On Optimal Geographical Caching in Heterogeneous Cellular Networks

Berksan Serbetci
Stochastic Operations Research
University of Twente, The Netherlands
b.serbetci@utwente.nl

Jasper Goseling
Stochastic Operations Research
University of Twente, The Netherlands
j.goseling@utwente.nl

***Abstract*—In this work we investigate optimal geographical caching in heterogeneous cellular networks where different types of base stations (BSs) have different cache capacities. Users request files from a content library according to a known probability distribution. The performance metric is the total hit probability, which is the probability that a user at an arbitrary location in the plane will find the content that it requires in one of the BSs that it is covered by.**

We consider the problem of optimally placing content in all BSs jointly. As this problem is not convex, we provide a heuristic scheme by finding the optimal placement policy for one type of base station conditioned on the placement in all other types. We demonstrate that these individual optimization problems are convex and we provide an analytical solution. As an illustration, we find the optimal placement policy of the small base stations (SBSs) depending on the placement policy of the macro base stations (MBSs). We show how the hit probability evolves as the deployment density of the SBSs varies. We show that the heuristic of placing the most popular content in the MBSs is almost optimal after deploying the SBSs with optimal placement policies. Also, for the SBSs no such heuristic can be used; the optimal placement is significantly better than storing the most popular content. Finally, we show that solving the individual problems to find the optimal placement policies for different types of BSs iteratively, namely repeatedly updating the placement policies, does not improve the performance.

I. Introduction

Recent years have seen extreme growth in data traffic over cellular networks. The demand is expected to further increase in the upcoming years such that current network infrastructures [1] will not be able to support this data traffic [2]. One of the bottlenecks is formed by the backhaul and a promising means of reducing backhaul traffic is by reserving some storage capacity at base stations (BSs) and use these as caches [3]. In this way, part of the data is stored at the wireless edge and the backhaul is used only to refresh this stored data. Data replacement will depend on the users' demand distribution over time. Since this distribution is varying slowly, the stored data can be refreshed at off-peak times. Thus, caches containing popular content can serve users without incurring a load on the backhaul.

Recently, there has been growing interest in caching in cellular networks. In [4] Shanmugam *et al.* focus on the content placement problem and analyze which files should be cached by which helpers for the given network topology and file popularity distribution by minimizing the expected total file delay. In [5] Poularakis *et al.* provide an approximation algorithm for the problem of minimizing the user content requests routed to macrocell base stations with constrained cache storage and bandwidth capacities. In [6] Błaszczyszyn *et al.* revisit the optimal content placement in cellular caches by assuming a known distribution of the coverage number and provide the optimal probabilistic placement policy which guarantees maximal total hit probability. In [7] Maddah-Ali *et al.* developed an information-theoretic lower bound for the caching system for local and global caching gains. In [8] Ioannidis *et al.* propose a novel mechanism for determining the caching policy of each mobile user that maximize the system's social welfare. In [9] Poularakis *et al.* consider the content storage problem of encoded versions of the content files. In [10] Bastug *et al.* couple the caching problem with the physical layer. In [11] Altman *et al.* compare the expected cost of obtaining the complete data under uncoded and coded data allocation strategies for caching problem. Cache placement with the help of stochastic geometry and optimizing the allocation of storage capacity among files in order to minimize the cache miss probability problem is presented by Avrachenkov *et al.* in [12]. A combined caching scheme where part of the available cache space is reserved for caching the most popular content in every small base station, while the remaining is used for cooperatively caching different partitions of the less popular content in different small base stations, as a means to increase local content diversity is proposed by Chen *et al.* in [13]. In [14] Dehghan *et al.* associate each content a utility, which is a function of the corresponding content hit probabilities and propose utility-driven caching by formulating an optimization problem to maximize the sum of utilities over all contents.

The *main contribution* of this paper is to find optimal placement strategies that maximize total hit probability in heterogeneous cellular networks. Our focus is on heterogeneous cellular networks in which an operator wants to jointly optimize the cached content in macro base stations (MBSs) and small base stations (SBSs) with different storage capacities. This problem is not convex. Therefore, we provide a heuristic scheme and optimize only one type of cache (e.g.

978-1-5090-4183-1/17/$31.00 ©2017 IEEE

SBS) with the information coming from other types of caches (e.g. MBS and other SBSs with different cache capacities) at each iteration step.

We show that whether MBSs use the optimal deployment strategy or store "the most popular content", has no impact on the total hit probability after deploying the SBSs with optimal content placement policies. We show that it is crucial to optimize the content placement strategy of the SBSs in order to maximize the overall performance. We show that heuristic policies like storing the popular content that is not yet available in the MBSs result in significant performance penalties.

