# Fair distributed user-traffic association in cache equipped cellular networks

Jonatan Krolikowski*, Anastasios Giovanidis† and Marco Di Renzo*
*CNRS-L2S, CentraleSupélec, Université Paris Sud, Université Paris-Saclay
†CNRS-LIP6, Université Pierre et Marie Curie, Sorbonne Universités, Paris, France
Email: jonatan.krolikowski@telecom-paristech.fr, anastasios.giovanidis@lip6.fr, marco.direnzo@lss.supelec.fr

***Abstract*—Caching of popular content on wireless nodes is recently proposed as a means to reduce congestion in the backbone of cellular networks and to improve Quality of Service. From a network point of view, the goal is to offload as many users as possible from the backbone network to the wireless caches while at the same time offering good service to cache-unrelated users. Aggressive offloading can lead to an unbalanced user association. Some wireless nodes can be overloaded by cache-related traffic while the resources of others remain underused.**

**Given a fixed content placement, this work proposes an efficient distributed algorithm to control and balance the association of cache-related traffic among cellular cache memories. The algorithm allows the network to achieve the globally optimal solution and can be executed on base stations using a limited amount of information exchange between them. It is based on a novel algorithm we call *Bucket-filling*. The solution limits the cache-users per node by balancing the total load among the nodes in a fair way. The improvement compared to common user assignment policies is highlighted for single- as well as for multi-tier random networks.**

## I. Introduction

By 2020, wireless data traffic is estimated to reach roughly the 8-fold of its volume of 2015 [1]. Such increase in data demand will be satisfied by densifying the network with new tiers as well as by allowing cooperation among stations. However, this increase of wireless traffic can pose new problems to the wireless backhaul network that are related to congestion. A trending strategy to ease the backhaul is to equip wireless nodes with large cheap cache memories [2]. Content can be stored on them based on some knowledge of the popularity of the content requested by the users in the cellular network. The content is thereby brought closer to the user and the costly usage of backhaul bandwidth is reduced.

Since not all requests can be served by the caches, wireless traffic will be divided into (1) cache-related and (2) cache-unrelated traffic. The former can retrieve the requested content from the cache of a node whose signal can be sufficiently received. The latter will typically connect to the node with the strongest signal. As a result, wireless nodes with popular content could be overloaded with cache-related traffic, leaving no resources for cache-unrelated users. At the same time, other nodes with less popular content could be underused.

A solution for such imbalance can be to limit the maximum cache-related load per wireless node and redistribute, when possible, the remaining load among neighboring nodes. This way, resources can be kept available for cache-unrelated traffic which is less predictable. Note that although multicasting of cached content could reduce the effective traffic load, this approach also has its limitations due to service delay tolerance [3]. Thus, the solution of the above association problem remains necessary.

The existing literature has not adequately dealt with such questions yet. It focuses on how to determine good content placement policies with respect to certain optimality criteria such as the hit ratio or user delays. User association is handled in different ways: In [4], the authors use caching to minimize bandwidth cost in a tree-like network. The routing decisions of users are, however, independent of each other. The authors of [5] associate users to any covering station that caches their requested content without balancing the traffic loads. In other works [6][7], users are associated to the closest base station, not knowing if the requested content is stored in the cache or not. The authors of [8] maximize the hit ratio by means of integer optimization. They introduce a bandwidth constraint limiting the amount of users that can be connected to each cellular station. While this model avoids overloading cellular stations, it cannot guarantee a balanced routing among them. In [9], user association is balanced between a cached and an uncached path. Association to the individual caches is modeled by shortest distance, again not allowing control over the use of the separate resources. The model in [10] includes both fractional content placement and routing variables and allows for the balancing of user traffic loads at the cache-equipped base stations. In the solution of the problem, however, the convergence to an optimal routing is dependent on iterative fractional content placement updates, which is decided for the entire network.

In this paper, we model the problem of associating users to stations with given cached content by introducing a utility function per station which puts soft limitation on the served user load. The related optimization problem allows to enforce a load fairness criterion between caches. We further develop a policy to optimally solve this problem in a distributed way. For a given cache placement, the resulting policy guarantees that all resources are used as evenly as possible. The calculations can be executed on the individual stations requiring a limited amount of information exchange. We show that the policy is beneficial both in single- and in multi-tier networks.

The remainder of this paper is organized as follows: Section II presents the network model and the problem formu-

978-3-901882-90-6/17 ©2017 IFIP

lation. In Section III, we introduce the solution techniques: Augmented Lagrangian, Diagonal Quadratic Approximation and the novel Bucket-filling algorithm. Numerical evaluations of the resulting policy are presented in Section IV for random networks (single-tier and 2-tier) with varying coverage or node density. Finally, Section V concludes our work.

