STEM+: A Fair and Efficient Algorithm for Scheduling on the DSCH of UMTS Networks

R. M. Karthik and Joy Kuri
Centre for Electronics Design and Technology
Indian Institute of Science, Bangalore
{rmkartik,kuri}@cedt.iisc.ernet.in

***Abstract*— In Universal Mobile Telecommunication Systems (UMTS), the Downlink Shared Channel (DSCH) can be used for providing streaming services. The traffic model for streaming services is different from the commonly used *continuously-backlogged* model. Each connection specifies a required service rate over an interval of time, k, called the "control horizon". In this paper, our objective is to determine how k DSCH frames should be shared among a set of I connections. We need a scheduler that is *efficient* and *fair* and introduce the notion of *discrepancy* to balance the conflicting requirements of aggregate throughput and fairness. Our motive is to schedule the mobiles in such a way that the schedule minimizes the discrepancy over the k frames. We propose an optimal and computationally efficient algorithm, called STEM+. The proof of the optimality of STEM+, when applied to the UMTS rate sets is the major contribution of this paper. We also show that STEM+ performs better in terms of *both* fairness and aggregate throughput compared to other scheduling algorithms. Thus, STEM+ achieves both fairness and efficiency and is therefore an appealing algorithm for scheduling streaming connections.**

I. Introduction and Related Work

UMTS [1] is one of the major third generation mobile communication systems, which uses the WCDMA as the radio access scheme. We consider the Downlink Shared Channel (DSCH) [1], a time-shared channel for high bit-rate transmission. The DSCH capacity is time-shared on a frame-by-frame basis, with each frame having duration, $\tau = 10\ ms$.

A streaming application (for example, video on demand) results in a data flow that is almost entirely one-way. We consider I mobiles in a cell who specify the "contracted" rates, $r_1, r_2, \ldots, r_I$, respectively. The DSCH must be shared among the I flows over k frames (a period referred to henceforth as the "control horizon"), in such a way that each receives a total service as close as possible to its contracted service, while at the same time ensuring high system efficiency. The required service for user i is $r_i k\tau$. Because of the contracted rates of the mobiles and the explicit *control horizon*, our traffic model is different from the widely used *continuously-backlogged* model. We introduce the notion of "discrepancy criterion" to capture the difference between the required service and the received service. We use the norm, $f(.)$ in I-dimensional space to measure the discrepancy between the received and required service. Clearly, we require a small value of discrepancy. Thus, our objective is to schedule the mobiles so as to minimize the discrepancy over the control horizon. We use the terms "discrepancy" and the "objective function" interchangeably. In Section V, we show how the choice of $f(.)$ influences scheduling decisions, and thereby allows us to trade off fairness against efficiency. We assume that Rayleigh fading and channel variations due to shadow fading and user mobility are absent over the control horizon. Similar assumptions have been made by several authors [2].

The problem of scheduling on the downlink of wireless networks has received enormous attention in the literature. Some papers consider the scheduling of *data* traffic, utilizing the knowledge of the file sizes to be transferred [2], while others [3], [4] *adapt* the wireline packet-fair queuing algorithms to the wireless context. Some papers [5], [6] propose scheduling algorithms and evaluate their performance by simulations. Our approach is different in that we do not start with algorithms and then obtain their performance metrics. Rather, we start with an optimization problem and propose an algorithm to solve it. Moreover, our interpretation of fairness is different from that considered in most papers. In [7], the problem of intra-cell scheduling was considered, with the objective of minimizing the total energy consumed over the control horizon T. While there is no consideration of fairness in [7], we explicitly consider fairness through the discrepancy criterion. In [8], the opportunistic scheduling problem with short term fairness constraints was considered for a continuously backlogged model. However, the problem was not solved with contracted rates and finite control horizon.

In this paper, we propose STEpwise Minimizer+ (STEM+), an optimal and computationally efficient scheduling algorithm for minimizing the discrepancy, when applied to UMTS Rate Sets. The same problem was attempted in [9]. However, the algorithm proposed there, was optimal only for a subset of the DSCH rate set, whereas STEM+ is optimal for the entire rate set. We compare the objective function value and the aggregate throughput achieved using STEM+ with those of Adaptive Proportional Fairness Scheduler (APFS) proposed in [10], the Credit Based Algorithm (CREDIT) proposed in [11] and Wireless Credit Based Fair Queuing (WCFQ) proposed in [12]. We show that STEM+ outperforms the others in both fairness (lower discrepancy) as well as system efficiency (higher aggregate throughput).