In Section II we start the paper with model and problem definition. In Section III we present the optimal content placement strategy problem and give required tools to solve it. We conclude this section by providing an iterative heuristic scheme for the joint optimization problem. In Section IV we continue with performance evaluation of the optimal placement strategies for different probabilistic deployment scenarios. In Section V we conclude the paper with discussions.

II. Model and Problem Definition

Throughout the paper we will be interested in two types of base stations, namely MBSs and SBSs. However, in this section we will give a more general formulation as it is possible to have SBSs with different storage capacities in some topologies. We consider a heterogeneous cellular network with L-different types of base stations in the plane. These base stations are distributed according to a homogeneous spatial process [15]. All base stations have a coverage radius r. Let $\mathcal{N}_i$, $i = 1, \dots, L$, denote the number of base stations of type i that are covering a user at an arbitrary location in the plane. Furthermore, let $p_{n_i}^{(i)} := \mathbb{P}[\mathcal{N}_i = n_i]$ and $p_{\boldsymbol{n}} := \mathbb{P}[\boldsymbol{\mathcal{N}} = \boldsymbol{n}]$, where $\boldsymbol{\mathcal{N}} = (\mathcal{N}_1, \dots, \mathcal{N}_L)$ and $\boldsymbol{n} = (n_1, \dots, n_L)$. Each base station is equipped with a cache and we will refer to caches in base stations of type i as type$-i$ caches. Caches of type i have capacity $K_i \geq 1$, $i = 1, \dots, L$, meaning that they can store K_i files from the content library $\mathcal{C} = \{c_1, c_2, \dots, c_J\}$. All files have equal size.

Users independently request files from $\mathcal{C}$ according to probability distribution $a_1, \dots, a_J$, where a_j is the probability of requesting file c_j. The probability that file c_j is requested is denoted as a_j. Without loss of generality, $a_1 \geq a_2 \geq \cdots \geq a_J$.

We use the probabilistic content placement policy of [6]. In this policy each cache is storing a random subset of the files in the content library. In particular, the probability that a type$-i$ cache is storing file c_j is independent and identically distributed over all type$-i$ caches. Placement is also independent over caches of different types. We denote the probability that the content c_j is stored at a type$-i$ cache as

$$b_j^{(i)} := \mathbb{P}\left(c_j \text{ stored in type}-i \text{ cache}\right), \tag{1}$$

and the content placement strategy $\boldsymbol{b}^{(i)} = (b_1^{(i)}, \dots, b_J^{(i)})$ as a J-tuple for any type$-i$ cache.

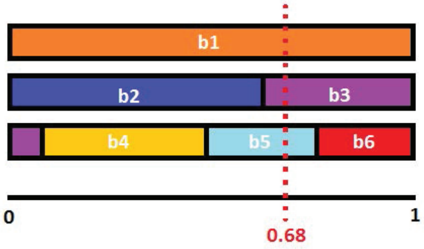

Fig. 1. A realization of the probabilistic placement policy ($J = 6$ and $K = 3$). A random number is picked (0.68 in this case) and the vertical line intersects with the optimal $\boldsymbol{b}^{(i)}$ values. From this figure, we conclude that the content subset $\{c_1, c_3, c_5\}$ will be cached.

One means obtaining a placement according this distribution is as follows. The memory of a type$-i$ cache is divided into K_i continuous memory intervals of unit length. Then $b_j^{(i)}$ values fill the cache memory sequentially and continue filling the next slot if not enough space is available in the memory slot that it has started filling in as in the end completely covering the K_i memory intervals. Then, for any type$-i$ cache, a random number from the interval $[0, 1]$ is picked and the intersecting K_i files are cached. An example is shown in Figure 1.

III. Content Placement Problem - Individual Optimization

A. *Formulation of the problem*

We start this section with defining our performance metric and the formulation of the optimization problem. The performance metric is the total hit probability which is the probability that a user will find the content that he requires in one of the *caches* that he is covered by. We assume that the content placement strategy for the probability distribution over J files through all $L-1$ types is fixed and known and we will solve the optimization problem for only one type. In this section, without loss of generality, we consider the optimization of type-1. For notational convenience, superscript ^{c} in the notation $b_j^{(i)^c}$ indicates that the content placement strategy for type$-i$ is known and constant. Then, the total hit probability is given by

$$f\left(\boldsymbol{b}^{(1)}\right) = 1 - \sum_{j=1}^{J} a_j \sum_{n_1=0}^{\infty} (1 - b_j^{(1)})^{n_1} p_{n_1}^{(1)} q(j, n_1), \tag{2}$$

where

$$q(j, n_1) = P(\text{non type}-1 \text{ caches miss the file } c_j)$$
$$= \sum_{n_2=0}^{\infty} \cdots \sum_{n_L=0}^{\infty} \prod_{l=2}^{L} (1 - b_j^{(l)^c})^{n_l} P(\boldsymbol{\mathcal{N}}_2^L = \boldsymbol{n}_2^L | N_1 = n_1),$$

where $\boldsymbol{n}_2^L = (n_2, \dots, n_L)$ and $\boldsymbol{\mathcal{N}}_2^L = (\mathcal{N}_2, \dots, \mathcal{N}_L)$ are $(L-1)$-tuples representing the coverage vectors of non type-1 caches.

