



(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。 Int. Cl. H04B 7/26 (2006.01)	(45) 공고일자 (11) 등록번호 (24) 등록일자	2007년05월21일 10-0719794 2007년05월14일
--	-------------------------------------	--

(21) 출원번호 (22) 출원일자 심사청구일자	10-2004-0095739 2004년11월22일 2004년11월22일	(65) 공개번호 (43) 공개일자	10-2005-0049420 2005년05월25일
----------------------------------	---	------------------------	--------------------------------

(30) 우선권주장	60/523,971 10/983,499	2003년11월21일 2004년11월08일	미국(US) 미국(US)
------------	--------------------------	----------------------------	------------------

(73) 특허권자

모토로라 인코포레이티드
미국, 일리노이 60196, 샤움버그, 이스트 앨공킨 로드 1303

노오쓰웨스턴 유니버시티
미국 일리노이 60208 에반스톤클라크 스트리트 633

(72) 발명자

서브래매니언비제이지.
미국, 일리노이 60657, 시카고, #2, 클리프톤 애비뉴 3027 엔.

애그레이웰래지브
미국, 일리노이 60062, 노스브루크, 캐로셀 2929

베리랜달
미국, 일리노이 60614, 시카고, #414, 링컨 파크 웨스트 2000

(74) 대리인

장훈
이법래

(56) 선행기술조사문헌
US6597705 B1

심사관 : 정재우

전체 청구항 수 : 총 10 항

(54) 자원 할당 및 스케줄링을 위한 방법 및 장치

(57) 요약

무선 통신 시스템 내에서 자원 할당 및 스케줄링을 위한 방법 및 장치가 본원에 제공된다. 자원 할당 동안, 최대 총 시스템 전송 전력(P), 이용가능한 최대 코드들의 수(N), 시스템내의 각 사용자를 위한 최대 코드들의 수($N=(N_1, \dots, N_d)$), 시스템

내의 각 사용자를 위한 최대 SINR 값($\mathbf{S}=(S_1, \dots, S_d)$), 및 시스템내의 각 사용자를 위한 전송 전력의 와트 당 SINR($\mathbf{e}=(e_1, \dots, e_d)$)가 스케줄러에 의해 수신된다(201). 그리고 나서, 스케줄러는 사용자 당 최적의 코드들의 수($\mathbf{n}$) 및 사용자 당 전력 레벨들($\mathbf{p}$)을 출력한다(215).

대표도

도 1

특허청구의 범위

청구항 1.

자원 할당 방법에 있어서,

최대 총 시스템 전송 전력(P)을 수신하는 단계;

이용가능한 최대 시스템 코드들의 수(N)를 수신하는 단계;

상기 시스템 내의 각 사용자를 위한 최대 코드들의 수(N)를 수신하는 단계;

상기 시스템 내의 각 사용자를 위한 최대 SINR 값(S)을 수신하는 단계;

상기 시스템 내의 각 사용자를 위한 전송 전력 값의 와트 당 SINR(e)을 수신하는 단계;

P, N, N, S 및 e에 기초하여 사용자 당 최적의 코드들의 수(n) 및 사용자 당 최적의 전력 레벨들(p)을 결정하는 단계; 및,
n 및 p에 기초하여 동작 파라미터들로 사용자들을 스케줄링하는 단계를 포함하는, 자원 할당 방법.

청구항 2.

제1항에 있어서, n 및 p에 기초하여 각 사용자를 위한 최적의 레이트(r)를 결정하는 단계를 더 포함하는, 자원 할당 방법.

청구항 3.

제1항에 있어서, n 및 p를 결정하는 단계는 스케줄링 시간(t)에서 선택될 레이트 벡터($\mathbf{r}_t$) 및 처리량/대기열 길이 (throughput/queue length) $\mathbf{w}_t=(w_1(W_{1,t}, Q_{1,t}), K, w_d(W_{d,t}, Q_{d,t}))$ 의 함수로서 가중 벡터의 내적을 최대화하는 것을 포함하고,

여기서:

$\mathbf{w}_t=(w_{1,t}, \dots, w_{d,t})$ 는 가중 벡터이고,

$\mathbf{W}_t=(W_{1,t}, \dots, W_{d,t})$ 는 사용자 처리량의 추정치들의 벡터이고,

$\mathbf{Q}_t=(Q_{1,t}, \dots, Q_{d,t})$ 는 사용자 대기열 길이들 또는 지연들의 벡터이고,

$\mathbf{r}_t=(r_{1,t}, \dots, r_{d,t})$ 는 선택된 레이트 벡터이고,

d는 사용자들의 수인, 자원 할당 방법.

청구항 4.

제1항에 있어서, $\mathbf{n}$ 및 $\mathbf{p}$ 를 결정하는 상기 단계는:

$$\sum_i n_i \leq N$$

$$\sum_i p_i \leq P$$

에 따라

$$EQ \ V(s) := \max_{(n,p) \in X} V(n,p)$$

를 결정하는 단계를 포함하고,

여기서,

$$EQ \ V(n,p) := \sum_i w_i n_i \ln(1 + F(p_i, e_i, n_i)),$$

$$EQ \ X = \{(n,p) \geq 0: n_i \leq N_i, p_i \leq S_i n_i / e_i \ \forall i\},$$

이고,

$\mathbf{w}$ 는 가중들(w_i s)의 벡터이고,

$\mathbf{e}$ 는 전송 전력(e_i s)의 와트 당 SINR의 벡터이고,

$\mathbf{S}$ 는 최대 SINR(S_i s)의 벡터이고,

$$w_i = f_i(W_i, Q_i) \text{ 이고,}$$

여기서,

$\mathbf{W} = (W_1, \dots, W_d)$ 는 모든 사용자들을 위한 추정된 처리량 벡터이고,

$\mathbf{Q} = (Q_1, \dots, Q_d)$ 는 모든 사용자들을 위한 대기열 길이 또는 지연 벡터이고,

f_i 는 사용자(i)를 위한 처리량 및/또는 대기열 길이의 어떤 함수이고, 또한 사용자(i)를 위한 QoS 클래스(class) 및 파라미터들에 의존할 수 있는, 자원 할당 방법.

청구항 5.

자원 할당 방법에 있어서,

통신 시스템 내에서 사용자 당 최적의 코드들의 수($\mathbf{n}$)에 대한 값을 결정하는 단계; 및,

$\mathbf{n}$ 에 기초하여 사용자당 최적의 레이트($\mathbf{r}$) 및 사용자당 최적의 전력($\mathbf{p}$)을 결정하는 단계를 포함하는, 자원 할당 방법.

청구항 6.

제5항에 있어서, 사용자 당 최적의 코드들의 수($\mathbf{n}$)를 결정하는 상기 단계는:

$\mathbf{n} = \mathbf{n}^*(\lambda^*, \mu^*(\lambda^*))$ 를 계산하는 단계를 포함하고,

여기서,

λ^* 는 전력 제약에 대한 최적의 라그랑주 승산자(Lagrange multiplier)이고,

$\mu^*(\lambda)$ 는 (전력 제약 λ 에 대한 상기 라그랑주 승산자의 함수로서)코드 제약에 대한 최적의 라그랑주 승산자 μ 이고,

$\mathbf{n}^*(\lambda, \mu)$ 는 λ, μ 의 함수로서 최적의 코드 할당인, 자원 할당 방법.

청구항 7.

제5항에 있어서, $\mathbf{n}$ 에 기초하여 사용자당 최적의 레이트($\mathbf{r}$) 및 사용자당 최적의 전력($\mathbf{p}$)을 결정하는 상기 단계는:

$\mathbf{n} = \mathbf{p}^*(\mathbf{n}) = \mathbf{p}^*(\mathbf{n}, \lambda^*(\mathbf{n}))$ 를 계산하는 단계를 포함하고,

여기서,

$\lambda^*(\mathbf{n})$ 는 전력 제약에 대한 최적의 라그랑주 승산자이고,

$\mathbf{p}^*(\mathbf{n}, \lambda)$ 는 $\mathbf{n}, \lambda$ 의 함수로서 최적의 전력 할당인, 자원 할당 방법.

청구항 8.

제5항에 있어서, $\mathbf{n}$ 에 대한 값을 결정하는 상기 단계는:

w_i 및 e_i 의 함수인 메트릭(metric)에 기초하여 내림차순으로 사용자들을 분류하는 단계로서, 상기 e_i 는 전송 전력 값의 와트당 SINR이고, w_i 는 시스템 내의 각 사용자(i)를 위한 기율기/가중치인, 상기 분류 단계;

$\sum_{i=1}^{j^*-1} N_i < N \leq \sum_{i=1}^{j^*} N_i$ 인 가장 작은 j^* 를 결정하는 단계;

$$n_i = \begin{cases} N_i & i < j^* \\ N - \sum_{i=1}^{j^*-1} N_i & i = j^* \\ 0 & i > j^* \end{cases}$$
 인 $\mathbf{n}$ 을 결정하는 단계를 포함하는, 자원 할당 방법.

청구항 9.

최대 총 시스템 전송 전력(P), 이용가능한 최대 시스템 코드들의 수(N), 상기 시스템 내의 각 사용자를 위한 최대 코드들의 수(N), 상기 시스템 내의 각 사용자를 위한 최대 SINR 값(S), 및 상기 시스템내의 각 사용자를 위한 전송 전력 값의 와트 당 SINR($\mathbf{e}$)를 수신하는 스케줄러를 포함하고, 상기 스케줄러는 P, N, N, S 및 $\mathbf{e}$ 에 기초하여 사용자 당 최적의 코드들의 수($\mathbf{n}$) 및 사용자 당 최적의 전력 레벨들($\mathbf{p}$)을 출력하는, 장치.

청구항 10.

제9항에 있어서, n 및 p 에 기초하여 물리층 동작 포인트들(physical layer operating posints)로 사용자들로의 전송들을 스케줄링하는 제어기를 더 포함하는, 장치.

명세서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

관련 출원(들)의 참조(들)

본 출원은 공동 소유되고 본원에 참조되어 있는 2003년 11월 21일에 출원된 발명의 명칭이 "자원 할당 및 스케줄링을 위한 방법 및 장치(METHOD AND APPARATUS FOR RESOURCE ALLOCATION AND SCHEDULING)"인 가출원 제60/523,971호의 우선권을 주장한 것이다.

