 각 마이크로셀의 모든 트래픽을 합하여 벌금을 부여하게 된다.

이에 따라 이 문제는 다음과 같이 수식화된다.

$$\begin{aligned} \text{Minimize} \quad & c_1 \sum_m hc_m^+ + c_2 \sum_k sc_k^+ + c_3 \sum_i \sum_j h_{ij} z_{ij} \\ & + c_4 \sum_i \sum_j h_{ij} (w_{ij} - z_{ij}) + c_5 \sum_i \sum_k a_{ik} TD_i x_{ik} \\ \text{Subject to} \quad & \\ & \sum_k x_{ik} = 1 \quad \forall i \quad (1) \\ & w_{ijk} \leq x_{ik}, \quad w_{ijk} \leq x_{jk}, \quad w_{ijk} \geq x_{ik} + x_{jk} - 1 \quad \forall i, j, k \quad (2) \\ & w_{ij} = 1 - \sum_k w_{ijk} \quad \forall i, j \quad (3) \\ & y_{im} = \sum_{k \in SEC_m} x_{ik} \quad \forall m \quad (4) \\ & z_{ijm} \leq y_{im}, \quad z_{ijm} \leq y_{jm}, \quad z_{ijm} \geq y_{im} + y_{jm} - 1 \quad \forall i, j, m \quad (5) \\ & z_{ij} = 1 - \sum_m z_{ijm} \quad \forall i, j \quad (6) \\ & hc_m = \sum_i TD_i y_{im} - HC_m \quad \forall m \quad (7) \\ & hc_m = hc_m^+ - hc_m^- \quad \forall m \quad (8) \\ & sc_k = \sum_i TD_i x_{ik} - SC_k \quad \forall k \quad (9) \\ & sc_k = sc_k^+ - sc_k^- \quad \forall k \quad (10) \\ & x_{ik}, w_{ijk}, z_{ijm} \in \{0, 1\} \quad \forall i, j, k, m \quad (11) \\ & hc_m^+, hc_m^-, sc_k^+, sc_k^- \geq 0 \quad \forall k, m \quad (12) \end{aligned}$$

목적식은 2.3절에서 설명한 바와 같이 다섯 가지의 비용 요소로 구성되어 있다. 식 (1)은 각 마이크로셀은 하나의 섹터에만 속한다는 조건이며, 식 (2)는 두 마이크로셀이 같은 섹터에 속할 때에만, 즉, $x_{ik} = x_{jk} = 1$ 일 때에만 $w_{ijk} = 1$ 이 됨을 나타낸다. 마찬가지로, 식 (4)는 각 마이크로셀은 하나의 VBS에만 속한다는 조건을 나타내며, 식 (5)는 두 마이크로셀이 같은 VBS에 속할 때에만, 즉, 식 (5)는 $y_{im} = y_{jm} = 1$ 인 경우에만 $z_{ijm} = 1$ 이 됨을 나타낸다.

본 논문에서 다루고 있는 섹터화 문제로 널리 알려진 그룹화 문제의 특수한 경우이다. 일반적으로 그룹화 문제는 NP-hard라고 알려져 있다 (Garey, 1976). 이것은 정확한 알고리즘이 문제의 크기에 따라 지수적인 수행시간이 소요된다는 것을 의미한다. 대부분의 이러한 알고리즘은 실제 문제에 적용하는 경우에 부적합한 경우가 많다. 따라서 실제적으로 사용 가능한 알고리즘을 찾기 위해 섹터화 문제에 대해서 타부서치를 적용하고자 한다.

4. 섹터화 알고리즘을 위한 타부서치

타부서치의 일반적인 구성요소 (Glover, 1989; Rolland, 1996; Glover, 1997)는 1) 단기메모리와 장기메모리, 2) 타부 제약과 열망조건, 3) 집중화와 다양화 전략의 세 가지이다. 집중화 전략은 단기메모리 함수를 사용하여 현재 상태보다 좋은 해를 탐색하도록 환경을 맞춘다. 이러한 전략은 전략적으로 제한된 영역에서 최상의 해를 적극적으로 탐색하는 데 초점을 둔다. 다양화 전략은 장기메모리 함수를 이용하여 해공간에서 방문하지 않았던 영역을 탐색하도록 한다.

단기메모리 함수는 타부 목록과 열망조건으로 구현된다. 타부 목록은 최근에 방문했던 곳의 move를 기억하여 다시 탐색하지 않도록 한다. move가 일정기간 (혹은 타부 사이즈) 동안 목록에 남아 있어서 보다 좋은 해를 적극적으로 탐색하는 데 도움을 준다. 열망조건은 move의 타부 상태를 극복하도록 하는 것으로 그 move가 충분히 좋으면 현재의 타부 상태를 무시하고라도 move를 수행하도록 하는 것이다.