We define the optimization problem to find the optimal content placement strategy maximizing the total hit probability for a type-1 cache as follows:

Problem 1.

$$\begin{aligned}&\max\ f\left(\boldsymbol{b}^{(1)}\right)\\ &\textbf{s.t.}\quad b_1^{(1)}+\cdots+b_J^{(1)}=K_1,\quad b_j^{(1)}\in[0,1],\quad\forall j.\end{aligned}\tag{3}$$

B. Solution of the optimization problem

In this section, we will analyze the structure of the optimization problem.

Lemma 1. *Problem 1 is a convex optimization problem.*

Proof. The objective function is separable with respect to (w.r.t.) $b_1^{(1)},\ldots,b_J^{(1)}$ (Replace $\sum_{j=1}^{J}a_j=1$ and rewrite the objective function (2).). $1-f\left(\boldsymbol{b}^{(1)}\right)$ is increasing and convex in $b_j^{(1)}$, $\forall j$. Hence, it is a convex function of $\left(b_1^{(1)},\ldots,b_J^{(1)}\right)$. □

We already showed that $1-f\left(\boldsymbol{b}^{(1)}\right)$ is convex by Lemma 1 and the constraint set is linear as given in (3). Thus, the Karush-Kuhn-Tucker (KKT) conditions provide necessary and sufficient conditions for optimality. The Lagrangian function corresponding to Problem 1 becomes

$$\begin{aligned}L\left(\boldsymbol{b}^{(1)},\nu,\boldsymbol{\lambda},\boldsymbol{\omega}\right)&=\sum_{j=1}^{J}a_j\sum_{n_1=0}^{\infty}(1-b_j^{(1)})^{n_1}p_{n_1}^{(1)}q(j,n_1)\\ &+\nu\left(\sum_{j=1}^{J}b_j^{(1)}-K_1\right)-\sum_{j=1}^{J}\lambda_j b_j^{(1)}+\sum_{j=1}^{J}\omega_j\left(b_j^{(1)}-1\right),\end{aligned}\tag{4}$$

where $\boldsymbol{b}^{(1)}$, $\boldsymbol{\lambda}$, $\boldsymbol{\omega}\in\mathbb{R}_+^J$ and $\nu\in\mathbb{R}$.

Let $\bar{\boldsymbol{b}}^{(1)}$, $\bar{\boldsymbol{\lambda}}$, $\bar{\boldsymbol{\omega}}$ and $\bar{\nu}$ be primal and dual optimal. The KKT conditions for Problem 1 state that

$$\sum_{j=1}^{J}\bar{b}_j^{(1)}=K_1,\tag{5}$$

$$0\le\bar{b}_j^{(1)}\le 1,\quad\forall j=1,\ldots,J,\tag{6}$$

$$\bar{\lambda}_j\ge 0,\quad\forall j=1,\ldots,J,\tag{7}$$

$$\bar{\omega}_j\ge 0,\quad\forall j=1,\ldots,J,\tag{8}$$

$$\bar{\lambda}_j\bar{b}_j^{(1)}=0,\quad\forall j=1,\ldots,J,\tag{9}$$

$$\bar{\omega}_j\left(\bar{b}_j^{(1)}-1\right)=0,\quad\forall j=1,\ldots,J,\tag{10}$$

$$-a_j\sum_{n_1=0}^{\infty}n_1(1-b_j^{(1)})^{n_1-1}p_{n_1}^{(1)}q(j,n_1)+\bar{\nu}-\bar{\lambda}_j+\bar{\omega}_j=0,\quad\forall j=1,\ldots,J.\tag{11}$$