발명의 분야

본 발명은 일반적으로, 자원 할당 및 스케줄링에 관한 것이며, 특히 공중을 통한 통신 시스템에서 자원 할당 및 스케줄링에 관한 것이다.

발명의 배경

통신 시스템들 내에서의 자원 할당 및 스케줄링은 일반적으로 종래 기술에 공지되어 있다. 자원 할당 및 스케줄링 동안, 시스템 제어기는 데이터 전송을 위한 스케줄과 함께, 각 사용자를 위한 채널 파라미터들을 결정한다. 예를 들어, 통신 시스템 내의 각 사용자는 코딩 방식, 전력, 이용되는 다수의 코드들, 및 데이터 레이트를 할당받을 수 있다. 공지된 바와 같이, 통신 시스템은 일반적으로 전송 파라미터들을 각각 갖는 많은 사용자들을 포함한다. 각 사용자를 적절하게 스케줄링하여 각 사용자를 위한 적절한 시스템 파라미터들을 선택함으로써 통신 시스템을 최적화하는 문제에 직면한다. 그러므로, 전반적으로 시스템 수행성능을 최적화하는 통신 시스템 내에서 스케줄링 및 자원 할당을 위한 방법 및 장치에 대한 필요성이 존재한다.

발명이 이루고자 하는 기술적 과제

자원 할당 및 스케줄링에 대한 필요성을 해소하기 위하여, 무선 통신 시스템 내에서 자원 할당 및 스케줄링을 위한 방법 및 장치가 본원에 제공된다. 자원 할당 동안, 최대 총 시스템 전송 전력(P), 이용가능한 최대 코드들의 수(N), 시스템내의 각 사용자를 위한 최대 코드들의 수($N=(N_1, \dots, N_d)$), 시스템 내의 각 사용자를 위한 최대 SINR 값($S=(S_1, \dots, S_d)$), 및 시스템 내의 각 사용자를 위한 전송 전력의 와트 당 SINR($e=(e_1, \dots, e_d)$)가 스케줄러에 의해 수신된다. 그리고 나서, 스케줄러는 P , N , N 및 S 에 기초하여 사용자 당 최적의 코드들의 수(n) 및 사용자 당 전력 레벨들(p)을 출력한다.

본 발명은 자원 할당하는 방법을 포함한다. 이 방법은 최대 총 시스템 전송 전력(P)을 수신하는 단계, 이용가능한 최대 시스템 코드들 수(N)를 수신하는 단계, 시스템 내의 각 사용자를 위한 최대 코드들의 수(N)를 수신하는 단계, 시스템 내의 각 사용자를 위한 최대 SINR 값(S)을 수신하는 단계, 시스템 내의 각 사용자를 전송 전력 값의 와트당 SINR(e)을 수신하는 단계, P , N , N , S 및 e 에 기초하여 사용자 당 최적의 코드들의 수(n) 및 사용자 당 최적의 전력 레벨들(p)을 결정하는 단계, 및 n 및 p 에 기초하여 동작 파라미터들로 사용자들을 스케줄링하는 단계를 포함한다.

본 발명은 부가적으로, 자원 할당하는 방법을 포함한다. 이 방법은 통신 시스템 내에서 사용자 당 최적의 코드들의 수(n)에 대한 값을 결정하는 단계 및 n 에 기초하여 사용자당 최적의 레이트(r) 및 사용자당 최적의 전력(p)을 결정하는 단계를 포함한다.

본 발명은 부가적으로, 최대 총 시스템 전송 전력(P), 이용가능한 최대 시스템 코드들의 수(N), 시스템내의 각 사용자를 위한 최대 코드들의 수(N), 시스템 내의 각 사용자를 위한 최대 SINR 값(S), 및 시스템내의 각 사용자를 위한 전송 전력의 와트 당 SINR(e)를 수신하는 스케줄러를 포함하고, 상기 스케줄러가 P, N, N, S 및 e에 기초하여 사용자 당 최적의 코드들의 수(n) 및 사용자 당 전력 레벨들(p)을 출력하는 장치를 포함한다.

발명의 구성

지금부터, 전체 도면들을 참조하면, 동일한 구성요소들에는 동일한 번호들이 병기되어 있으며, 도1은 전송 회로(100)의 블록도이다. 도시된 바와 같이, 전송 회로(100)는 스케줄러/최적화기(101), 제어기(103) 및 다수의 송신기들(105)을 포함한다. 동작 동안, 시스템 파라미터들은 스케줄러(101)에 입력된다. 그리고 나서, 스케줄러(101)는 각 송신기(105)를 위한 최적화된 전송 파라미터들의 집합을 계산하여 이 정보를 제어기(103)로 통과시킨다. 데이터 전송 동안, 제어기(103)는 각 사용자들에 지정된 데이터를 수신하고 최적화된 전송 파라미터들의 집합을 사용하여 송신기들(105)을 적절하게 제어한다. 데이터가 송신기들(105)로 통과될 때, 송신기들(105)은 전송 파라미터들을 사용하여 데이터를 각 사용자들에게 적절하게 전송한다.

일 예로서, 스케줄러/최적화기(101)는 제어기(103)로부터 사용자들의 집합을 수신하고 전송을 위하여 사용자들을 선택한다. 선택된 사용자들 각각에 대해서, 스케줄러/최적화기(101)는 특정 변조 및 코딩 방식(MCS), 특정 전송 전력, 특정 코드들의 수 및 특정 정보 레이트를 포함하는 물리층 동작 포인트(PLOP)를 선택한다. 각 PLOP는 서로 다른 량의 전체 코드 및 전력 자원들을 소모한다. 스케줄러/최적화기(101)는 각 사용자를 위한 "최적의" PLOP를 선택하여, 전체 시스템을 최적화한다. 특히, 각 사용자를 위한 최적의 PLOP가 선택되어, 사용자들에 할당된 레이트들의 가중된 조합의 합을 최대화시킨다. 가중 벡터에 대한 바람직한 값은 현재 처리량 추정시 유틸리티 함수의 기울기이다.

상술된 바와 같이, 스케줄러/최적화기(101)는 입력 파라미터들의 주어진 집합에 기초하여 전체 시스템 수행성능을 최적화하는 작업을 행한다. 본 발명의 바람직한 실시예에서, 이는 스케줄링 및 자원 할당 알고리즘을 최적화함으로써 성취된다. 이하의 내용은, 1) 자원 할당 알고리즘의 생성 및 2) 자원 할당 알고리즘의 최적화를 서술한다.

1. 자원 할당 알고리즘의 생성

요약하면, 스케줄링 알고리즘 최적화는 각 스케줄링 시간(t)에서 선택된 레이트 벡터(rate vector)($\mathbf{r}_t$) 및 $\nabla U(\mathbf{W}_t, \mathbf{Q}_t) := \phi \nabla_{\mathbf{W}} U(\mathbf{W}_t, \mathbf{Q}_t) - \nabla_{\mathbf{Q}} U(\mathbf{W}_t, \mathbf{Q}_t)$ 로서 정의된 "유틸리티의 기울기"(처리량 및 대기열 길이의 함수로서)의 내적을 최대화하는 것인데, 여기서:

U는 유틸리티 함수이고,

$\mathbf{W}_t = (W_{1,t}, \dots, W_{d,t})$ 는 사용자 처리량의 추정치이고,

$\mathbf{Q}_t = (Q_{1,t}, \dots, Q_{d,t})$ 는 사용자 대기열 길이 또는 지연이고,

$\nabla_{\mathbf{W}} U(\mathbf{W}_t, \mathbf{Q}_t)$ 는 $\mathbf{W}$ 에 대한 U의 부분 도함수들의 벡터이고,

$\nabla_{\mathbf{Q}} U(\mathbf{W}_t, \mathbf{Q}_t)$ 는 $\mathbf{Q}$ 에 대한 U의 부분 도함수들의 벡터이고,

$\mathbf{r}_t = (r_{1,t}, \dots, r_{d,t})$ 는 선택된 레이트 벡터이고,

d는 사용자들의 수이다.

즉, 유틸리티의 기울기에 대해 최고 프로젝션을 갖는 상태 의존 레이트 영역 $\mathbf{R}(\mathbf{e}_t)$ 로부터 레이트 벡터를 선택하고자 한다. 이 레이트 영역은 채널 상태를 따른다. 채널 상태가 완전하게 알려지지 않지만, 품질(예를 들어, CQI) 피드백 등을 통해서

이용가능한 채널 상태를 일부 추정한다. 특히, 각 사용자(i)에 대한 $EQ e_{i,t}$ 의 와트 당 예측된 SINR을 인지하고 $\mathbf{e}_t = (e_{1,t}, \dots, e_{d,t})$ 가 채널 상태 $\nabla U(\mathbf{W}_t, \mathbf{Q}_t) \leq \mathbf{r}_t = \max_{r_i \in \mathcal{R}(e_i)} \nabla \log(\mathbf{U}(\mathbf{S}(\mathbf{e}_t)))_i(\mathbf{W}_{i,t}, \mathbf{Q}_{i,t}) r_{i,t}$ 를 포착한다고 하자.

어떤 클래스의 유틸리티 함수들에 대해선, 다음 식(1)을 갖는다.

$$\mathcal{U}_i(\mathbf{W}_i, \mathbf{Q}_i) := \phi \frac{\partial}{\partial \mathbf{W}_i} U_i(\mathbf{W}_i, \mathbf{Q}_i) - \frac{\partial}{\partial \mathbf{Q}_i} U_i(\mathbf{W}_i, \mathbf{Q}_i) = \phi c_i \mathbf{W}_i^{\alpha-1} + b_i \mathbf{Q}_i^{p-1} =: \mathbf{w}_i \quad (1)$$

여기서,

c_i 는 사용자(i)에 대한 처리량과 관련된 QoS 가중치이고,

d_i 는 사용자(i)에 대한 대기열 길이와 관련된 QoS 가중치이고,

ϕ 는 처리량과 관련된 필터 계수이고,

α 는 처리량과 관련된 페어니스 파라미터(fairness parameter)이고,

p 는 대기열과 관련된 페어니스 파라미터이다.