4.1 초기해

모든 mBS들을 섹터에 적절히 할당하기 위해 다음과 같은 초기해 알고리즘을 사용한다. 먼저 가장 트래픽이 많은 mBS를 섹터 1에 할당하고, 섹터의 SC와 해당 VBS의 HC를 초과하지 않는 범위 내에서 이미 할당된 mBS와 핸드오프 트래픽이 가장 많은 순서대로 추가로 할당한다. SC나 HC를 넘기는 경우에는 다음 섹터로 넘어가서 아직 할당되지 않은 mBS들 중에서 트래픽이 가장 많은 것을 할당하고 같은 과정을 반복한다. 마지막 섹터까지 탐색하여 할당되지 못한 mBS가 남아있다면, 남아있는 것 중에서 가장 트래픽이 많은 mBS를 찾아 뒤의 섹터부터 할당하게 된다. 이렇게 하는 이유는 처음부터 마지막 섹터까지 할당해 나가갈 때 뒤쪽 섹터로 갈수록 할당된 mBS들의 수가 적게 되기 때문이다. 즉, 1번 섹터에 mBS들이 할당이 되고 2번 섹터로 진행하게 되면 2번 섹터에서는 처음에 어느 하나의 mBS가 할당된 후에 이 mBS와 핸드오프 트래픽이 있는 mBS들을 탐색하게 되는데, 이들 중에서 이미 1번 섹터에 할당되어 2번 섹터에 할당할 수 없는 경우가 발생할 수 있다. 이러한 경향은 뒷번호의 섹터로 갈수록 커지게 되어 뒤쪽의 섹터는 거의

할당을 하지 못하는 경우가 발생한다. 이렇기 때문에 아직 할당되지 않은 mBS중에서 가장 트래픽이 많은 mBS를 뒤쪽 섹터부터 할당하도록 하여 전체적인 트래픽이 균형을 이루도록 한다. 이 과정을 요약하면 다음과 같다.

```

FOR (i = 1 : i ≤ # of sectors)
  IF (Sector i is empty)
    Find the mBS j which has the largest traffic among unassigned set of mBSs.
  ELSE
    Find the mBS j which has the largest handoff traffic with assigned mBSs in Sector
    Assign mBS j to Sector i until they do not exceed the capacity
    i = i + 1
END FOR
IF (there exist unassigned mBSs)
  Sort mBSs in largest traffic order.
  FOR (i = # of sectors ; i = 1)
    Assign the largest mBS among unassigned set of mBSs to Sector i until they do not exceed the capacity
    i = i - 1
  END FOR
  IF (there remains mBSs after assignment)
    Assign remaining mBSs to Sector 1
END IF
    
```

그림 3. 초기화 알고리즘.

4.2 단기메모리 함수를 이용한 집중화 전략

move의 형태는 Swap와 Delete-then-Add (DTA)의 두 가지로 정의된다. <그림 4>에서 보는 바와 같이, Swap은 서로 다른 섹터에 속해 있는 두 개의 마이크로셀을 바꾸는 것이고, DTA는 한 섹터에 속한 마이크로셀을 다른 섹터로 옮기는 것이다. 이러한 두 가지의 move 중에서 어떠한 것을 선택하는가는 섹터간 트래픽의 분산을 측정하여 결정한다. 섹터간 분산을 작게 하는 것, 즉 섹터간 트래픽 부하의 균형을 맞추는 것이 중요하기

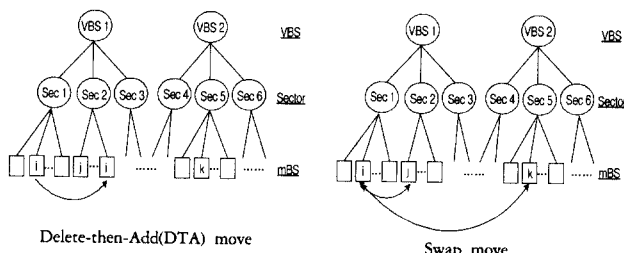


그림 4. move의 두 가지 형태(DTA move와 Swap move).

때문에 분산이 일정량의 한계치 (threshold)보다 크면 DTA를 실행하고, 그렇지 않으면 Swap을 실행한다.

단기메모리 함수는 Swap와 DTA를 각각 따로 관리하는 두 개의 타부 목록으로 구현된다. 최근에 탐색했던 해로 움직이는 것을 방지하기 위해 일정한 시간 (iteration)동안 목록을 가지게 된다. 타부 목록에는 Swap move의 경우 서로 교환된 마이크로셀 번호가 기록되며, DTA move의 경우에는 움직인 마이크로셀 번호와 이전에 속해있던 섹터 번호가 기록된다. 즉 다음과 같이 기록된다.

$$\text{Tabu_DTA}(x, \text{from_sector}) = \text{Iteration} + \text{Tabu_DTA_Tenure}$$

$$\text{Tabu_Swap}(x, y) = \text{Iteration} + \text{Tabu_Swap_Tenure}$$

섹터간 트래픽 분산의 한계치와 Tabu_DTA_Tenure, Tabu_Swap_Tenure는 실험적으로 결정하기로 한다. 타부 상태에서 벗어나기 위한 열망조건은 현재까지의 가장 좋은 해보다 좋은 경우에 타부 상태에 있더라도 이를 극복하여 move가 진행될 수 있도록 하는 Improved Best 전략 (Glover, 1997)을 사용하도록 한다.