## II. System Model and Problem statement

Consider a cellular communications network with a finite set $\mathcal{M} = \{m_1, \dots, m_M\}$ of Cached Base Stations (CBSs). Every CBS $m$ is equipped with a cache memory the size of $K_m \in \mathbb{Z}_{\geq 0}$ memory units. Note that other base stations not equipped with caches (uncached base stations, UBSs) can be part of the network (not considered in $\mathcal{M}$). Each CBS has a coverage area, and areas of different CBSs can overlap.

A user arrives in a position on the plane and requests a file $f \in \mathcal{F}$. The user will be covered (or not) by some CBS. In case he/she is covered by some CBS with the requested content in the cache, the user will be served from the caches. This traffic is called *cache-related*. The user association is assumed unique in the sense that an association to two or more CBSs is not allowed. When a user request occurs in the overlap of two or more coverage areas, it can be associated to any one of the covering CBSs. The exact distance of the user from the covering station (e.g. near the center or at the edge of the cell) is not taken into consideration. The users who do not find their requested content cached, and the users not covered by any CBS at all, constitute the *cache-unrelated* traffic. This traffic will be served either from covering CBSs or uncached base stations (UBSs). In this work, we will not treat the question how to distribute its load among the stations. The boundaries of the overlapping coverage areas of the CBSs partition the plane into a set of disjoint regions: Two points are in the same region if and only if they are in the coverage areas of exactly the same subset of CBSs. Let $\mathcal{S}$ denote the set of all regions covered by at least one CBS.

The cache placement is given a priori. Tuples of region $s$ and file $f$ are called *region-files* $q = (s, f)$ if there is a CBS covering $s$ with $f$ in its cache. We denote the complete set of region-files by $\mathcal{Q}$. Not included in $\mathcal{Q}$ are requests for files not stored in a covering CBS. The subset of CBSs that cache content $f$ and cover region $s$ is denoted by $\mathcal{M}(q) \subseteq \mathcal{M}$. Conversely, $\mathcal{Q}(m) \subseteq \mathcal{Q}$ are the region-files which can be served by the cache of $m$.

For delay and backhaul congestion reasons, it is always more beneficial to serve a request from a cache than through the backhaul network. Therefore, in an optimal solution of the routing problem, all traffic of $q \in \mathcal{Q}$ is served from the cache of a CBS: For each region-file $q = (s, f) \in \mathcal{Q}$ there is a station covering $s$ with $f$ in its cache. As a consequence, all user requests related to region-files in $\mathcal{Q}$ (and only they) are cache-related traffic.

For each region-file $q \in \mathcal{Q}$, the expected number of user requests is denoted by $N_q \in \mathbb{R}_+$. The vector of these popularity values is denoted by $\mathbf{N} := (N_q), q \in \mathcal{Q}$. The information on file popularity is considered locally available at each covering CBS. Note that the model does not assume spatially uniform traffic and that the vector $\mathbf{N}$ is general.

We introduce the routing variable $y_{m,q} \geq 0$ for all $m \in \mathcal{M}, q \in \mathcal{Q}(m)$. Each $y_{m,q}$ with $q = (s, f)$ denotes the part of expected traffic for content file $f$ in region $s$ associated to CBS $m$. The vector of all routing variables is denoted by $\mathbf{y}$. For each $q \in \mathcal{Q}$, the sum of all routing variables towards the covering CBSs must be smaller than or equal to $N_q$. Formally, for every feasible solution:

$$\sum_{m \in \mathcal{M}(q)} y_{m,q} \leq N_q, \ \ \forall\, q \in \mathcal{Q}. \tag{1}$$

The total traffic volume associated with CBS $m$ depends on the partial vector $\mathbf{y}_m := (y_{m,q}), q \in \mathcal{Q}(m)$ and can be expressed as follows:

$$v_m(\mathbf{y}_m) := \sum_{q \in \mathcal{Q}(m)} y_{m,q}.$$

The aim of our model is to put a soft limit on the maximum traffic associated with a CBS while balancing the cache-related traffic volume among the CBSs. This approach also guarantees the *usefulness* of each cache. We measure the usefulness of a CBS $m$ by the utility function $\mathrm{U}_m$ which takes the traffic volume routed to the CBS as its argument.

Following the concept of diminishing returns, $\mathrm{U}_m$ is increasing and concave as well as continuously differentiable. Additionally, we can choose the utility function such that for volumes greater than a certain $V_m$, the derivative of $\mathrm{U}_m$ becomes close to zero. Such soft limitation entails that it is not beneficial to further route users to this station. The overall objective is to maximize the sum of utilities subject to routing constraints. The Convex Program formulation of the Cache Routing Problem is

$$\begin{aligned} \text{(CRP)} \max_{0 \leq \mathbf{y}_m \leq \mathbf{N}_m, m \in \mathcal{M}} \quad & \sum_{m \in \mathcal{M}} \mathrm{U}_m(v_m(\mathbf{y}_m)) \\ \text{s. t.} \quad & \sum_{m \in \mathcal{M}(q)} y_{m,q} = N_q, \quad \forall\, q \in \mathcal{Q}, \end{aligned}$$

where $\mathbf{N}_m := (N_{m,q}), q \in \mathcal{Q}(m)$ is the vector of expected requests for the region-files covered by $m$. Note that we introduce an equality constraint instead of the inequality in (1), since the utility functions are increasing. There is, thus, no cache-related traffic in the optimal solution that remains unrouted.

