

# 음성 및 데이터 서비스를 지원하는 CDMA/TDD 시스템을 위한 실시간 통합 타임슬롯 및 코드 할당 체계

†장근녕\* · 이기동\*\*

Real-time Integrated Timeslot and Code Allocation Scheme  
for the CDMA/TDD System Supporting Voice and Data Services

†Kun-Nyeong Chang\* · Ki-Dong Lee\*\*

## ■ Abstract ■

CDMA/TDD with asymmetric capacity allocation between uplink and downlink is a highly attractive solution to support the next generation mobile systems. This is because flexible asymmetric allocation of capacity to uplink and downlink usually improves the utilization of the limited bandwidth. In this paper, we mathematically formulate an optimal timeslot and code allocation problem, which is to maximize the total utility considering the numbers of codes(channels) allocated to each data class and the forced terminations of previously allocated codes. We also suggest a real-time integrated timeslot and code allocation scheme using Lagrangean relaxation and subgradient optimization techniques. Experimental results show that the proposed scheme provides high-quality solutions in a fast time.

Keywords : CDMA/TDD, Lagrangean Relaxation, Subgradient Optimization, Timeslot and Code Allocation

논문접수일 : 2007년 07월 06일    논문게재확정일 : 2008년 06월 05일

\* 이 논문은 2005년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2005-41-B00172).

\*\* 연세대학교 정경대학 경영학과

\*\*\* LGE Mobile Research

† 교신저자

## 1. 서론

CDMA(code division multiple access)는 음성, 비디오, 고속 인터넷접속서비스 등 보다 다양한 멀티미디어 서비스를 제공하는 3세대 및 4세대 이동통신시스템을 위한 매우 유망한 기술이다. 특히, DS-SS-CDMA with TDD mode(direct sequence CDMA with time division duplex mode : CDMA/TDD)가 마이크로 셀(micro cell) 또는 피코 셀(pico cell) 환경 하에서 멀티미디어 서비스를 효율적으로 제공하기 위한 매우 유망한 기술로 알려지고 있다[8, 9, 12].

이동통신망의 주요한 특징 중의 한가지로 상향링크(uplink : 이동국(mobile station)에서 기지국(base station)으로의 링크)과 하향링크(downlink : 기지국에서 이동국으로의 링크)간의 트래픽 불균형을 들 수 있다. 이러한 불균형은 음성 및 데이터 서비스를 지원하는 3세대 및 4세대 이동통신시스템에서는 더욱 심화된다[9]. 즉, 이동국에서 기지국으로 전송되는 데이터는 짧은 메시지인 경우가 많고, 기지국에서 이동국으로 전송되는 데이터는 상대적으로 용량이 큰 경우가 많다. 이러한 환경 하에서 상향링크와 하향링크에 동일한 대역폭(bandwidth)이 할당된다면 시스템 용량은 하향링크에 의해 결정되게 되고, 결국 대역폭 낭비와 시스템 용량 감소 결과를 초래하게 된다.

이러한 환경 하에서는 두 링크에 할당되는 타임슬롯 수를 달리하는 것이 용이한 CDMA/TDD 시스템이 그렇지 못한 CDMA/FDD(DS-SS-CDMA with FDD(frequency division duplex) mode) 시스템에 비해 매우 유용한 방법이라 할 수 있다. FDD의 경우에는 상향링크에 할당되는 주파수와 하향링크에 할당되는 주파수가 일정한 간격을 두고 따로 분리되어 있고, TDD의 경우에는 상향링크와 하향링크에 별개의 주파수를 할당하지 않고 타임슬롯(timeslot) 단위로 구분하여 사용한다[9]. 따라서 FDD에서는 채널간의 상호교환이 어려운 반면에 TDD에서는 상향링크와 하향링크가 동일한 주파수 대역을 사용하기 때문에 채널간의 상호교환이 용이하다

는 특성이 있다. 즉, CDMA/TDD 시스템은 유연하게 자원을 할당할 수 있기 때문에 상향링크와 하향링크간의 트래픽 불균형이 심하고 자주 변하는 멀티미디어 통신환경 하에서 주파수 이용률을 극대화할 수 있다[5, 9, 16, 19].

CDMA/TDD 시스템은 모든 타임슬롯을 각 셀에서 상향링크용 또는 하향링크용으로 동일하게 사용하는 경우와, 특정 타임슬롯에 대해서 일부 셀에서는 상향링크용으로 사용하고 다른 셀에서는 하향링크용으로 사용하는 경우로 나눌 수 있다. 시스템 성능 측면에서는 후자가 보다 유리하다. 모든 셀에서 하향링크용으로만 사용되는 타임슬롯을 하향링크 타임슬롯(downlink timeslot), 모든 셀에서 상향링크용으로만 사용되는 타임슬롯을 상향링크 타임슬롯(uplink timeslot), 일부 셀에서는 상향링크용으로 사용되고 다른 셀에서는 하향링크용으로 사용될 수 있는 타임슬롯을 교차 타임슬롯(crossed timeslot)이라 한다.

CDMA/TDD 시스템에서 자원 할당은 타임슬롯과 코드(code) 단위로 이루어진다. 3GPP에서는 CDMA/TDD 시스템에서의 자원 할당을 slow DCA와 fast DCA로 구분하고 있다[1]. slow DCA는 셀별로 상향링크용 타임슬롯과 하향링크용 타임슬롯을 결정하는 것이고, fast DCA는 베어러 서비스(bearer service)에 특정 타임슬롯과 코드를 할당하는 것이다. fast DCA를 위해서는 호 수락 제어 체계가 필요하다. 이에 따라 CDMA/TDD 시스템의 효율적인 자원 운용을 위한 타임슬롯 할당(timeslot allocation) 체계와 호 수락 제어(call admission control) 및 코드 할당(code allocation) 체계 수립에 대한 연구가 많이 진행되고 있다.

효율적인 타임슬롯 할당 체계 수립을 위한 연구를 살펴보면, 먼저 Esmailzadeh et al.[10]은 상향링크와 하향링크에 동일한 수의 타임슬롯을 할당한다는 가정 하에 전송출력제어 체계를 제시하고 있다. Nasreddine and Lagrange[22], Yomo et al.[25], Zhang and Zeira[26] 등은 최적화의 개념을 고려하지 않은 단순한 휴리스틱 타임슬롯 할당 체

계를 제시하고 있는데, 음성 및 데이터 서비스를 동시에 고려하지 못하고 있다. Jeon and Jeong[15]은 2개의 셀로 구성된 시스템을 대상으로 타임슬롯 별로 간섭을 분석하고 최적 타임슬롯 할당의 효과를 제시하고 있지만, 각 셀에서의 트래픽을 서비스 유형별로 구분하지 않고 통합하여 하나의 트래픽으로 간주하고 있다. Chang et al.[7]은 멀티셀(multicell) CDMA/TDD 시스템을 대상으로 최적 타임슬롯 할당 체계를 제시하고 있지만, 음성 및 데이터 서비스를 동시에 고려하지 못하고 있다. Chang and Lee[6]는 음성과 데이터 서비스를 지원하는 2개의 셀로 구성된 시스템을 대상으로 최적 타임슬롯 할당 체계를 제시하고 있지만, 실시간으로 활용 가능한 동적 타임슬롯 할당 체계를 제시하지 못하고 있다. Kobayashi et al.[20]은 각 링크에서의 전송률과 목표 임계값을 고려하여 상향링크용 타임슬롯 수와 하향링크용 타임슬롯 수를 결정하는 실시간 분산 타임슬롯 할당 체계를 제시하고 있는데, 이 방법은 설정된 목표 임계값에 의해 성능이 좌우되고 타임슬롯 할당이 최적화된다는 보장을 하지 못한다. Lee et al.[21]은 음성과 데이터 서비스를 제공하는 CDMA/TDD 시스템을 대상으로 총효율을 최대화하는 실시간 분산 타임슬롯 할당 체계를 제시하고 있다. 그런데, 이 연구에서는 다양한 유형의 음성 및 데이터 서비스를 고려하지 않고 있고, 기존에 할당된 코드의 강제종료는 고려하지 않고 있다.

일반적으로 코드 할당을 위한 호 수락 제어는 셀 별로 상향링크 타임슬롯 수, 하향링크 타임슬롯 수, 교차 타임슬롯 수 등이 주어진 상황 하에서 코드 요구가 있을 경우 간섭 정도를 고려하여 이용 가능한 코드를 할당하는 방식으로 이루어지고 있다[2-4]. 즉, 새로운 코드를 할당하더라도 간섭 수준이 임계값 이하이면 호를 수락하고, 그렇지 않으면 호를 수락하지 않는다. 특히 Cao et al.[2]에서는 여러 개의 코드 할당이 필요한 데이터 서비스의 경우, 멀티코드(multicode) 방식과 멀티슬롯(multislot) 방식의 할당이 가능한지 확인하는 호 수락 제어 체계

를 제시하고 있다. 그런데 이들 연구들은 타임슬롯 할당과 호 수락 제어를 통한 코드 할당을 통합적으로 고려하지 못하고 있고, 또한 현재 할당된 채널의 강제종료(forced termination)를 통한 신규 코드 수요에 대한 할당 문제를 고려하지 못하고 있다.

본 논문에서는 다양한 유형의 음성 및 데이터 서비스를 제공하는 CDMA/TDD 시스템을 대상으로, 서비스별 코드 수요와 기존에 할당된 코드의 강제종료를 고려하면서 총효율을 최대화하는 통합 타임슬롯 및 코드 할당 방법을 찾기 위한 최적 타임슬롯 및 코드 할당 모형을 정형화한다. 또한, 라그랑지안 이완 기법과 서브그라디언트 최적화 기법을 이용하여 실시간 통합 타임슬롯 및 코드 할당 체계를 제시한다.

본 논문은 다음과 같이 구성되어 있다. 제 2장에서는 CDMA/TDD 시스템의 모형을 살펴보고, 상향링크 타임슬롯, 하향링크 타임슬롯, 교차 타임슬롯에 대한 간섭을 분석한다. 제 3장에서는 서비스별 코드 수요와 기존에 할당된 코드의 강제종료를 고려하면서 총효율을 최대화하는 통합 타임슬롯 및 코드 할당 방법을 찾기 위한 최적 타임슬롯 및 코드 할당 모형을 정형화한다. 제 4장에서는 라그랑지안 이완 기법과 서브그라디언트 최적화 기법을 이용한 실시간 통합 타임슬롯 및 코드 할당 체계를 제시한다. 제 5장에서는 제 4장에서 제시한 할당 체계의 성능을 실험을 통해 분석하고, 제 6장에서는 결론을 제시한다.

## 2. 시스템 모형과 간섭 분석

본 논문은 하나의 TDD 프레임이  $N$ 개의 타임슬롯으로 나누어지는 CDMA/TDD 시스템을 대상으로 한다.  $N$ 개의 타임슬롯은  $N^d$ 개의 하향링크 타임슬롯,  $N^u$ 개의 상향링크 타임슬롯,  $N^c$ 개의 교차 타임슬롯으로 나누어진다( $N = N^d + N^u + N^c$ ). 하향링크 타임슬롯과 상향링크 타임슬롯은 모든 셀에서 각각 하향링크용과 상향링크용으로 동일하게 사용된다, 반면에 교차 타임슬롯은 셀의 상황에 따

라 셀별로 다르게 사용될 수 있다. 즉, 하향링크용으로 사용될 수도 있고, 상향링크용으로 사용될 수도 있다. 셀  $i$ 에서  $N^c$ 개의 교차 타임슬롯 중에  $N_i^{d(c)}$ 개는 하향링크용으로 사용되고(하향링크용 교차 타임

슬롯),  $N_i^{u(c)}$ 개는 상향링크용으로 사용된다(상향링크용 교차 타임슬롯)( $N^c = N_i^{d(c)} + N_i^{u(c)}$ ).