4.3 장기메모리를 이용한 다양화 전략

다양화 전략은 해공간에서 방문하지 않았던 새로운 영역을 탐색하기 위한 방법이다. 이 전략은 지역 최적해를 벗어나기 위해 사용되며 보다 넓은 해 영역을 탐색하도록 구현된다. 이의 구현은 어느 정도의 iteration이 진행된 후에 선택된 마이크로셀의 적용된 move 횟수 (frequency)에 따라 그 횟수가 가장 적은 마이크로셀을 선택하여 목적 함수를 가장 작게 하는 move (Swap or DTA)를 결정하여 움직이게 된다. 이러한 다양화의 단계는 수 차례를 반복하게 되며, 다음 다양화 단계에서는 직전 다양화 단계에서의 선택된 마이크로셀의 move 횟수에 기반하여 진행되도록 한다. 마이크로셀의 frequency는 다음과 같이 기록된다.

If microcell i is selected by DTA move

$$\text{Then Frequency}[i] = \text{Frequency}[i] + 1$$

If microcell i and j are selected by Swap move

$$\text{Then Frequency}[i] = \text{Frequency}[i] + 1,$$

$$\text{Frequency}[j] = \text{Frequency}[j] + 1$$

이러한 타부 서치의 과정을 두 알고리즘으로 나누어 구현한다. 두 알고리즘 모두 각각 3개의 Phase를 거치게 된다.

Phase 1은, 각각의 섹터의 트래픽을 측정하여 트래픽이 가장 많은 섹터 (Max_Traffic_Sector)와 트래픽이 가장 적은 섹터 (Min_Traffic_Sector)를 선택하고 섹터간 트래픽의 분산에 따라 move의 형태 (Swap, DTA)를 결정하고, 3장의 수리 모형에서 설명한 목적 함수를 최소화하도록 두 섹터간에서 move를 수행하

도록 한다. 이 과정은 주로 섹터 용량(SC)을 고려하여, 해를 탐색하는 초기에 섹터간 트래픽의 균형을 맞추어 주는 과정이다. 물론 타부 목록과 NoImprove 파라미터의 업데이트가 이루어진다. NoImprove는 현재 iteration의 해가 직전 iteration의 해보다 좋으면, 즉 목적 함수의 값을 작게 만들면 NoImprove 값을 0으로 하고, 그렇지 않으면 그 값에 1을 더하여 해가 어느 정도 개선이 되고 있는지, 또 해가 얼마 동안의 iteration 동안 개선이 되지 않는지를 나타내는 파라미터이다.

Phase 2 역시 Phase 1과 마찬가지로 각각의 섹터의 트래픽을 측정하여 트래픽이 가장 많은 섹터(Max_Traffic_Sector)와 트래픽이 가장 적은 섹터(Min_Traffic_Sector)를 선택하는 과정을 동일하게 반복한다. 하지만 여기서는 지금까지의 해보다 개선된 해가 나올 경우에 서로 다른 VBS간의 move를 수행하도록 하여 더 이상 개선되지 않을 때까지 진행하도록 한다. Phase 1에서는 SC를 고려한 섹터간 트래픽의 균형을 이루게 하는 기능만을 하는 반면에, Phase 2에서는 SC뿐만 아니라 VBS의 용량(HC)과 핸드오프 비용까지 고려하여 보다 최적해에 다가가려는 단계이다. 지금까지의 해보다 개선된 해가 나온 다음의 과정은 타부 목록과 NoImprove 파라미터의 업데이트가 이루어지지 않는다.

두 알고리즘 모두 지금까지의 두 Phase를 공통적으로 사용한다. 하지만 Phase 3의 과정을 다르게 적용하도록 한다. 알고리즘 1에서는 다양화(diversification) 전략을 수행하도록 설계되고, 알고리즘 2에서는 이러한 과정이 없이 move를 보다 폭 넓게 적용하는 방법을 사용하기로 한다.

알고리즘 1에서는 4.3에서 설명한 방법과 같이, Phase 2까지 진행한 후에 지금까지 마이크로셀이 선택된 횟수(frequency)에 바탕을 두어 가장 적게 선택된(frequency가 가장 작은) 마이크로셀을 찾아 목적 함수를 가장 적게 하는 move(Swap or DTA)를 결정하여 움직이게 된다. 다양화단계는 수 차례 반복하도록 하며, 각각의 다양화 단계에서는 이전 단계에서의 frequency를 참조하도록 한다.

알고리즘 2에서는 Phase 2까지와 다르게 보다 폭 넓은 move를 적용하도록 한다. 즉, 각각의 마이크로셀에 대해 가능한 DTA move와 Swap move를 모두 평가한 후에 목적 함수를 가장 작게 하는 move를 선택하도록 한다. Phase 2까지는 근사적인 방법으로 최적해 방향으로 진행하는 단계이지만, Phase 3에서는 보다 정확하게 최적해의 방향으로 진행하도록 하는 방법을 나타내고 있다.

이 과정은 다음과 같이 정리된다.

Procedure Tabu Search (Algorithm 1)

Step 1. 초기해를 구한다.

Step 2. 각 파라미터를 초기화한다.

Step 3. Phase 1 move and Phase 2 move

NoImprove가 Phase1_NoImprove + Phase2_Noimprove가 될 때까지 (a) - (g)를 반복한다.