The remainder of the paper is organized as follows. Section II defines the problem and the Exhaustive Search method. Section III proposes STEM+ and Section IV proves its optimality. Section V compares STEM+ with other scheduling algorithms and Section VI concludes the work.

II. Problem Formulation and Exhaustive Search

A. Problem Definition

We assume that a DSCH has been set up between the Radio Network Controller (RNC) and the mobiles. There are I mobiles. The sets of feasible rates for the I mobiles are $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_I$ respectively, where $\mathcal{S}_i$ has the structure, $\mathcal{S}_i = \{0, a, 2a, 2^2a, \ldots, 2^{(m_i-1)}a\}$. This structure of the rate set is specific to UMTS technology [1] and we refer to such sets as "UMTS rate sets". m_i is the number of rates feasible for user i and each rate is twice the immediately lower one. We have k frames and need to obtain scheduling decisions for each frame. The time duration between the epochs l and $(l+1)$ defines the frame $(l+1)$, $0 \le l \le (k-1)$. The scheduling decision at epoch l determines the mobile and the rate at which it should be scheduled in frame $(l+1)$. Let $j_i(l)$, $0 \le l \le (k-1)$, $1 \le i \le I$, be binary decision variables that indicate whether mobile i is scheduled in frame $(l+1)$ and $x_i(l)$ denote the rate assigned to i (if it is scheduled) in frame $(l+1)$. The vector of required services $(r_1k\tau, r_2k\tau, \ldots, r_Ik\tau)^t$:=$\boldsymbol{r}(k)$ can be thought of as a point in I-dimensional space, where $(.)^t$ indicates the transpose of a row vector. Let $S_i(k) = \tau \sum_{l=0}^{k-1} j_i(l)x_i(l)$, indicate the total service allotted to mobile i up to and including frame k. Then, $(S_1(k), S_2(k), \ldots, S_I(k))^t$:=$\boldsymbol{S}(k)$ is the vector of allotted services up to frame k. The p^{th} norm, $f(\boldsymbol{r}(k), \boldsymbol{S}(k)) = \left(\sum_{i=1}^{I} |r_ik\tau - S_i(k)|^p\right)^{1/p}$, $p \ge 1$ can be used to measure the discrepancy between the required and the allotted service.

The overall objective is to obtain the *best* approximation to the vector $\boldsymbol{r}(k)$, *i.e.,* determine the vector $\boldsymbol{S}(k)$ which is closest to $\boldsymbol{r}(k)$, using the sets $\mathcal{S}_i, i = 1, 2, \ldots, I$. With the above objective, we pose the following problem:

$$\text{minimize} \quad f(\boldsymbol{r}(k), \boldsymbol{S}(k))$$

subject to

$$x_i(l) \in \mathcal{S}_i, l = 0, 1, \ldots, (k-1), i = 1, 2, \ldots, I \quad (1)$$

$$\sum_{i=1}^{I} j_i(l) \le 1, l = 0, 1, \ldots, (k-1), i = 1, 2, \ldots, I \quad (2)$$

$$j_i(l) \in \{0, 1\}$$

The unknowns to be solved for are $j_i(l)$, and $x_i(l)$, $0 \le l \le (k-1)$, $1 \le i \le I$. The constraint in Equation (2) ensures that at most *one* mobile is scheduled in each frame.

B. Exhaustive Search

A naive approach for solving the above problem is the exhaustive search procedure. If m_i is the number of rates feasible for user i, then the number of points to be searched to find the optimal solution can be shown to be $\binom{\sum_{i=1}^{I} m_i + k - 1}{k}$. Obviously, the computational complexity increases drastically with increasing I and k. In practice, heuristics are required.