CDMA/TDD 시스템에서 지원하는 서비스의 전송률(data rate)은 코드 한 개의 전송률의 배수로

<표 1> 기호 정의

기 호	정 의
$T$	서비스 클래스의 집합
$S_i^t$	셀 $i$ 의 서비스 클래스 $t$ 사용자의 집합
$A_i$	셀 $i$ 의 인접셀의 집합
$N$	TDD 프레임당 타임슬롯 수( $N = N^d + N^u + N^c$ )
$N^d$	하향링크 타임슬롯 수
$N^u$	상향링크 타임슬롯 수
$N^c$	교차 타임슬롯 수( $N^c = N_i^{d(c)} + N_i^{u(c)}$ )
$N_i^{d(c)}$	셀 $i$ 에서 하향링크용으로 사용되는 교차 타임슬롯 수
$N_i^{u(c)}$	셀 $i$ 에서 상향링크용으로 사용되는 교차 타임슬롯 수
$M_i^t$	셀 $i$ 에서 서비스 클래스 $t$ 에 할당된 채널의 수
$M_i^{d,t}$	셀 $i$ 의 하향링크 타임슬롯들을 이용하여 서비스 클래스 $t$ 에 할당한 하향링크용 채널의 수
$M_i^{u,t}$	셀 $i$ 의 상향링크 타임슬롯들을 이용하여 서비스 클래스 $t$ 에 할당한 상향링크용 채널의 수
$M_i^{d(c),t}$	셀 $i$ 의 교차 타임슬롯들을 이용하여 서비스 클래스 $t$ 에 할당한 하향링크용 채널의 수
$M_i^{u(c),t}$	셀 $i$ 의 교차 타임슬롯들을 이용하여 서비스 클래스 $t$ 에 할당한 상향링크용 채널의 수
$L_{ik}^{d,t}$	특정 하향링크 타임슬롯에서 셀 $i$ 의 서비스 클래스 $t$ 사용자 $k$ 에게 할당되는 코드의 수
$L_i^{d,t}$	특정 하향링크 타임슬롯에서 셀 $i$ 의 서비스 클래스 $t$ 사용자들에게 할당되는 코드의 수( $L_i^{d,t} = \sum_{k \in S_i^t} L_{ik}^{d,t}$ )
$L_{ik}^{u,t}$	특정 상향링크 타임슬롯에서 셀 $i$ 의 서비스 클래스 $t$ 사용자 $k$ 에게 할당되는 코드의 수
$L_i^{u,t}$	특정 상향링크 타임슬롯에서 셀 $i$ 의 서비스 클래스 $t$ 사용자들에게 할당되는 코드의 수( $L_i^{u,t} = \sum_{k \in S_i^t} L_{ik}^{u,t}$ )
$L_{ik}^{d(c),t}$	특정 하향링크용 교차 타임슬롯에서 셀 $i$ 의 서비스 클래스 $t$ 사용자 $k$ 에게 할당되는 코드의 수
$L_i^{d(c),t}$	특정 하향링크용 교차 타임슬롯에서 셀 $i$ 의 서비스 클래스 $t$ 사용자들에게 할당되는 코드의 수( $L_i^{d(c),t} = \sum_{k \in S_i^t} L_{ik}^{d(c),t}$ )
$L_{ik}^{u(c),t}$	특정 상향링크용 교차 타임슬롯에서 셀 $i$ 의 서비스 클래스 $t$ 사용자 $k$ 에게 할당되는 코드의 수
$L_i^{u(c),t}$	특정 상향링크용 교차 타임슬롯에서 셀 $i$ 의 서비스 클래스 $t$ 사용자들에게 할당되는 코드의 수( $L_i^{u(c),t} = \sum_{k \in S_i^t} L_{ik}^{u(c),t}$ )
$W$	확산대역폭(spreading bandwidth)(MHz)
$\bar{r}$	코드 하나의 전송률
$r^{d,t}$	코드 하나의 전송률에 대한 서비스 클래스 $t$ 의 하향링크 전송률의 비율(정수)
$r^{u,t}$	코드 하나의 전송률에 대한 서비스 클래스 $t$ 의 상향링크 전송률의 비율(정수)

〈표 1〉 계속

기 호	정 의
$g_i^{d,t}$	셀 $i$ 의 하향링크 타임슬롯(또는 하향링크용으로 사용되는 교차 타임슬롯)에서 서비스 클래스 $t$ 의 확산이득 (processing gain)
$g_i^{u,t}$	셀 $i$ 의 상향링크 타임슬롯(또는 상향링크용으로 사용되는 교차 타임슬롯)에서 서비스 클래스 $t$ 의 확산이득
$p_{iikk}^{d,t}$	셀 $i$ 의 서비스 클래스 $t$ 사용자 $k$ 가 셀 $i$ 기지국으로부터 수신하는 신호전력
$p_{iikl}^{d,t}$	하향링크 타임슬롯(또는 하향링크용 교차 타임슬롯)에서 셀 $i$ 의 서비스 클래스 $t$ 사용자 $k$ 가 셀 $i$ 의 서비스 클래스 $t$ 사용자 $l$ 로 인해 수신하는 신호전력( $k \neq l$ )
$p_{iikl}^{d,v}$	하향링크 타임슬롯(또는 하향링크용 교차 타임슬롯)에서 셀 $i$ 의 서비스 클래스 $t$ 사용자 $k$ 가 셀 $i$ 의 서비스 클래스 $v$ 사용자 $l$ 로 인해 수신하는 신호전력( $v \neq t$ )
$p_{ijkl}^{d,v}$	하향링크 타임슬롯(또는 셀 $i$ 와 셀 $j$ 가 하향링크용으로 사용하는 교차 타임슬롯)에서 셀 $i$ 의 서비스 클래스 $t$ 사용자 $k$ 가 셀 $j$ 의 서비스 클래스 $v$ 사용자 $l$ 로 인해 수신하는 신호전력( $i \neq j$ )
$p_{ijkl}^{u(c),v}$	셀 $i$ 에서는 하향링크용으로, 셀 $j$ 에서는 상향링크용으로 사용하는 교차 타임슬롯에서 셀 $i$ 의 서비스 클래스 $t$ 사용자 $k$ 가 셀 $j$ 의 서비스 클래스 $v$ 사용자 $l$ 로 인해 수신하는 신호전력( $i \neq j$ )
$p_{iikk}^{u,t}$	셀 $i$ 기지국이 셀 $i$ 의 서비스 클래스 $t$ 사용자 $k$ 로부터 수신하는 신호전력
$p_{iikl}^{u,t}$	상향링크 타임슬롯(또는 상향링크용 교차 타임슬롯)에서 셀 $i$ 기지국이 셀 $i$ 의 서비스 클래스 $t$ 사용자 $l$ 로 인해 수신하는 신호전력( $k \neq l$ )
$p_{iikl}^{u,v}$	상향링크 타임슬롯(또는 상향링크용 교차 타임슬롯)에서 셀 $i$ 기지국이 셀 $i$ 의 서비스 클래스 $v$ 사용자 $l$ 로 인해 수신하는 신호전력( $v \neq t$ )
$p_{ijkl}^{u,v}$	상향링크 타임슬롯(또는 셀 $i$ 와 셀 $j$ 가 상향링크용으로 사용하는 교차 타임슬롯)에서 셀 $i$ 기지국이 셀 $j$ 의 서비스 클래스 $v$ 사용자 $l$ 로 인해 수신하는 신호전력( $i \neq j$ )
$p_{ijkl}^{d(c),v}$	셀 $i$ 에서는 상향링크용으로, 셀 $j$ 에서는 하향링크용으로 사용하는 교차 타임슬롯에서 셀 $i$ 기지국이 셀 $j$ 의 서비스 클래스 $v$ 사용자 $l$ 로 인해 수신하는 신호전력( $i \neq j$ )
$\delta_{ij}^d$	셀 $i$ 의 하향링크 타임슬롯으로부터의 간섭에 대한 셀 $j$ 의 하향링크 타임슬롯으로부터의 간섭 비율의 기대값
$\delta_{ij}^u$	셀 $i$ 의 상향링크 타임슬롯으로부터의 간섭에 대한 셀 $j$ 의 상향링크 타임슬롯으로부터의 간섭 비율의 기대값
$\delta_{ij}^{d(c)}$	셀 $i$ 의 상향링크용 교차 타임슬롯으로부터의 간섭에 대한 셀 $j$ 의 하향링크용 교차 타임슬롯으로부터의 간섭 비율의 기대값
$\delta_{ij}^{u(c)}$	셀 $i$ 의 하향링크용 교차 타임슬롯으로부터의 간섭에 대한 셀 $j$ 의 상향링크용 교차 타임슬롯으로부터의 간섭 비율의 기대값

결정된다고 가정한다. 코드 한 개의 전송률을  $\bar{r}$ 라 할 때, 특정 서비스의 하향링크와 상향링크 전송률은 각각  $r^{d,t}\bar{r}$ 와  $r^{u,t}\bar{r}$ 가 된다( $r^{d,t}$ 와  $r^{u,t}$ 는 정수).

이제 CDMA/TDD 시스템의 하향링크 타임슬롯, 상향링크 타임슬롯, 교차 타임슬롯에 대한 간섭(interference)은 다음과 같이 분석할 수 있다[2, 6, 7, 15, 21].

## 2.1 하향링크 타임슬롯

모든 셀에서 하향링크용으로 사용되는 하향링크

타임슬롯을 고려하자. 하향링크 타임슬롯에서 셀  $i$ 의 서비스 클래스  $t$  사용자  $k$ 의 비트당 에너지 대 잡음 비율(bit energy-to-noise density ratio)  $\left(\frac{E_b}{N_o}\right)_{ik}^{d,t}$ 는 다음과 같이 정의된다.

$$\left(\frac{E_b}{N_o}\right)_{ik}^{d,t} = p_{iikk}^{d,t} g_i^{d,t} \left[ (L_{ik}^{d,t} - 1) p_{iikk}^{d,t} + \sum_{l(\neq k) \in S_i^t} L_{il}^{d,t} p_{iikl}^{d,t} + \sum_{v(\neq t) \in T} \sum_{l \in S_i^v} L_{il}^{d,v} p_{iikl}^{d,v} \right]^{-1} + \sum_{j \in A} \sum_{v \in T} \sum_{l \in S_j^v} L_{jl}^{d,v} p_{ijkl}^{d,v} \left\}^{-1}$$

하나의 코드에 대한 셀  $i$  기지국의 전송출력은 서비스 클래스나 사용자에게 관계없이 동일하다고 가정하면,  $p_{iikk}^{d,t} = p_{iikl}^{d,t} = p_{iikl}^{d,v}$ 이다.  $\delta_{ij}^d$ 를 셀  $i$ 의 하향링크 타임슬롯으로부터의 간섭에 대한 셀  $j$ 의 하향링크 타임슬롯으로부터의 간섭 비율의 기대값이라 하면,  $\left(\frac{E_b}{N_o}\right)_{ik}^{d,t}$ 는 다음과 같이 정의된다.

$$\left(\frac{E_b}{N_o}\right)_{ik}^{d,t} = g_i^{d,t} \left[ (L_{ik}^{d,t} - 1) + \sum_{l(\neq k) \in S_i^t} L_{il}^{d,t} + \sum_{v(\neq t) \in T} \sum_{l \in S_i^v} L_{il}^{d,v} + \sum_{j \in A_i} \delta_{ij}^d \sum_{v \in T} \sum_{l \in S_j^v} L_{jl}^{d,v} \right]^{-1}$$

이제 하향링크 타임슬롯에서 셀  $i$ 의 서비스 클래스  $t$  사용자의 비트당 에너지 대 잡음 비율  $\left(\frac{E_b}{N_o}\right)_i^{d,t}$ 는 다음과 같이 정의된다.

$$\left(\frac{E_b}{N_o}\right)_i^{d,t} = g_i^{d,t} \left[ \sum_{v \in T} L_i^{d,v} - 1 + \sum_{j \in A_i} \delta_{ij}^d \sum_{v \in T} L_j^{d,v} \right]^{-1} \quad (1)$$

## 2.2 상향링크 타임슬롯

모든 셀에서 상향링크용으로 사용되는 상향링크 타임슬롯을 고려하자. 상향링크 타임슬롯에서 셀  $i$ 의 서비스 클래스  $t$  사용자  $k$ 의 비트당 에너지 대 잡음 비율  $\left(\frac{E_b}{N_o}\right)_{ik}^{u,t}$ 는 다음과 같이 정의된다.

$$\left(\frac{E_b}{N_o}\right)_{ik}^{u,t} = p_{iikk}^{u,t} g_i^{u,t} \left[ (L_{ik}^{u,t} - 1) p_{iikk}^{u,t} + \sum_{l(\neq k) \in S_i^t} L_{il}^{u,t} p_{iikl}^{u,t} + \sum_{v(\neq t) \in T} \sum_{l \in S_i^v} L_{il}^{u,v} p_{iikl}^{u,v} + \sum_{j \in A_i} \sum_{v \in T} \sum_{l \in S_j^v} L_{jl}^{u,v} p_{ijkl}^{u,v} \right]^{-1}$$