- (a) Max_Traffic_Sector, Min_Traffic_Sector, Inter_Sector_Traffic_Variance를 찾는다.
- (b) 만일 Inter_Sector_Traffic_Variance가 Threshold보다 크면 DTA move, Threshold보다 작으면 Swap move로 결정한다.
 - (b-1) DTA move이면 Max_Traffic_Sector에서 목적 함수를 최소화하는 mBS를 하나선택하여 Min_Traffic_Sector로 옮긴다.
 - (b-2) Swap move이면 목적 함수를 최소화하는 mBS를 Max_Traffic_Sector에서 하나, Min_Traffic_Sector에서 하나를 각각 선택하여 두 mBS를 바꾼다.
- (c) 타부 목록을 체크하고, 열망 조건(Aspiration Criteria)을 체크한다.
- (d) Current_Solution이 Best_Solution보다 작으면 Best_Solution을 업데이트한다.
- (e) Phase 2 move
NoImprove > Phase1_NoImprove이고, Current_Solution < Best_Solution이면, 소속된 VBS가 다른 mBS들에 대해서 목적 함수를 최소화하는 move를 선택하여 해가 더 이상 개선이 되지 않을 때까지 (Best_Solution이 업데이트되지 않을 때까지) 지속한다.
- (f) Frequency와 NoImprove를 업데이트한다.
- (g) Goto Step 4.

Step 4. Phase 3 move

NoImprove := 0; Div_Frequency(j) := 0

NoImprove가 Phase3_NoImprove가 될 때까지 (a) - (d)를 반복한다.

- (a) 이번 루프에서 선택되지 않은 mBS중에서 minimum Frequency(j)를 가지는 mBS를 선택한다.
- (b) 선택된 mBS를 포함하는 가장 좋은 move를 찾는다.
- (c) (b)에서 선택된 mBS(들)의 Div_Frequency, Diversification, NoImprove, Best_Solution을 업데이트한다.
- (d) Diversification이 D_{max} 보다 작으면 Frequency(j) := Div_Frequency(j), Goto (a).

알고리즘 2의 과정은 마지막 단계에서만 알고리즘 1과 다르게 구성된다. 알고리즘 2는 다음과 같이 진행된다.

Procedure Tabu Search (Algorithm 2)

Step 1 - Step 3 : 알고리즘 1과 동일.

Step 4. Phase 3 move

NoImprove := 0;

NoImprove가 Phase3_NoImprove가 될 때까지 (a) - (c)를 반복

- (a) 모든 mBS들에 대해 목적 함수를 최소화하는 move를 선택한다.
- (b) 타부 목록을 체크하고, 열망 조건을 체크한다.
- (c) Current_Solution이 Best_Solution보다 작으면 Best_Solution을 업데이트한다.

제시된 타부 서치 알고리즘이 구한 값을 $TS(i)$ 라 하면 다음과 같은 $GAP(i)$ 가 정의된다.

$$GAP(i) = \frac{TS(i) - OPT(i)}{OPT(i)} \times 100(\%)$$

또한 전체의 성능을 나타내는 Average GAP은 각 문제에 대한 $GAP(i)$ 의 평균을 나타내며, 다음과 같이 정의된다.

$$Average\ GAP = \frac{1}{N} \sum_{i=1}^N GAP(i)$$

where N = 전체 문제의 수

이 논문에서 다루고 있는 문제는 최소화 (minimization) 문제이므로 타부 서치가 구한 해 $TS(i)$ 는 항상 $OPT(i)$ 보다 크거나 같다. 따라서 $GAP(i)$ 는 항상 0보다 크거나 같은 값을 가지게 되며, $GAP(i)$ 가 작을수록 보다 우수한 해임을 나타낸다. 또한 Average GAP 역시 작을수록 우수한 값을 나타낸다.

5. 실험 결과

5.1 실험 환경

이 장에서는 타부 서치를 이용한 색터화 알고리즘의 효율성을 실험하도록 한다. 앞에서 설명한 알고리즘은 Windows 95 기반의 Intel Pentium MMX 200 MHz, 64 Mbytes RAM 환경에서 Visual C++ (Version 6.0)로 구현하였다. 실험 환경은 <그림 5>에서와 같이 육각형 모양의 마이크로셀이 각각 12개, 19개, 37개가 있는 환경이며 이에 대한 세부 사항은 <표 1>에서 나타내고 있다. <표 1>에 있는 number of permutation 항목은 가능한 색터화 (sectorization)의 경우의 수를 말하고 있으며, 마이크로셀의 수가 증가할수록 가능한 경우의 수가 지수적으로 증가함을 알 수가 있다. 본 논문에서 제시하고 있는 타부 서치 알고리즘의 효율성은 3장의 모형을 풀 수 있는 CPLEX (1998)의 결과와 비교하도록 한다.

5.3 파라미터 설명 (Parameters Description)

3장에서 설명하고 있는 목적 함수의 비용 요소를 채택하였으며, 각 계수의 값은 $c_1 = 10$, $c_2 = 5$, $c_3 = 2$, $c_4 = c_5 = 1$ 이다. 색터화 (sectorization) 문제에서의 일차적인 목표는 색터의 용량 (SC)과 VBS의 용량 (HC)에 의해 발생하는 호 차단 비용을 최소화하는 것이기 때문에 c_1 과 c_2 를 위와 같이 할당하였고, 소프트 핸드오프 비용 계수인 c_3 는 소프트 핸드오프 비용 계수인 c_4 의 두 배를 할당하였다. 그 이유는 소프트 핸드오프는 채널 요소 (CE)를 하나만 사용하는 반면에 소프트 핸드오프는 두 개의 채널 요소를 사용하기 때문이다. 강제적 핸드오프는 소프트/소프트 핸드오프와 마찬가지로 하나 혹은 두 개의 채널 요소를 필요로 한다. 하지만 강제적 핸드오프는 재구성 주기의 초기에만 나타나기 때문에 c_5 는 소프트 핸드오프와 동일하게 할당하도록 한다.