Theorem 1. *The optimal content placement strategy for Problem 1 is*

$$\bar{b}_j^{(1)}=\begin{cases}1, & \text{if } \bar{\nu}<a_j p_1^{(1)}q(j,1)\\ 0, & \text{if } \bar{\nu}>a_j\sum_{n_1=0}^{\infty}n_1 p_{n_1}^{(1)}q(j,n_1),\\ \phi(\bar{\nu}), & \text{otherwise},\end{cases}\tag{12}$$

where $\phi(\bar{\nu})$ is the solution over $b_j^{(1)}$ of

$$a_j\sum_{n_1=0}^{\infty}n_1(1-b_j^{(1)})^{n_1-1}p_{n_1}^{(1)}q(j,n_1)=\bar{\nu},\tag{13}$$

and $\bar{\nu}$ can be obtained as the unique solution to the additional constraint

$$\bar{b}_1^{(1)}+\cdots+\bar{b}_J^{(1)}=K_1.\tag{14}$$

Proof. The proof is provided in Appendix A. □

Finding the optimal content placement strategy for all types of caches jointly is an interesting problem. However, this optimization problem is *not convex*. Hence, we provide a heuristic algorithm to see how the overall system performance is improved. The procedure is as follows. At each iteration step we find the optimal strategy for a specific type of cache assuming that the placement strategies for other types of caches are known and fixed. Then we continue with the same procedure for the next type and we continue iterating over different types. In the next section we will evaluate the performance of this scheme.

IV. Performance Evaluation

In this section we will specify different network coverage models and show the performances of the placement strategies. In both models we use Zipf distribution for the file popularities. The probability that a user will ask for content c_j is equal to

$$a_j=\frac{j^{-\gamma}}{\sum_{j=1}^{J}j^{-\gamma}},\tag{15}$$

where $\gamma>0$ is the Zipf parameter.

A. Poisson Point Process (PPP) Model

The caches follow a two-dimensional (2D) spatial homogeneous Poisson process with type$-i$ caches independently distributed in the plane with density $\lambda_i>0$, $i=1,\ldots,L$. The number of type$-i$ caches within radius r follows a Poisson distribution with parameter $t_i=\lambda_i\pi r^2$, i.e.,

$$\begin{aligned}p_{n_i}^{(i)}&=P(n_i\text{ type}-i\text{ caches within radius }r)\\ &=\frac{t_i^{n_i}}{n_i!}e^{-t_i}.\end{aligned}\tag{16}$$

The user is covered by n_i type$-i$ caches and distribution of the different type of caches is independent of each other. Therefore, the total coverage distribution probability mass function $p_{\boldsymbol{n}}$ will be the product of individual probability distributions

$$p_{\boldsymbol{n}}=p_{n_1}^{(1)}p_{n_2}^{(2)}\ldots p_{n_L}^{(L)}.\tag{17}$$

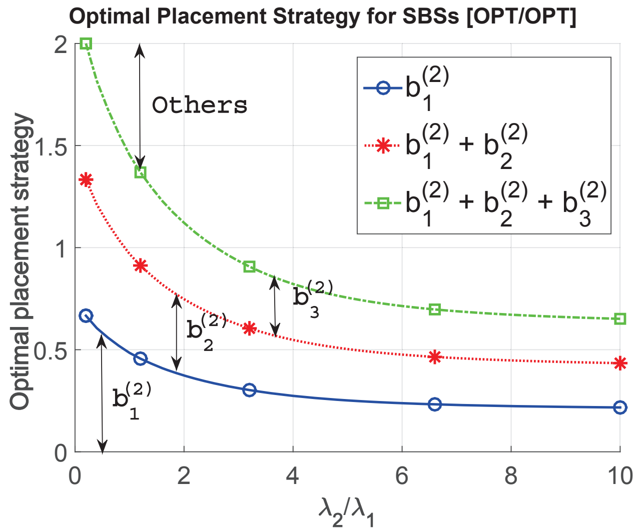

Fig. 2. Optimal content placement strategy $\bar{\boldsymbol{b}}^{(2)}$ for SBSs with different deployment densities λ_2 [OPT/OPT] (PPP).

Consider the case of two types of caches in the plane. Type-1 caches represent MBSs and type-2 caches represent SBSs with K_1 and K_2-slot cache memories, respectively. The content library size is $J = 100$. We set $K_1 = 1$ and $K_2 = 2$. We set the Zipf parameter $\gamma = 1$ and taking a_j according to (15). Also we set $\lambda_1 = 0.5$ and $r = 1$. We will consider various values for the deployment densities of SBSs, namely λ_2. Our aim is to compare the optimal content placement strategy with various heuristics. Therefore, we will compare various scenarios in the remainder of this subsection.