전송출력 제어가 완전하게 이루어지면, 셀  $i$ 의 기지국이 모든 코드로부터 수신하는 신호전력은 동일하다. 즉,  $p_{iikk}^{u,t} = p_{iikl}^{u,t} = p_{iikl}^{u,v}$ 이다.  $\delta_{ij}^u$ 를 셀  $i$ 의 상

향링크 타임슬롯으로부터의 간섭에 대한 셀  $j$ 의 상향링크 타임슬롯으로부터의 간섭 비율의 기대값이라 하면,  $\left(\frac{E_b}{N_o}\right)_{ik}^{u,t}$ 는 다음과 같이 정의된다.

$$\left(\frac{E_b}{N_o}\right)_{ik}^{u,t} = g_i^{u,t} \left[ (L_{ik}^{u,t} - 1) + \sum_{l(\neq k) \in S_i^t} L_{il}^{u,t} + \sum_{v(\neq t) \in T} \sum_{l \in S_i^v} L_{il}^{u,v} + \sum_{j \in A_i} \delta_{ij}^u \sum_{v \in T} \sum_{l \in S_j^v} L_{jl}^{u,v} \right]^{-1}$$

이제 상향링크 타임슬롯에서 셀  $i$ 의 서비스 클래스  $t$  사용자의 비트당 에너지 대 잡음 비율  $\left(\frac{E_b}{N_o}\right)_i^{u,t}$ 는 다음과 같이 정의된다.

$$\left(\frac{E_b}{N_o}\right)_i^{u,t} = g_i^{u,t} \left[ \sum_{v \in T} L_i^{u,v} - 1 + \sum_{j \in A_i} \delta_{ij}^u \sum_{v \in T} L_j^{u,v} \right]^{-1} \quad (2)$$

## 2.3 교차 타임슬롯

먼저 하향링크용으로 사용되는 교차 타임슬롯을 고려하자. 하향링크용으로 사용되는 셀  $i$ 의 교차 타임슬롯에서 셀  $i$ 의 서비스 클래스  $t$  사용자  $k$ 의 비트당 에너지 대 잡음 비율  $\left(\frac{E_b}{N_o}\right)_{ik}^{d(c),t}$ 는 다음과 같이 정의된다.

$$\left(\frac{E_b}{N_o}\right)_{ik}^{d(c),t} = p_{iikk}^{d,t} g_i^{d,t} \left[ (L_{ik}^{d(c),t} - 1) p_{iikk}^{d,t} + \sum_{l(\neq k) \in S_i^t} L_{il}^{d(c),t} p_{iikl}^{d,t} + \sum_{v(\neq t) \in T} \sum_{l \in S_i^v} L_{il}^{d(c),v} p_{iikl}^{d,v} + \sum_{j \in A_i} \sum_{v \in T} \sum_{l \in S_j^v} L_{jl}^{d(c),v} p_{ijkl}^{d,v} z_j^d + \sum_{j \in A_i} \sum_{v \in T} \sum_{l \in S_j^v} L_{jl}^{u(c),v} p_{ijkl}^{u(c),v} z_j^u \right]^{-1}$$

여기서  $z_j^d$ 는 셀  $j$ 에서 교차 타임슬롯이 하향링크용으로 사용되면 1, 상향링크용으로 사용되면 0

의 값을 가지는 이진변수이고,  $z_j^u$ 는 셀  $j$ 에서 교차 타임슬롯이 상향링크용으로 사용되면 1, 하향링크용으로 사용되면 0의 값을 가지는 이진변수이다 ( $z_j^d + z_j^u = 1$ ).

$\delta_{ij}^{u(c)}$ 를 하향링크용으로 사용되는 셀  $i$ 의 교차 타임슬롯으로부터의 간섭에 대한 상향링크용으로 사용되는 셀  $j$ 의 교차 타임슬롯으로부터의 간섭 비율의 기대값이라 하면,  $\left(\frac{E_b}{N_o}\right)_{ik}^{d(c),t}$ 는 다음과 같이 정의된다.

$$\begin{aligned} \left(\frac{E_b}{N_o}\right)_{ik}^{d(c),t} &= g_i^{d,t} \left[ (L_{ik}^{d(c),t} - 1) + \sum_{l(\neq k) \in S_i^t} L_{il}^{d(c),t} \right. \\ &\quad + \sum_{v(\neq t) \in T} \sum_{l \in S_i^v} L_{il}^{d(c),v} \\ &\quad + \sum_{j \in A_i} \delta_{ij}^d z_j^d \sum_{v \in T} \sum_{l \in S_i^v} L_{jl}^{d(c),v} \\ &\quad \left. + \sum_{j \in A_i} \delta_{ij}^{u(c)} z_j^u \sum_{v \in T} \sum_{l \in S_i^v} L_{jl}^{u(c),v} \right]^{-1} \end{aligned}$$

이제 하향링크용으로 사용되는 셀  $i$ 의 교차 타임슬롯에서 셀  $i$ 의 서비스 클래스  $t$  사용자의 비트당 에너지 대 잡음 비율  $\left(\frac{E_b}{N_o}\right)_i^{d(c),t}$ 는 다음과 같이 정의된다.

$$\begin{aligned} \left(\frac{E_b}{N_o}\right)_i^{d(c),t} &= g_i^{d,t} \left[ \sum_{v \in T} L_i^{d(c),v} - 1 + \sum_{j \in A_i} \delta_{ij}^d z_j^d \sum_{v \in T} L_j^{d(c),v} \right. \\ &\quad \left. + \sum_{j \in A_i} \delta_{ij}^{u(c)} z_j^u \sum_{v \in T} L_j^{u(c),v} \right]^{-1} \quad (3) \end{aligned}$$

다음으로 상향링크용으로 사용되는 교차 타임슬롯을 고려하자. 상향링크용으로 사용되는 셀  $i$ 의 교차 타임슬롯에서 셀  $i$ 의 서비스 클래스  $t$  사용자  $k$ 의 비트당 에너지 대 잡음 비율  $\left(\frac{E_b}{N_o}\right)_{ik}^{u(c),t}$ 는 다음과 같이 정의된다.

$$\left(\frac{E_b}{N_o}\right)_{ik}^{u(c),t} = p_{iikk}^{u,t} g_i^{u,t} \left[ (L_{ik}^{u(c),t} - 1) p_{iikk}^{u,t} \right.$$

$$\begin{aligned} &\quad + \sum_{l(\neq k) \in S_i^t} L_{il}^{u(c),t} p_{iikk}^{u,t} \\ &\quad + \sum_{v(\neq t) \in T} \sum_{l \in S_i^v} L_{il}^{u(c),v} p_{iikk}^{u,v} \left. \right] \\ &\quad + \sum_{j \in A_i} \sum_{v \in T} \sum_{l \in S_i^v} L_{jl}^{d(c),v} p_{ijkl}^{d,v} z_j^d \\ &\quad + \sum_{j \in A_i} \sum_{v \in T} \sum_{l \in S_i^v} L_{jl}^{u(c),v} p_{ijkl}^{u,v} z_j^u \left. \right]^{-1} \end{aligned}$$

$\delta_{ij}^{d(c)}$ 를 상향링크용으로 사용되는 셀  $i$ 의 교차 타임슬롯으로부터의 간섭에 대한 하향링크용으로 사용되는 셀  $j$ 의 교차 타임슬롯으로부터의 간섭 비율의 기대값이라 하면,  $\left(\frac{E_b}{N_o}\right)_{ik}^{u(c),t}$ 는 다음과 같이 정의된다.

$$\begin{aligned} \left(\frac{E_b}{N_o}\right)_{ik}^{u(c),t} &= g_i^{u,t} \left[ (L_{ik}^{u(c),t} - 1) + \sum_{l(\neq k) \in S_i^t} L_{il}^{u(c),t} \right. \\ &\quad + \sum_{v(\neq t) \in T} \sum_{l \in S_i^v} L_{il}^{u(c),v} \\ &\quad + \sum_{j \in A_i} \delta_{ij}^{d(c)} z_j^d \sum_{v \in T} \sum_{l \in S_i^v} L_{jl}^{d(c),v} \\ &\quad \left. + \sum_{j \in A_i} \delta_{ij}^u z_j^u \sum_{v \in T} \sum_{l \in S_i^v} L_{jl}^{u(c),v} \right]^{-1} \end{aligned}$$

이제 상향링크용으로 사용되는 셀  $i$ 의 교차 타임슬롯에서 셀  $i$ 의 서비스 클래스  $t$  사용자의 비트당 에너지 대 잡음 비율  $\left(\frac{E_b}{N_o}\right)_i^{u(c),t}$ 는 다음과 같이 정의된다.

$$\begin{aligned} \left(\frac{E_b}{N_o}\right)_i^{u(c),t} &= g_i^{u,t} \left[ \sum_{v \in T} L_i^{u(c),v} - 1 + \sum_{j \in A_i} \delta_{ij}^{d(c)} z_j^d \sum_{v \in T} L_j^{d(c),v} \right. \\ &\quad \left. + \sum_{j \in A_i} \delta_{ij}^u z_j^u \sum_{v \in T} L_j^{u(c),v} \right]^{-1} \quad (4) \end{aligned}$$

### 3. 최적 타임슬롯 및 코드 할당 모형

여기에서는 서비스별 코드 수요와 기존에 할당된 코드의 강제종료를 고려하면서 총효율을 최대화하는 통합 타임슬롯 및 코드 할당 방법을 찾기

위한 최적 타임슬롯 및 코드 할당 모형을 정형화한다. 목적식은 서비스 클래스별로 할당된 채널(코드) 수에 따른 효용과 기존에 할당된 채널(코드)의 강제종료로 인한 효용 감소의 합으로 정의한다. 셀  $i$ 에서 서비스 클래스  $t$ 에 할당된 채널의 수가  $M_i^t$ 라 할 때, 이에 따른 효용은 효용함수  $U_i^t(\cdot)$ 를 이용하여  $\sum_t U_i^t(w^t M_i^t)$ 로 정의한다. 기존에 서비스 클래스  $t$ 에 할당된 채널 중 강제종료되는 채널의 수가  $y_i^t$ 라 할 때, 이에 따른 효용 감소는 효용함수  $V_i^t(\cdot)$ 를 이용하여  $\sum_t V_i^t(w^t y_i^t)$ 로 정의한다. 결국 목적식은  $\sum_t U_i^t(w^t M_i^t) - \sum_t V_i^t(w^t y_i^t)$ 로 정의한다. 여기서  $w^t$ 는 서비스 클래스에 관계없이 동일하게 1로 가정할 수도 있고,  $r^{d,t} + r^{u,t}$ 로 가정할 수도 있다. 채널의 수가  $M_i^t$ 일 때,  $r^{d,t} M_i^t$ 는 하향링크용 코드 수를 나타내고,  $r^{u,t} M_i^t$ 는 상향링크용 코드 수를 나타낸다. 효용함수  $U_i^t(\cdot)$ 와  $V_i^t(\cdot)$ 는 서비스 클래스별 우선 순위를 감안하여 결정한다. 본 논문에서는 효용함수  $U_i^t(\cdot)$ 를 순증가 오목함수(strictly increasing concave function)로,  $V_i^t(\cdot)$ 는 순감소 볼록함수(strictly decreasing convex function)로 가정한다.