5.2 성능 평가 함수 (Performance Measure)

제시된 알고리즘의 성능을 평가하는 방법으로는 CPLEX가 구한 최적해와 값의 차이를 나타내는 함수를 사용하도록 한다. 문제 i 에 대해서 CPLEX가 구한 최적해 값을 $OPT(i)$ 라 하고,

그리고, tabu tenure (Tabu_DTA_Tenure, Tabu_Swap_Tenure), 색터간 트래픽 분산에 따라 move의 형태를 결정하기 위해 사용되는 분산의 한계치 (threshold), Phase1_NoImprove, Phase2_NoImprove, Phase3_NoImprove, 다양화의 횟수를 나타내는 D_{max} 는 실험적으로 결정하도록 한다. 전체 세 가지 크기의 문제 중에서 마이크로셀이 19개인 문제에 대해서 이들 파라미터들을 결정하는 방법을 예시하고자 한다.

먼저, 타부 서치에서 가장 중요한 파라미터인 tabu tenure에 대해서는 다음과 같은 그림으로 나타낸다. 이 논문에서는 tabu tenure를 DTA move와 Swap move에 대해 다르게 적용하기 때문에 각기 다르게 최적화를 수행하여야 한다. 이 경우에는 알고리즘 2의 경우에 바탕을 두고 진행한다.

<그림 6>과 <그림 7>은 tabu tenure에 따른 Average GAP (%)의 변화를 나타내고 있다. <그림 6>은 Tabu_Swap_Tenure를 변화시킨 값이고, <그림 7>은 Tabu_DTA_Tenure를 변화시킨

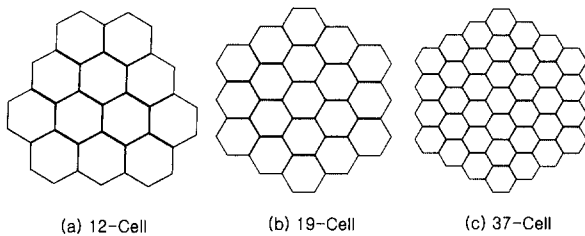


그림 5. 실험 환경 (12-cell, 19-cell, 37-cell).

표 1. 실험 환경 (12-cell, 19-cell, 37-cell)의 세부 사항

	12 microcells	19 microcells	37 microcells
Number of microcells	12 microcells	19 microcells	37 microcells
Number of VBSs	1 VBS	2 VBS	3 VBS
Number of sectors	3 sectors	6 sectors	9 sectors
Average traffic/microcell	9 Erlangs	12 Erlangs	9 Erlangs
Number of permutations	$3^{12} \approx 5.31 \times 10^5$	$6^{19} \approx 6.09 \times 10^{14}$	$9^{37} \approx 2.03 \times 10^{35}$

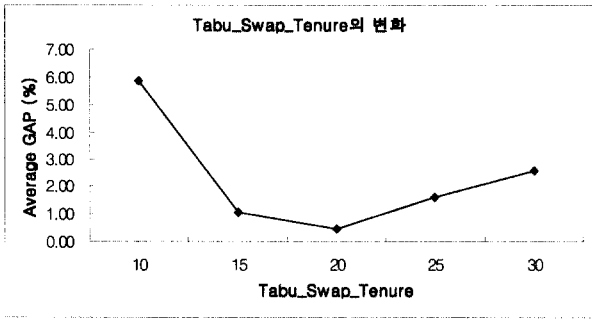


그림 6. Tabu_Swap_Tenure의 변화에 따른 Average GAP (%)의 변화 (Tabu_DTA_Tenure = 17).

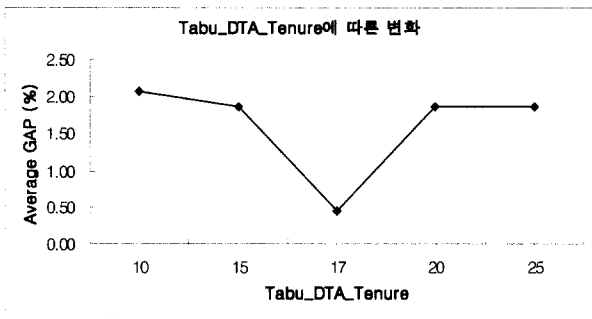


그림 7. Tabu_DTA_Tenure의 변화에 따른 Average GAP (%)의 변화 (Tabu_Swap_Tenure = 20).

값을 나타낸다. 위의 결과에 따라 tabu tenure를 결정하도록 하며, 19개의 마이크로셀 문제에서는 Tabu_Swap_Tenure=20, Tabu_DTA_Tenure=17로 결정하도록 한다. 마찬가지로 12-cell 문제의 경우는 Tabu_Swap_Tenure=15, Tabu_DTA_Tenure=10으로 하고, 37-cell의 문제에서는 각각 50과 30으로 하였다.