따라서, 최적화 문제는 다음과 같이 된다.

$$\max_{\mathbf{r} \in \mathcal{R}(\mathbf{e})} \sum_i \mathbf{w}_i r_i$$

여기서, $\mathbf{w}_i$ 는 상기 식(1)로 주어지고, 간편성을 위하여 시간(t)에서 의존성(dependence)을 억제한다. 레이트(r_i)의 선택은 코드들의 수(n_i) 및 전력(p_i)을 포함하는 물리층 동작 포인트(PLOP)의 선택과 결합된다. 이들 PLOP 변수들은 시스템을 충족시키기 위하여 필요로 되고 사용자 마다 다음과 같이 제약된다.

$$n_i \leq N_i$$

$$\sum_i n_i \leq N$$

$$\sum_i p_i \leq P$$

이들은 상태 의존 레이트 영역 $\mathcal{R}(\mathbf{e})$ 의 디스크립션 부분이다. 아게웰(Agawal) 등이 "무선 통신 시스템에서의 패킷 스케줄링 및 라디오 자원 할당을 위한 방법(Method for Packet Scheduling and Radio Resource Allocation in a Wireless Communication System)"이라는 명칭으로 출원한 미국 특허 공개번호 US2002/0147022호(이하, "아그레이웰(Agrawal)

특허 공개"라 한다)에 서술된 바와 같이, 코드 당 레이트는 $\frac{r_i}{n_i} = \Gamma(\text{SINR}_i)$ 이 되도록 코드 당 신호 대 잡음비(SINR)와 관련하여 기록될 수 있다.

여기서, 코드 당 SINR은 $\text{SINR}_i = \frac{p_i}{n_i} e_i$ 로 주어진다.

여기서, Γ 는 어떤 적절한 함수이다. 함수 Γ 의 적당한 선택은 널리 공지된 정보 이론 채널 식 $\Gamma(x) = \text{Blog}(1+x)$ 이다. 레이트 및 SINR 간의 이와 같은 함수 관계는 HSDPA가 $B=3840000/16=240000$ symbols/sec인 심볼 레이트(칩 레이트/스프레

딩 팩터)와 동일한 B를 지닌 고속 다운링크 패킷 액세스(HSDPA)에 양호한 곡선 적합(Ccurve fit)을 제공한다. 최적의 곡선 적합을 위하여, SINR은 A=1.7의 팩터 만큼 감소되는데, 즉 $\Gamma(x) \approx B \log(1+x/A)$ 이 사용된다는 것을 주지하라. 이 식에서 플러깅(plugging)은 e_i , n_i , p_i 와 관련하여 레이트(r_i)에 대해 다음 식(2)으로 된다.

$$r_i = n_i B \log\left(1 + \frac{p_i e_i}{A n_i}\right) \quad (2)$$

HS-PDSCH 코드에 대한 전송 전력의 와트 당 SINR은 다음과 같이 주어진다라는 것을 주지하라.

$$e_i = \left(\frac{E_c}{N_t} \right)_{pilot} \frac{16}{P_{pilot}}$$

표기법을 간단히 하기 위하여, e_i 는 e_i/A 로 다시정의되며, HS-PDSCH 코드에 대한 전송 전력의 와트 당 "유효" SINR은 다음과 같이 주어진다.

$$e_i = \left(\frac{E_c}{N_t} \right)_{pilot} \frac{16}{P_{pilot}} \frac{1}{A}$$

$w_i = \frac{B(\phi_i W_i^{\alpha-1} + b_i Q_i^{p-1})}{\ln 2}$ 가 되도록 다시정의함으로써, 이 레이트 영역은 다음과 같이 주어진다.

$$EQ \mathcal{R}(e) = \{r \geq 0: r_i = n_i B \log(1 + \frac{1}{A} (p_i e_i, n_i)), n_i \leq N_i \forall i, \forall su(i, n_i) \leq M, \forall su(i, p_i) \leq P\}$$

여기서,

$\mathbf{w} = (w_1, \dots, w_d)$ 는 기울기 벡터이고,

$\mathbf{W} = (W_1, \dots, W_d)$ 는 모든 사용자들을 위한 추정된 처리량 벡터이고,

$\mathbf{Q} = (Q_1, \dots, Q_d)$ 는 모든 사용자들을 위한 추정된 처리량 벡터이고,

c_i 는 사용자(i)에 대한 처리량과 관련된 QoS 가중치(가능한 0)이고,

d_i 는 사용자(i)에 대한 대기열 길이와 관련된 QoS 가중치(가능한 0)이고,

ϕ 는 처리량과 관련된 필터 계수이고,

α 는 처리량과 관련된 페어니스 파라미터(fairness parameter)이고,

p 는 대기열과 관련된 페어니스 파라미터이고,

B는 비트들들에 식을 정합시키기 위한 스케일링 상수이다.

이는 가중 벡터 $\mathbf{w} = (w_1, \mathbf{K}, w_d)$ 의 특정 예이다. 그러나, 일반적으로, w_i 는 처리량을 증가시키지 않고 대기열 길이를 감소시키지 않는 처리량(W_i) 및 대기열 길이(Q_i)의 임의의 함수일 수 있다. 이 함수는 또한, 지연 요구조건 QoS 클래스, 대역폭 요구조건, 등과 같은 각종 QoS 관련된 정보를 파라미터들로서 취할 수 있다.

$$\sum_i n_i \leq N$$

최적화 문제는, $\sum_i p_i \leq P$ 에 따라 다음 식(3)으로서 재기록될 수 있다.

$$\text{EQ } V^*(*) := \max_{(n, p) \in X} V(n, p) \quad (3)$$

여기서,

$$\begin{aligned} \text{EQ } V(n, p) &:= \sum_i w_i n_i \ln(1 + F(p_i e_i, n_i)), \\ \text{EQ } X &:= \{(n, p) \geq 0: n_i \leq N_i \quad \forall i\}, \end{aligned} \quad (4)$$

n 은 코드 할당들의 벡터 즉, 사용자 당 코드들을 포함하는 벡터이고,

p 는 전력 할당들의 벡터, 즉 사용자당 전력 레벨들을 포함하는 벡터이고,

w 는 w_i s의 벡터이고,

e 는 e_i s의 벡터이다.

제약 집합(constraint set) X 는 컨벡스(convex)라는 점에 유의하라. 이는 또한, V 가 (n, p) 에 컨케이브(concave)라는 것을 입증할 수 있다. 사용자(i)를 위하여 재전송하는 경우에, 이 사용자의 코드(e_i)의 와트 당 "유효" SINR은 적절한 팩터로 댄프 업(bump up)될 수 있다.

상기 포착된 제약들 이외에도, 사용자 당 다음의 전력 제약들이 허용되는데, 이들 제약들중 어떤 제약들은 파라미터들을 적절하게 설정함으로써 리던던트(redundant)한 것으로 간주될 수 있다.

· 피크 전력 제약:

$$\text{EQ } p_i \leq P_i, \quad \forall i.$$

· SINR 제약:

$$\text{SINR}_i = \frac{p_i e_i}{n_i} \leq S_i \Leftrightarrow p_i \leq S_i \frac{n_i}{e_i}, \quad \forall i.$$

· 코드 당 레이트(고정된 변조 순서(modulation order)를 추정하는 엔코더 레이트는 상수 $\times$ 상기 순서인데, 즉 엔코더 레이트는 변조 순서(2 또는 4) $\times$ 심볼 레이트의 곱(product)(HSDPA에 대해 240000 symbols/sec)인 코딩된 출력 레이트에 의해 나뉘진 코드 당 레이트라는 것을 주지하라).

$$\frac{r_i}{n_i} = \ln\left(1 + \frac{p_i e_i}{n_i}\right) \leq (R/N)_i \Leftrightarrow p_i \leq (e^{(R/N)_i} - 1) \frac{n_i}{e_i} \quad \forall i.$$

· 레이트

$$r_i = n_i \ln\left(1 + \frac{p_i e_i}{n_i}\right) \leq R_i \Leftrightarrow p_i \leq (e^{R/n_i} - 1) \frac{n_i}{e_i} \quad \forall i. \quad (5)$$

사용자 당 모든 전력 제약들은, 다음 식(6)의 특수한 경우들이다.

$$EQ SINR_i = \{F(p_i, e_i, n_i) \leq s_i(n_i), \quad \forall i\}, \quad (6)$$

여기서 함수(s_i)는 또한 (최적화 문제에 대해) 고정된 $EQ P_i, e_i, S_i, R_i, (R/N)_i$ 에 의존한다. 실제로, 상기 제약들 모두는

$$s_i(n_i) = \min \left\{ \frac{P_i e_i}{n_i}, S_i, (e^{(R/N)_i} - 1), (e^{R_i/n_i} - 1) \right\}$$

에 의해 단일 제약(6)과 결합될 수 있다.

관심을 둔 2가지 특수한 경우들은 다음과 같다.

1. $s_i(n_i) \equiv s_i$ 는 n_i 에 좌우되지 않는다.

2. $s_i(n_i) \equiv s_i = \infty$

사용자 당 전력 제약들로 인해, 제약 집합(X)은, 다음 식(7)로 또한 제한된다.

$$EQ X = \{(n, p) \geq 0: n_i \leq N_i, \quad p_i \leq s_i(n_i) n_i / e_i \quad \forall i\} \quad (7)$$

제약 집합은 $s_i(n_i) n_i$ 가 n_i 의 컨케이브 함수이면 계속해서 컨벡스 되도록 한다. $s_i(n_i) n_i$ 가 실제로 상술된 2가지 특수한 경우 (1-2)에 대해서 n_i 의 컨케이브 함수라는 것을 주지하라. 그러나, 레이트 제약 경우(식 18)에, $s_i(n_i) n_i$ 는 n_i 에서 컨벡스이다.

1.1 재전송들 처리

앞서 언급된 바와 같이, 재전송들로 인한 사용자들은 2가지 면들에서 서로 다르게 처리될 필요가 있다.

1. 코드 당 와트 당 이들의 유효 SINR은 증가될 필요가 있다.

2. 이들은 레이트 $r_i = R_i$ 에 대해 동등한 제약을 갖는다.