한편, 적절한 전송품질을 유지하기 위해서 하향링크 타임슬롯, 상향링크 타임슬롯, 하향링크용 교차 타임슬롯, 상향링크용 교차 타임슬롯은 각각 다음의 제약식을 만족해야 한다.

$$\left(\frac{E_b}{N_o}\right)_i^{d,t} \geq \gamma_i^{d,t}, \forall t \quad (5)$$

$$\left(\frac{E_b}{N_o}\right)_i^{u,t} \geq \gamma_i^{u,t}, \forall t \quad (6)$$

$$\left(\frac{E_b}{N_o}\right)_i^{d(c),t} \geq \gamma_i^{d,c,t}, \forall t \quad (7)$$

$$\left(\frac{E_b}{N_o}\right)_i^{u(c),t} \geq \gamma_i^{u,c,t}, \forall t \quad (8)$$

식 (5)~식 (8)은 식 (1)~식 (4)를 이용하여 다

음과 같이 바꿀 수 있다.

$$\sum_{v \in T} L_i^{d,v} + \sum_{j \in A_i} \delta_{ij}^d \sum_{v \in T} L_j^{d,v} \leq \min\{\alpha_i^{d,t}, t \in T\} \quad (9)$$

$$\sum_{v \in T} L_i^{u,v} + \sum_{j \in A_i} \delta_{ij}^u \sum_{v \in T} L_j^{u,v} \leq \min\{\alpha_i^{u,t}, t \in T\} \quad (10)$$

$$\sum_{v \in T} L_i^{d(c),v} + \sum_{j \in A_i} \delta_{ij}^d z_j^d \sum_{v \in T} L_j^{d(c),v} + \sum_{j \in A_i} \delta_{ij}^{u(c)} z_j^u \sum_{v \in T} L_j^{u(c),v} \leq \min\{\alpha_i^{d,c,t}, t \in T\} \quad (11)$$

$$\sum_{v \in T} L_i^{u(c),v} + \sum_{j \in A_i} \delta_{ij}^{u(c)} z_j^u \sum_{v \in T} L_j^{u(c),v} + \sum_{j \in A_i} \delta_{ij}^d z_j^d \sum_{v \in T} L_j^{d(c),v} \leq \min\{\alpha_i^{u,c,t}, t \in T\} \quad (12)$$

여기서  $\alpha_i^{d,t} = 1 + g_i^{d,t} / \gamma_i^{d,t}$ ,  $\alpha_i^{u,t} = 1 + g_i^{u,t} / \gamma_i^{u,t}$ 이다.  $L_i^{d,v}$ 는 정수이므로 식 (9)는 다음과 같이 변형될 수 있다.

$$\sum_{v \in T} L_i^{d,v} \leq \beta_i^d \quad (13)$$

여기서  $\beta_i^d = \left\lfloor \min\{\alpha_i^{d,t}, t \in T\} - \sum_{j \in A_i} \delta_{ij}^d \sum_{v \in T} L_j^{d,v} \right\rfloor$ 이다.  $\sum_{j \in A_i} \delta_{ij}^d \sum_{v \in T} L_j^{d,v}$ 는 셀  $i$ 의 하향링크 타임슬롯 간섭에 대한 인접셀 하향링크 타임슬롯의 총간섭의 비율을 나타낸다. 각 셀별로 분산해서 타임슬롯 할당 방법을 결정하고, 이 값을 추정해서 사용한다고 가정한다. 결국,  $\beta_i^d$ 는 한 개의 하향링크 타임슬롯으로 실제 할당 가능한 코드의 수를 나타내고, 주어진 값으로 보고 분석하면 된다.

하향링크 타임슬롯의 코드는 서비스 클래스에 구분 없이 할당이 가능하므로 아래의 식 (14)를 만족하는  $M_i^{d,v}$ 가 존재하면, 각 하향링크 타임슬롯에 대한 식 (13)을 만족하는  $L_i^{d,v}$ 도 존재한다. 참고로  $L_i^{d,v}$ 는 모든 하향링크 타임슬롯에서 동일할 필요는 없다.

$$\sum_{v \in T} r^{d,v} M_i^{d,v} \leq \beta_i^d N^d \quad (14)$$

여기서  $\beta_i^d N^d$ 는 하향링크 타임슬롯을 이용하여 할



당 가능한 하향링크용 코드 수를 나타낸다.

마찬가지로 식 (10)~식 (12)로부터 다음의 식 (15)~식 (17)을 유도할 수 있다.

$$\sum_{v \in T} r^{u,v} M_i^{u,v} \leq \beta_i^u N^u \quad (15)$$

$$\sum_{v \in T} r^{d,v} M_i^{d(c),v} \leq \beta_i^{d(c)} N_i^{d(c)} \quad (16)$$

$$\sum_{v \in T} r^{u,v} M_i^{u(c),v} \leq \beta_i^{u(c)} N_i^{u(c)} \quad (17)$$

$$\begin{aligned} \beta_i^u &= \left[ \min\{\alpha_i^{u,t}, t \in T\} - \sum_{j \in A_i} \delta_{ij}^u \sum_{v \in T} L_j^{u,v} \right], \\ \beta_i^{d(c)} &= \left[ \min\{\alpha_i^{d,t}, t \in T\} - \sum_{j \in A_i} \delta_{ij}^d z_j^d \sum_{v \in T} L_j^{d(c),v} \right. \\ &\quad \left. - \sum_{j \in A_i} \delta_{ij}^{u(c)} z_j^u \sum_{v \in T} L_j^{u(c),v} \right], \\ \beta_i^{u(c)} &= \left[ \min\{\alpha_i^{u,t}, t \in T\} - \sum_{j \in A_i} \delta_{ij}^{d(c)} z_j^d \sum_{v \in T} L_j^{d(c),v} \right. \\ &\quad \left. - \sum_{j \in A_i} \delta_{ij}^u z_j^u \sum_{v \in T} L_j^{u(c),v} \right] \end{aligned}$$

이다.  $\beta_i^u$ ,  $\beta_i^{d(c)}$ ,  $\beta_i^{u(c)}$  등도  $\beta_i^d$ 와 마찬가지로 주어진 값으로 보고 분석하면 된다.  $\beta_i^u N^u$ 는 상향링크 타임슬롯을 이용하여 할당 가능한 상향링크용 코드 수를 나타내고,  $\beta_i^{d(c)} N_i^{d(c)}$ 와  $\beta_i^{u(c)} N_i^{u(c)}$ 는 각각 교차 타임슬롯을 이용하여 할당 가능한 하향링크용 코드 수와 상향링크용 코드 수를 나타낸다.

$N^d$ 와  $N^u$ 가  $\overline{N^d}$ 와  $\overline{N^u}$ 로 주어졌을 때, 셀  $i$ 에서의 최적 타임슬롯 및 코드 할당 모형은 다음과 같이 정형화된다. 전체 타임슬롯 중 일부 타임슬롯은 모든 셀에서 하향링크 타임슬롯 또는 상향링크 타임슬롯으로 미리 정해진다. 물론  $\overline{N^d} = \overline{N^u} = 0$ 일 수도 있다.

(ITCAM<sub>0</sub>)

$$\begin{aligned} & \text{Max} \sum_t U_i^t(w^t M_i^t) - \sum_t V_i^t(w^t y_i^t) \\ & \text{s.t.} \sum_{t \in T} r^{d,t} M_i^{d,t} \leq \beta_i^d \overline{N^d} \\ & \sum_{t \in T} r^{u,t} M_i^{u,t} \leq \beta_i^u \overline{N^u} \\ & \sum_{t \in T} r^{d,t} M_i^{d(c),t} \leq \beta_i^{d(c)} N_i^{d(c)} \\ & \sum_{t \in T} r^{u,t} M_i^{u(c),t} \leq \beta_i^{u(c)} N_i^{u(c)} \end{aligned}$$

$$M_i^t \leq M_i^{d,t} + M_i^{d(c),t}, \forall t$$

$$M_i^t \leq M_i^{u,t} + M_i^{u(c),t}, \forall t$$

$$M_i^t - \overline{M_i^t} = x_i^t - y_i^t, \forall t$$

$$M_i^{tL} \leq M_i^t \leq \widehat{M_i^t}, \forall t$$

$$\overline{N^d} + \overline{N^u} + N_i^{d(c)} + N_i^{u(c)} = N$$

$$N_i^{d(c)}, N_i^{u(c)}, M_i^t, M_i^{d,t}, M_i^{u,t},$$

$$M_i^{d(c),t}, M_i^{u(c),t} \in Z_+ \cup \{0\}.$$

여기서  $\overline{M_i^t}$ 는 셀  $i$ 에서 서비스 클래스  $t$ 에 현재 할당된 채널 수를 나타낸다.  $M_i^{tL}$ 은 셀  $i$ 에서 서비스 클래스  $t$ 에 할당되어야 하는 최소한의 채널 수를 나타내고,  $\widehat{M_i^t}$ 는 셀  $i$ 에서 서비스 클래스  $t$ 에 필요한 채널 수를 나타낸다.  $\widehat{M_i^t}$ 은 셀  $i$ 에서 서비스 클래스  $t$ 에 평균적으로 필요한 채널 수를 예측한 값이거나, 현재 사용 중인 채널 수에 신규 채널 수요를 더한 값이다. 여기서 신규 채널 수요는 새로 도착한 호들에 대한 채널 또는 이전에 도착하여 대기하고 있는 호들을 위한 채널이다. 지연에 민감하지 않은 서비스의 경우 전용 채널이 할당되지 않고 채널을 공유하면서 경쟁방식으로 전송한다면, 전체 필요한 전송량을 감안한 필요 채널 수 관점에서 신규로 필요한 채널의 수를 결정할 수 있다. 시스템에 따라 다르지만 하나의 프레임의 길이는 10ms이고, 지연에 민감한 서비스의 경우에 50ms(5 프레임) 정도의 짧기는 하지만 프레임 길이 보다는 좀 더 긴 액세스 지연(access delay)이 허용되기도 한다[8, 23].

교차 타임슬롯의 코드를 상향링크 타임슬롯 또는 하향링크 타임슬롯의 코드와 구분 없이 사용할 수 있다고 가정하면, 위 모형은 다음과 같이 변형된다.

$$\begin{aligned} & \text{Max} \sum_t U_i^t(w^t M_i^t) - \sum_t V_i^t(w^t y_i^t) \\ & \text{s.t.} \sum_{t \in T} r^{d,t} M_i^t \leq \beta_i^d \overline{N^d} + \beta_i^{d(c)} N_i^{d(c)} \\ & \sum_{t \in T} r^{u,t} M_i^t \leq \beta_i^u \overline{N^u} + \beta_i^{u(c)} N_i^{u(c)} \end{aligned}$$

$$\begin{aligned}
M_i^t - \overline{M}_i^t &= x_i^t - y_i^t, \forall t \\
M_i^{tL} &\leq M_i^t \leq \widehat{M}_i^t, \forall t \\
\overline{N}^d + \overline{N}^u + N_i^{d(c)} + N_i^{u(c)} &= N \\
N_i^{d(c)}, N_i^{u(c)}, M_i^t &\in \mathbb{Z}_+ \cup \{0\}.
\end{aligned}$$

이제  $C_i^t = M_i^t - M_i^{tL}$ 라 하면, 위 모형은 다음과 같이 변형된다.

(ITCAMr)

$$\begin{aligned}
& \text{Max} \sum_t \overline{U}_i^t(C_i^t) \\
& \text{s.t.} \sum_{t \in T} r^{d,t} C_i^t \leq \beta_i^d \overline{N}^d + \beta_i^{d(c)} N_i^{d(c)} \\
& \quad - \sum_{t \in T} r^{d,t} M_i^{tL} \\
& \sum_{t \in T} r^{u,t} C_i^t \leq \beta_i^u \overline{N}^u + \beta_i^{u(c)} N_i^{u(c)} \\
& \quad - \sum_{t \in T} r^{u,t} M_i^{tL} \\
& C_i^t \leq \widehat{M}_i^t - M_i^{tL}, \forall t \\
& \overline{N}^d + \overline{N}^u + N_i^{d(c)} + N_i^{u(c)} = N \\
& N_i^{d(c)}, N_i^{u(c)}, C_i^t \in \mathbb{Z}_+ \cup \{0\}
\end{aligned}$$

여기서

$$\overline{U}_i^t(C_i^t) = \begin{cases} U_i^t(w^t(C_i^t + M_i^{tL})) \\ \quad - V_i^t(w^t(\widehat{M}_i^t - C_i^t - M_i^{tL})), \\ \text{if } C_i^t \leq \widehat{M}_i^t - M_i^{tL} \\ U_i^t(w^t(C_i^t + M_i^{tL})), \\ \text{if } \widehat{M}_i^t - M_i^{tL} < C_i^t \leq \widehat{M}_i^t - M_i^{tL} \end{cases}$$

이다.  $U_i^t(\cdot)$ 가 순증가 오목함수이고  $V_i^t(\cdot)$ 가 순감소 볼록함수이므로,  $\overline{U}_i^t(\cdot)$ 는 순증가 오목함수이다.

#### 4. 실시간 통합 타임슬롯 및 코드 할당 체계

(ITCAMr)은  $N_i^{d(c)}$ 와  $N_i^{u(c)}$ 가 주어지면 보다 쉽게 해결할 수 있는 문제가 된다. 즉,  $N_i^{d(c)}$ 와  $N_i^{u(c)}$

가  $\overline{N}_i^{d(c)}$ 와  $\overline{N}_i^{u(c)}$ 로 주어질 경우 (ITCAMr)은 다음과 같이 단순화된 모형 (ITCAMs)로 변형된다.