For the [OPT/OPT] scenario first we find the optimal solution for MBSs assuming that there are no SBSs in the plane. Solving this optimization problem gives $\bar{\boldsymbol{b}}^{(1)} = (0.7136, 0.2723, 0.0141, 0, \ldots, 0)$. The optimal content placement strategy results in storing the first three most popular files into MBSs with given probabilities and the resulting hit probability is $f\left(\bar{\boldsymbol{b}}^{(1)}\right) = 0.1649$. Then we set $\boldsymbol{b}^{(1)^c} = \bar{\boldsymbol{b}}^{(1)}$, take it as an input, add SBSs into the plane and find the optimal content placement strategy $\bar{\boldsymbol{b}}^{(2)}$ for SBSs. We observe that applying the iterative optimization procedure, namely repeatedly updating $\bar{\boldsymbol{b}}^{(1)}$ and $\bar{\boldsymbol{b}}^{(2)}$, does not improve the hit probability. From Figure 2, after deploying SBSs we see that when the deployment density of SBSs is low compared to MBSs, only the first three most popular files are stored in SBSs. As the deployment density of SBSs increases, we see that probability of storing less popular files in the caches increases.

For the [POP/OPT] scenario we consider a heuristic for the placement in MBSs: The most popular content c_1 is stored at MBSs with probability (w.p.) 1, i.e., $\bar{\boldsymbol{b}}^{(1)} = (1, 0, \ldots, 0)$. The resulting hit probability is $f\left(\bar{\boldsymbol{b}}^{(1)}\right) = 0.1527$. It is important to note that the hit probability under this heuristic policy is not significantly different from the optimal hit probability.

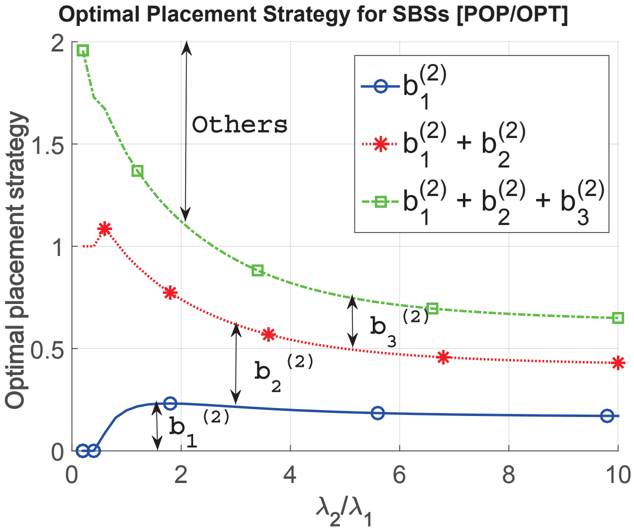

Fig. 3. Optimal content placement strategy $\bar{\boldsymbol{b}}^{(2)}$ for SBSs with different deployment densities λ_2 [POP/OPT] (PPP).

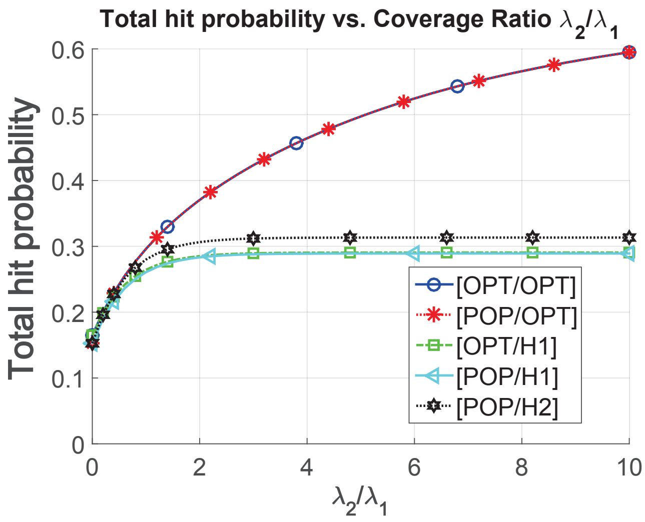

Fig. 4. The total hit probability evaluation for different SBS deployment densities (PPP).

Next we add SBSs. We set $\boldsymbol{b}^{(1)^c} = \bar{\boldsymbol{b}}^{(1)}$, take it as an input, add SBSs into the plane and find the optimal content placement strategy $\bar{\boldsymbol{b}}^{(2)}$ for SBSs. From Figure 3, after deploying SBSs we see that when the deployment density of SBSs is low compared to MBSs, only c_2 and c_3 are stored in SBSs with probability 1 in order to increase the chance of the user to get these files. As the deployment density of SBSs increases, we see that probability of storing c_1 in SBSs slightly increases. However, we see that since we cannot get c_2 and c_3 from any of the MBSs, probabilities of deploying c_2 and c_3 are always higher than c_1. Probability of storing other files also increases as the λ_2 increases.

From Figure 4, we see that the difference between the hit probabilities using the [OPT/OPT] and the [POP/OPT] heuristics decreases as the density of SBSs increases, i.e. we can use a heuristic placement policy for MBSs as finding the optimal content placement strategy for SBSs compensate the performance penalty caused by ill-adjusted content placement of MBSs. However, as we will see next, no straightforward heuristic seems to exist for the content placement policy in SBSs.