또한, Phase1_NoImprove는 Phase 1 move를 수행하게 되는 기

간을 나타내는 것으로, NoImprove가 Phase1_NoImprove에 도달하면 SC를 만족시키는 해를 찾는 과정에서 지역해(local optimum)에 빠진 것으로 판별해도 된다. 따라서 초기의 탐색 기간인 Phase 1에서는 이 기간을 tabu tenure와 같은 기간으로 정의하도록 하며, Tabu_DTA_Tenure와 Tabu_Swap_Tenure의 크기가 다르므로 둘 중에서 큰 것을 적용하도록 한다. Phase2_NoImprove는 VBS간 move를 수행하는 기간을 나타내는 것으로, 실험적으로 이 기간 중에 2~3번 정도 Best_Solution보다 우수한 Current_Solution을 찾아서 Phase 2 move를 적용할 수 있도록 결정하였으며, 이 값은 12-cell, 19-cell, 37-cell 모두에 대해 30을 적용한다. Phase3_NoImprove는 알고리즘 1과 알고리즘 2에 다르게 적용하도록 하였으며, 다양화 전략을 제공하는 알고리즘 1에서는 12, 19, 37-cell 문제 모두 200을 적용하였고, 알고리즘 2에서는 하나의 iteration이 알고리즘 1보다는 성능이 뛰어나게 나타나지만, 시간이 많이 소요되는 것을 감안하여 세 가지의 문제 크기에 대해 모두 100을 적용하였다.

한계치(threshold)에 관한 파라미터는, 섹터간 트래픽의 분산이 대략 섹터 개수에 비례한다고 볼 수 있으므로, 섹터에 소속된 평균 개수와 평균 트래픽을 감안하여 각 문제 크기에 대해 20, 30, 50을 적용하였다.

다양화의 횟수를 결정하게 되는 D_{max} 에 대해서는, 평균적으로 수행되는 횟수로 10번을 수행하도록 하였다. 이 정도 기간이면 다양화의 과정을 충분히 거쳤다고 판단할 수도 있으며, 또한 10보다 큰 값을 부여하더라도 Phase3_NoImprove 파라미터의 제약에 걸려서 그 과정을 충분히 수행하지 못하고 멈추게 된다. 이러한 파라미터들을 바탕으로 하여 각 문제에 대해 실험을 수행하도록 한다.

<표 2>는 12개의 마이크로셀이 있는 경우의 결과를 나타내고 있다. 총 20개의 문제를 테스트하였으며, OPT는 CPLEX가

표 2. 12-Cell 문제의 계산 결과

Prob #	OPT	Time	Algorithm1	Time	GAP	Algorithm2	Time	GAP
1	94.03	3.13	94.03	0.44	0.00	94.03	0.49	0.00
2	335.84	23.07	335.84	0.55	0.00	335.84	0.49	0.00
3	65.95	3.51	65.95	0.49	0.00	65.95	0.50	0.00
4	54.65	2.15	54.65	0.44	0.00	54.65	0.49	0.00
5	291.13	6.53	291.13	0.44	0.00	291.13	0.55	0.00
6	99.98	3.90	99.98	0.43	0.00	99.98	0.55	0.00
7	70.96	2.19	70.96	0.44	0.00	70.96	0.44	0.00
8	34.57	2.03	34.57	0.39	0.00	34.57	0.43	0.00
9	156.51	2.20	156.51	0.55	0.00	156.51	0.55	0.00
10	53.98	4.83	53.98	0.50	0.00	53.98	0.55	0.00
11	123.47	4.11	123.47	0.54	0.00	123.47	0.44	0.00
12	105.85	2.64	105.85	0.44	0.00	105.85	0.49	0.00
13	262.88	9.28	262.88	0.44	0.00	262.88	0.44	0.00
14	153.72	8.73	153.72	0.49	0.00	153.72	0.50	0.00
15	59.18	3.79	59.18	0.44	0.00	59.18	0.50	0.00
16	273.07	4.84	273.07	0.49	0.00	273.07	0.50	0.00
17	165.47	5.28	165.47	0.49	0.00	165.47	0.49	0.00
18	95.63	3.35	95.63	0.49	0.00	95.63	0.50	0.00
19	111.61	4.12	111.61	0.49	0.00	111.61	0.55	0.00
20	33.52	2.47	33.52	0.44	0.00	33.52	0.44	0.00
Average		5.11		0.47	0.00		0.49	0.00