전자는 이들의 e_i 를 적절하게 조정함으로써 손쉽게 처리될 수 있다. 후자의 제약을 처리하기 위하여, 레이트 제약이 다음 식(8)과 등가라는 것에 주지하자.

$$r_i = \frac{B}{\ln 2} n_i \ln \left(1 + \frac{p_i e_i}{n_i} \right) = R_i \Leftrightarrow p_i = \frac{n_i}{e_i} \left(e^{\frac{R_i \ln 2}{B n_i}} - 1 \right) \quad (8)$$

2. 스케줄러 /최적화기 동작

스케줄러/최적화기(101)는 파라미터들의 집합을 수신하여 식(3)의 자원 할당 알고리즘을 최적화함으로써 V^* 에 대해 푼다. 특히, 스케줄러/최적화기(101)는 다음을 수신한다.

P = 최대 총 시스템 전송 전력

N = 시스템에 이용가능한 최대 총 코드들

$\mathbf{N}$ = 시스템에서 각 사용자를 위한 최대 코드들(예를 들어, 스프레딩 코드들)의 수를 포함하는 벡터

$\mathbf{R}$ = 시스템에서 각 사용자를 위한 최대 데이터 레이트를 포함하는 벡터

R/N = 시스템에서 각 사용자를 위한 코드 당 최대 데이터 레이트를 포함하는 벡터

P = 시스템에서 각 사용자를 위한 최대 전송 전력을 포함하는 벡터

S = 시스템에서 각 사용자를 위한 최대 잡음(예를 들어, SINR)을 포함하는 벡터

e = 시스템에서 각 사용자를 위한 전력 당 잡음(예를 들어, SINR)(예를 들어, 와트 당 SINR)을 포함하는 벡터

W = 시스템에서 각 사용자를 위한 추정된 처리량 벡터를 포함하는 벡터

Q = 시스템에서 각 사용자를 위한 대기열 길이 벡터를 포함하는 벡터

c = 시스템에서 각 사용자를 위한 처리량과 관련된 QoS 가중치를 포함하는 벡터

b = 시스템에서 각 사용자를 위한 대기열 길이와 관련된 QoS 가중치를 포함하는 벡터.

본 발명의 바람직한 실시예에서, 스케줄러/최적화기(101)는 각 사용자(i)를 위한 r_i , n_i 및 p_i 의 최적 값들을 결정하여, 통신 시스템(100)을 위한 벡터들 r , n 및 p 을 출력한다. 상술된 바와 같이, 통신 시스템(100)을 위한 벡터들 r , n 및 p 은 상술된 입력 파라미터들의 집합이 제공된 식(3)에서 V^* 를 풀음으로써 결정된다. 스케줄러/최적화기(101)는 식(7)에서 상술된 부가적인 제약들에 따라 V^* 에 대해 푼다.

2.1 제1 실시예

$\sum_i N_i \leq N$,
 i 인 경우, $n_i = N_i$ 는 최적의 코드 할당이며, 나중에 도시된 바와 같이 매우 간단하게 해결될 수 있는 전력 최적화 문제만이 남게된다. 지금부터, 이것이 그 경우가 아닌 경우, 즉 $\sum_i N_i > N$ 인 경우를 다룰 것이다. 또한, 정수 최적화 문제에 대한 완화로서 n_i 가 비정수 값들을 취하도록 한다. 대부분의 경우들에서, 최적의 해법은 정수값들을 n_i 에 할당하는 것으로 판명되었다.

듀얼 포물레이션(dual formulation)을 고찰함으로써 최적화 문제를 해결한다. 다음과 같은, 라그랑지안(lagrangian)을 정의하자.

$$EQ L(p, n, \lambda, \mu) = \sum_i (w_i n_i \ln(1 + F(p_i, e_i, n_i))) + \lambda (P - \sum_i p_i) + \mu (N - \sum_i n_i)$$

이를 토대로 다음 식(9)의 듀얼 함수(dual function)를 정의할 수 있다.

$$EQ L(\lambda, \mu) := \max_{(n, p) \in \mathcal{X}} L(p, n, \lambda, \mu) \quad (9)$$

여기서,

$\mathcal{X}$ 는 제약 집합(7)으로 주어진 (n, p) 에 대한 유효한 선택들의 집합이고,

λ 는 전력 제약에 대한 라그랑주 승산자(lagrange multiplier)이고,

μ 는 코드 제약에 대한 라그랑주 승산자이다.

이중 문제는 다음을 구하기 위한 것이다.

$$EQ L^{\sup}(*):= \min_{(\lambda, \mu) \geq 0} L(\lambda, \mu) \quad (10)$$

또한, 다음을 정의하자.

$$EQ L(\lambda) := \min_{\mu \geq 0} L(\lambda, \mu) = \max_{(n, p) \in \mathcal{X}} L(p, n, \lambda, \mu) \quad (11)$$

이 해법의 부분에 대해서, 레이트 제약(5)(재전송들을 위한 (8)) 무시 또는 이를 "레이트 제한들을 처리"하는 후술되는 바와 같은 선형 제약으로 대체할 것이다.

주: 레이트 제한들 처리

레이트 제한(5) 또는 (8)을 지닌 제약 집합은 비컨벡스인데, 이는 n_i 에 대해서 매우 손쉽게 최적화하지 못하도록 한다. 문제들을 간단하게 하기 위하여, 우측의 제2 항에서 $n_i = N_i$ 라고 추정할 수 있다. 따라서, (5)는 다음과 같이 대체된다고 추정한다.

$$p_i \leq \frac{n_i}{e_i} \left(e^{\frac{R_i \ln 2}{BN_i}} - 1 \right)$$

제약(8)의 경우에, 이를 다음과 같이 대체할 수 있다.

$$p_i = \frac{n_i}{e_i} \left(e^{\frac{R_i \ln 2}{BN_i}} - 1 \right).$$

이는 몇가지 상이한 이유들로 정당화될 수 있다. 첫번째, 많은 경우들에서, 최적화 $n_i = N_i$ 라는 것이 관찰된다. 두번째, $n_i = N_i$ 치환은 이 사용자에게 어느정도 나쁘게 하지만, (재전송들의 경우에) e_i 에서 범프 업은 아마도 여전히 이 사용자 선택을 지원하는 이 사용자의 $w_i e_i$ 를 증가시킬 것이다. 상기 제약에 대해 서로 다른 선형 근사화를 시도할 수 있다. 이 제약은 코드 할당 부분에만 필요로 된다는 점에 유의하라. 일부 다른 발견적방법들(heuristics)이 또한 가능하다. 예를 들어, 상술된 바와 같은 레이트 제약을 대체할 수 있고 이 사용자를 선택하고 $n_i = N_i$ 를 획득하면, 어떤것도 거의 희생하지 않을 것이다. 유사하게, 사용자를 선택하도록 하는 제2 항에서 $n_i = 1$ 이 되도록 시도할 수 있다. 여전히 선택되지 않으면, 또 다시, 아마도 이 근사화에 의해 아마도 어떤 것도 희생되지 않는다. 그러나, 중간 경우들에 대해서(이는 아마도 드물다), 이 사용자에게 n_i 를 명시적으로 가변시키고 최적화하길 원할 수 있다. 코드 할당이 행해지면, 전력들에 대한 최적화들을 위하여 원래의 레이트 제약으로 작업할 수 있다는 점에 유의하라.

목적 함수(objective function)가 컨케이브(concave)이고 제약 집합이 컨벡스(convex)이기 때문에, 듀얼리티 갭(duality gap)이 존재하지 않아 제1의 해법을 계산하기 위하여 듀얼 해법을 사용할 수 있다.

이를 토대로, 스케줄러/최적화기(101)는 다음 단계들(또한 도2에 도시)을 통해서 $\mathbf{r}$, $\mathbf{n}$ 및 $\mathbf{p}$ 에 대한 적절한 값들을 결정한다.

1. 모든 입력 파라미터들(P , N , $\mathbf{N}$, $\mathbf{R}$, $\mathbf{R}/N$, $\mathbf{P}$, $\mathbf{S}$, $\mathbf{e}$, $\mathbf{W}$, $\mathbf{Q}$, $\mathbf{c}$, $\mathbf{b}$)을 수집하라(단계 201).
2. 고정된 λ 및 μ 에 대해, 분석적으로 최적의 $(\mathbf{n}, \mathbf{p})$ 를 구하라(단계 203).
3. λ 를 고정된 채로 유지하고, 최적의 $\mu^*(\lambda)$ 를 구하라(단계 205).
4. 또한, 식(11)에 규정된 바와 같이 λ 를 고정된 채로 유지하라(단계 207).
5. 최적의 $\lambda = \lambda^*$ 를 수치적으로 탐색, 즉 L^* 를 구하기 위하여 $L(\lambda)$ 를 최소화하는 λ 의 값을 탐색하라 (단계 209).
6. $\mathbf{n} = \mathbf{n}^*(\lambda^*, \mu^*(\lambda^*))$ 를 계산하라(단계 211)

7. $\mathbf{n}$ 을 고정된 채로 유지하면서 최적의 $\mathbf{p}$ 및 $\mathbf{r}$ 을 계산하라(단계 213).

도2는 스케줄러/최적화기(101)의 동작을 도시한 순서도이다. 논리 흐름은 단계(201)에서 시작되는데, 여기서, e, P, N, N 및 $\mathbf{S}$ 는 최적화기(101)에 의해 수집된다. 단계(203)에서, $\mathbf{n}$ 및 $\mathbf{p}$ 는 λ 및 μ 를 고정된 채로 유지하면서 분석적으로 계산된다. 특히, λ 및 μ 는 고정되지만 임의적이고, $\mathbf{n}$ 및 $\mathbf{p}$ 는 λ 및 μ 의 함수로서 계산되어 라그랑지안을 최적화하여 식(9)에 도시된 바와 같은 듀얼 함수를 구한다.