Traffic association is balanced if the available resources are used in a fair way. Some notions of fairness are max-min, $\alpha-$ and proportional fairness. Each of them is achieved by appropriate choice of the utility functions (see [11][12]). E.g. for proportional fairness, utilities could be chosen as (weighted) logarithms, depending also on the soft limit we want to achieve.

In the next section, we derive the procedure that achieves the solution to the CRP for general utilities, leaving their specific choice to the network designer.

## III. Solution

The CRP is solved with a distributed algorithm which consists of three nested loops.

### A. Dual method for the Augmented Lagrangian

The CRP is solved using the dual method on the Augmented Lagrangian (see [13], Section 3.4.4). We use the Augmented instead of the regular Lagrangian to achieve a distributed solution. In our case, the regular Lagrangian is not appropriate since it is not strictly concave in the primal variables and hence the primal solution is not unique. This creates conflicts when different stations compete for the same users and convergence cannot be guaranteed. Like the regular Lagrangian, the Augmented one relaxes constraints of the CRP and introduces a *price* $\lambda_q$ for the violation of each constraint. The difference between them is an additional quadratic term penalizing the violation of each constraint together with a factor $\varrho > 0$. This penalty guarantees strict concavity in the primal variables. Denoting the Augmented Lagrangian by $L^{(\varrho)}$, we get

$$\begin{aligned} L^{(\varrho)}(\mathbf{y}, \boldsymbol{\lambda}) = & \sum_{m \in \mathcal{M}} \mathrm{U}_m(v_m(\mathbf{y}_m)) \\ & - \sum_{q \in \mathcal{Q}} \lambda_q (N_q - \sum_{m \in \mathcal{M}(q)} y_{m,q}) \\ & - \frac{\varrho}{2} \sum_{q \in \mathcal{Q}} (N_q - \sum_{m \in \mathcal{M}(q)} y_{m,q})^2, \end{aligned} \tag{2}$$

where $\boldsymbol{\lambda} := (\lambda_q), q \in \mathcal{Q}$ is the price vector. The domains of the dual variables are $\lambda_q \in \mathbb{R}$ for all $q \in \mathcal{Q}$, since the respective constraints are equalities.

The Duality theorem (see [13], Appendix C) applies, which means that the duality gap is 0, and the dual method can be used. The objective function of the dual problem is

$$D^{(\varrho)}(\boldsymbol{\lambda}) := \max_{0 \le \mathbf{y}_m \le \mathbf{N}_m, m \in \mathcal{M}} L^{(\varrho)}(\mathbf{y}, \boldsymbol{\lambda}) = L^{(\varrho)}(\mathbf{y}^*(\boldsymbol{\lambda}), \boldsymbol{\lambda}),$$

where

$$\mathbf{y}^*(\boldsymbol{\lambda}) = \operatorname*{arg\,max}_{0 \le \mathbf{y}_m \le \mathbf{N}_m, m \in \mathcal{M}} L^{(\varrho)}(\mathbf{y}, \boldsymbol{\lambda}) \tag{3}$$

is the primal maximum of (2) for a given price vector $\boldsymbol{\lambda}$. The dual problem is then defined as

$$\text{(CRP-dual)} \qquad \min_{\boldsymbol{\lambda} \in \mathbb{R}^{\mathcal{Q}}} D_\varrho(\boldsymbol{\lambda}).$$

Starting from an arbitrary initial dual vector $\boldsymbol{\lambda}(0)$, the dual vector is iteratively updated according to

$$\lambda_q(t+1) = \lambda_q(t) + \varrho\Big( \sum_{m \in \mathcal{M}(q)} N_q - y^*_{m,q}(\boldsymbol{\lambda}(t)) \Big), \tag{4}$$

where the steplength $\varrho > 0$ is the penalty used in (2). The convergence of this method is well known (see [14] or Section 3.4.4 of [13]).

For practical implemetation issues of each update step of the region-file price $\lambda_q, q \in \mathcal{Q}$, only the primal solutions of the covering CBSs need to be known. Thus, for a distributed implementation, exchange of such information among neighboring stations is sufficient.

The next subsection presents the distributed solution for the primal problem (3), which needs to be found for every iteration of the dual algorithm.