표 3. 19-Cell 문제의 계산 결과

Prob #	OPT	Time	Algorithm1	Time	GAP	Algorithm2	Time	GAP
1	358.87	900.89	358.87	0.98	0.00	358.87	8.40	0.00
2	353.00	683.66	365.83	2.03	3.63	353.00	7.58	0.00
3	340.04	605.33	343.70	2.03	1.08	340.04	10.61	0.00
4	157.72	508.88	159.83	2.04	1.34	157.72	8.02	0.00
5	220.37	2777.81	241.63	2.25	9.65	239.72	14.89	8.78
6	162.41	665.70	162.41	1.92	0.00	162.41	10.88	0.00
7	167.23	513.00	175.16	1.87	4.74	167.23	8.79	0.00
8	163.63	470.49	163.63	2.03	0.00	163.63	10.55	0.00
9	177.48	547.66	183.01	2.53	3.12	177.48	7.86	0.00
10	132.93	209.76	132.93	2.20	0.00	132.93	7.80	0.00
11	126.71	401.84	126.71	2.03	0.00	126.71	7.53	0.00
12	448.73	716.61	536.46	1.87	19.55	448.73	10.05	0.00
13	172.90	497.29	220.08	1.86	27.29	172.90	7.80	0.00
14	165.73	743.20	165.73	2.25	0.00	165.73	8.41	0.00
15	160.34	438.09	160.34	2.09	0.00	160.34	10.44	0.00
16	160.83	412.49	178.99	2.28	11.29	160.83	8.07	0.00
17	236.46	764.89	236.46	1.98	0.00	236.46	7.47	0.00
18	206.33	2436.00	245.21	2.19	18.84	206.48	8.67	0.07
19	178.48	1402.25	178.48	1.76	0.00	178.48	17.58	0.00
20	168.51	997.06	168.51	1.92	0.00	168.51	7.14	0.00
Average		834.65		2.01	5.03		9.43	0.44

구한 최적해, $OPT(i)$ 를 나타내고, Algorithm1과 Algorithm2는 각각 알고리즘 1과 알고리즘 2가 구한 해를 나타내며, Time은 각각을 구하는 데 소요된 시간(초), GAP은 이 알고리즘과 최적해 값과의 차이, 즉 $GAP(i)$ (%)를 나타낸다. 이 결과에서 나타내고 있는 바와 같이, CPLEX를 이용하여 최적해를 구하는 데에 걸리는 시간은 평균 5.11초가 소요되었지만, 알고리즘 1은 평균 0.47초가 소요되어 최적해를 구하는 시간의 10%도 걸리지 않았으며, 최적해와 같은 해를 구하고 있다. 알고리즘 2의 경우를 보면 알고리즘 1과 마찬가지로, 최적해와 같은 해를 구했으며, 평균 시간은 0.49초로 최적해를 구하는 시간의 약 10% 정도가 소요되었다.

<표 3>은 19개의 마이크로셀이 있는 경우의 결과를 나타내고 있다. 이 문제의 경우 최적해를 구하는 데 걸리는 시간이 때

우 늘어났다는 것을 알 수 있다. CPLEX를 이용하여 최적해를 구하는 시간은 평균 834.65초가 소요되었고, 알고리즘 1은 평균 2.01초, 알고리즘 2는 평균 9.43초가 소요되었다. 반면 최적해와의 차이가 알고리즘 1은 평균 5.03%, 알고리즘 2는 평균 0.44%로 나타나, 두 알고리즘 모두가 최적해를 구하는 시간보다는 훨씬 빠르게 나타났으며, 두 알고리즘을 비교했을 때에는 12개, 19개의 마이크로셀의 문제를 보았을 때, 알고리즘 1이 속도면에서는 더 우수하고, 해의 값을 보았을 때에는 알고리즘 2가 우수한 면을 보이고 있다.

<표 4>는 37개의 마이크로셀이 있는 경우의 결과를 보여주고 있다. CPLEX의 시간 제한을 1시간으로 두었는데, 그 이유는 이 논문에서 다루는 문제가 신속히 처리되어야 하며 적어도 1시간 안에 해를 구해야 하는 제약을 담고 있기 때문이다. 문제

표 4. 37-Cell 문제의 계산 결과

Prob #	Algorithm1	Time	Algorithm2	Time	Algorithm1/Algorithm2
1	443.54	10.22	429.78	80.85	1.03
2	328.15	16.26	344.46	57.23	0.95
3	386.17	14.17	397.11	92.44	0.97
4	525.95	11.92	414.52	51.24	1.27
5	460.76	11.43	440.98	64.98	1.04
6	446.92	14.67	439.53	80.08	1.02
7	239.72	19.06	244.36	49.16	0.98
8	281.37	14.56	267.62	36.85	1.05
9	351.41	11.20	307.55	39.32	1.14
10	351.45	10.93	238.81	119.30	1.47
11	297.11	12.09	309.52	41.85	0.96
12	340.57	11.21	400.29	55.20	0.85
13	437.11	16.59	462.91	47.73	0.94
14	330.88	14.06	304.28	47.67	1.09
15	414.99	14.28	397.83	92.93	1.04
16	302.14	10.87	262.60	59.04	1.15
17	819.23	9.89	734.57	88.21	1.12
18	482.41	11.91	444.78	48.39	1.08
19	361.62	11.59	279.25	50.14	1.29
20	378.14	11.86	306.49	43.77	1.23
Average		12.94		62.32	1.08

의 크기가 커짐에 따라 CPLEX는 최적해를 전혀 찾지 못했다. 그래서 알고리즘 1과 알고리즘 2의 결과를 비교해보도록 한다. 표 6에서 Algorithm1/Algorithm2는 알고리즘 1과 알고리즘 2의 값의 비율을 나타내고 있다. 이 값이 1보다 크면 알고리즘 2의 결과가 더 우수하며, 1보다 작은 경우에는 알고리즘 1이 더 우수한 결과이다. 표에서 보는 바와 같이 Algorithm1/Algorithm2의 평균은 1.08로 나타나 알고리즘 2가 알고리즘 1보다 약 8.5% 정도가 우수한 결과를 나타냄을 알 수 있다. 하지만 시간을 보았을 때에는 알고리즘 1이 평균 12.94초를 보여 평균 62.32초를 보인 알고리즘 2보다 우수한 결과를 보이고 있다.

