On the Efficiency of Lightweight Content Placement Heuristics for Cache-Enabled Networks

Vaggelis G. Douros, Janne Riihijärvi, Petri Mähönen
Institute for Networked Systems, RWTH Aachen University
Kackertstrasse 9, 52072 Aachen, Germany
E-mail:{vaggelis.douros, jar, pma}@inets.rwth-aachen.de

***Abstract*—Cache-enabled networks have received increasing attention in both wired and wireless settings. A big challenge for the operator of such networks is to solve efficiently the content placement problem, *i.e.,* to decide how many caches to deploy in the network and in which nodes. We study the content placement problem for two classes of network optimisation objectives, the first focusing on the minimisation of the sum of the shortest paths and the second capturing the cost vs. benefit trade-off to deploy a cache. We know from the state-of-the-art that, even in small networks with few caches, it is unrealistic to find the optimal solution in a reasonable timescale for similar optimisation problems. In order to cope with this challenge, we present an approach under the prism of network analysis. We introduce a family of lightweight heuristic algorithms that use graph-theoretic metrics that identify the most important nodes of the network. We evaluate the performance of the heuristics using real network datasets, showing that the best heuristics are based on the metrics of betweenness centrality and degree centrality. Finally, we provide a randomised version of the heuristics noticing that the same metrics present again the best performance across the different datasets. Moreover, we find out that, in general, the deterministic version of each heuristic outperforms its randomised version.**

I. Introduction

Nowadays, due to the increasing demand for mobile multimedia content, operators face the challenge of redesigning the wireless networks in order to enable high data-rate and low-latency content delivery. Introducing cache-enabled wireless networks that store popular contents at the network edge (gateway routers, base stations of different sizes, end-user devices) has emerged as a promising candidate for future 5G wireless networks since it has the potential to significantly reduce the load at the backhaul [1].

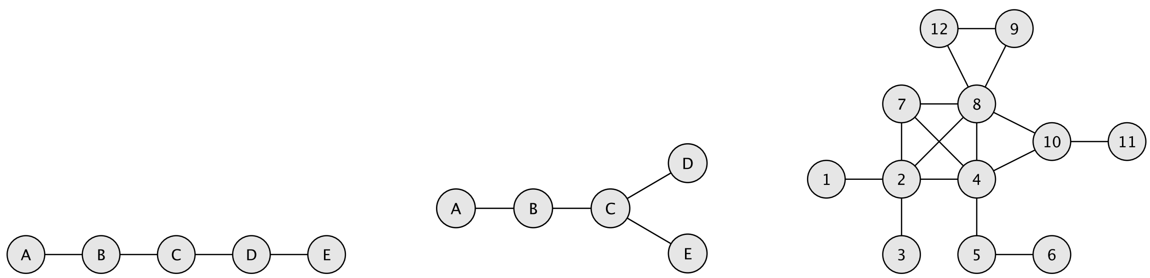
The idea of cache-enabled networks has already been used extensively in the wired domain; Content Delivery Networks (CDNs) and Information-Centric Networks (ICNs) are examples of this. A CDN is an overlay network which consists of a set of distributed servers placed in strategic locations that replicate the content of the original server with the view to decrease its load and reduce the latency by hosting contents close(r) to the end-users [2]. ICN supports in-network caching mechanisms to enhance content delivery, and thus each router has storage space to cache frequently requested content [3].

In both wired and wireless cache-enabled networks, the content placement problem is a key design challenge; the operator needs to decide where and how many caches to deploy. The decision depends on the optimisation metric such as the minimisation of the network cost or the satisfaction of the Quality-of-Service (QoS) of end-users. Three families of approaches have been used to solve the content placement problem: i) Find the exact solution of the optimisation problem. The main disadvantage is that even some simple instances of the content placement problem are NP-hard [4], meaning that the exact solution cannot be computed in a reasonable timescale and therefore finding the optimal solution has very limited practical interest. ii) Solve a relaxation problem with an approximation algorithm that has performance guarantees of the returned solution to the optimal one in the original problem. The main disadvantage is that many approximation algorithms face difficult implementation issues or present improved running time performance (over exact algorithms) only on impractically large inputs. iii) Propose a heuristic algorithm. This approach is simpler but there is no performance guarantee of the heuristic.