III. The STEM+ (STEpwise Minimizer+) Algorithm

We explain STEM (STEpwise Minimizer) and STEM+, its augmented version in this section. The *updated requirements* of mobile i at epoch l in STEM and STEM+ are denoted by $u_i(l)$ and $u_i^{(S)}(l)$ respectively. The *updated requirements* of mobile i at epoch l, $u_i(l)$, is equal to its required service, $r_ik\tau$ minus the service allotted to it till epoch l, $S_i(l)$. $u_i(l) < 0$ means that the service allotted to mobile i till epoch l exceeds its requirement. The vector of updated requirements of mobiles at epoch l in STEM and STEM+ are denoted by $\boldsymbol{u}(l) = (u_1(l), u_2(l), \ldots, u_I(l))^t$ and $\boldsymbol{u}^{(S)}(l)$, respectively. U denotes the set of users considered for allocation of the frames and $c(U)$ denotes its cardinality.

A. STEM (STEpwise Minimizer) Algorithm

Let $l \in \{0, 1, \ldots, (k-1)\}$ and $\boldsymbol{u}(l)$ be given. At epoch l, STEM determines the *best one step* approximation to $\boldsymbol{u}(l)$, using rates from the set $\mathcal{S}_i$ for each user i. Let $i^*(l)$ indicate the mobile chosen at epoch l and $x_{i^*(l)}(l)$ be the rate assigned to it. STEM obtains $i^*(l)$ and $x_{i^*(l)}(l)$ as the solution of

$$\text{minimize} \quad f\Big(\boldsymbol{u}(l), (\tau j_1(l)x_1(l), \ldots, \tau j_I(l)x_I(l))^t\Big) \quad (3)$$

subject to Equations 1 and 2 for a given l over U. The updated requirements are altered as

$$u_{i^*(l)}(l+1) = u_{i^*(l)}(l) - \tau x_{i^*(l)}(l)$$
$$u_i(l+1) = u_i(l), \quad i \ne i^*(l)$$

Now Equation 3 is solved at successive epochs $(l+1), (l+2), \ldots, (k-1)$. STEM finds the solution after a search among $k\sum_{i=1}^{I} m_i$ points. Now we state Proposition 3.1, which specifies the condition under which STEM chooses the same action as a given optimal policy.

Proposition 3.1: Given that at epoch l, STEM chooses $i^*(l)$, the quantum of service is $q^*_{i^*(l)}$ and $\left(u_{i^*(l)}(l) - q^*_{i^*(l)}\right) > 0$, then this action, $\left(i^*(l), x_{i^*(l)}(l)\right)$ chosen by STEM is also chosen by the optimal policy.

That is, as long as the action chosen by STEM leaves a *positive* updated requirement at the next epoch, the action is optimal. The proof is based on the following argument: Given any policy, where the action at epoch l is different from $i^*(l)$ and $q_{i^*(l)}$, keeping all the other users' (excluding $i^*(l)$) allocations unchanged, we can show that the objective function can be strictly improved by assigning $q^*_{i^*(l)}$ to $i^*(l)$.

If $\boldsymbol{u}(l)^t > \boldsymbol{0}, \forall l \in \{0, 1, \ldots, (k-1)\}$, then STEM is optimal applying Proposition 3.1 repeatedly. If $\boldsymbol{u}(l)^t \not> \boldsymbol{0}$, STEM does not result in the optimal value of the discrepancy always. Table I shows 2 examples, where STEM is not optimal. The 1$^{\text{st}}$ column shows the control horizon k. This is followed by the Rate Set for the users and the contracted rates in the next two columns. "Discrepancy" refers to the discrepancy evaluated using the Exhaustive Search method and STEM. We now explain how to use $\boldsymbol{u}(l)$, the number of frames remaining, $rem = (k-l)$ and $\mathcal{S}_i$, $i = 1, 2, \ldots, I$ to obtain the optimal policy, STEM+. STEM+ is explained in subsection III-C.

B. Single user with negative updated requirement: towards the optimal policy for a single user

Consider a single user with feasible rate set $\mathcal{S}_i$. Let $\tau = 1$ and $u(l) < 0$. $u(l) < 0$ means the allocation at epoch $(l-1)$, $x(l-1)$ is greater than $u(l-1)$. The number of frames remaining, $rem = (k-l)$.

k	RateSet (kbps)	r_i(kbps)	Discrepancy	
			Exhaust	STEM
5	{30, …1920} {30} {30} {30,…,240}	368.000 12.000 10.000 40.000	877.496	916.515
7	{30, …960} {30,…480} {30, …480} {30, …480}	128.571 74.286 74.286 71.429	600.000	624.500

TABLE I

COMPARING EXHAUSTIVE SEARCH WITH STEM.