(ITCAMs)

$$\begin{aligned}
& \text{Max} \sum_t \overline{U}_i^t(C_i^t) \\
& \text{s.t.} \sum_{t \in T} r^{d,t} C_i^t \leq \overline{\beta}_i^d \\
& \sum_{t \in T} r^{u,t} C_i^t \leq \overline{\beta}_i^u \\
& C_i^t \leq \widehat{M}_i^t - M_i^{tL}, \forall t \\
& C_i^t \in \mathbb{Z}_+ \cup \{0\}.
\end{aligned}$$

여기서  $\overline{\beta}_i^d = \beta_i^d \overline{N}^d + \beta_i^{d(c)} \overline{N}_i^{d(c)} - \sum_{t \in T} r^{d,t} M_i^{tL}$ ,  $\overline{\beta}_i^u = \beta_i^u \overline{N}^u + \beta_i^{u(c)} \overline{N}_i^{u(c)} - \sum_{t \in T} r^{u,t} M_i^{tL}$ 이다. 결국 (ITCAMs)의 최적 목적식 값을 가장 크게 하는  $N_i^{d(c)}$ 와  $N_i^{u(c)}$ 을 구하면 (ITCAMr)의 최적해를 구하게 된다.

따라서 최적 타임슬롯 및 코드 할당 모형 (ITCAMr)은 두 개의 프로시저를 통해 해결한다. 즉, 주어진  $N_i^{d(c)}$ 와  $N_i^{u(c)}$  하에서 (ITCAMs)를 해결하는 프로시저와,  $N_i^{d(c)}$ 와  $N_i^{u(c)}$ 의 갱신을 통해 최적의  $N_i^{d(c)}$ 와  $N_i^{u(c)}$ 을 구하는 프로시저를 이용하여 해결한다.

#### 4.1 (ITCAMs) 해결 프로시저

우선 (ITCAMs)를 해결하기 위해 라그랑지안 이완 기법(Lagrangian relaxation technique)을 적용한다. 라그랑지 승수(Lagrange multiplier)  $\pi = (\pi^d, \pi^u)$ 로 제약식  $\sum_{t \in T} r^{d,t} C_i^t \leq \overline{\beta}_i^d$ 와  $\sum_{t \in T} r^{u,t} C_i^t \leq \overline{\beta}_i^u$ 를 이완하여 다음의 라그랑지안 이완 문제 (LITCAMs)를 만든다.

(LITCAMs)

$$\begin{aligned}
L(\pi) &= \text{Max} \sum_t \overline{U}_i^t(C_i^t) + \pi^d (\overline{\beta}_i^d - \sum_{t \in T} r^{d,t} C_i^t) \\
& \quad + \pi^u (\overline{\beta}_i^u - \sum_{t \in T} r^{u,t} C_i^t)
\end{aligned}$$

각  $\pi \geq 0$ 에 대해서  $L(\pi)$ 는 (ITCAMS)의 최적 목적식 값에 대한 상한(upper bound)이 된다[11].  $L(\pi)$ 를 최소화하는  $\hat{\pi}$ 을 구하면 가장 작은 상한을 구할 수 있다. 즉,  $L(\hat{\pi}) = \min_{\pi \geq 0} L(\pi)$ 이다.

(LITCAMS)는 아래의 각 서비스 클래스  $t$ 에 대한 하위 문제로 나누어서 풀면 된다.

$$\begin{aligned} & \text{Max } \overline{U}_i^t(C_i^t) - (\pi^d r^{d,t} + \pi^u r^{u,t}) C_i^t \\ & \text{s.t. } C_i^t \leq \widehat{M}_i^t - M_i^{tL} \\ & C_i^t \in Z_+ \cup \{0\} \end{aligned}$$

$\overline{U}_i^t(\cdot)$ 가 순증가 오목함수이므로 각 서비스 클래스  $t$ 에 대한 하위 문제의 최적해  $C_i^{t*}$  다음과 같이 구할 수 있다.

$$C_i^{t*} = \min \left\{ \widehat{M}_i^t - M_i^{tL}, \min \left\{ k \mid \Delta \overline{U}_i^t(k) - (\pi^d r^{d,t} + \pi^u r^{u,t}) \leq 0 \right\} \right\}$$

여기서  $\Delta \overline{U}_i^t(k) = \overline{U}_i^t(w^t(k+1) + w^t M_i^{tL}) - \overline{U}_i^t(w^t k + w^t M_i^{tL})$ 이다.

$L(\pi)$ 를 최소화하는  $\pi$ 를 구하기 위해 서브그라디언트 최적화 기법(subgradient optimization technique)을 사용한다. 주어진  $\pi_m$  하에서의 (LITCAMS)의 최적해가  $C_i^{t*}$  일 때,  $\pi_m$  하에서 서브그라디언트  $\gamma_m$ 는 다음과 같다.

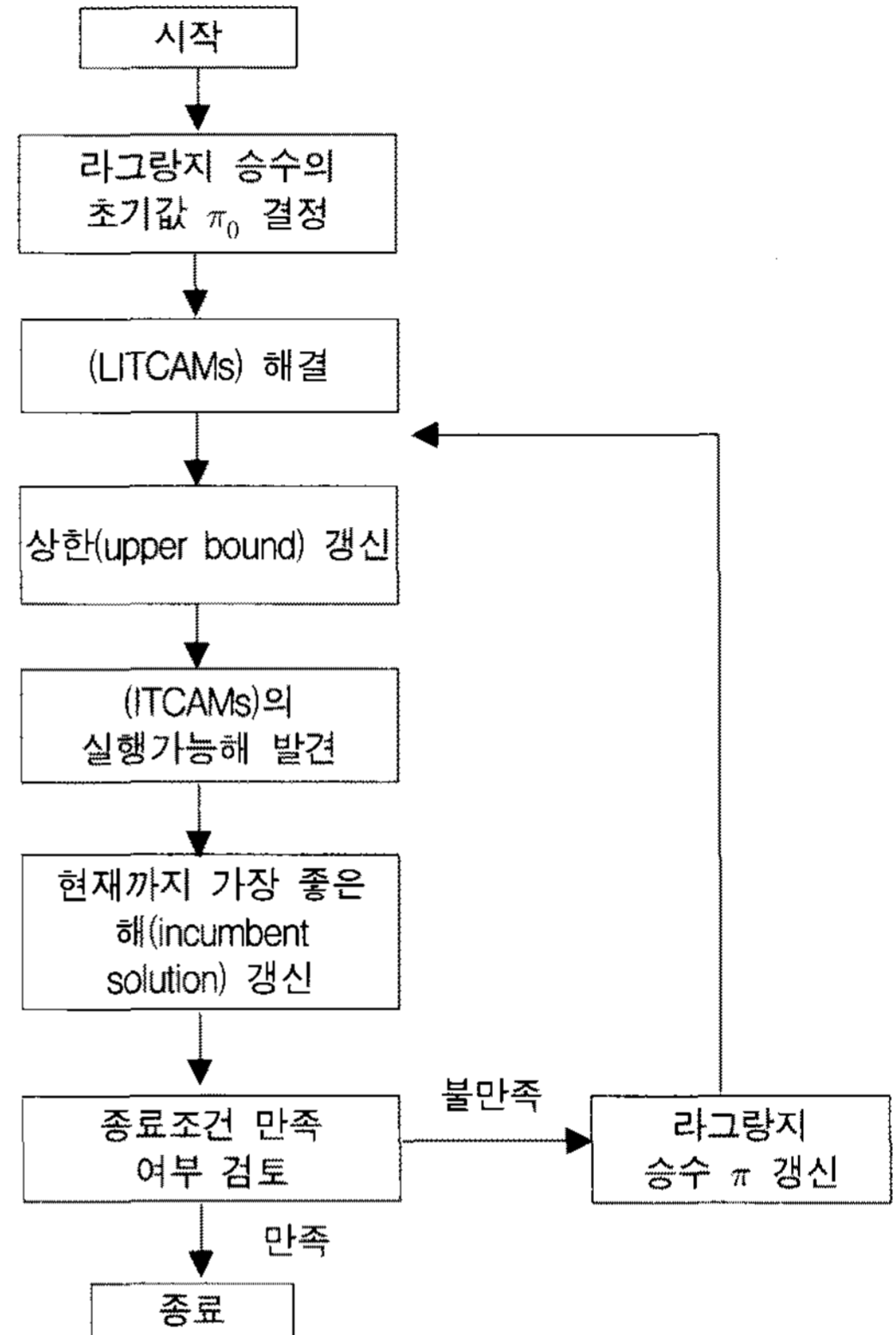
$$\gamma_m = \left( \overline{\beta}_i^d - \sum_t r^{d,t} C_i^{t*}, \overline{\beta}_i^u - \sum_t r^{u,t} C_i^{t*} \right)$$

한편, 라그랑지 승수  $\pi$ 는 다음과 같이 갱신된다.

$$\pi_{m+1} = \max \{0, \pi_m - s_m \gamma_m\}$$

여기서  $s_m$ 는 stepsize를 나타낸다. Polyak이 제시한 stepsize 갱신 프로시저가 널리 활용되고 있고 [14], 또한 이 방법을 개선한 많은 stepsize 갱신 프로시저가 제시되었다[14, 18].

라그랑지안 이완 기법과 서브그라디언트 최적화 기법을 이용한 (ITCAMS) 해결 프로시저의 흐름도는 <그림 1>과 같다.



<그림 1> (ITCAMS) 해결 프로시저의 흐름도

라그랑지 승수의 초기값  $\pi_0$ 는 다음과 같이 구한다. 이용 가능한 코드를 전부 다 사용하고 있지 않으면 프로시저 (Find\_Pi)를 이용하여 구한다. 이용 가능한 코드를 전부 다 사용하고 있으면 다음의 식을 이용하여 구한다.

$$\overline{U}_i^{t'}(\overline{M}_i^t) = \pi^d r^{d,t} + \pi^u r^{u,t}, \forall t$$

여기서  $\overline{U}_i^{t'}(\overline{M}_i^t)$ 는 1차 미분값을 나타낸다. 다만  $\overline{U}_i^{t'}(\cdot)$ 가 미분불가능일 경우에는 서브그라디언트를 이용한다. 위의 식을 동시에 만족하는  $\pi$ 는 일반적으로 없기 때문에, 최소자승법을 이용하여 구한

다. 결국 초기값  $\pi_0$ 는 다음과 같이 구해진다.

$$\begin{bmatrix} \pi_0^d \\ \pi_0^u \end{bmatrix} = \begin{bmatrix} \sum_t (r^{d,t})^2 & \sum_t r^{d,t} r^{u,t} \\ \sum_t r^{d,t} r^{u,t} & \sum_t (r^{u,t})^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_t r^{d,t} \overline{U_i^{t'}}(\overline{M_i^t}) \\ \sum_t r^{u,t} \overline{U_i^{t'}}(\overline{M_i^t}) \end{bmatrix}$$

#### Procedure Find\_Pi

Step 1: Initialize  $\pi_L^d, \pi_U^d, \pi_L^u, \pi_U^u$ .

Step 2: Set  $\pi_0^d = (\pi_L^d + \pi_U^d)/2, \pi_0^u = (\pi_L^u + \pi_U^u)/2$ .

$$C_i^t = \min \left\{ \widehat{M}_i^t - M_i^{tL}, \min \left\{ k \left| \Delta U_i^t(k) - (\pi_0^d r^{d,t} + \pi_0^u r^{u,t}) \right| \leq 0 \right\} \right\}$$

Step 3: If  $(\sum_t r^{d,t} C_i^t > \overline{\beta}_i^d)$ , set  $\pi_L^d = \pi_0^d$ .

$$\pi_0^d = (\pi_0^d + \pi_U^d)/2,$$

else if  $(\sum_t r^{d,t} C_i^t < \overline{\beta}_i^d)$ , set  $\pi_U^d = \pi_0^d$ ,

$$\pi_0^d = (\pi_0^d + \pi_L^d)/2.$$

If  $(\sum_t r^{u,t} C_i^t > \overline{\beta}_i^u)$ , set  $\pi_L^u = \pi_0^u$ ,

$$\pi_0^u = (\pi_0^u + \pi_U^u)/2,$$

elseif  $(\sum_t r^{u,t} C_i^t < \overline{\beta}_i^u)$ , set  $\pi_U^u = \pi_0^u$ ,

$$\pi_0^u = (\pi_0^u + \pi_L^u)/2.$$

Step 4: If  $(\sum_t r^{d,t} C_i^t = \overline{\beta}_i^d, \sum_t r^{u,t} C_i^t = \overline{\beta}_i^u)$  or

$(\pi_U^d - \pi_L^d \leq \epsilon_\pi, \pi_U^u - \pi_L^u \leq \epsilon_\pi)$ , terminate,  
otherwise, go to Step 2.