단계(205)에서, λ 는 고정되지만 임의적이고, 최적의 $\mu^*(\lambda)$ 가 결정되어, 듀얼 함수 $L(\lambda, \mu)$ 를 최소화하여 $L(\lambda)$ 를 구한다. $L(\lambda)$ 는 단계(207)에서 식(23)에 따라서 계산되고 최적의 $\lambda = \lambda^*$ 에 대한 수치 탐색은 단계(209)에서 발생된다. 다른 말로서, 단계(209)에서, $\lambda = \lambda^*$ 가 결정되어, $L^* = L(\lambda^*) = \min_{\lambda \geq 0} L(\lambda)$ 이 되도록 한다. 단계(211)에서, $\mathbf{n} = \mathbf{n}^*(\lambda^*, \mu^*(\lambda^*))$ 가 식(14)에 따라서 계산된다. 벡터($\mathbf{R}$)는 수집되고, $\mathbf{r}$ 및 $\mathbf{p}$ 는 $\mathbf{n}$ 을 고정된 채로 유지하면서 계산된다(단계 213). 특히, $\mathbf{n}$ 은 단계 5에서 구해진 값으로 고정되고, $\mathbf{p}$ 및 $\mathbf{r}$ 은 실제 레이트 제약($\mathbf{R}$)을 고려하여 식(21) 및 (2)에 따라서 계산된다. 최종적으로, 단계(215)에서, $\mathbf{n}, \mathbf{p}$ 및 $\mathbf{r}$ 값들은 제어기(103)로 출력되고 데이터를 스케줄하도록 사용되어 송신기들(105)을 통해서 이 데이터를 전송한다.

이하에 도2에서 언급된 단계들을 보다 상세히 설명한다.

2.1.1. 분석적으로 최적의 ($\mathbf{n}, \mathbf{p}$)를 구하는 고정된 λ 및 μ 에 대한 단계(203)

고정된 λ, μ 및 $\mathbf{n}$ 에 대한 (듀얼에서) 라그랑지안을 최적화(이는 고정된 가능한 $\mathbf{n}$ 을 추정하는 전력들에 대한 최적화를 포함)가 획득된다.

$$p_i^* = \frac{n_i}{e_i} \begin{cases} 0, & w_i e_i \leq \lambda \\ \frac{w_i e_i}{\lambda} - 1, & \frac{w_i e_i}{1 + s_i(n_i)} \leq \lambda < w_i e_i \\ s_i(n_i), & \lambda < \frac{w_i e_i}{1 + s_i(n_i)} \end{cases}$$

또는, 등가적으로

$$p_i^* = \frac{n_i}{e_i} \left[\min \left\{ \frac{w_i e_i}{\lambda} - 1, s_i(n_i) \right\} \right]^+ \quad (12)$$

이로 인한 코드 당 SINR은 다음과 같이 주어진다.

$$\frac{p_i^* e_i}{n_i} = s^* \left(\frac{w_i e_i}{\lambda}, s_i(n_i) \right) := \left[\min \left\{ \frac{w_i e_i}{\lambda} - 1, s_i(n_i) \right\} \right]^+ \quad (13)$$

전력들에 대해 사용자 당 제약들을 갖지 않을 때 마지막 경우는, 비워지는데, 즉 $s_i(n_i) = \infty$ 가 된다. $s_i(n_i) \equiv s_i$ 가 n_i 에 의존하지 않는 경우에, 이로 인한 코드 당 SINR은 또한 코드들의 수(n_i)에 의존하지 않고 전력 요구조건은 코드들의 수 면에서 선형적으로 스케일된다.

다음을 정의하자.

$$h(w_i e_i, s_i(n_i), \lambda) := \begin{cases} 0, & w_i e_i \leq \lambda \\ \left[\frac{\lambda}{w_i e_i} - 1 - \ln \frac{\lambda}{w_i e_i} \right], & \frac{w_i e_i}{1 + s_i(n_i)} \leq \lambda < w_i e_i \\ \left[\ln(1 + s_i(n_i)) - \frac{\lambda}{w_i e_i} s_i(n_i) \right], & \lambda < \frac{w_i e_i}{1 + s_i(n_i)} \end{cases}$$

경우(1)에 대해서, $s_i(n_i) \equiv s_i$ 가 n_i 에 의존하지 않는다. 게다가,

$h(w_i e_i, s_i(n_i), \lambda) = h(w_i e_i, s_i, \lambda)$ 는 또한 n_i 에 의존하지 않는다. 이 경우에, n_i 에 대해서 라그랑지안을 더욱 최적화하여 듀얼 함수를 계산하는 것이 용이하게 된다. 특히, 다음 식을 구한다.

$$n_i^* = n_i^*(\lambda, \mu) = \begin{cases} 0, & w_i h(w_i e_i, s_i, \lambda) < \mu \\ 0 \leq n_i \leq N_i, & w_i h(w_i e_i, s_i, \lambda) = \mu \\ N_i, & w_i h(w_i e_i, s_i, \lambda) > \mu \end{cases} \quad (14)$$

$\mu = w_i h(w_i e_i, s_i, \lambda)$ 경우에, 임의의 $0 \leq n_i \leq N_i$ 가 최적화된다는 점에 유의하라. 듀얼 함수는 다음 식과 같이 주어진다.

$$L(\lambda, \mu) = \sum_i [\mu_i(\lambda) - \mu]^+ N_i + \mu N + \lambda P \quad (15)$$

여기서, $\mu_i(\lambda) := w_i h(w_i e_i, s_i, \lambda)$

2.1.2 λ 고정시키고 μ 를 최적화하는 단계(205)

μ 에 대해서 듀얼 함수(15)를 최적화하기 위하여, 사용자들이 $\mu_i(\lambda)$ 의 순서를 분류하게 한다. j^* 는 다음과 같이 되도록 하는 최소 정수라 하자.

$$\sum_{i=1}^{j^*} N_i \geq N$$

$$L(\lambda) := \min_{\mu \geq 0} L(\lambda, \mu) = \sum_{i=1}^{j^*-1} \mu_i(\lambda) N_i + \mu_{j^*}(\lambda) \left(N - \sum_{i=1}^{j^*-1} N_i \right) + \lambda P \quad (16)$$

그리고, 최소화하는 μ 는 다음 식으로써 주어진다.

$$\mu^*(\lambda) := \mu_{j^*}(\lambda) \quad (17)$$

상기 순서에 의해 $\mu_j(\lambda) \geq \mu_{j+1}(\lambda)$ 가 된다는 점에 유의하라 따라서, $\mu^*(\lambda)$ 는 임계값이다. $\mu_i(\lambda) > \mu^*(\lambda)$ 를 갖는 사용자들은 자신들의 전체 코드 성능을 획득하고 $\mu_i(\lambda) < \mu^*(\lambda)$ 를 갖는 사용자들은 아무것도 획득하지 못한다. $w_i \geq w_j$ 및 $e_i \geq e_j$ 일때

$\mu_i(\lambda) \geq \mu_j(\lambda)$ 이고 사용자(i)가 어떤 코드들을 사용자(j)에게 할당하기 전 전체 코드 할당을 제공받는다는 점에 유의하라. 이는 페어니스 파라미터 $\alpha=1$ ("max C/I 스케줄러") 및 모든 사용자들이 동일한 QoS 가중치를 가질 때, w_i 's는 사용자에게 대해서 일정하게 되고 동일하게 되어, 이들의 e_i 's의 순서에서 코드 버짓(code budget)내로 사용자들 패킹이 최적화된다는 것을 의미한다. 이는 본 발명의 또 다른 실시예의 설명과 관련하여 후술된다.

$\mu_{j-1}^*(\lambda) > \mu_j^*(\lambda) > \mu_{j+1}^*(\lambda)$ 경우에, 증가성을 지닌 합 코드 제약을 충족하고, $\mu = \mu^*(\lambda)$ 에 대한 라그랑지안을 최적화하는 특정한 가능한 $\mathbf{n}^*$ 이 존재한다. 이는 다음과 같이 주어진다.

$$n_i^* = \begin{cases} N_i, & i < j^* \\ N - \sum_{i=1}^{j^*-1} N_i, & i = j^* \\ 0, & i > j^* \end{cases} \quad (18)$$

그러나, 이것이 그 경우가 아니라면, 타이(tie)가 존재하고 라그랑지안을 최적화하는 다수의 $\mathbf{n}^*$ 이 존재한다. 그러나, (18) 및 (12)에 의해 주어진 $\mathbf{n}^*$ 및 얻어진 $\mathbf{p}^*$ 의 이들 선택들 전부가 가능하지는 않을 것이다. 가능성(feasibility)을 성취하기 위하여, 또한 부가적인 코드 및 전력 제약들을 충족하여야만 한다. $\mathbf{n}^*(\lambda)$ 가 상기 구성된 임의의 최대 $\mathbf{n}$ 을 가리킨다고 하자.

최적의 코드 할당은 다음의 특성들을 갖는 것으로 도시되어 있다.

1. $N_i = N$ 의 경우에, 기껏 2명의 사용자들이 스케줄될 것이다.
2. 모든 N_i 가 동일하다면, 기껏 $\lceil N/N_i \rceil + 1$ 사용자들이 스케줄될 것이다. 단지 2명의 사용자들은 자신들의 전체 코드 할당을 가질 것이다.
3. 일반적으로 단지 2명의 사용자들이 자신들의 전체 코드 할당을 가질 것이다.

2.1.3 $\mathbf{n}$ 을 고정된 채로 유지하면서 $\mathbf{r}$ 및 $\mathbf{p}$ 를 계산하는 단계 213

단계 213에서, 코드 할당 $\mathbf{n}$ 이 제공된 최적의 전력들에 대해 해결한다. $\mathbf{n}$ 은 단계들(203 및 205)과 관련하여 상술된 바와 같이 도출될 수 있거나, 대안적으로, $\mathbf{n}$ 은 사전 결정되어 전력-최적화 알고리즘으로 입력으로 제공될 수 있다. $\sum_i p_i \leq P$ 에 따라 다음과 같이 표시된다.

$$V^*(\mathbf{n}) := \max_{\{p_i \geq 0: p_i \leq s_i(n_i) n_i / e_i \forall i\}} V(\mathbf{n}, \mathbf{p})$$

이는 상술된 바와 같이 듀얼 포물레이션을 사용하여 $\lambda^*(\mathbf{n})$ 을 구하고 나서 최적의 $\mathbf{p}^*(\mathbf{n})$ 을 계산함으로써 해결된다. $\mathbf{n}$ 이 고정되어 있기 때문에, 부(negative)가 되지 않도록 한다는 것을 제외하면 함수 $n_i s_i(n_i)$ 에 대한 제약들은 존재하지 않는다. 따라서, 이 단계에서 수행되는 계산들에 레이트 제약들을 결합시킬 수 있다.