### B. Distributed solution for the primal problem

The solution for (3) is unique since the domain of $\mathbf{y}$ is convex and compact and, for any fixed feasible vector $\boldsymbol{\lambda}$, the Augmented Lagrangian $L^{(\varrho)}$ is strictly concave. We use the Diagonal Quadratic Approximation Method (DQA) [14] to derive seperate problems which can be solved by each cache. A limited amount of exchanged information between neighboring caches is required.

The DQA overcomes the problem that the objective function $L^{(\varrho)}(\mathbf{y}, \boldsymbol{\lambda})$ of (3) is not easily separable among the variables related to the different CBSs, since it contains quadratic terms combining different variables $y_{m,q}$ (see (2)). To achieve this, we introduce the functions $L_m^{(\varrho)} : \mathbb{R}^{Q_m} \times \mathbb{R}^{\sum_{\bar{m}} Q_{\bar{m}}} \times \mathbb{R}^Q \to \mathbb{R}$ for all $m \in \mathcal{M}$:

$$\begin{aligned} L_m^{(\varrho)}(\mathbf{y}_m, \tilde{\mathbf{y}}, \boldsymbol{\lambda}) := & \mathrm{U}_m(v_m(\mathbf{y}_m)) + \sum_{q \in \mathcal{Q}(m)} \lambda_q y_{m,q} \\ & - \frac{\varrho}{2} \sum_{q \in \mathcal{Q}(m)} (\bar{N}_q^m(\tilde{\mathbf{y}}) - y_{m,q})^2, \end{aligned}$$

where $\bar{N}_q^m(\tilde{\mathbf{y}}) := N_q - \sum_{\substack{\bar{m} \in \mathcal{M}(q) \\ \bar{m} \neq m}} \tilde{y}_{\bar{m},q}$ is the number of requests in $q$ not associated with caches other than $m$ in the routing vector $\tilde{\mathbf{y}} = (\tilde{y}_{m,q}), m \in \mathcal{M}, q \in \mathcal{Q}(m)$ which is here seen as a parameter. The primal problem to be solved by each cache $m$ is defined as

$$\text{(CRP-primal-}m\text{)} \qquad \max_{0 \le \mathbf{y}_m \le \mathbf{N}_m} L_m^{(\varrho)}(\mathbf{y}_m, \tilde{\mathbf{y}}, \boldsymbol{\lambda}). \tag{5}$$

Since $L_m^{(\varrho)}(\mathbf{y}_m, \tilde{\mathbf{y}}, \boldsymbol{\lambda})$ is strictly concave in $\mathbf{y}$ and the domain is compact, CRP-primal-$m$ has a unique solution which we call $\tilde{\mathbf{y}}^*_m$. The vector containing the solutions of CRP-primal-$m$ for all caches is $\tilde{\mathbf{y}}^*$.

The DQA method consists of parallel execution of CRP-primal-$m$ at the caches with consecutive update of the vector $\tilde{\mathbf{y}}$ in the fashion of a nonlinear Jacobi algorithm. It produces a succession of vectors $\tilde{\mathbf{y}}(0), \tilde{\mathbf{y}}(1), \tilde{\mathbf{y}}(2), \ldots$. Starting from some given vector $\tilde{\mathbf{y}}(0)$, the vector $\tilde{\mathbf{y}}(\tau+1)$ is defined as the convex combination of $\tilde{\mathbf{y}}(\tau)$ and $\tilde{\mathbf{y}}^*(\tau)$. Given a constant $0 < \alpha \le 1$, we get

$$\tilde{\mathbf{y}}(\tau+1) = \tilde{\mathbf{y}}(\tau) + \alpha(\tilde{\mathbf{y}}^*(\tau) - \tilde{\mathbf{y}}(\tau)). \tag{6}$$

In [14] it is shown that the DQA method converges. Observe that the convergence depends on the uniqueness of the primal solutions $\tilde{\mathbf{y}}^*(\tau)$. For every update (6), each station only requires results from its neighboring stations that cover a common region-file.

### C. Bucket-filling

What is left is to find an efficient solution to CRP-primal-$m$ (5) running on each CBS separately: In order to do so, we develop a novel technique we call *Bucket-filling*. First, we can simplify our notation since the problem is separated by cache, and the vectors $\tilde{\mathbf{y}}$ and $\boldsymbol{\lambda}$ are parameters. In this subsection, $\mathbf{y} := \mathbf{y}_m$, $\mathrm{U} := \mathrm{U}_m$, $N := N_m$, $v := v_m$, $\mathcal{Q} := \mathcal{Q}(m)$, $\bar{N}_q := \bar{N}_q^m(\tilde{\mathbf{y}})$. Then, the solution to (5) can be rewritten as

$$\begin{aligned}\mathbf{y}^* &= \underset{0\leq\mathbf{y}\leq\mathbf{N}}{\arg\max}\ \mathrm{U}(v(\mathbf{y})) + \sum_{q\in\mathcal{Q}} \lambda_q y_q - \frac{\varrho}{2}\sum_{q\in\mathcal{Q}}(\bar{N}_q - y_q)^2 \\ &= \underset{0\leq\mathbf{y}\leq\mathbf{N}}{\arg\max}\ \mathrm{U}\Big(\sum_{q\in\mathcal{Q}} y_q\Big) - \sum_{q\in\mathcal{Q}}\Big[\frac{\varrho}{2}y_q^2 - (\lambda_q + \varrho\bar{N}_q)y_q\Big].\end{aligned}$$