Epoch	Allocation
$(l-1)$	$\frac{x(l-1)}{2}$
l	$\frac{1}{2}\frac{x(l-1)}{2}$
$\vdots$	$\vdots$
$(l+b-1)$	$\frac{1}{2^b}\frac{x(l-1)}{2}$

TABLE II

ALLOCATIONS IN THE MODIFIED POLICY

Since $x(l-1)$ was the optimizing action, we must have $(u(l-1) - x(l-1)/2) > (x(l-1) - u(l-1))$. Therefore $u(l-1) > \frac{3x(l-1)}{4}$. Hence $u(l-1) \in \left(\frac{3x(l-1)}{4}, x(l-1)\right)$. We can partition $\left(\frac{3x(l-1)}{4}, x(l-1)\right)$ into sub-intervals as shown in Fig 1.

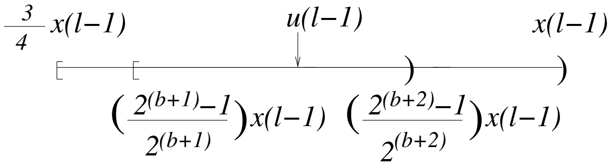

Fig. 1. Subinterval in which $u(l-1)$ lies

Proposition 3.2: If $|u(l)| > a/2$ and

$$\left(\frac{2^{(b+1)}-1}{2^{(b+1)}}\right)x(l-1) \leq u(l-1) < \left(\frac{2^{(b+2)}-1}{2^{(b+2)}}\right)x(l-1) \quad (4)$$

then there *is* a modified policy with the allocations from epochs $(l-1)$ to $(l+b-1)$ given in Table II to obtain an updated requirement at epoch $(l+b)$, $u_{mod}(l+b)$, such that

1) $u_{mod}(l+b) \geq 0$ and it is the least possible value of the updated requirement at epoch $(l+b)$ and
2) b is the minimum number of additional frames required to obtain $u_{mod}(l+b)$, *i.e.,* $b = min\{e : 0 \leq u_{mod}(l+e) < |u(l)|\}$.

The sub-interval's left end point is obtained by adding the values in the set $\{\frac{x(l-1)}{2}, \frac{x(l-1)}{4}, \ldots, \frac{x(l-1)}{2^{(b+1)}}\}$. Similarly, the right end point is equal to $\left(\frac{x(l-1)}{2} + \frac{x(l-1)}{4} + \ldots + \frac{x(l-1)}{2^{(b+2)}}\right)$. All the actions till epoch $(l-2)$, obtained using STEM are retained. The allocations in Table II are obtained by approximating the modified updated requirement at each epoch, $(l-1), l, \ldots, (l+b-1)$ from below, using continuously shrinking rate sets. For example, the rate sets to be used at epochs l and $(l+b)$ are $R = \{0, a, \ldots, x(l-1)/4\}$ (removing the rate, $x(l-1)/2$ used at epoch $(l-1)$) and $L = \{0, a, \ldots, \frac{x(l-1)}{2^{(b+2)}}\}$, respectively. Now we state Proposition 3.3, which specifies the conditions, at which the optimal point is reached.

Proposition 3.3: For the single user problem, if any one of the following 3 conditions occurs, the optimal policy is apparent immediately: *(i)* $b \geq rem$, *(ii)* $rem = 1$ and *(iii)* The rate set to be used for the user shrinks to $\{0, a\}$. We refer to conditions *(i)* – *(iii)* as "boundary conditions".

In case *(i)*, any further allocation will only increase the discrepancy. This follows from Proposition 3.2. Case *(ii)* corresponds to the 1-step problem for which STEM is optimal, while in case *(iii)*, even a single allocation from $\{0, a\}$ will push the updated requirement to the minimum possible value.