6. 결론 및 향후 연구과제

이 논문에서는 최근 관심을 많이 끌고 있는 CDMA 기반의 PCS 시스템에서 비용을 절감하고, 용량을 확대하며 서비스 영역을 확장할 수 있도록 하기 위해, 기존의 매크로셀 시스템과 일반 마이크로셀 시스템의 대안으로 제시되고 있는 광마이크로셀룰러 CDMA 시스템에서의 동적 섹터화 문제를 다루고 있다. 이러한 시스템에서는 스펙트럼을 효율적으로 사용하고 높은 수준의 품질을 제공하며, 높은 수준의 전송 속도를 제공하여 사용자들이 요구하는 새로운 서비스를 제공할 수 있도록 한다. 하지만, 이 시스템에서는 시간에 따라 동적으로 신속하게 섹터화를 위한 결과를 출력하는 것이 매우 중요하다. 5장에서 보는 바와 같이 마이크로셀의 수가 조금만 커져도 최적해를 구하기에는 시간이 너무 많이 소요된다. 이에 이 논문에서는 타부 서치를 이용하여 채널 자원을 고려한 동적 섹터화 알고리즘 두 가지를 제시하고 있다. 이 알고리즘에서는 두 개의 타부 목록으로 운영되는 두 가지 형태의 move를 이용하고 있으며, 이러한 move는 섹터간 트래픽의 분산에 따라 동적으로 적용시키도록 한다. 제안하고 있는 알고리즘은 3개의 Phase를 가지고 있으며, 마지막 Phase 3를 다르게 적용함으로써 두 가지의 알고리즘을 제시하고 있다. 각각의 알고리즘이 시간과 해의 값의 상충 관계를 보이고 있다. 해의 질을 생각한다면 알고리즘 2가 더 우수하게 나오고, 속도를 비교한다면 알고리즘 1이 우수한 결과를 보이고 있다. 실험 결과를 보면 제시한 알고리즘들이 최적해를 구하는 시간에 비해 그 계산 시간이 훨씬 짧으며 최

적해에 거의 근접하는 해를 구하는 매우 효율적인 알고리즘임을 알 수 있다.

앞으로 문제의 크기를 더욱 확대하여 실험하는 과정이 필요하며, 또한 알고리즘 2에서 Phase 3에서의 시간이 많이 소요되기 때문에 이러한 문제점을 해결하고, 알고리즘 1과 알고리즘 2를 적절히 혼합하여 어느 정도의 해의 품질과 속도를 줄 수 있는 알고리즘을 개발할 필요가 있다.

참고문헌

- Ariyavisitakul S. S. and Greenstein L. J. (1998), A Method for Estimating Local Traffic Load in a Simulcast Radio System, *IEEE Transactions on Vehicular Technology*, 47(1).
- Cheong, J. M., Kim, T. G., Park, T. H., Choi, J. H., and Park S., A Dynamic Resource Management Scheme (DRMS) on Hybrid Fiber Radio (HFR) CDMA Microcellular Systems, *to appear VTC*.
- Cheong, J. M., Seo, S. H., Kim, J. S., Koo, H. S., Park, H. S., Choi, J. H., and Park, S. (1999), A Novel CDMA-based Fiber-Optic Microcellular System: FoMiCellTM, *Proc. 49th IEEE VTC*, 2200-2203.
- CPLEX 4.0, CPLEX Optimization Inc., (1998).
- Garey, M. R., Johnson, D. S., and Stockmeyer L. (1976), Some simplified NP-complete graph problems, *Theoretical Computer Science*, 1, 237-267.
- Glover, F. (1989), Tabu Search, *ORSA Journal on Computing*, 1(3), 190-206.
- Glover, F. and Laguna, M. (1997), *Tabu Search*, Kluwer Academic Publishers.
- Katzela, I. and Naghshineh, M. (1996), Channel Assignment Schemes for Cellular Mobile Telecommunication Systems: A Comprehensive Survey, *IEEE Personal Communications Magazine*, 10-31.
- Kuek, S. S., Wong, W. C., Vijayan, R., and Goodman, D. J. (1993), A Predictive Load Sharing Scheme in a Microcellular Radio Environment, *IEEE Transactions on Vehicular Technology*.
- Ohmoto, R., Ohtsuka, H. and Ichikawa, H. (1993), Fiber-Optic Microcell Radio Systems with a Spectrum Delivery Scheme, *IEEE Journal on Selected Areas in Communications*, 11(7), 1108-1117.
- Rolland, E., Pirkul, H., and Glover, F. (1996), Tabu Search for Graph Partitioning, *Annals of Operations Research*, 63, 209-232.
- Sarnecki, J., Vinodari, C., Javed, A., O'Kelly P. and Dick, K. (1993), Microcell Design Principles, *IEEE Communications Magazine*, 31.
- Shanankarannarayanan, N. K., Philips, M. R., Darcie, T. E., and Ariyavisitakul, S. (1995), Multipoint Wireless System Using Fiber/Coaxial Networks for Personal Communication Services and Subscriber Loop Application, *Proc. IEEE GLOBECOM'95*, 977-981.