The last step comes from the development of the quadratic term and from omitting the additive constants which do not affect the optimal solution. Further defining $a_q := \lambda_q + \varrho\bar{N}_q$, the CRP-primal-$m$ can be stated as

$$\mathbf{y}^* = \underset{0\leq\mathbf{y}\leq\mathbf{N}}{\arg\max}\quad g(\mathbf{y}) \tag{7}$$

with $g(\mathbf{y}) := \mathrm{U}\left(\sum_{q\in\mathcal{Q}} y_q\right) - \sum_{q\in\mathcal{Q}}\left[\frac{\varrho}{2}y_q^2 - a_q y_q\right]$.

We will now show that one known value in the optimal routing vector is sufficient to derive the entire solution of CPR-primal-$m$. This observation allows to develop the Bucket-filling technique.

Given the optimal vector $\mathbf{y}^*$, we define the function $g_q(y) : [0, N_q] \to \mathbb{R}$ for $q \in \mathcal{Q}$ as

$$\begin{aligned}g_q(y) :=\ &\mathrm{U}\Big(y + \sum_{\tilde{q}\neq q} y^*_{\tilde{q}}\Big) - \Big[\frac{\varrho}{2}y^2 - a_q y\Big] \\ &- \sum_{\tilde{q}\neq q}\Big[\frac{\varrho}{2}(y^*_{\tilde{q}})^2 - a_{\tilde{q}}y^*_{\tilde{q}}\Big].\end{aligned}$$

The maximum of this function is the optimal solution $y_q^*$. In other words, $\arg\max_{0\leq y\leq N_q} g_q(y) = y_q^*$.

Furthermore, note that $g_q$ is continuously differentiable, strictly concave and defined on a compact set which implies that it has a unique maximum. We denote the unique root of its derivative

$$g_q'(y) = \mathrm{U}'\Big(y + \sum_{\tilde{q}\neq q} y^*_{\tilde{q}}\Big) - [\varrho y - a_q] \tag{8}$$

by $y_q'$. Then we get

$$\begin{aligned}0 \leq y_q' \leq N_q &\implies y_q^* = y_q', \\ y_q' < 0 &\implies y_q^* = 0, \\ y_q' > N_q &\implies y_q^* = N_q.\end{aligned} \tag{9}$$

Now, we can formulate the following key lemma:

**Lemma 1.** *Let* $\mathbf{y}^*$ *be defined as in* (7). *Let* $i, j \in \mathcal{Q}$ *with* $0 < y_i^* < N_i$, $0 < y_j^* < N_j$. *Then*

$$y_i^* = \frac{a_i - a_j}{\varrho} + y_j^*.$$

*Proof.* From (9) we know that $g_i'(y_i^*) = 0$ and $g_j'(y_j^*) = 0$ and thus $g_i'(y_i^*) = g_j'(y_j^*)$. From (8), we get

$$\varrho y_i^* - a_i = \varrho y_j^* - a_j$$

The statement follows from a simple calculation. □

The Lemma allows us to characterize the entire optimal solution just from one value $0 < y_q^* < N_q$.

**Corollary 1.** *The steepest increase of* $g$ *(as in* (7)*) is achieved by increasing the region-file* $q$ *with the highest value* $a_q$*. For two different* $q, \hat{q} \in \mathcal{Q}$*, the gradients of* $g_q$ *and* $g_{\hat{q}}$ *become equal once* $y_q$ *is increased by the value* $(a_q - a_{\hat{q}})/\varrho$*.*

We can now describe the novel Bucket-filling algorithm that efficiently finds the optimal solution to CRP-primal-$m$:

**Algorithm 1** Bucket-filling

```
1: Sort $\mathcal{Q}$ by $a_q$ non-increasingly.
2: Declare all $q \in \mathcal{Q}$ with maximum $a_q$ as active.
3: Increase $y_q$ of active $q$ equally (because of Lemma 1) until
     • another $q$ becomes active by Corollary 1,
     • a $q$ becomes inactive by $y_q$ reaching $N_q$, or
     • $g'(\mathbf{y}) = 0$ for all active region-files $q$.
4: If the last condition is fulfilled, or $y_q = N_q$ for all $q$, then
   terminate. Otherwise, go to step 3.
```

The algorithm is illustrated in Fig. 1. It terminates and returns the optimal solution $\mathbf{y}^*$ defined in (7).