C. STEM+ explanation

The STEM+ algorithm is designed by extending the modified policy described in Section III-B to the multiple user case. If $|u_i^{(S)}(l)| < \tau a/2$ for user i, the discrepancy for the i^{th} user cannot be reduced and therefore, we set $u_i^{(S)}(k)$ to $u_i^{(S)}(l)$ and apply STEM+ only on the set $U \setminus \{i\}$ in all the remaining epochs. Let $\boldsymbol{u}(l)^t \ngeq \mathbf{0}$ for the first time at epoch l and $u_n(l) < 0$ for some n. $u_n(l) < 0$ means that user n has been selected at epoch $(l-1)$ by STEM and $\tau x_n(l-1) > u_n(l-1)$. If

$$\left(\frac{2^{(b_n+1)}-1}{2^{(b_n+1)}}\right)x_n(l-1)\tau \leq u_n(l-1) < \left(\frac{2^{(b_n+2)}-1}{2^{(b_n+2)}}\right)x_n(l-1)\tau \quad (5)$$

by Proposition 3.2, b_n additional frames are required to obtain $u_n^{(S)}(l+b_n)$, such that $0 \leq u_n^{(S)}(l+b_n) < |u_n(l)|$. The number of frames remaining is $rem = (k-l)$. The value V_o is calculated by applying STEM+ on the set $U \setminus \{n\}$ for the rem frames from epoch l to $(k-1)$ and the corresponding schedule is stored. There are 3 possibilities: *(i)*: $b_n > rem$, *(ii)*: $b_n = rem$, *(iii)*: $b_n < rem$. If $b_n > rem$, then the objective function is given by V_o and the corresponding schedule is used.

If $b_n = rem$, then V_n is calculated by allocating rem frames to user n after changing the allocation at epoch $(l-1)$ to $x_n(l-1)/2$. The objective function value Obj is calculated as the minimum of $\{V_n, V_o\}$ and the schedule corresponding to the smaller of V_n and V_o is chosen.

If $b_n < rem$, then V_n is calculated as the objective function obtained by allocating the first b_n out of rem frames to user n (after changing the allocation at epoch $(l-1)$ to $x_n(l-1)/2$) and applying STEM+ on U for $(rem - b_n)$ frames. $Obj = min\{V_n, V_o\}$ and the schedule corresponding to the smaller of V_n and V_o is chosen.

IV. OPTIMALITY OF STEM+

In the proof of optimality of STEM+, we make use of the fact that when $\boldsymbol{u}(l)^t \ngeq \mathbf{0}$ for the first time at epoch l, all the actions till epoch $(l-2)$ are still part of the optimal policy.

Proposition 4.1: For given k, r_1, r_2, …, r_I and $\mathcal{S}_i$, $i = 1, 2, \ldots, I$, STEM+ results in optimal $f(\boldsymbol{r}(k), \boldsymbol{S}(k))$.

Proof of Proposition 4.1: The proof is based on induction. Assume that STEM+ results in optimum $f(\boldsymbol{r}(k), \boldsymbol{S}(k))$ when applied to $(M-1)$ users for given k, $\boldsymbol{r}(k)$ and $\mathcal{S}_1, \ldots, \mathcal{S}_{(M-1)}$. Suppose STEM+ is applied to M users with the M^{th} user having a requirement $r_M k\tau$ and feasible rate set $\mathcal{S}_M$ for the same k and the values for the other users unchanged.

If because of M, $\boldsymbol{u}(l)^t > \mathbf{0}$, $\forall l \in \{0, 1, \ldots, (k-1)\}$, then STEM+ is optimal (by Proposition 3.1). Let l be the first epoch at which $\boldsymbol{u}(l)^t \not> \mathbf{0}$ and $u_n(l) < 0$ for some n. V_o is the objective function value evaluated by applying STEM+ on the set $U \setminus \{n\}$ for the remaining frames, $rem = (k-l)$. If $|u_n(l)| < \tau a/2$, then user n can be omitted from the set of users U, and V_o is the objective function value, which is optimal by the induction hypothesis. We have to prove optimality of STEM+ when $|u_n(l)| > \tau a/2$ and b_n is known, for the 3 CASES *(i)* $b_n > rem$, *(ii)* $b_n = rem$ and *(iii)* $b_n < rem$.