The first heuristic is to use is to store the two most popular files in SBSs, denoted by [OPT/H1] and [POP/H1] for an optimal and a heuristic policy in MBS, respectively. The second heuristic is to store the second and third most popular files in SBSs [POP/H2]. From Figure 4 it is clear that these heuristic policies achieve significantly lower hit probability than the optimal policy.

B. M-or-none deployment model

Type-1 caches represent MBSs and follow a two-dimensional (2D) spatial homogeneous Poisson process with density $\lambda_1 > 0$, and number of MBSs within radius r follows a Poisson distribution satisfying (16) with $i = 1$.

We assume that if a user is covered by at least one MBS (type-1 cache), then it should be covered by M caches in total. As a result, network operators serve users with providing them M caches as long as they are connected to one of the macro base stations. If a user is not in the coverage area of a MBS, then it doesn't receive any service from other caches either at all. Therefore, we have

$$P\left(\boldsymbol{\mathcal{N}}_2^L = \boldsymbol{n}_2^L | N_1 = 0\right) = \begin{cases} 0 & \text{if } \max\{\boldsymbol{n}_2^L\} > 0, \\ 1 & \text{if } \max\{\boldsymbol{n}_2^L\} = 0, \end{cases}$$

and

$$P\left(\boldsymbol{\mathcal{N}}_2^L = \boldsymbol{n}_2^L | N_1 = n_1\right) = \begin{cases} 0 & \text{if } \sum_{l=2}^L n_l \neq (M - n_1)^+, \\ 1 & \text{if } \sum_{l=2}^L n_l = (M - n_1)^+, \end{cases}$$

when $n_1 > 0$.

Consider the case of two types of caches in the plane. Type-1 caches represent MBSs and type-2 caches represent SBSs with K_1 and K_2-slot cache memories, respectively. The content library size is $J = 100$. We set $K_1 = 1$ and $K_2 = 2$. Also, we set the Zipf parameter $\gamma = 1$ and taking a_j according to (15). Also we set $\lambda_1 = 0.5$ and $r = 1$.

For the [OPT/OPT] scenario, optimal content placement strategy for MBSs is $\bar{\boldsymbol{b}}^{(1)} = (0.7136, 0.2723, 0.0141, 0, \ldots, 0)$ and the resulting hit probability is $f\left(\bar{\boldsymbol{b}}^{(1)}\right) = 0.1649$. Then we set $\boldsymbol{b}^{(1)^c} = \bar{\boldsymbol{b}}^{(1)}$, take it as an input and find the optimal content placement strategy $\bar{\boldsymbol{b}}^{(2)}$ for different M values. We observe that using the iterative optimization procedure, namely repeatedly updating $\bar{\boldsymbol{b}}^{(1)}$ and $\bar{\boldsymbol{b}}^{(2)}$, does not improve the hit probability. From Figure 5, we see that the probability of storing less popular files increases as M increases.

For the [POP/OPT] scenario, we have $\bar{\boldsymbol{b}}^{(1)} = (1, 0, \ldots, 0)$ and the resulting hit probability is $f\left(\bar{\boldsymbol{b}}^{(1)}\right) = 0.1527$. Then we set $\boldsymbol{b}^{(1)^c} = \bar{\boldsymbol{b}}^{(1)}$, take it as an input, add SBSs into the plane and find the optimal content placement strategy $\bar{\boldsymbol{b}}^{(1)}$ for SBSs. From Figure 6, as c_1 is stored in MBSs w.p. 1 and SBSs are present in the system only when $n_1 > 0$, c_1 is never stored at SBSs. We see that probability of storing c_2 and c_3 decreases and probability of storing other files increases as M increases.

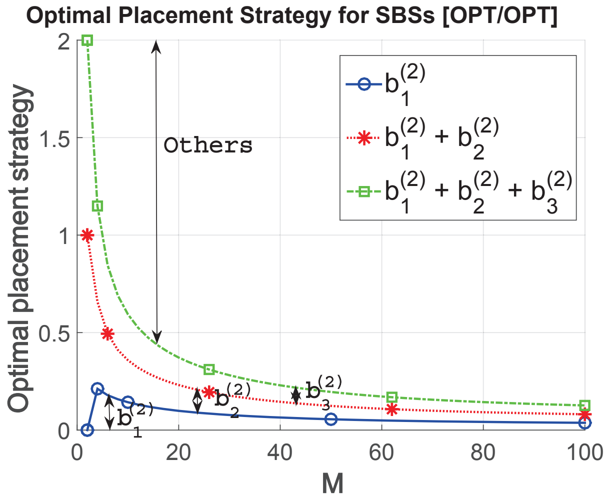

Fig. 5. Optimal content placement strategy $\bar{\boldsymbol{b}}^{(2)}$ for SBSs for different M values [OPT/OPT] (M-or-None).