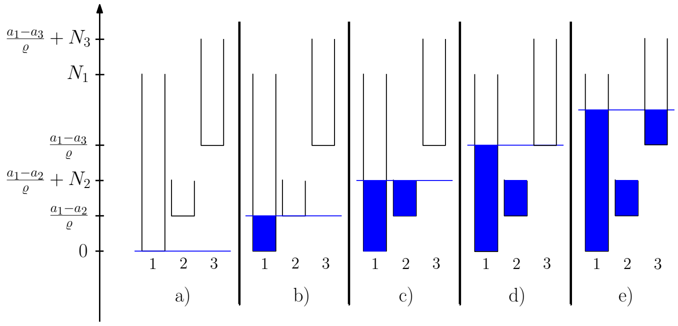
$\frac{a_1-a_3}{\varrho} + N_3$
$N_1$
$\frac{a_1-a_3}{\varrho}$
$\frac{a_1-a_2}{\varrho} + N_2$
$\frac{a_1-a_2}{\varrho}$
0
1 2 3
a)
b)
c)
d)
e)

Fig. 1. Illustration of the Bucket-filling algorithm. Each bucket $q$ is placed with its bottom at level $\frac{a_{\max} - a_q}{\varrho}$ and has the height $N_q$. The buckets are filled to a common level – or until they are full. The algorithm halts when further increasing the water levels starts decreasing the objective value.

Observe that step 1 can be done in $O(|\mathcal{Q}|\log(|\mathcal{Q}|))$ operations with a sorting algorithm such as quicksort. Step 2 runs in $O(|\mathcal{Q}|)$. Step 3 is executed up to $2|\mathcal{Q}|$ times, since every region-file is activated and deactivated no more than one time each. Assuming that $f$ can be evaluated in $O(1)$, this implies $O(|\mathcal{Q}|)$ as the asymptotic runtime for steps 3 and 4. This shows that the overall runtime is dominated by sorting in step 1 and thus is $O(|\mathcal{Q}|\log(|\mathcal{Q}|))$.

This algorithm is executed in every inner loop of the DQA method, which is why its efficiency is paramount.

### *D. Algorithm*

The complete algorithm that finds the globally optimal solution to the CRP is summed up in Algorithm 2.

**Algorithm 2** Solve CRP

```
1: Choose dual vector $\boldsymbol{\lambda}(0)$, $t = 0$, $\varepsilon > 0$.
2: while $\|\boldsymbol{\lambda}(t) - \boldsymbol{\lambda}(t-1)\| > \varepsilon$ do
3:     Choose $\tilde{\mathbf{y}}(0)$, $\tau = 0$.
4:     while $\|\tilde{\mathbf{y}}(\tau) - \tilde{\mathbf{y}}(\tau-1)\| > \varepsilon$ do
5:         Find $\tilde{\mathbf{y}}^*_m(\tau)$ with Algorithm 1 at every $m \in \mathcal{M}$ separately.
6:         Exchange results among neighboring stations, set $\tilde{\mathbf{y}}(\tau+1)$ as in (6), $\tau = \tau + 1$.
7:     end while
8:     Exchange results among neighboring stations, set $\boldsymbol{\lambda}(t+1)$ as in (4) , $t = t + 1$.
9: end while
```

The choice of the first dual vector $\boldsymbol{\lambda}(0)$ in line 1 is arbitrary. The first primal vector $\tilde{\mathbf{y}}(0)$ in line 3 can be chosen as the last primal vector of the iteration before.

## IV. Numerical Evaluation

For the simulation of the algorithm, we consider an urban area of 2.5 km × 2.5 km with uniform user distribution. A catalog of 6 files of equal size is known. The popularity of the files follows a Zipf distribution with parameter 1. Throughout the simulations, CBSs are placed in the area following a Poisson Point Process (PPP) with density $8\frac{\text{CBS}}{km^2}$. This means that their total number in each run is a random Poisson realization, and their position is uniform in the simulation window. In this way we avoid the bias of testing only particular network topologies. We run two simulation scenarios, the first for single-tier and the second for two-tier netowrks. Each scenario consists of 1000 simulation runs and we consider the averaged results over the runs. Coverage follows the Boolean model where a disc area is centered on each wireless station with some defined radius. The surface of different overlapping areas is found in each run by the Monte-Carlo method. The users are routed to the CBSs following three different policies:

1) The policy from Algorithm 2 with logarithmic utility function for each cache. This policy guarantees a proportionally fair (and also max-min fair) solution. We call this policy *fair*.
2) The *closest-available* policy, which associates each user with the closest CBS that both covers its position and has the requested content cached.
3) The *unsplittable* policy which associates all users in a region requesting the same file with a unique random CBS among all covering CBSs having this content.

We want to evaluate the proportion of user traffic served by different CBSs over the total traffic routed to CBSs for each policy. In this way we can compare the policies based on how (un)equally they associate traffic load among the available CBSs. Observe that for all three policies, the total traffic volume associated to CBSs is the same, because in all scenarios the CBSs store the same cached content, and traffic is routed to a CBS whenever possible. Hence, the comparison is fair.