(LITCAMs)의 최적해  $C_i^{t*}$ 를 이용하여 (ITCAMs)의 실행가능해를 구하게 되는데, 이 때 프로시저 (Find\_Feasible)을 이용한다. Step 2는 (LITCAMs)의 최적해에 따라 코드를 할당할 경우, 이용가능한 하향링크용 코드 수  $\overline{\beta}_i^d$  또는 상향링크용 코드 수  $\overline{\beta}_i^u$ 를 초과하는 경우 수행된다. 여기에서는 총효용의 감소가 상대적으로 적도록 각 데이터 클래스에 할당된 코드 수를 줄여서 실행가능해를 구한다. Step 3에서는 이용가능한 하향링크용 코드 수  $\overline{\beta}_i^d$  또는 상향링크용 코드 수  $\overline{\beta}_i^u$ 에 여유가 있는 경우, 총효용이 상대적으로 많이 증가할 수 있도록 각 데이터 클래스에 할당된 코드 수를 증가시킨다.

#### Procedure Find\_Feasible

Step 1: If  $(\sum_t r^{d,t} C_i^{t*} > \overline{\beta}_i^d)$  or  $(\sum_t r^{u,t} C_i^{t*} > \overline{\beta}_i^u)$ ,

go to Step 2,

otherwise, go to Step 3.

Step 2: Repeat the followings until  $\sum_t r^{d,t} C_i^{t*} \leq \overline{\beta}_i^d$

and  $\sum_t r^{u,t} C_i^{t*} \leq \overline{\beta}_i^u$ .

$$\hat{t} = \operatorname{argmin}_t \left\{ \frac{\Delta U_i^t(C_i^{t*}-1)}{s^d + s^u} \mid C_i^{t*} > 0 \right\}$$

where  $s^d = r^{d,t} / (\sum_t r^{d,t} C_i^{t*} - \overline{\beta}_i^d)$

if  $\sum_t r^{d,t} C_i^{t*} > \overline{\beta}_i^d$ ,

otherwise,  $s^d = 0$

$s^u = r^{u,t} / (\sum_t r^{u,t} C_i^{t*} - \overline{\beta}_i^u)$

if  $\sum_t r^{u,t} C_i^{t*} > \overline{\beta}_i^u$ ,

otherwise,  $s^u = 0$ .

$$C_i^{\hat{t}*} = C_i^{\hat{t}*} - 1.$$

Step 3: Repeat until the following  $\hat{t}$  exists.

$$\hat{t} = \operatorname{argmax}_t \left\{ \frac{\Delta U_i^t(C_i^{t*})}{r^{d,t}/\overline{\beta}_i^d + r^{u,t}/\overline{\beta}_i^u} \mid C_i^{t*} < \widehat{M}_i^t - M_i^{tL}, \right.$$

$$\left. \sum_v r^{d,v} C_i^{v*} + r^{d,t} \leq \overline{\beta}_i^d, \sum_v r^{u,v} C_i^{v*} + r^{u,t} \leq \overline{\beta}_i^u \right\}.$$

$$C_i^{\hat{t}*} = C_i^{\hat{t}*} + 1.$$

<그림 1>에서 종료조건으로는 다음의 세 가지를 사용한다.

$$T1 : \gamma_m = 0$$

$$T2 : \frac{UB-LB}{LB} \leq \epsilon$$

$$T3 : \text{No. of Iterations} \geq TN$$

종료조건 T1의 경우,  $\gamma_m = 0$ 이면 (LITCAMs)의 최적해가 (ITCAMs)의 최적해이므로 종료한다. 종료조건 T2에서,  $LB$ 는 지금까지 구한 (ITCAMs)의 가장 좋은 목적식 값을 의미하고,  $UB$ 는 지금까지 구한 (ITCAMs)에 대한 가장 작은 상한을 나타낸다. 한편 라그랑지 승수는 Kim et al.[18]이 제시한 방법을 활용하여 갱신한다.

4.2 (ITCAMr) 해결 프로시저( $N_i^{d(c)}$  및  $N_i^{u(c)}$ 

## 갱신 프로시저)

모든 타임슬롯이 사용되고 있으면 현재의  $N_i^{d(c)}$ 와  $N_i^{u(c)}$ 를 초기값으로 사용하고, 그렇지 않으면  $N_i^{d(c)}$ 와  $N_i^{u(c)}$ 의 초기값은 다음의 식이 성립하도록 정한다.

$$\frac{\beta_i^d \bar{N}^d + \beta_i^{d(c)} N_i^{d(c)}}{\beta_i^u \bar{N}^u + \beta_i^{u(c)} N_i^{u(c)}} = \frac{\sum_t r^{d,t} \Delta \bar{U}_i^t(0)}{\sum_t r^{u,t} \Delta \bar{U}_i^t(0)}$$

결국  $N_i^{d(c)}$ 와  $N_i^{u(c)}$ 의 초기값은 다음과 같다.

$$N_i^{d(c)} = \frac{(N - \bar{N}^d - \bar{N}^u) \beta_i^{u(c)} \sum_t r^{d,t} \Delta \bar{U}_i^t(0) + \bar{N}^u \beta_i^u \sum_t r^{d,t} \Delta \bar{U}_i^t(0) + \bar{N}^d \beta_i^d \sum_t r^{u,t} \Delta \bar{U}_i^t(0)}{\beta_i^{u(c)} \sum_t r^{d,t} \Delta \bar{U}_i^t(0) + \beta_i^{d(c)} \sum_t r^{u,t} \Delta \bar{U}_i^t(0)}$$

$$N_i^{u(c)} = (N - \bar{N}^d - \bar{N}^u) - N_i^{d(c)}$$

다음으로  $N_i^{d(c)}$ 와  $N_i^{u(c)}$ 의 갱신은 프로시저 (Find\_Timeslot)을 통해 이루어진다. 프로시저에서  $V_{N_i^{d(c)}, N_i^{u(c)}}$ 는 주어진  $N_i^{d(c)}$ 와  $N_i^{u(c)}$  하에서 (ITCAMs) 해결 프로시저를 이용하여 구한 (ITCAMs)의 최적 목적식 값을 나타낸다.

## Procedure Find\_Timeslot

Step1: Set  $\bar{x}^d$  and  $\bar{x}^u$  be the initial values of  $N_i^{d(c)}$  and  $N_i^{u(c)}$ .

Solve (ITCAMs) with  $\bar{x}^d$  and  $\bar{x}^u$ .

If  $(\beta_i^d - \sum_t r^{d,t} C_i^{t*} < \beta_i^u - \sum_t r^{u,t} C_i^{t*})$ ,

go to Setp2, otherwise, go to Setp3.

Step2: Set  $x^d = \bar{x}^d$  and  $x^u = \bar{x}^u$ .

Step2.1: Set  $x^d = x^d + 1$  and  $x^u = x^u - 1$ .

Solve (ITCAMs) with  $x^d$  and  $x^u$ .

Step2.2: If  $(V_{x^d, x^u} > V_{x^d-1, x^u+1})$ , go to Setp2.1,

otherwise, go to Setp2.3.

Step2.3: Solve (ITCAMs) with  $\bar{x}^d + 1$  and  $\bar{x}^u - 1$ .  
Terminate with the best solution found.

Step3: Set  $x^d = \bar{x}^d$  and  $x^u = \bar{x}^u$ .

Step3.1: Set  $x^d = x^d - 1$  and  $x^u = x^u + 1$ .

Solve (ITCAMs) with  $x^d$  and  $x^u$ .

Step3.2: If  $(V_{x^d, x^u} > V_{x^d+1, x^u-1})$ , go to Step3.1, otherwise, go to Setp3.3

Step3.3: Solve (ITCAMs) with  $\bar{x}^d + 1$  and  $\bar{x}^u - 1$ .  
Terminate with the best solution found.

## 5. 실험 결과

여기에서는 앞에서 제시한 실시간 통합 타임슬롯 및 코드 할당 체계의 성능을 실험을 통해 분석한다. TDD 프레임당 타임슬롯 수  $N$ 은 16으로, 주파수 대역폭  $W$ 는 10MHz로, 서비스 클래스의 집합  $T = \{1, 2, 3, \dots, 10\}$ 로 가정하였다.  $\bar{r}$ 는 10kbps로 가정하였고,  $r^{d,t} = (1, 2, 4, 2, 4, 1, 0, 1, 2, 4)$ ,  $r^{u,t} = (1, 1, 1, 2, 2, 1, 1, 0, 0, 0)$ 로 가정하였다.  $\gamma_i^{d,t} = \gamma_i^{u,t} = 5\text{dB}$ 로 설정하였다.  $\bar{N}^d = \bar{N}^u = 4$ 로 가정하였고, 효율함수는 다음과 같이 가정하였다.

$$U_i^t(l) = a_i^t \ln(1+l),$$

$$\text{단 } a_i^t = (1, 1, 1, 1, 1, 0.1, 0.1, 0.1, 0.1, 0.1),$$

$$V_i^t(l) = b_i^t \times l,$$

$$\text{단 } b_i^t = (0.01, 0.01, 0.01, 0.01, 0.01, 0.0001,$$

$$0.0001, 0.0001, 0.0001, 0.0001).$$

또한,  $\beta_i^d$ ,  $\beta_i^u$ ,  $\beta_i^{d(c)}$ ,  $\beta_i^{u(c)}$ 는 다음과 같이 가정하였다.

$$\beta_i^d = \lfloor p^d \times \min\{\alpha_i^{d,t}, t \in T\} \rfloor.$$

$$\beta_i^u = \lfloor p^u \times \min\{\alpha_i^{u,t}, t \in T\} \rfloor.$$

$$\beta_i^{d(c)} = \lfloor p^{d(c)} \times \min\{\alpha_i^{d,t}, t \in T\} \rfloor.$$

$$\beta_i^{u(c)} = \lfloor p^{u(c)} \times \min\{\alpha_i^{u,t}, t \in T\} \rfloor.$$

여기서  $p^d$ ,  $p^u$ ,  $p^{d(c)}$ ,  $p^{u(c)}$ 는 0.5, 0.6, 0.7, 0.8 또는 0.9로 가정하였다.  $M_i^{tL} = 0$ 로 가정하였고, 종료조건에서  $\epsilon = 10^{-4}$ ,  $TN = 100$ 로 두었다.

5.1 (ITCAMs) 해결 프로시저 실험 결과

<표 2>~<표 5>와 <그림 2>에는 라그랑지안 이완 기법과 서브그래디언트 최적화 기법을 이용하는 (ITCAMs) 해결 프로시저에 대한 실험 결과가 제시되어 있다. 여기서  $p^d = p^u = p^{d(c)} = p^{u(c)} = 0.8$ 로 가정하였다. <표 2>, <표 3>, <그림 2> 등에는  $\overline{M}_i^t = 0, \widehat{M}_i^t = 500/(r^{d,t} + r^{u,t})$ 로 가정한 경우의 결과가 정리되어 있다. <표 2>에는  $w^t = 1, \forall t$ 인 경우,  $\overline{N}_i^{d(c)}$ 와  $\overline{N}_i^{u(c)}$  값에 따른 총효용, 상한, GAP 등이 제시되어 있고, <표 3>에는  $w^t = r^{d,t} + r^{u,t}, \forall t$ 인 경우,  $\overline{N}_i^{d(c)}$ 와  $\overline{N}_i^{u(c)}$  값에 따른 총효용, 상한, GAP 등이 제시되어 있다. <표 2>와 <표 3>에 제시된 GAP을 통해 제시한 (ITCAMs) 해결 프로시저가 최적 목적식 값에 가까운 좋은 해를 제공한다는 것을 알 수 있다. <그림 2>는 하향링크용 타임슬롯 수  $\overline{N}_i^d + \overline{N}_i^{d(c)}$ 가 증가함에 따라 총효용이 어떻게 변하는지를 보여주고 있다. 하향링크용 타임슬롯 수가 10개가 될 때까지는 총효용이 증가하고, 그 이후에는 총효용이 감소한다. 결국 하향링크용 타임슬롯 수가 10개일 때 총효용이 가장 크다.