CASE *(i)*: Here, user n is not considered for any further allocation since the discrepancy for user n cannot be reduced, and the objective function value is V_o. By the induction hypothesis, STEM+ results in the optimum value of $f(\boldsymbol{r}(k), \boldsymbol{S}(k))$.

CASE *(ii)*: $V_n = f(\boldsymbol{u}^{(S)}(k), \mathbf{0}^t)$ with $u_i^{(S)}(k) = u_i(l), i \neq n$ and $u_n^{(S)}(k) = u_n(l) + \frac{x_n(l-1)\tau}{2^{(b_n+1)}}$. STEM+ calculates the discrepancy as $Obj = min\{V_n, V_o\}$. The paths that can be traced by the optimal policy are shown in Fig. 2. If $V_o = min\{V_n, V_o\}$, then STEM+ is optimal by the induction hypothesis, because we have a problem with $(M-1)$ users. Otherwise, $V_n = min\{V_n, V_o\}$ implies that the overall objective function can be reduced below V_o by reducing user n's discrepancy, and this is possible only by allocating *all* the remaining frames to user n. Thus in either case, STEM+ is optimal.

CASE *(iii)*: V_n is the objective function value, if b_n frames are allocated to user n and STEM+ is applied to the set U for $(rem - b_n)$ frames. Here, $u_n^{(S)}(l + b_n) = u_n(l) + \frac{x_n(l-1)\tau}{2^{(b_n+1)}}$, $u_i^{(S)}(l + b_n) = u_i(l)$, $i \neq n$ and $\boldsymbol{u}^{(S)}(l + b_n) > \mathbf{0}^t$. Now applying STEM+ on the set U for the $(rem - b_n)$ frames, several times if necessary, ultimately one of the boundary conditions mentioned in Proposition 3.3 will be reached. Therefore using Proposition 3.1, V_n is the least objective function value if the first b_n frames out of rem frames are allocated to user n and STEM+ is applied on U for the remaining $(rem - b_n)$ frames. STEM+ calculates Obj as $min\{V_n, V_o\}$. The paths that can be followed by the optimal policy after epoch $(l-2)$ are shown in Fig. 2. Applying the same argument as in CASE *(ii)*, we see that STEM+ ensures optimality.

V. Performance comparison

Here, we compare STEM+ with the Exhaustive search method as well as the scheduling policies APFS, CREDIT and WCFQ and show that it performs better in terms of computational efficiency, fairness and the aggregate throughput.

A. STEM+ with Exhaustive Search method

We compare STEM+ with Exhaustive Search in terms of the discrepancy and the execution time using the DSCH rate sets. We show that STEM+ is optimal and computationally efficient.

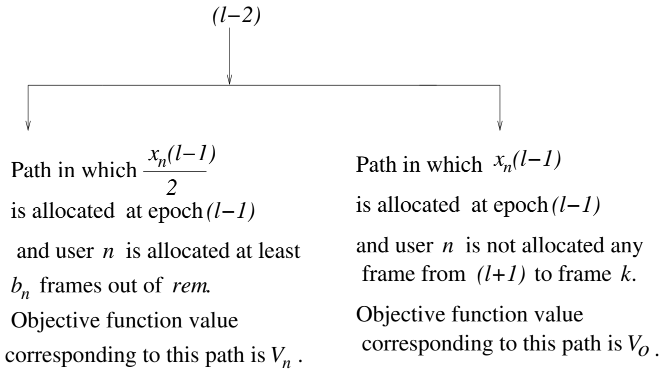

Fig. 2. Paths that can be followed by the optimal policy after epoch $(l-2)$ in CASES *(ii)* and *(iii)*.