일반성(generality)을 손실함이 없이, 제로 코드 할당들을 지닌 어떠한 사용자들을 제거한다. M이 정의(positive) 코드 할당을 지닌 다수의 사용자들이라 하자. 우선 합 전력 제약이 비활성, 즉 $\sum_i p_i = \sum_i \frac{n_i}{e_i} s_i(n_i) \leq P$ 인지를 검사할 필요가 있다.

이것이 그 경우라면, 최적의 전력 할당들은 단지 개개의 전력 제약들 $p_i^* = \frac{n_i}{e_i} s_i(n_i)$ 이고 행해진다.

지금부터, 이 제약이 활성화되는 경우를 진행시킨다. 이 경우에, 합 전력 제약은 최적의 전력들을 위한 등가성으로 충족되어야만 한다는 것을 보여준다(그렇지 않다면, 사용자들의 전력들중 적어도 한 전력은 증가되어, 보다 높은 최초 값 함수로 된다). 현재 단지 합 전력 제약에서만 라그랑지안을 구성할 수 있다.

$$EQ L_p(\mathbf{p}, \lambda) := \sum_i w_i n_i \ln(1 + F(p_i, e_i, n_i)) + \lambda(P - \sum_i p_i)$$

$\sum_i n_i = N$ 경우에, 2개의 라그랑지안들은 동일하다. 듀얼 함수는 다음과 같이 주어진다.

$$L_p(\lambda) := \max_{\{\mathbf{p} \geq 0: p_i \leq s_i(n_i) n_i / e_i \forall i\}} L_p(\mathbf{p}, \lambda).$$

전력들에 대해서 최적화하기 위하여, 이 제약 집합은 함수들 $s_i(n_i) n_i$ 에 관계없이 항상 컨벡스된다. $\mathbf{p}$ 에 대해서 $L_p(\mathbf{p}, \lambda)$ 를 최대화하면 근본적으로 앞서의 장에서 언급된 바와 같은 $L(\mathbf{p}, \mathbf{n}, \lambda, \mu)$ 의 문제와 동일하게 된다. 최적의 $\mathbf{p}$ 는 전과 같이 (12)에 의해 다음과 같이 주어진다.

$$p_i^*(\mathbf{n}, \lambda) = \frac{n_i}{e_i} \left[\min \left\{ \frac{w_i e_i}{\lambda} - 1, s_i(n_i) \right\} \right]^+ \quad (19)$$

에그레이웰(Agrawal) 특허 공개에서, $\lambda^*(\mathbf{n})$ 은 대응하는 $\mathbf{p}^*(\mathbf{n}, \lambda)$ 가 $\sum_i p_i^* = P$ 를 충족하는 경우 및 이 경우에만 $L_p(\lambda)$ 를 최적화한다. (19)로부터 치환하면, $\lambda^*(\mathbf{n})$ 은 식의 어떤 해법에 의해 주어진다.

$$\lambda = \frac{\sum_i n_i w_i 1_{\left\{ \frac{w_i e_i}{1+s_i(n_i)} \leq \lambda < w_i e_i \right\}}}{P - \sum_i \frac{n_i s_i(n_i)}{e_i} 1_{\left\{ \lambda < \frac{w_i e_i}{1+s_i(n_i)} \right\}} + \sum_i \frac{n_i}{e_i} 1_{\left\{ \frac{w_i e_i}{1+s_i(n_i)} \leq \lambda < w_i e_i \right\}}} \quad (20)$$

따라서, 최적의 전력 할당 벡터 $\mathbf{p}^*(\mathbf{n})$ 는 다음과 같이 주어진다.

$$p_i^*(\mathbf{n}) = p_i^*(\mathbf{n}, \lambda^*(\mathbf{n})) = \frac{n_i}{e_i} \left[\min \left\{ \frac{w_i e_i}{\lambda^*(\mathbf{n})} - 1, s_i(n_i) \right\} \right]^+ \quad (21)$$

λ 가 $\lambda^*(\mathbf{n})$ 을 해결하는 식(20)의 양 측들상에서 발생되기 때문에, 이는 곤란한 것으로 보일 수 있다. 이에 대한 알고리즘 해법이 다음과 같이 제공된다.

사용자들 $i = 1, \dots, M$ 에 대해서, 다음과 같이 정의한다.

$$b_i = w_i e_i$$

$$a_i = \frac{w_i e_i}{1 + s_i(n_i)}.$$

$a_i < b_i$ 라는 점에 유의하라. 집합 $\{a_i, b_i; i=1, K, 2M\}$ 을 감소하는 수들의 집합 $\{x[l]; l=1, \dots, 2M\}$ 으로 분류하라.

각 $i=1, \dots, 2M$ 에 대해서, 또한, $x[l]=b_i$ 또는 $x[l]=a_i$ 인 경우, $y[l]=i$ 라고 하고,

$$EQ z[l] := \begin{cases} b_i & \text{if } x[l] = a_{y[l]}, \\ a_i & \text{otherwise} \end{cases}$$

라고 하자.

타이들은 임의적으로 분해된다. 다음에, 가장 큰 $\lambda \in \{x[l]; l=1, K, 2M\}$ 을 구하여 $\sum_i p_i^*(n, \lambda) \geq P$, 가 되도록 하는데, 즉 가장 작은 l 을 구하여 $\sum_i p_i^*(n, x[l]) \geq P$. 이 되도록 한다. 애그레이웰, 알.(Agrawal, R.), 서브래매니언, 브이.(Subramanian, V.), 및 베리, 알(Berry, R)이 영국 캠브리지에서 2004년 3월 WiOpt04 회의에서 발표한 "CDMA 시스템에서의 조인트 스케줄링 및 자원 할당(Joint Scheduling and Resource Allocation in CDMA Systems)"이란 제목의 논문(이하부터, "애그레이웰(Agrawal) 논문"이라 한다)에서, $\lambda^*(n) \in \{x[l], x[l-1]\}$ 을 보여주고 식(20)의 해법에 의해 제공된다. 게다가, 상기 범위에서 λ 에 대하여, 식(20)에서 표시자들은 λ 의 특정 값에 의존하지 않으므로 λ 에 대한 보다 용이한 계산가능한 표현을 갖는다.

상기 식(20)에 대한 해법을 구하는 알고리즘을 위한 의사-코드(pseudo-code)가 이하에 제공된다. 이 알고리즘 복잡성은 분류로 인해 $O(M \log M)$ 이 된다.

바람직한 실시예의 해법을 위한 의사-코드 $\lambda^*(n)$

```
lambda()
{
    Psum = 0;
    Ppossum = 0;
    Pmaxsum = 0;
    nwpossum = 0;

    for (i=1, i <= M, i++){
        pos[i] = 0;
        max[i] = 0;
    }

    l = 1;

    while (Psum < P && l < 2M){
        i = y[l]
        if (z[l] = 1){
            max[i] = 1;
            Pmaxsum +=  $\frac{n_i}{e_i} s_i(n_i)$ ;

            //Pmaxsum =  $\sum_i \left( \frac{n_i}{e_i} * s_i(n_i) * max[i] \right)$ 

            Ppossum -=  $\frac{n_i}{e_i}$ ;

            //Ppossum =  $\sum_i \left( \frac{n_i}{e_i} * (pos[i] - max[i]) \right)$ 

            nwpossum -=  $n_i * w_i$ ;
        }
        l++;
    }
}
```

```

//nwpossum =  $\sum_i (n_i * w_i * (pos[i] - max[i]))$ 
else{
    pos[i] = 1;
    Ppossum +=  $\frac{n_i}{e_i}$ ;
    //Ppossum =  $\sum_i (\frac{n_i}{e_i} * (pos[i] - max[i]))$ 
    nwpossum +=  $n_i * w_i$ ;
    //nwpossum =  $\sum_i (n_i * w_i * (pos[i] - max[i]))$ 
}
l++;
Psum =  $\sum_i (\frac{n_i}{e_i} * s^*(\frac{w_i e_i}{x[l]}, s_i(n_i)))$ ;
}
lambda = Ppossum/(P - Pmaxsum);
if (lambda > x[l-1] or lambda < x[l]){
    error!
}
else{
    return(lambda)
}
}

```

여기서 $s^*\left(\frac{w_i e_i}{\lambda}, s_i(n_i)\right)$ 는 식(13)에 정의되어 있다.

2.1.3.1 사용자마다 등가성 제약들

레이트가 고정될 때 예를 들어 재전송시에 발생될 수 있는 바와 같은, 어떤 사용자(i)에 대해, 비등가성 $p_i \leq e_i s_i(n_i) n_i$ 이 등가성 제약 $p_i = e_i s_i(n_i) n_i$ 로 대체되면, P로부터 P_i 를 감산한 후 문제로부터 이 사용자를 간단히 제거하고 전과 같이 진행한다.

2.1.3.2. 사용자마다 없는 전력 제약들

$s_i(n_i) = \infty$ 인 경우에, 다음 보다 간단한 알고리즘은 $\lambda^*(\mathbf{n})$ 을 구한다. 우선, 일반성을 손실함이 없이, 사용자들이 $w_i e_i$ 의 감소하는 순서에서 번호가 매겨진다고 추정하자. 다음을 계산하라.

$$EQ \setminus O(\lambda, S(\%))_j = \setminus F(\setminus \setminus su(i=1,j,) w_i n_i P + \setminus \setminus su(i=1,j,) \setminus F(n_i, e_i)) \quad \forall j \in \{1, 2, \dots, M\}.$$

다음과 같이 $\lambda^*(\mathbf{n})$ 을 결정한다.

1. j=1 설정
2. $w_{j+1} e_{j+1} > \%_j, j++$ 동안
3. $\lambda^* = \%_j$

2.1.4 λ^* 을 구하는 단계 209

λ 에 대해 $L(\lambda)$ 를 최소화하기 위하여 최소화하거나 시도되는 어떤 알고리즘이 고려된다. 바람직한 실시예에서, 반복적으로 행해지는데, 이 반복의 각 단계 k 에서, 우리는 $k=0$ 에서 시작하도록 하는 방식으로 최적의 λ^* 의 범위 $[\lambda_k^{LB}, \lambda_k^{UB}]$ 를

$$0 \leq \lambda_0^{LB} \leq \lambda_0^{UB} < \infty \text{로 좁히고, 어떤 } d \geq 1 \text{ 및 } 0 \leq \rho < 1 \text{에 대해서, } [\lambda_{(k+1)d}^{UB} - \lambda_{(k+1)d}^{LB}] \leq \rho [\lambda_{kd}^{UB} - \lambda_{kd}^{LB}] \quad \forall k \geq 0. \text{이다.}$$

다수의 이와 같은 방식들은 최적의 λ^* 에 기하학적으로 고속 수렴하도록 설계될 수 있다. 2가지 이와 같은 실시예들이 후술된다.