### A. Single-tier Networks

In an ideal situation, all stations would serve exactly the same amount of traffic. This, however, is normally not possible in a random network. A routing policy is better than another, when the maximum load of a CBS is lower and at the same time the minimum load share is higher than in the other policy. This way, an overload of the stations is avoided while the usefulness of all stations is achieved. Simulating single-tier networks, we want to verify that the fair policy provides a more balanced distribution of traffic to CBSs than the other two. The coverage radius of the CBSs is varied between 62.5 m and 500 m. This can be translated to an expected number of covering CBSs per user between 1 and 6. This mapping comes from the Boolean model [6]. Two different sets of content files with different popularities are placed uniformly randomly into the caches. Since we are only interested in traffic associated with CBSs, we disregard users not covered by any CBS.

Fig. 2 illustrates how the routing decisions of each of the three policies affect the distribution of load shares among all CBSs. It displays the average maximum (upper curve) and minimum load share (lower curve) over the mean number of CBSs a covered user can see.

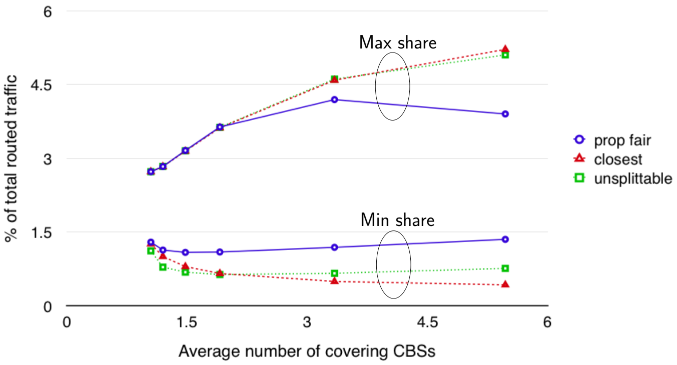

Fig. 2. Minimum and maximum load share of a CBS in the network depending on the mean coverage number.

The results show that with an increasing average number of covering stations, the fair policy achieves a lower maximum load as well as a higher minumum load. Since the overall traffic routed to the CBSs is the same for all policies, we can conclude that the fair policy makes the most balanced use of the available resouces. The resources which remain available for potential cache-unrelated traffic are spread evenly across the network.

### B. Two-tier Network

In a second scenario, we simulate an area covered by two tiers [15]: one of large and one of small coverage. We show that the fair policy is better at offloading traffic from larger to smaller CBSs than the other policies (see Fig. 3). While the fair policy burdens the small stations with a higher load, we demonstrate that it distributes the load more evenly among them so that no individual station is overburdened (see Fig. 4). The first tier consists of large CBSs having a 187.5 m coverage radius while the second tier of small CBSs has 62.5 m coverage radius. The large CBSs are equipped with caches and the two most popular files are stored in all of them. The smaller CBSs get one of these two files assigned uniformly randomly in their cache.

In Fig. 3, we show the percentage of all traffic routed to large CBSs for each of the three policies. The x-coordinate increases with the ratio of small stations over large stations in the network. When less traffic is routed to the large CBSs, then the policy provides a more efficient offloading of traffic towards the small CBSs. The figure shows that, increasing the amount of small CBSs, the fair policy offloads significantly more traffic to the small stations than the unsplittable policy, and slightly more than the closest-available policy.

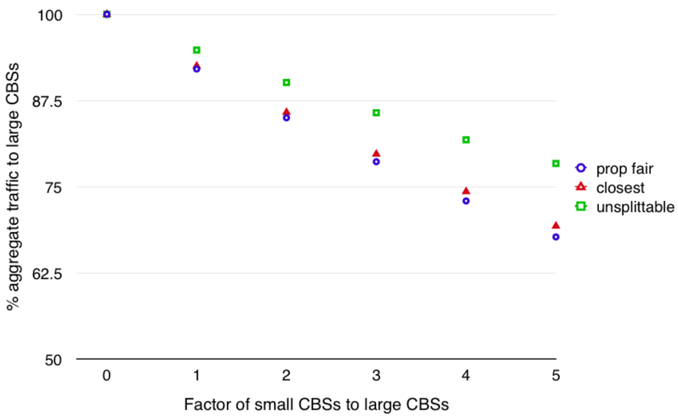

Fig. 3. Aggregate traffic share of large CBSs depending on the amount of small CBSs.

Fig. 4 shows (as in Fig. 2) the maximum and minimum traffic load routed to a small CBS depending on each policy. Even though for the fair policy more users are routed to the small CBSs overall, the maximum load share that one small CBS takes is almost the same for all policies. The increase in traffic load by the fair policy is distributed to the less loaded small CBSs. This is indicated by the higher minimum load among the stations. When applying either the closest-available or the unsplittable policy these CBSs are underused. Thus, the fair policy utilizes the available resources better.