In this paper, we study the content placement problem in cache-enabled networks where the objective is either the minimisation of the sum of the shortest paths or a metric that quantifies the trade-off from the benefit of adding one cache to the network with the cost to deploy it. Our contributions are two-fold: i) We introduce a family of heuristic algorithms that use network analysis metrics in order to identify the most important nodes of the network [5]. These heuristics have two appealing design features: First, they can be used to solve both optimisation objectives, and, in principle, they are directly applicable (with minor modifications) to any set of similar optimisation objectives, even in adjacent fields (e.g., the function placement problem in edge networks [6]). Second, they are lightweight, since they can solve the content placement problem in milliseconds, which is an appealing property especially for wireless networks where the content changes dynamically. ii) We compare the performance of the heuristics using real network datasets and find out that two particular heuristics, based on the betweenness centrality and the degree centrality [5], perform better than the other heuristics in almost all network topologies. Moreover, we present a randomised version of the heuristics and find out that the same conclusion holds. We also compare the randomised version with the deterministic version and show that the latter outperforms the former for the majority of the network topologies.

II. Related Work

Authors of [4] survey content placement approaches in the context of ICN networks. Many optimisation problems lead to



(a) Chain network. (b) Y-network. (c) A complex network.

Fig. 1: Three networks to illustrate how the metrics identify the most important nodes.

NP-hard formulations and a number of heuristics have been proposed [7]. References [8] and [9] are the ones closest to our work. The authors study the content placement problem in a particular type of ICN called Content Centric Networking-CCN. In order to decide where to place the content, they take into account the path length reduction, the probability of a content being requested by a user and the storage constraints. They show that the optimisation problem is equivalent to a variation of the general knapsack problem, which is NP-hard. They then propose heuristics by choosing where to add the content based on a number of graph-theoretic centrality metrics.

In our work, we also use three centrality metrics (degree centrality, betweenness centrality, closeness centrality) that the authors in [8] and [9] have used. One difference is that they focus on different optimisation objectives; moreover, ours are generic, whereas in [8] and [9], the objective is CCN-specific. Another difference is that we analyse the performance of the heuristics based on datasets from real network topologies. The authors in [8] and [9] focus heavily on synthetic topologies; they just use one dataset from a real network in [9]. Finally, in our work, with the second objective function we take into account the cost to deploy a cache. This factor is not captured in [8], [9], since they consider physical storage constraints without examining any techno-economic parameters so that the operator can decide how many caches to deploy.

Reference [10] presents a comprehensive survey of content placement algorithms in the context of CDNs. In some cases, multiple algorithms are proposed; for example, [10] proposes two heuristics where an initial placement of the servers take place, following with a refinement procedure in order to remove redundant contents and to reduce the operational cost further. In general, the state-of-the-art does not consider centrality metrics in the context of CDN. The only exception is [11], where the authors use the betweenness centrality metric in order to assign the end-user to a particular cache based on his QoS requirements. This user-centric application of a centrality metric is complementary to our network-centric analysis. Finally, the authors in [12] introduce a different notion of centrality, which is based on the content and assumes knowledge of the content popularity.

In the wireless domain, the deployment of cache-enabled base stations should take into account a number of wireless performance metrics, such as the outage probability and average delivery rate as well as the effects of mobility [13]. In heterogeneous wireless networks, a user can retrieve the requested content from many network endpoints, therefore neighbouring caches should cooperate and avoid storing the same objects multiple times [14]. Moreover, the problem of caching has been studied along with multicast in [15], authors showing that it is NP-hard and deriving heuristics with performance guarantees. Centrality metrics have not been considered in the context of wireless cache-enabled networks.

III. Metrics that Identify the Most Important Nodes in a Network

In this section, we model a cache-enabled network as a graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ with $\mathbb{V}$, $\mathbb{E}$, being the set of nodes and links respectively. Since there is a bidirectional link between any two connecting nodes, we consider only undirected graphs. In this graph $\mathbb{G}$, the operator is interested in identifying the most important nodes. Towards this direction, we introduce the following 4 centrality metrics [5].