Table III compares STEM+ and the Exhaustive Search method in terms of the execution time. The Euclidean norm is used for the discrepancy measure. All the entries except "Time" carry the same meaning as in Table I. 'Time" indicates the average execution time. The remaining columns show the performance comparison for different sets of r_i and $\mathcal{S}_i$, $i = 1, 2, \ldots, I$ for the same k. I and II refer to the groups of users. The first comparison with $k = 6$ shows the perfect case where $f(\boldsymbol{r}(k), \boldsymbol{S}(k)) = 0$. Here, the allotted service of each mobile is equal to its requirement and the two points $\boldsymbol{r}(k)$ and $\boldsymbol{S}(k)$ coincide. We reiterate that Exhaustive search searches a total of $\binom{\sum_{i=1}^{I} m_i + k - 1}{k}$ points while STEM+ in certain cases, obtains the optimum value by searching $k \sum_{i=1}^{I} m_i$ points, which shows the enormous difference in complexity. This is validated from Table III from the execution times, which are very large for Exhaustive search for $k = 17$ and for higher values of I and m_i, while they are negligible for STEM+.

B. Numerical comparison with APFS, CREDIT and WCFQ

We compare the performance of STEM+ with APFS, CREDIT and WCFQ in terms of the discrepancy and the aggregate throughput provided to the users, since they provide a measure of fairness and system efficiency respectively. Fig. 3 shows how STEM+ performs better compared to APFS, CREDIT and WCFQ under different channel conditions. STEM+ achieves a lower discrepancy and higher aggregate throughput. Lower discrepancy means fairer operation, while higher aggregate throughput implies more efficient operation.

Table IV shows the discrepancy and the aggregate throughput achieved with 1-norm, Euclidean norm, 4-norm and the infinity norm. The first 3 columns carry the same meaning as in Table III. The terms, "objective function value" and "aggregate throughput" refer to the discrepancy and the sum of the throughputs achieved by all users, respectively. We observe that the discrepancy and the aggregate throughput are non-increasing from left to right of Table IV. Lower values of $f(.)$ imply fairer operation, at the cost of lower aggregate throughput. Depending on whether fairness or efficiency is more important, we can choose the corresponding norm.

k	$\mathcal{S}_i$ (kbps)	r_i (kbps)	Discrepancy		Time (ms)		$\mathcal{S}_i$ (kbps)	r_i (kbps)	Discrepancy		Time (ms)	
			Exhaust	STEM+	Exhaust	STEM+			Exhaust	STEM+	Exhaust	STEM+
6	{30, 60} {30,...,240} {30,...,120}	20.00 15.00 10.00	0	0	50	0.4	I:{30,60,120} II:{30,...,240} I:6 II:4 users	12.50 228.75	20474	20474	47740	45
	{30,...,960} {30,...,480} {30,60} {30,...,240}	460.15 675.36 174.23 268.21	29893	29893	1300	0.7	{30,...,480} {30,...,240} {30,...,480} {30,...,120} {30,...,960}	388.15 914.05 485.37 912.57 508.08	80803	80803	6290	0.8
17	{30,...,120} {30,...,480}	297.68 210.12	38040	38040	11960	0.7	I:{30,60,120} II:{30,...,240} I:9,II:11 users	15.00 250.00	128988	128988	5×10^7	20

TABLE III

OBJECTIVE FUNCTION VALUES & EXECUTION TIMES OF EXHAUSTIVE SEARCH & STEM+ RUN ON A P-IV MACHINE AT 1.51 GHZ WITH 1 GB RAM.

k	$\mathcal{S}_i$ (kbps)	r_i (kbps)	objective function value				aggregate throughput			
			1-NORM	2-NORM	4-NORM	IF-NORM	1-NORM	2-NORM	4-NORM	IF-NORM
5	{30,...,120} {30,...,240} {30,...,120}	220.000 350.000 460.000	39500.00	25523.50	20492.40	17500.00	1200	960	720	600
13	{30,...,480} {30,...,960} {30,...,240}	630.000 550.000 200.000	96400.00	62972.20	48983.20	39000.00	9600	8640	8160	8160

TABLE IV

OBJECTIVE FUNCTION VALUE AND AGGREGATE THROUGHPUT WITH DIFFERENT NORMS.