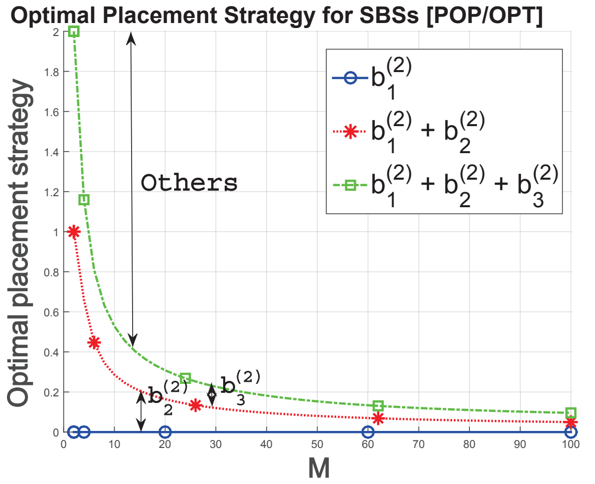

Fig. 6. Optimal content placement strategy $\bar{\boldsymbol{b}}^{(2)}$ for SBSs for different M values [POP/OPT] (M-or-None).

From Figure 7, we see that the hit probabilities under the [OPT/OPT] and the [POP/OPT] are identical and increase as M increases, i.e. we can use a heuristic placement policy for MBSs. For heuristic SBS deployment policies, [OPT/H1], [POP/H1] and [POP/H2] policies achieve significantly lower hit probability than the optimal policy.

V. Discussion and Conclusion

In this paper we have shown that whether MBSs use the optimal deployment strategy or store "the most popular

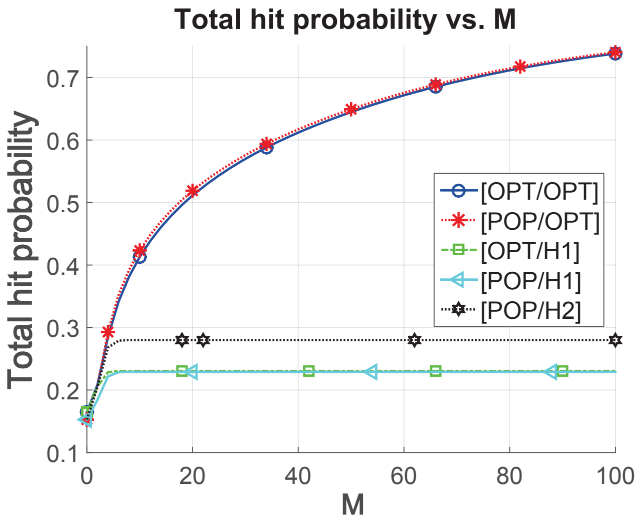

Fig. 7. The total hit probability evaluation for different M values (M-or-None).

content", has very limited impact on the total hit probability if the SBSs are using the optimal deployment strategy. It is important to optimize the content placement strategy of the SBSs and the total hit probability is increased significantly when the SBSs use the optimal deployment strategy. It is shown that heuristic policies for SBSs like storing the popular content that is not yet available in the MBS results in significant performance penalties. To conclude, using the optimal deployment strategy for the SBSs is crucial and it ensures the overall network to have the greatest possible total hit probability independent of the deployment policy of MBSs. Finally, we have shown that solving the individual problem to find optimal content placement strategy for different types of base stations iteratively, namely repeatedly updating the placement strategies of the different types, does not improve the hit probability.

References

[1] "Cisco VNI Global Mobile Data Traffic Forecast, 2015-2020", San Jose, CA, USA, Feb. 2016.

[2] J. Andrews, H. Claussen, M. Dohler, S. Rangan, M. Reed, "Femtocells: Past, Present and Future", *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 3, pp. 497-508, 2012.

[3] N. Golrezaei, A. F. Molisch, A. G. Dimakis, G. Caire, "Femtocaching and Device-to-Device Collaboration: A New Architecture for Wireless Video Distribution", *IEEE Communications Magazine*, vol. 51, no. 4, pp. 142-149, 2013.

[4] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, G. Caire, "FemtoCaching: Wireless Content Delivery Through Distributed Caching Helpers", *IEEE Transactions on Information Theory*, vol. 59, no.12, pp. 8402-8413, 2013.

[5] K. Poularakis, G. Iosifidis, L. Tassiulas, "Approximation Algorithms for Mobile Data Caching in Small Cell Networks", *IEEE Transactions on Communications*, vol. 62, no.10, pp. 3665-3677, 2014.