- Degree centrality (DC): The degree of a node is its most basic structural property, *i.e.,* the number of its incident links. DC chooses the nodes based on the analogy that someone who has many friends is the most important.
- Betweenness centrality (BC): The node betweenness expresses how often the node lies on the shortest path between all the other nodes of the network. In case that there are more than one shortest paths between two nodes, BC corresponds to the fraction of them that run through a particular node.
- Closeness centrality (CC): Closeness centrality of a node n is defined as the inverse of the sum of the shortest distances between n and every other node in the network. The lower the total distance from all other nodes, the more central is considered the node.
- Eigenenvector centrality (EC): Eigenvector centrality is based on the premise that a node's importance is determined by how important its neighbours are. Therefore, it takes into consideration not only how central the node is but also how central its neighbours are, using similar ideas with the Google's PageRank algorithm [5]. A high value of EC implies that a node is connected to a lot of nodes who themselves have high EC.

In Tables I and II, we present two numerical examples where we compute the above metrics for a chain network and a

TABLE I: Centrality metrics for the chain network.

Metric	A	B	C	D	E
DC	1	2	2	2	1
BC	0	3	4	3	0
CC	0.1	0.14	0.17	0.14	0.1
EC	0.5	0.87	1	0.87	0.5

TABLE II: Centrality metrics for the Y-network.

Metric	A	B	C	D	E
DC	1	2	3	1	1
BC	0	3	5	0	0
CC	0.11	0.17	0.2	0.13	0.13
EC	0.41	0.77	1	0.54	0.54

Y-network that depicted in Figs. 1a, 1b. In these examples, the ranking of the nodes is quite similar for all metrics. However, there are some interesting variations. In the chain network, all intermediates nodes have the same DC. However, when we use the other metrics, node C which is in the middle of the topology becomes the most important node. Consider e.g. the BC metric; there are 4 shortest paths that pass through C instead of 3 that pass through nodes B and D. In the Y-network, though nodes A, D and E which are at the edge of the network have the same DC and BC (the lowest one in the network), nodes D and E are considered more important than node A when it comes to EC. The reason is that they have as a direct neighbour node C, which is more important than the unique neighbour of node A, *i.e.,* node B.

Besides these centrality metrics, we also use two metrics that capture some local topological properties in a graph: the coreness metric (CM) and the local clustering (LC) coefficient. CM identifies tightly interlinked groups within a network: a k-core of a graph $\mathbb{G}$ is a maximal subgraph of $\mathbb{G}$ in which all nodes have degree at least k. Since we focus on connected networks, by definition, all nodes have degree at least 1 and belong to 1-core. Then, in order to compute the 2-core, we need to remove nodes of degree 1 until everything that has left has degree at least 2. On the other hand, the local clustering coefficient of a node n is defined as the number of pairs of neighbours of n connected by edges over the number of pairs of its neighbours [5]. LC measures the extent to which one's friends are also friends of each other.

We illustrate these metrics in Table III using the network of Fig. 1c (we do not use the chain and the Y-network, since these metrics need more complex networks in order to classify the nodes into different groups). A general comment is that each node who belongs to 1-core has always a LC of 0. Other than this, we notice that though node 5 has a degree 2, it belongs to 1-core, since its neighbour 6 has a degree 1. Regarding LC, node 7 has 3 neighbours and each pair of them are also neighbours, therefore its LC is 1, which is the maximum possible value.

IV. Network Optimisation Objectives and Our Heuristic Algorithms

Let $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ be a cache-enabled network and $\mathbb{K}$ be the set of nodes with a cache. We assume that the caches are identical. We denote with l_n the shortest path between a node $n \in \mathbb{V}$ and a cache in $\mathbb{K}$ (l_n=0 if $n \in \mathbb{K}$). In this work, we study the content placement problem for the following two network objectives.

TABLE III: CM and LC for the network of Fig. 1c.

	1	2	3	4	5	6	7	8	9	10	11	12
CM	1	3	1	3	1	1	3	3	2	2	1	2
LC	0	0.3	0	0.4	0	0	1	0.33	1	0.33	0	1

- Problem A: For a given cardinality K of the set $\mathbb{K}$, find whether a node n belongs to $\mathbb{K}$ so that the sum of the shortest paths $L = \sum_n l_n$ is minimised.
- Problem B: Given a fixed cost c per cache, find the cardinality K of the set $\mathbb{K}$ and whether a node n belongs to $\mathbb{K}$ so that the quantity $\sum_n l_n + cK$ is minimised.