2.1.4.1 λ^* 를 구하기 위한 제1 실시예

이 알고리즘은 앞서 규정된 함수들 $L(\lambda)$ (11), $\mathbf{n}^*(\lambda)$ (18) 및 타이 레졸루션(tie resolution), $\lambda^*(\mathbf{n})$ (20), $\mathbf{p}^*(\mathbf{n})$ (21) 및 $V(\mathbf{p}, \mathbf{n})$ (4)을 사용한다. 함수 $\mathbf{r}(\mathbf{p}, \mathbf{n})$ 은 단지, (2)에서 제공된 전력들 및 코드들이 제공된 레이트이다. $\lambda^*(\mathbf{n})$ 을 계산하는 의사-코드들은 또한, 단계(213)에서 제공된다.

애그레이월(Agrawal) 논문에서, $L(\lambda)$ 가 컨벡스이고 최적의 $\lambda^* = \argmin_{\lambda \geq 0} L(\lambda) \in [\lambda^{\min}, \lambda^{\max}]$ 이라는 것을 보여주며, 여기서, $\lambda^{\min} = 0$ 이고 $\lambda^{\max} = \max_i w_i e_{i0}$ 이다

k 번째 반복에서, λ_k^* 로 주어진 λ^* 의 추정치를 지닌 범위 $[\lambda_k^{LB}, \lambda_k^{UB}]$ 에서 λ^* 를 식별한다. 어떤 시간 k 에서 이 알고리즘의 상태는 다음에 의해 주어진다.

$$S_k = \{ [\lambda_k^{LB}, \lambda_k^{LB}], [\lambda_k^*, \lambda_k^*, \mathbf{n}_k^*], [\lambda_k^{UB}, \lambda_k^{UB}] \}$$

단계 0은 이 알고리즘의 상태를 초기화하고 최적의 할당을 운 좋게 맞힌 경우 출구 검사(exit check)를 제공한다.

단계 1 이 알고리즘의 상기 상태에 대해 반복을 행하고 또한 출구 검사를 사용한다.

단계 2 부가적인 제약들을 고려하며, 전력들을 다시최적화하며, 레이트들/TBS 계산, 등등을 행함으로써 최종마무리한다.

이 알고리즘은 다음과 같이 작업한다.

단계 0 : 초기화

$$EQ \quad L_k \backslash s \backslash up6(*) = L(\lambda_k \backslash s \backslash up6(*)) \leq \min \quad \backslash b$$

$$(a) \lambda_0^{LB} = 0$$

$$(b) \text{적어도 한 사용자에게 대해 사용자 당 전력 제약들이 존재하지 않으면, } L(\lambda_0^{LB}) = +\infty$$

$$(c) \text{그 밖에는, } L(\lambda_0^{LB}) \text{를 계산}$$

$$(d) w_i e_i \text{의 감소 순서에서 사용자들을 분류}$$

$$(e) \sum_{i=1}^{j^*-1} N_i < N \leq \sum_{i=1}^{j^*} N_i \text{이 되도록 가장 작은 } j^* \text{를 구함}$$

(f) 다음의 n_0^* 를 정의

$$EQ\ n_{0,i}^{sup6(*)} = \min\{N_p, i \leq j^{sup6(*)}, N - \sum_{i=1}^{j-1} N_p, i=j^{sup6(*)}, 0, i > j^{sup6(*)}\}$$

(g) j^* 에 대한 타이의 경우에, 보다 큰 w_i 를 지닌 사용자를 선택하고 w_i s가 또한 동일한 경우, 둘 중 하나 선택

(h) $n_{j,0}^* = N_j^*$ 인 경우,

$$EQ\ P_{req} = \sum_{i=1}^{j^{sup6(*)}} F(n_{i,0}, e_i) \min((F(w_i e_i, w_{j^{sup5(*)}+1} e_{j^{sup5(*)}+1}) - 1), s_i)$$

(i) 그 밖에는,

$$EQ\ P_{req} = \sum_{i=1}^{j^{sup6(*)}} F(n_{i,0}, e_i) \min((F(w_i e_i, w_{j^{sup5(*)}} e_{j^{sup5(*)}}) - 1), s_i)$$

(j) $P_{req} \geq P$ 이면, 단계 2로 진행

(k) $n_{j,0}^* = N_j^*$ 이면, $\lambda_0^{UB} = w_{j+1}^* e_{j+1}^*$, 그 밖에는, $\lambda_0^{UB} = w_j^* e_j^*$.

(l) $L_0^{UB} = L(\lambda_0^{UB})$ 를 계산

(m) $L_0^{LB} \leq L_0^{UB}$ 이면, $\lambda_0^* = \lambda_0^{LB}$; $L_0^* = L_0^{LB}$.

(n) 그 밖에는, $\lambda_0^* = \lambda_0^{UB}$; $L_0^* = L_0^{UB}$.

단계1: ($k=0$, $k <$ 반복 제한, $k++$)에 대해 반복

(a) $L_k^* > \min(L_k^{LB}, L_k^{UB})$ 이면, 예러!

(b) $\lambda_{k+1}^{CAND} = \lambda^*(n_k^*)$ 를 계산

(c) $L_{k+1}^{CAND} = L(\lambda_{k+1}^{CAND})$, $n_{k+1}^{CAND} = n^*(\lambda_{k+1}^{CAND})$ 를 계산

(d) $\lambda_{k+1}^* = \lambda^*(n_{k+1}^{CAND})$, $p_{k+1}^{CAND} = p^*(n_{k+1}^{CAND})$ 를 계산

(e) $V(n_{k+1}^{CAND})$ 를 계산

(f) $V(n_{k+1}^{CAND}) > (1 - [\epsilon]) \min\{L_{k+1}^{CAND}, L_k^*\}$ 이면, 단계2로 진행

(g) $L_{k+1}^{CAND} < L_k^*$ 와 $\lambda_{k+1}^{CAND} > \lambda_k^{LB} + [\text{epsilon}](\lambda_k^{UB} - \lambda_k^{LB})$ 와 $\lambda_{k+1}^{CAND} < \lambda_k^{UB} - [\text{epsilon}](\lambda_k^{UB} - \lambda_k^{LB})$ 와 $\lambda_{k+1}^{CAND} \neq \lambda_k^*$ 이면,

보다 빠르게 트랙을 시도(TRY FASTER TRACK)

i. $\lambda_{k+1}^{CAND} < \lambda_k^*$ 이면,

$$\begin{aligned} \lambda_{k+1}^{UB} &= \lambda_k^* \\ \lambda_{k+1}^* &= \lambda_{k+1}^{CAND} \\ \lambda_{k+1}^{LB} &= \lambda_k^{LB} \end{aligned}$$

ii. 그 밖에는,

$$\begin{aligned} \lambda_{k+1}^{LB} &= \lambda_k^* \\ \lambda_{k+1}^* &= \lambda_{k+1}^{CAND} \\ \lambda_{k+1}^{UB} &= \lambda_k^{UB} \end{aligned}$$

(h) 그 밖에는,

이등분을 행함(DO BISECTION)

i. $\lambda_k^* - \lambda_k^{LB} \geq \lambda_k^{UB} - \lambda_k^*$ 이면,

A. $\lambda_{k+1}^{CAND} = (\lambda_k^{LB} + \lambda_k^*)/2$ 를 할당

B. $L_{k+1}^{CAND} = L(\lambda_{k+1}^{CAND})$, $n_{k+1}^{CAND} = n^*(\lambda_{k+1}^{CAND})$ 를 계산

C. $\lambda^*(n_{k+1}^{CAND})$, $p_{k+1}^{CAND} = p^*(n_{k+1}^{CAND})$ 를 계산

D. $V(n_{k+1}^{CAND})$ 를 계산

E. $V(n_{k+1}^{CAND}) > (1 - [\text{epsilon}]) \min\{L_{k+1}^{CAND}, L_k^*\}$ 이면, 단계 2로 진행

F. $L_{k+1}^{CAND} \leq L_k^*$ 이면, 다음과 같이 알고리즘 변경

$$\begin{aligned} \lambda_{k+1}^{UB} &= \lambda_k^* \\ L_{k+1}^{UB} &= L_k^* \\ \lambda_{k+1}^* &= \lambda_{k+1}^{CAND} \end{aligned}$$

$$\begin{aligned} L_{k+1}^* &= L_{k+1}^{CAND} \\ n_{k+1}^* &= n_{k+1}^{CAND} \\ \lambda_{k+1}^{LB} &= \lambda_k^{LB} \\ L_{k+1}^{LB} &= L_k^{LB} \end{aligned}$$

G. 그 밖에는, 다음과 같이 변경

$$\begin{aligned} \lambda_{k+1}^{LB} &= \lambda_{k+1}^{CAND} \\ L_{k+1}^{LB} &= L_{k+1}^{CAND} \\ \lambda_{k+1}^* &= \lambda_k^* \\ L_{k+1}^* &= L_k^* \\ n_{k+1}^* &= n_k^* \\ \lambda_{k+1}^{UB} &= \lambda_{k+1}^{UB} \\ L_{k+1}^{UB} &= L_k^{LB} \end{aligned}$$

ii. 그 밖에는,

A. $\lambda_{k+1}^{CAND} = (\lambda_k^{UB} + \lambda_k^*)/2$ 를 할당

B. $L_{k+1}^{CAND} = L(\lambda_{k+1}^{CAND}), n_{k+1}^{CAND} = n^*(\lambda_{k+1}^{CAND})$ 를 계산

C. $\lambda^*(n_{k+1}^{CAND}), p_{k+1}^{CAND} = p^*(n_{k+1}^{CAND})$ 를 계산

D. $V(n_{k+1}^{CAND})$ 를 계산

E. $V(n_{k+1}^{CAND}) > (1 - [\text{epsilon}]) \min\{L_{k+1}^{CAND}, L_k^*\}$ 이면, 단계 2로 진행

F. $L_{k+1}^{CAND} \leq L_k^*$ 이면, 다음과 같이 알고리즘 상태를 변경

$$\begin{aligned} \lambda_{k+1}^{LB} &= \lambda_k^* \\ L_{k+1}^{LB} &= L_k^* \\ \lambda_{k+1}^* &= \lambda_{k+1}^{CAND} \\ L_{k+1}^* &= L_{k+1}^{CAND} \\ n_{k+1}^* &= n_{k+1}^{CAND} \\ \lambda_{k+1}^{UB} &= \lambda_k^{UB} \\ L_{k+1}^{UB} &= L_k^{UB} \end{aligned}$$