[6] B. Błaszczyszyn, A. Giovanidis, "Optimal Geographic Caching In Cellular Networks", *IEEE International Conference on Communications (ICC) 2015*, pp. 3358 - 3363, London, UK, 2015.

[7] M. A. Maddah-Ali, U. Niesen "Fundamental Limits of Caching", *IEEE Transactions on Information Theory*, vol. 60, pp. 2856 - 2867, 2014.

[8] S. Ioannidis, L. Massoulie, A. Chaintreau, "Distributed Caching over Heterogeneous Mobile Networks", *SIGMETRICS 2010*, pp. 311-322, NY, USA, 2010.

[9] K. Poularakis, L. Tassiulas, "Exploiting User Mobility for Wireless Content Delivery", *IEEE International Symposium on Information Theory Proceedings (ISIT) 2013*, pp. 1017-1021, Istanbul, Turkey, 2013.

[10] E. Baştuğ, M. Bennis, M. Debbah, "Cache-enabled Small Cell Networks: Modeling and Tradeoffs", *11th International Symposium on Wireless Communications Systems*, pp. 649-653, 2014.

[11] E. Altman, K. Avrachenkov, J. Goseling, "Distributed Storage in the Plane", *Networking Conference, IFIP 2014*, pp. 1-9, Trondheim, Norway, 2014.

[12] K. Avrachenkov, X. Bai, J. Goseling, "Optimization of Caching Devices with Geometric Constraints", *arXiv preprint arXiv: 1602.03635*, 2016.

[13] Z. Chen, J. Lee, T. Q. S. Quek, M. Kountouris, "Cooperative Caching and Transmission Design in Cluster-Centric Small Cell Networks", *arXiv preprint arXiv: 1601.00321*, 2016.

[14] M. Dehghan, L. Massoulie, D. Towsley, D. Menasche, Y. C. Tay, "A Utility Optimization Approach to Network Cache Design", *Proc. IEEE Infocom*, San Francisco, CA, USA, 2016.

[15] F. Baccelli and B. Blaszczyszyn, *Stochastic Geometry and Wireless Networks, Volume I - Theory*, ser. Foundations and Trends in Networking Vol. 3: No 3-4, pp 249-449. NoW Publishers, 2009, vol. 1.

Appendix A
Proof of Theorem 1

From (9), (10) and (11), we have

$$\bar{\omega}_j = \bar{b}_j^{(1)} \left[a_j \sum_{n_1=0}^{\infty} n_1 (1-b_j^{(1)})^{n_1-1} p_{n_1}^{(1)} q(j,n_1) - \bar{\nu} \right], \tag{18}$$

which, when insterted into (10), gives

$$\bar{b}_j^{(1)} \left(\bar{b}_j^{(1)} - 1 \right) \left[a_j \sum_{n_1=0}^{\infty} n_1 (1-b_j^{(1)})^{n_1-1} p_{n_1}^{(1)} q(j,n_1) \right] = \bar{\nu}. \tag{19}$$

From (19), we see that $0 < \bar{b}_j^{(1)} < 1$ only if

$$\bar{\nu} = a_j \sum_{n_1=0}^{\infty} n_1 (1-b_j^{(1)})^{n_1-1} p_{n_1}^{(1)} q(j,n_1).$$

Since we know that $0 \leq b_j^{(i)} \leq 1$, this implies that

$$\bar{\nu} \in \left[a_j p_1^{(1)} q(j,1),\ a_j \sum_{n_1=0}^{\infty} n_1 p_{n_1}^{(1)} q(j,n_1) \right].$$

If $\bar{\nu} < a_j p_1^{(1)} q(j,1)$, we have

$$\bar{\omega}_j = \bar{\lambda}_j + a_j \sum_{n_1=0}^{\infty} n_1 (1-b_j^{(1)})^{n_1-1} p_{n_1}^{(1)} q(j,n_1) - \bar{\nu} > 0.$$

Thus, from (10), we have $\bar{b}_j^{(1)} = 1$. Similarly, if $\bar{\nu} > a_j \sum_{n_1=0}^{\infty} n_1 p_{n_1}^{(1)} q(j,n_1)$, we have

$$\bar{\lambda}_j = \bar{\omega}_j + \bar{\nu} - a_j \sum_{n_1=0}^{\infty} n_1 (1-b_j^{(1)})^{n_1-1} p_{n_1}^{(1)} q(j,n_1) > 0.$$

Hence, from (9), we have $\bar{b}_j^{(1)} = 0$.

Finally, since $\sum_{j=1}^{J} b_j^{(1)}$ is a decreasing function in ν, solving J equations of (13) satisfying (14) give the unique solution $\bar{\nu}$.