In order to solve these problems, we apply a heuristic algorithm with six variations, each one for the metrics defined in the previous section. The steps of the algorithm for the problem A are:

1) For a given cardinality K of the set $\mathbb{K}$ and a particular metric, the operator computes the nodes' values for this metric. The K nodes with the highest values belong to $\mathbb{K}$, breaking randomly the ties.
2) For each node $n \in \mathbb{V}$, the operator computes the minimum shortest path l_n.
3) The sum of the shortest paths $L = \sum_n l_n$ corresponds to the solution of the heuristic.

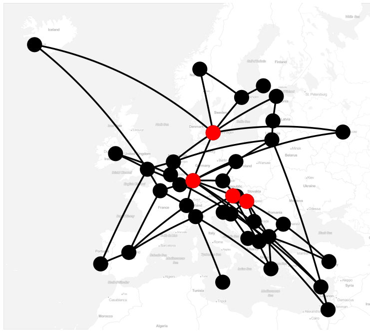
We stress that the solution corresponds to a local minimum, not a global minimum. The latter would demand the computation of all possible combinations of deploying K caches, which is practically too expensive, even when the operator is interested in deploying a small number of caches.

Regarding problem B, there are some changes in the heuristic algorithm. First, the operator needs to repeat V times the steps 1 and 2, since he does not know a priori whether the increase in the cardinality of set $\mathbb{K}$ will increase or decrease the value of the optimisation metric. Then, for each set of cardinality $K \in \{1, 2, \ldots, V\}$, he computes the quantity $\sum_n l_n + cK$ and chooses the set $\mathbb{K}$ that minimises it.

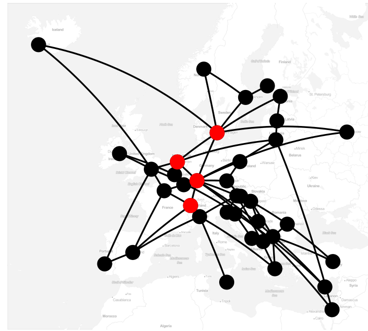
V. Performance Evaluation: Network Optimisation Problem A

For the performance evaluation, we use as a benchmark 15 network topologies from around the world which are available from the Internet Topology Zoo project (http://www.topology-zoo.org/): BICS (33 nodes), BT Europe (24 nodes), Canerie (32 nodes), Chinanet (42 nodes), DFN (58 nodes), Geant2012 (40 nodes), Grnet (37 nodes), HiberniaGlobal (55 nodes), Nsfnet (13 nodes), RedIris (19 nodes), RNP (31 nodes), Sinet (74 nodes), Sunet (26 nodes), VtlWavenet2011 (92 nodes), Xspedius (34 nodes). We use wired networks since we are interested in evaluating the heuristic algorithms in real topologies and there is lack of appropriate wireless network datasets.

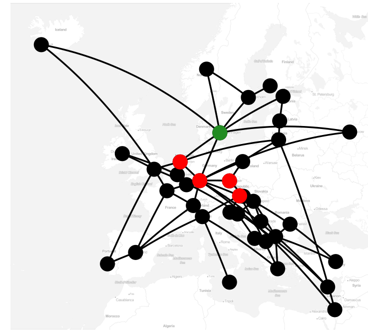
Fig. 2 presents six instances of the Geant2012 topology, where, for the optimisation problem A with $K = 4$, we have ran 1000 times the heuristic algorithm for each metric. The colour of a node visualises the number of times that a cache



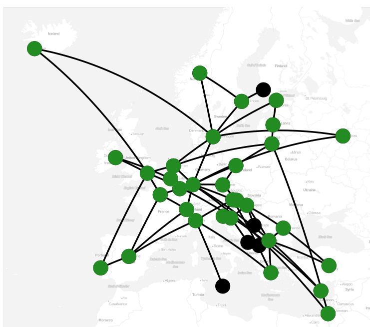
(a) BC. Shortest paths's sum $L = 54$.