<표 2> (ITCAMs) 해결 프로시저 실험 결과

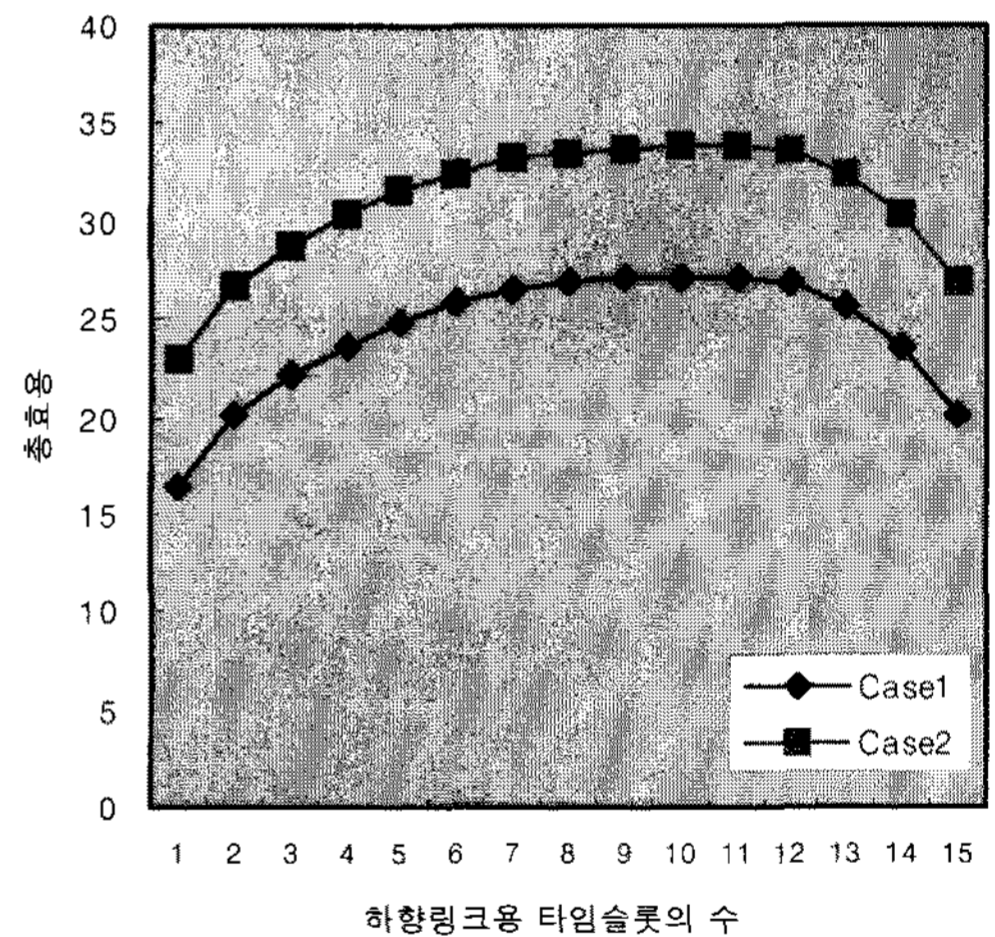
$(w^t = 1, \overline{M}_i^t = 0, \widehat{M}_i^t = 500/(r^{d,t} + r^{u,t}))$

$\overline{N}_i^{d(c)}$	$\overline{N}_i^{u(c)}$	상한	총효용	GAP*
0	8	23.68591	23.68301	0.00012
1	7	24.91888	24.86632	0.00211
2	6	25.83123	25.81434	0.00065
3	5	26.51988	26.51988	0.00000
4	4	26.84575	26.83735	0.00031
5	3	27.01238	27.01208	0.00009
6	2	27.10709	27.10704	0.00000
7	1	27.10184	27.10179	0.00000
8	0	26.90065	26.90027	0.00001

주) \* GAP = (상한-총효용)/총효용.

<표 3> (ITCAMs) 해결 프로시저 실험 결과( $w^t = r^{d,t} + r^{u,t}, \overline{M}_i^t = 0, \widehat{M}_i^t = 500/(r^{d,t} + r^{u,t})$ )

$\overline{N}_i^{d(c)}$	$\overline{N}_i^{u(c)}$	상한	총효용	GAP
0	8	30.34135	30.34095	0.00001
1	7	31.61924	31.53582	0.00265
2	6	32.51761	32.49327	0.00075
3	5	33.20497	33.20497	0.00000
4	4	33.53268	33.52403	0.00026
5	3	33.69956	33.69923	0.00008
6	2	33.79441	33.79435	0.00000
7	1	33.78918	33.78912	0.00000
8	0	33.58715	33.58676	0.00001



<그림 2> (ITCAMs) 해결 프로시저 실험 결과(Case 1 :  $w^t = 1$ , Case 2 :  $w^t = r^{d,t} + r^{u,t}$ )

<표 5>에는  $\overline{M}_i^t$ 를 <표 4>와 같이 가정하고,  $\widehat{M}_i^t = \overline{M}_i^t + 1$ 로 가정한 경우의 결과가 정리되어 있다. <표 5>에 제시된 GAP을 통해 제시한 (ITCAMs) 해결 프로시저가 최적 목적식 값에 가까운 좋은 해를 제공한다는 것을 알 수 있다. <표 4>와 <표 5>를 비교하여 살펴보면, 모든 문제에 대해서 신규 채널 할당과 기존 채널 회수를 통해 총효용이 증가하고 있음을 알 수 있다. <표 2>, <표 3>, <표 5> 등에 있는 결과는 Pentium IV(3.2GHz)에서 평균적으로 1.4ms 시간에 구하였다.

〈표 4〉 실험 데이터

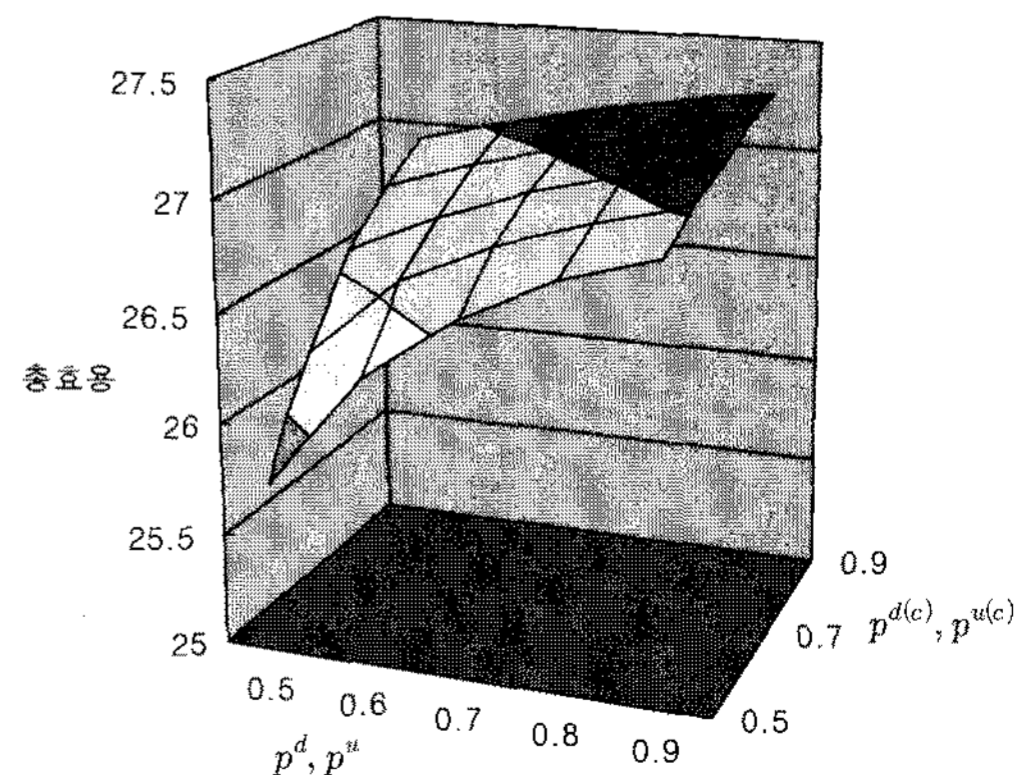
No.	$\overline{N}_i^{d(c)}$	$\overline{N}_i^{u(c)}$	$\overline{M}_i^t$										총효용
			189	94	46	94	47	18	500	19	9	4	
1	0	8	189	94	46	94	47	18	500	19	9	4	23.68301
2	1	7	236	118	58	118	58	23	500	23	11	5	24.86632
3	2	6	250	150	74	125	75	29	500	31	14	7	25.81434
4	3	5	250	166	100	125	83	51	500	51	25	12	26.51988
5	4	4	250	166	100	125	83	114	500	114	56	28	26.83735
6	5	3	250	166	100	125	83	175	500	177	88	44	27.01208
7	6	2	250	166	100	125	83	164	416	266	133	65	27.10704
8	7	1	250	166	100	125	83	130	198	362	180	89	27.10179
9	8	0	250	166	100	125	83	38	38	476	237	118	26.90027

〈표 5〉 (ITCAMs) 해결 프로시저 실험 결과( $w^t = 1$ ,  $\overline{M}_i^t$ 는 〈표 4〉의 값,  $\widehat{M}_i^t = \overline{M}_i^t + 1$ )

No.	$\overline{N}_i^{d(c)}$	$\overline{N}_i^{u(c)}$	$M_i^{t*}$										총효용	상한	GAP
			189	94	46	94	47	18	501	19	9	4			
1	0	8	189	94	46	94	47	18	501	19	9	4	23.68321	23.68514	0.00008
2	1	7	236	118	58	118	58	23	501	23	11	5	24.86652	24.86822	0.00007
3	2	6	251	150	74	126	75	29	501	32	14	6	25.81604	25.81856	0.00010
4	3	5	251	167	101	126	84	48	501	49	23	11	26.53295	26.53479	0.00007
5	4	4	251	167	101	126	84	109	501	114	56	26	26.86479	26.86703	0.00008
6	5	3	251	167	101	126	84	175	501	178	87	41	27.04395	27.04665	0.00010
7	6	2	251	167	101	126	84	164	409	267	132	62	27.13876	27.14032	0.00006
8	7	1	251	167	101	126	84	130	191	363	181	85	27.13292	27.13494	0.00007
9	8	0	251	167	101	126	84	30	39	477	238	116	26.91732	26.91937	0.00008

## 5.2 (ITCAMr) 해결 프로시저 실험 결과

〈표 6〉~〈표 7〉과 〈그림 3〉~〈그림 4〉에는 (ITCAMr) 해결 프로시저에 대한 실험결과가 제시되어 있다. 〈표 6〉과 〈그림 3〉에서는  $w^t = 1$ ,  $\overline{M}_i^t = 0$ ,  $\widehat{M}_i^t = 500/(r^{d,t} + r^{u,t})$ 로 가정하였다. 〈표 6〉에 의하면 프로시저 (Find\_Timeslot)에서 일반적으로 (ITCAMs)를 3번 풀어서 최적의  $N_i^{d(c)}$  및  $N_i^{u(c)}$ 를 구하고 있다는 것을 알 수 있다. 〈그림 3〉은  $p^d, p^u, p^{d(c)}, p^{u(c)}$  값에 따른 총효용 변화를 보여주고 있다. 이들 값이 클수록 할당 가능한 코드 수가 늘어나기 때문에 당연히 총효용이 증가한다.