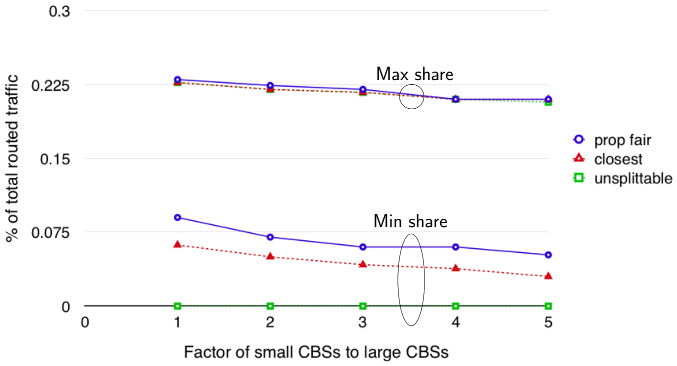

Fig. 4. Minimum and maximum load share of a small (2nd tier) CBS depending on the ratio of small CBS number over large CBS number.

## V. Conclusions

We propose a distributed algorithm that optimizes cache-related user traffic association among cache-equipped cellular stations for a given content placement. The main novelty of the approach is the balancing of cache-related traffic load to evenly use all available resources. The solution procedure makes use of the Augmented Lagrangian to solve a strictly concave problem. The novel Bucket-filling subroutine takes optimal routing decisions locally at each station. With limited information exchange, our policy achieves a fair distribution of users among the stations. The efficiency of the algorithm allows for large networks with different sizes of coverage areas and different cache sizes to be solved optimally. Simulations of single-tier and multi-tier networks show that our policy is superior to conventional user traffic association policies in balancing cache-related user traffic among stations.

## References

[1] Cisco visual networking index: Global mobile data traffic forecast update, 2015–2020 white paper. White Paper, 2 2016.

[2] K. Poularakis, G. Iosifidis, I. Pefkianakis, Leandros Tassiulas, and M. May. Mobile data offloading through caching in residential 802.11 wireless networks. *IEEE Transactions on Network and Service Management*, 13(1):71–84, 2016.

[3] K. Poularakis, G. Iosifidis, V. Sourlas, and L. Tassiulas. Exploiting caching and multicast for 5g wireless networks. *IEEE Transactions on Wireless Communications*, 15(4):2995–3007, April 2016.

[4] S. Borst, V. Gupta, and A. Walid. Distributed caching algorithms for content distribution networks. In *Proceedings of the 29th Conference on Information Communications*, INFOCOM'10, pages 1478–1486, Piscataway, NJ, USA, 2010. IEEE Press.

[5] K. Shanmugam, N. Golrezaei, A.G. Dimakis, A.F. Molisch, and G. Caire. Femtocaching: Wireless content delivery through distributed caching helpers. *Information Theory, IEEE Transactions on*, 59(12):8402–8413, Dec 2013.

[6] B. Blaszczyszyn and A. Giovanidis. Optimal geographic caching in cellular networks. In *Communications (ICC), 2015 IEEE International Conference on*, pages 3358–3363. IEEE, 2015.

[7] E. Bastug, M. Bennis, and M. Debbah. Cache-enabled small cell networks: Modeling and tradeoffs. In *Wireless Communications Systems (ISWCS), 2014 11th International Symposium on*, pages 649–653, Aug 2014.

[8] K. Poularakis, G. Iosifidis, and L. Tassiulas. Approximation algorithms for mobile data caching in small cell networks. *IEEE Transactions on Communications*, 62(10):3665–3677, 2014.

[9] M. Dehghan, A. Seetharam, Bo Jiang, Ting He, Th. Salonidis, J. Kurose, D. Towsley, and R. Sitaraman. On the complexity of optimal routing and content caching in heterogeneous networks. In *INFOCOM, IEEE Conference on Computer Communications*, 2015.

[10] K.P. Naveen, L. Massoulie, E. Baccelli, A. Carneiro Viana, and D. Towsley. On the interaction between content caching and request assignment in cellular cache networks. In *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*, AllThingsCellular '15, pages 37–42, New York, NY, USA, 2015. ACM.

[11] F. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 1997.

[12] J. Mo and J. Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Trans. Netw.*, 8(5):556–567, October 2000.

[13] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[14] A. Ruszczyński. On convergence of an augmented lagrangian decomposition method for sparse convex optimization. *Mathematics of Operations Research*, 20(3):634–656, 1995.

[15] M. Di Renzo, A. Guidotti, and G. E. Corazza. Average rate of downlink heterogeneous cellular networks over generalized fading channels: A stochastic geometry approach. *IEEE Trans. on Comm.*, 61(7):3050–3071, 2013.