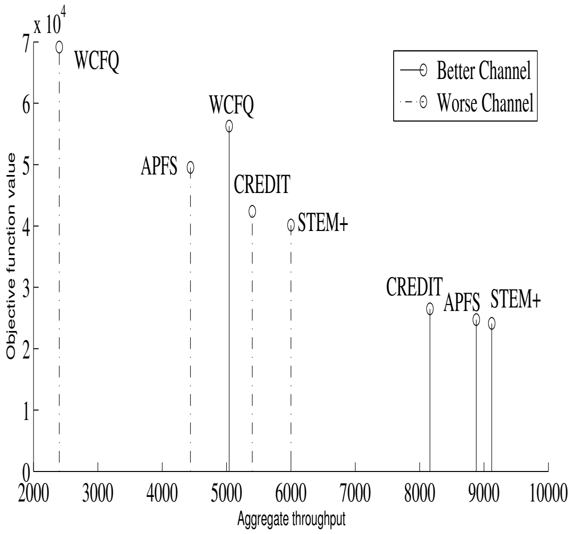

Fig. 3. Performance comparison under different channel conditions but same contracted rates. $k = 13$, $I = 3$, $r_1 = 230$, $r_2 = 550$, $r_3 = 200$ (in kbps) *(i)* Better channel condition: $\mathcal{S}_1 = \{30, 60, 120, 240, 480\}$, $\mathcal{S}_2 = \{30, 60, 120, 240, 480, 960\}$, $\mathcal{S}_3 = \{30, 60, 120, 240\}$ (in kbps). *(ii)* Worse channel condition: $\mathcal{S}_1 = \{30, 60, 120, 240\}$, $\mathcal{S}_2 = \{30, 60, 120, 240, 480\}$, $\mathcal{S}_3 = \{30, 60, 120\}$ (in kbps).

VI. CONCLUSION

We considered the problem of sharing k UMTS frames among a set of I connections and introduced the discrepancy criterion to obtain the best schedule, which balances fairness and system efficiency. We proposed the STEM+ algorithm, proved that it is optimal and computationally efficient, compared its performance with other algorithms and showed its superiority. With the flexibility available in the choice of norm for measuring the discrepancy, we can choose a norm which balances fairness and efficiency.

REFERENCES

[1] H. Holma and A. Toskala, *WCDMA for UMTS*. John Wiley & Sons, Ltd., 2000.

[2] N. Joshi, S. R. Kadaba, S. Patel, and G. S. Sundaram, "Downlink scheduling in CDMA data networks," in *Mobicom*, 2000, pp. 179–190.

[3] T. S. E. Ng, I. Stoica, and H. Zhang, "Packet fair queueing algorithms for wireless networks with location dependent errors," in *IEEE Infocom*, April 1998.

[4] S. Lu, V. Bhargavan, and R. Srikant, "Fair scheduling in wireless packet networks," *IEEE/ACM Transactions on Networking*, vol. 7, no. 4, pp. 473–489, August 1999.

[5] D. N. Skoutas and A. N. Rouskas, "A dynamic traffic scheduling algorithm for the downlink shared channel in 3G WCDMA," in *IEEE International Conference on Communications*, June 2004, pp. 2975–2979.

[6] B. Makarevitch, "Learning rate control for downlink shared channel in 3G WCDMA," in *PIMRC*, September 2003, pp. 2919–2922.

[7] F. Berggren, S.-L. Kim, R. Jäntti, and J. Zander, "Joint power control and intracell scheduling of DS-CDMA Nonreal time data," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 10, pp. 1860–1870, October 2001.

[8] S. S. Kulkarni and C. Rosenberg, "Opportunistic scheduling policies for wireless systems with short term fairness constraints," in *Globecom*, vol. 1, December 2003, pp. 533–537.

[9] J. Kuri and Karthik, "Scheduling streaming flows on the downlink shared channel of UMTS," in *International Journal of Network Management*, vol. 17, no. 2, March/April 2007, pp. 117–137.

[10] G. Aniba and S. Aïssa, "Adaptive proportional fairness for packet scheduling in HSDPA," in *Globecom*, November-December 2004, pp. 4033–4037.

[11] A. C. Kam, T. Minn, and K.-Y. S. Siu, "Supporting rate guarantee and fair access for bursty data traffic in WCDMA," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 11, pp. 2121–2130, November 2001.

[12] Y. Liu, S. Gruhl, and E. W. Knightly, "WCFQ: An opportunistic wireless scheduler with statistical fairness bounds," *IEEE Transactions on Wireless Communications*, vol. 2, no. 5, pp. 1017–1028, September 2003.