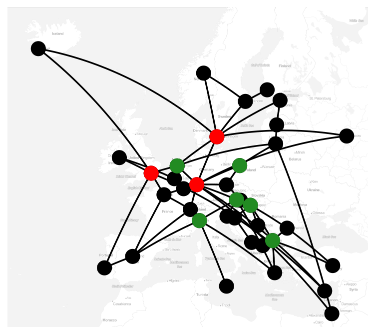
(b) EC. Shortest paths's sum $L = 71$.



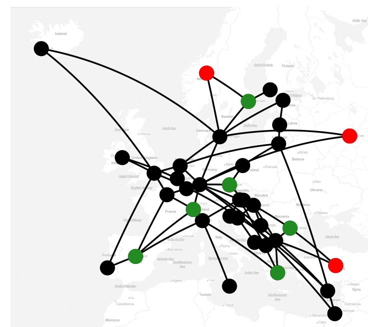
(c) CC. Shortest paths's sum $L = 66$.



(d) CM. Shortest paths's sum $L = 74.4$.

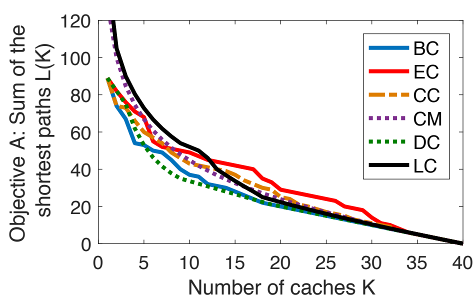


(e) DC. Shortest paths's sum $L = 62.2$.

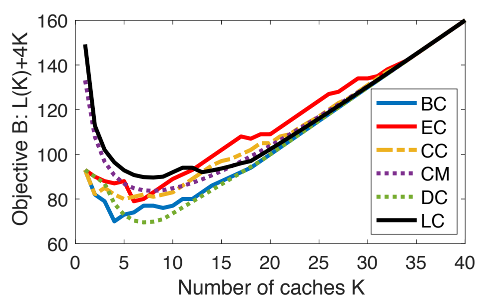


(f) LC. Shortest paths's sum $L = 80.6$.

Fig. 2: Example of the nodes that each heuristic chooses when 4 caches should be deployed in the Geant2012 topology. Green color means that this node gets a cache for less than 50% of the experiments, red that it gets a cache for more than 50% of the experiments, black that it never gets a cache.



(a) Problem A.



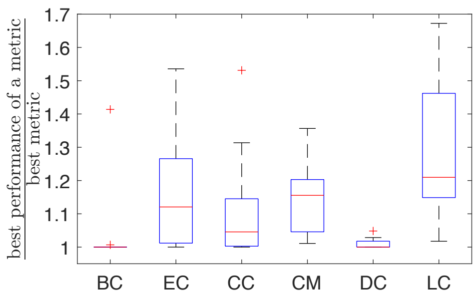
(b) Problem B.

Fig. 3: Performance of the heuristics for the Geant2012 topology for the objectives of problems A and B.

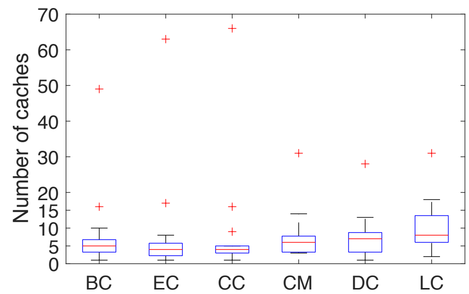
has been deployed in this node. We use red for the case that this node has been selected at least 500 times, green for 1-499 times and black when it is never selected. For BC and EC, the choice of the nodes was fixed; this means that 4 nodes had higher values for these metrics than all the other nodes. On the other hand, for the other metrics, we need to break ties and therefore more nodes have a chance to deploy a cache.

Regarding the performance of the heuristics, BC is the best heuristic, following by DC, CC, EC, CM and LC. We notice that BC shares two nodes with EC, CC and DC. Since BC performs better than the other metrics, we conclude that the other two nodes that it chooses are better candidates for the objective of problem A. Moreover, we note that LC has three fixed nodes which are found at the edges of the network and have no overlap with the choices of BC, EC, CC and DC. Apparently, these are not good choices for the optimisation metric. Finally, 32 nodes have the same CM and therefore the choice of the caches follows a uniform distribution among them, performing worse than BC, EC, CC and DC.