G. 그 밖에는, 다음과 같이 변경

$$\begin{aligned} \lambda_{k+1}^{LB} &= \lambda_{k+1}^{LB} \\ L_{k+1}^{LB} &= L_{k+1}^{LB} \\ \lambda_{k+1}^* &= \lambda_k^* \\ L_{k+1}^* &= L_k^* \\ n_{k+1}^* &= n_k^* \\ \lambda_{k+1}^{UB} &= \lambda_{k+1}^{CAND} \\ L_{k+1}^{UB} &= L_{k+1}^{CAND} \end{aligned}$$

단계 2:

(a) 라운드 n^* - 2명의 사용자들은 타이의 경우에 비정수 할당들을 가질 수 있다는 점에 유의하라. 이 경우에, 코드 요구조건 당 보다 큰 전력을 갖는 사용자를 라운드 다운시키고 다른 한 사용자를 라운드 업한다.

(b) 레이트 제약들을 포함한 모든 제약들을 고려하면서 $\mathbf{p}^*(\mathbf{n}^*)$ 을 재계산

(c) $r_i^* = \frac{240000}{\ln 2} n_i^* \ln(1 + \frac{p_i^* e_i}{n_i^*})$ 및 $TBS = r_i \times 0.002$ 로 제공된 레이트 $\mathbf{r}^* = \mathbf{r}(\mathbf{p}^*, \mathbf{n}^*)$ 을 계산. 최근접 바이트 또는 심지어 어떤 코서 양자화(coarser quantization)로 라운드 다운.

단계 1에서 단계들 (b)-(g)를 스킵하여 순수한 바이섹션 기반으로한 탐색 알고리즘을 획득한다.

2.1.4.2 λ^* 를 구하는 제2 실시예

최적의 λ^* 를 구하는 제2 실시예는 $L(\lambda)$ 의 서브그라디언트들을 사용하여 이의 최소화를 지원한다. 어떤 주어진 λ 에 대해, 서브그라디언트는 다음과 같이 손쉽게 구해질 수 있다.

1. 우선 식(18)에서 $\mathbf{n}^*(\lambda)$ 를 구하라. "타이"의 경우에, (식(18)이후에 설명된 바와 같이)등가성을 지닌 합 전력 제약을 충족시키고 라그랑지안을 최적화하는 모든 가능한 코드 할당들이 고려된다.

2. 다음에, $\mathbf{n}^*(\lambda)$ 을 사용하여, 식(12)에 의해 제공된 바와 같은 대응하는 $\mathbf{p}^*$ 를 구하라.

3. $P - \sum_i p_i^*$ 는 λ 에서 $L(\lambda)$ 에 대한 서브그라디언트 이다.

4. $L(\lambda)$ 가 컨벡스이기 때문에, λ^* 는 0이 $L(\lambda^*)$ 의 서브그라디언트인 경우 그리고 이 경우에만 최적화될 것이다.

5. 따라서, 0 서브그라디언트를 지닌 λ 를 수치적으로 탐색함으로써 $L(\lambda)$ 를 최소화하는 최적의 λ^* 를 구할 수 있다.

2.2 다른 실시예들

상기 결과들에 기초하여, 스케줄러/최적화기(101)는 다음과 같이 $\mathbf{r}$, $\mathbf{n}$ 및 $\mathbf{p}$ 에 대한 적절한 값들을 결정한다.

1. 모든 입력 파라미터들(P , N , N , R , R/N , P , S , e , W , Q , c , b)을 수집하라(단계 301).

2. $\mathbf{n}$ 을 계산하라(단계 303)

3. $\mathbf{n}$ 을 고정된채로 유지하면서 최적의 $\mathbf{p}$ 및 $\mathbf{r}$ 을 계산하라(단계 305)

이는 본 발명의 또 다른 실시예를 따른 스케줄러/최적화기(101)의 동작을 도시한 순서도인 도3에 도시되어 있다. 논리 흐름은 단계(301)에서 시작되는데, 여기서 $P, N, N, R, R/N, P, S, e, W, Q, c, b$ 는 최적화기(101)에 의해 수집된다. 제1 실시예에서 단계들 2-5는 실제로 최적의 n 계산을 지원한다는 것을 주지하라. 입력 파라미터들($P, N, N, R, R/N, P, S, e, W, Q, c, b$)에 기초하여 n 을 계산하는 어떤 다른 방법이 고려된다(단계 303). 이 때, r 및 p 가 n 을 고정된채로 유지하면서 계산된다(단계 305). 특히, n 은 단계(303)에서 구해진 값으로 고정되고, p 및 r 은 실제 레이트 제약(R)을 고려하면서 식들 (21) 및 (2)에 따라서 계산된다. 최종적으로, 단계(307)에서, n, p 및 r 에 대한 값들은 제어기(103)로 출력되고 데이터를 스케줄하여 송신기들(105)을 통해서 이 데이터를 전송하도록 사용된다.

입력 파라미터들(e, P, N, N 및 S)에 기초하여 n 을 계산하는 또 다른 방법이 이하에 제공된다.

1. w_i 및 e_i 를 포함하는 입력 파라미터들의 함수인 어떤 메트릭에 기초하여 사용자들을 분류하라.

2.
$$\sum_{i=1}^{j^*-1} N_i < N \leq \sum_{i=1}^{j^*} N_i$$
 되도록 가장 작은 j^* 를 구하라.

3. 다음과 같이 n_0 를 정의하라.

$$EQ_{n_0, j^*} = \min_{\{N_i, i \leq j^*\}} \{ \sum_{i=1}^{j^*} N_i - \sum_{i=1}^{j^*} N_i \}$$

4. j^* 에 대한 타이의 경우에, 보다 큰 w_i 를 지닌 사용자를 선택하고 w_i s가 또한 동일하다면, 어느 하나를 선택하라.

5. 선택적으로, 또한 앞서의 장에서 서술된 알고리즘의 한 번 이상 반복 실행할 수 있다.

본 발명이 특정 실시예들과 관련하여 도시되고 설명되었지만, 당업자는 본 발명의 원리 및 영역을 벗어남이 없이 각종 변경들을 행할 수 있다는 것을 이해할 것이다. 예를 들어, 상기 제공된 예들이 HSDPA 시스템 내에서 구현들과 관계하였지만, 코드들 및/또는 전력이 할당될 필요가 있는 임의의 통신 시스템에서 상기 실시예들이 수행될 수 있다는 것을 당업자는 인지할 것이다. 이와 같은 변경들은 이하의 청구범위의 영역내에 포함된다.

발명의 효과

본 발명은 시스템 수행성능을 최적화하는 통신 시스템 내에서 스케줄링 및 자원 할당을 위한 방법 및 장치를 제공한다.

도면의 간단한 설명

도1은 본 발명의 일 실시예를 따른 전송 회로의 블록도.

도2는 본 발명의 일 실시예를 따른 도1의 회로의 동작을 도시한 순서도.

도3은 본 발명의 또 다른 실시예를 따른 도1의 회로의 동작을 도시한 순서도.

도면의 주요 부분에 대한 부호의 설명

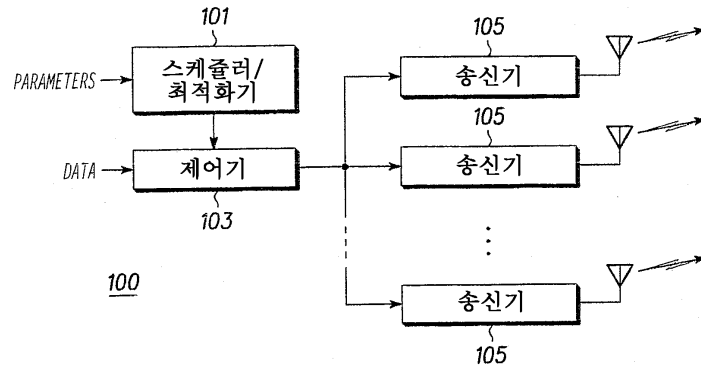
101: 스케줄러/최적화기

103: 제어기

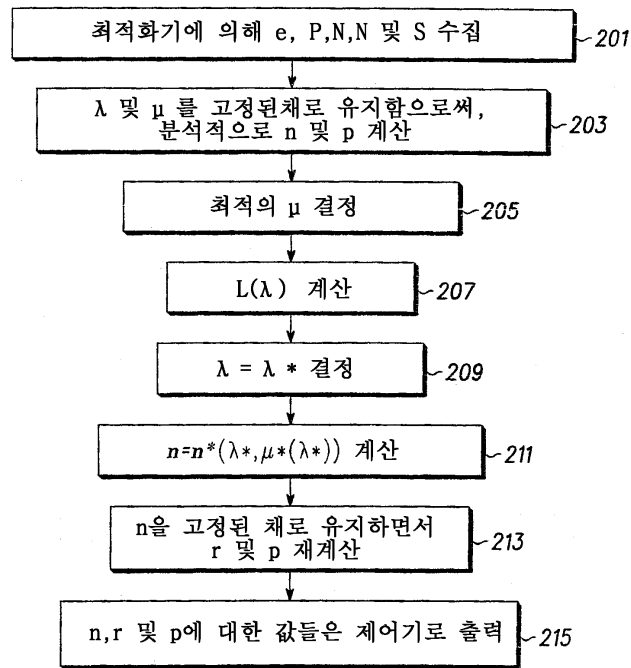
105: 송신기

도면

도면1



도면2



도면3