〈그림 3〉  $p^d, p^u, p^{d(c)}, p^{u(c)}$  값에 따른 총효용 변화  
( $w^t = 1$ ,  $\overline{M}_i^t = 0$ ,  $\widehat{M}_i^t = 500/(r^{d,t} + r^{u,t})$ )

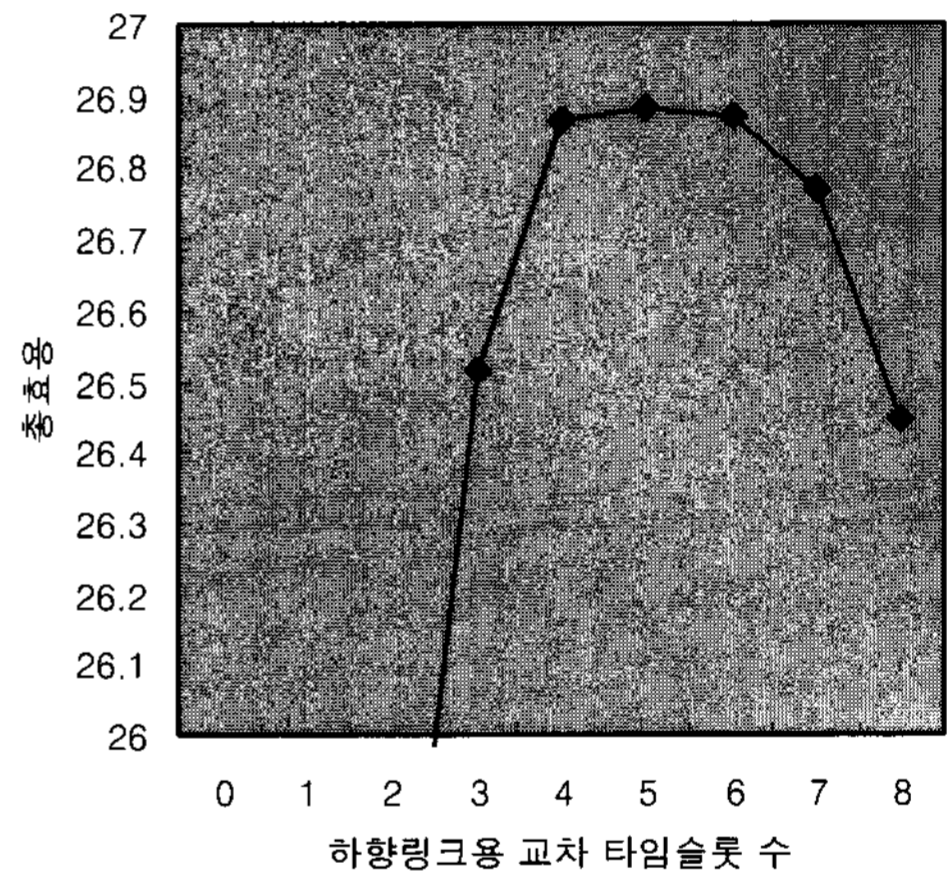
<표 6> (ITCAMr) 해결 프로시저 실험 결과

$$(w^t = 1, \overline{M}_i^t = 0, \widehat{M}_i^t = 500/(r^{d,t} + r^{u,t}))$$

$p^d, p^u, p^{d(c)}, p^{u(c)}$	$N_i^{d(c)}$	$N_i^{u(c)}$	총효용
0.9	7	1	27.25179
	6	2	27.24441
	8	0	27.11045
0.8	7	1	27.10179
	6	2	27.10704
	5	3	27.01208
0.7	7	1	26.88581
	6	2	26.91347
	5	3	26.81314
0.6	7	1	26.37162
	6	2	26.57869
	5	3	26.34366
0.5	7	1	25.46801
	6	2	25.76872
	5	3	25.43032

<표 7>에는  $p^d = p^u = p^{d(c)} = p^{u(c)} = 0.8$ 로,  $\overline{M}_i^t$  는 <표 4>와 같이 가정하고,  $\widehat{M}_i^t = \overline{M}_i^t + 1$ 로 가정한 경우의 결과가 정리되어 있다. <표 7>을 살펴보면, No.1에서 No.5의 경우, 하향링크용 교차 타임슬롯 수  $N_i^{d(c)}$ 를 한 개 증가시켜서 기존 할당 채널의 강제종료 없이 신규 채널 수요를 충족시키고 있다. 결국 No.1에서 No.5의 경우, <표 5>에 제시된 타임

슬롯 수를 고정한 경우의 결과와 비교해 볼 때, 총효용이 더 증가하였음을 알 수 있다. N.6에서 No. 9의 경우, <표 5>의 결과와 동일하다. 이는 타임슬롯 수 조정을 통해 더 나은 코드 할당 방법을 구할 수 없다는 것을 의미한다. <그림 4>는 <표 7>에 제시된 No.5에 대해 통합 타임슬롯 및 코드 할당 체계를 이용해 구한  $N_i^{d(c)}$  및  $N_i^{u(c)}$  값이 가능한 모든 교차 타임슬롯 할당 방법을 고려하더라도 최적이라는 것을 보여주고 있다. <표 6>과 <표 7>에 있는 결과를 구하는 데는 Pentium IV(3.2GHz)에서 평균적으로 3ms의 시간이 소요되었다.



<그림 4> 하향링크용 교차 타임슬롯 수에 따른 총효용 변화(No.5)

<표 7> (ITCAMr) 해결 프로시저 실험 결과( $w^t = 1, \overline{M}_i^t$ 는 <표 4>의 값,  $\widehat{M}_i^t = \overline{M}_i^t + 1$ )

No.	$N_i^{d(c)*}$	$N_i^{u(c)*}$	$M_i^{t*}$											총효용	상한	GAP
1	1	7	190	95	47	95	48	19	501	20	10	5	23.78884	23.78884	0.00000	
2	2	6	237	119	59	119	59	24	501	24	12	6	24.95266	24.95266	0.00000	
3	3	5	251	151	75	126	76	30	501	32	15	8	25.88393	25.88393	0.00000	
4	4	4	251	167	101	126	84	52	501	52	26	13	26.57461	26.57461	0.00000	
5	5	3	251	167	101	126	84	115	501	115	57	29	26.88395	26.88395	0.00000	
6	5	3	251	167	101	126	84	175	501	178	87	41	27.04395	27.04665	0.00010	
7	6	2	251	167	101	126	84	164	409	267	132	62	27.13876	27.14032	0.00006	
8	7	1	251	167	101	126	84	130	191	363	181	85	27.13292	27.13494	0.00007	
9	8	0	251	167	101	126	84	30	39	477	238	116	26.91732	26.91937	0.00008	



## 6. 결 론

본 논문에서는 다양한 유형의 음성 및 데이터 서비스를 제공하는 CDMA/TDD 시스템을 대상으로, 서비스별 코드 수요와 기존에 할당된 코드의 강제종료를 고려하면서 총효율을 최대화하는 통합 타임슬롯 및 코드 할당 방법을 찾기 위한 최적 타임슬롯 및 코드 할당 모형을 정형화하고, 라그랑지안 이완기법과 서브그라디언트 최적화 기법을 이용하여 실시간 통합 타임슬롯 및 코드 할당 체계를 제시하였다. 실험을 통해 제시한 자원 할당 체계가 매우 빠른 시간 내에 좋은 해를 제공한다는 것을 확인하였다. 주어진 타임슬롯 수 하에서는 평균적으로 1.4ms에, 최적의 타임슬롯 수도 결정해야 하는 경우에는 평균적으로 3ms에 최적해 또는 최적해에 아주 가까운 해를 구하였다. 결론적으로 본 논문에서 제시한 자원 할당 체계는 CDMA/TDD 시스템에서 실시간으로 타임슬롯 및 코드를 할당하기 위한 유용한 방법으로 활용될 수 있을 것이다.

## 참 고 문 헌

- [1] 3GPP TR 25.922 V5.0.0, *Radio resource management strategies*, Release 5, 2002.
- [2] Cao, Y., B. Zhou, and C. Li, "Admission control of integrated voice and data CDMA/TDD," *Proceedings of PIMRC*, (2003), pp. 896-900.
- [3] Casoni, M., G. Immovilli, and M.L. Merani, "Admission control in T/CDMA systems supporting voice and data applications," *IEEE Transactions on Wireless Communications*, Vol.1(2002), pp.540-548.
- [4] Casoni, M., G. Immovilli, and M.L. Merani, "Integrating voice and data applications in T/CDMA architectures," *IEEE Communications Letters*, Vol.5, No.9(2001), pp.378-380.
- [5] Chan, P.W.C. et al., "The evolution path of 4G networks : FDD or TDD?," *IEEE Communications Magazine*, Vol.44, No.12(2006), pp.42-50.
- [6] Chang, K.-N. and K.-D. Lee, "Capacity of CDMA/TDD systems supporting voice and multimedia data applications," *Telecommunications Review*, Vol.15, No.3(2005), pp.507-514.
- [7] Chang, K.-N., K.-D. Lee, and D. Kim, "Optimal timeslot and channel allocation considering fairness for multicell CDMA/TDD systems," *Computers and Operational Research*, Vol.33, No.11(2006), pp.3203-3218.
- [8] Chen, H.-H., C.-X. Fan, and W.W. Lu, "China's perspectives on 3G mobile communications and beyond : TD-SCDMA technology," *IEEE Wireless Communications*, Vol.9, No.2(2002), pp.48-59.
- [9] Esmailzadeh, R., M. Nakagawa, and A. Jones, "TDD-CDMA for the 4th generation of wireless communications," *IEEE Wireless Communications*, Vol.10, No.4(2003), pp.8-15.
- [10] Esmailzadeh, R., M. Nakagawa, and E. A. Sourour, "Time-division duplex CDMA communications," *IEEE Personal Communications*, Vol.4, No.2(1997), pp.51-56.
- [11] Geoffrion, A.M., "Lagrangian relaxation for integer programming," *Mathematical Programming Study*, Vol.2(1974), pp.82-114.
- [12] Bi, Q., George I. Zysman, and H. Menkes, "Wireless mobile communications at the start of the 21st century," *IEEE Communications Magazine*, Vol.39, No.1(2001), pp. 110-116.

- [13] Hasegawa, M., W. Gang, and M. Mizuni, "Applications of nonlinear prediction methods to the Internet traffic," *Proceedings of International Symposium on Circuits and Systems*, (2001), pp.169-172.
- [14] Held, M., P. Wolfe, and H. Crowder, "Validation of subgradient optimization," *Mathematical Programming*, Vol.6(1974), pp. 62-88.
- [15] Jeon, W.S. and D.G. Jeong, "Comparison of time slot allocation strategies for CDMA/TDD systems," *IEEE Journal on Selected Areas in Communications*, Vol.18(2000), pp. 1271-1278.
- [16] Jeong, D.G. and W.S. Jeon, "CDMA/TDD system for wireless multimedia services with traffic unbalance between uplink and downlink," *IEEE Journal on Selected Areas in Communications*, Vol.17, No.5(1999), pp. 939-946.
- [17] Jiang, Z. and V.C.M. Leung, "A predictive demand assignment multiple access protocol for Internet access over broadband satellite networks," *International Journal of Satellite Communication and Networking*, Vol.21(2003), pp.451-467.
- [18] Kim, S., H. Ahn, and S. Cho, "Variable target value subgradient method," *Mathematical Programming*, Vol.49(1991), pp.359-369.
- [19] Kim, D. and D.G. Jeong, "Capacity unbalance between uplink and downlink in spectrally overlaid narrowband and wideband CDMA mobile systems," *IEEE Transactions on Vehicular Technology*, Vol.49(2000), pp.1086-1093.
- [20] Kobayashi, Y., K. Mori, and H. Kobayashi, "Radio resource assignment scheme for asymmetric traffic in CDMA/shared-TDD cellular packet communications," *Proceedings of IEEE VTC*, (2004), pp.939-943.
- [21] Lee, K.D., K.-N. Chang, and V.C.M. Leung, "An Efficient Local Predictive Method for Distributed Timeslot Allocation in CDMA/TDD," accepted by *IEEE Transactions on Wireless Communications*.
- [22] Nasreddine, J. and X. Lagrange, "Time slot allocation based on a path gain division scheme for TD-CDMA TDD systems," *Proceedings of IEEE VTC*, (2003), pp.1410-1414.
- [23] Pérez-Romero, J., R. Agustí, O. Sallent, and F. Casadevall, "Packet transmission strategies to provide quality of service in a TDD-TD/CDMA system," *Proceedings of PIMRC*, (2001), pp.1-5.
- [24] Ryu, B., "Modeling and simulation of broadband satellite networks, Part II : Traffic Modeling," *IEEE Communications Magazine*, Vol.37, No.7(1999), pp.48-56.
- [25] Yomo, H., A. Nakata, and S. Hara, "An efficient slot allocation algorithm to accommodate multimedia traffic in CDMA/TDD-based wireless communications systems," *Proceedings of IEEE VTC*, 2001.
- [26] Zhang, G. and E. Zeira, "Fast permutation based time slot allocation for 3G WCDMA TDD systems," *Proceedings of IEEE VTC*, (2003), pp.1415-1419.