Then, we expand our analysis as the number of caches varies from 1 to 40 (all nodes have a cache). As expected, the sum of the shortest paths decreases with the number of caches, ending up to 0 for the (unrealistic) case that all nodes have a cache. Moreover, we notice from Fig. 3a that for a large number of caches, all metrics converge to the same point. This is reasonable since the heuristics tend to choose the same set



(a) Ranking of the heuristics.



(b) Number of caches.

Fig. 4: Relative performance of the heuristics and statistics on the number of caches.

of nodes to deploy a cache and the impact of having a couple of different nodes is small. For a smaller number of caches, DC (the simplest metric) and BC rank higher than the other metrics for the Geant2012 topology. We will identify whether this is a representative trend in the next section where we will study the network optimisation problem B.

VI. Performance Evaluation: Network Optimisation Problem B

We analyse the performance of the heuristics for the network optimisation problem B for the Geant2012 topology with a numerical example where the cost to deploy a cache is c=4. We ran the heuristics for 10000 times per topology. As expected, the optimisation metric is not a monotonic function of the number of caches. As we can see from Fig. 3b, when a lot of caches have been deployed, the addition of one more cache increases the value of the objective. This is reasonable since, in these cases,the marginal benefit from adding a cache is lower than the marginal cost. Moreover, we notice that the global minimum for each metric lies between 4 and 8 caches with DC and BC being again the best metrics.

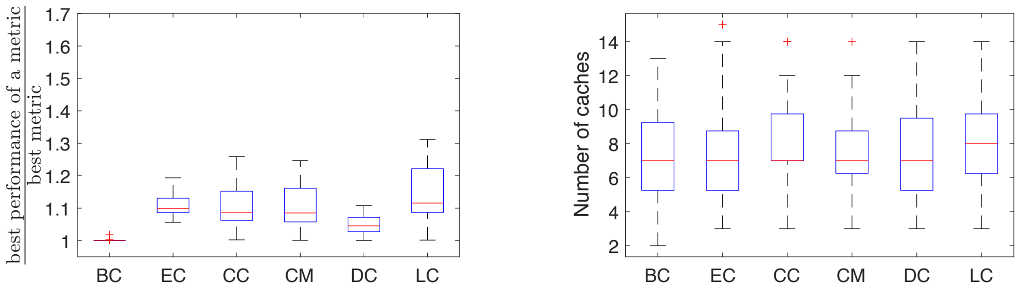
Next, we are interested in comparing the performance of the six heuristics across the 15 network topologies. Though the exact performance of the heuristics is topology dependent, we aim at identifying whether we can rank the metrics and whether there are some general trends. Towards this direction, we compute for each network topology the best performance for each metric. Then, we compute the normalised ratio y of the best performance of metric i over the best metric for this topology. Apparently, with this normalised metric y, the closer the value is to one, the better is the performance of the metric.

In order to visualise the performance of the six heuristics, we use boxplots. The bottom and top of the box are the first and third quartiles, and the red band inside the box is the median. The whiskers extend to the most extreme data points which are not outliers, and the outliers are plotted individually. From Fig. 4a, it is clear that BC is the best metric, performing better than the other metrics for 13 out of 15 network topologies. This is a clear indication that the way that BC chooses the nodes, evaluating how often the node lies on the shortest path between all the other nodes, is the best criterion for the optimisation problem B. Besides BC, DC is ranked 2^{nd} with a quite close performance to BC and very small variance. The third best metric is CC. Then, EC and CM are quite close with the former having a lower median value but a higher variance and whiskers than the latter. Finally, LC admits the worst performance both in terms of median value and in terms of the variance. It is worth mentioning that LC works very well for the topology VtlWavenet2011, which is the only one that BC underperforms. This topology consists of four back-to-back clusters where most nodes have degree 2 and local clustering coefficient 0. For such topology, it works better to choose the nodes from a uniform distribution and, therefore, DC and LC present the best performance. Finally, Fig. 4b depicts the boxplot of the number of caches for each metric. Ignoring the outliers that correspond to VtlWavenet2011, the optimal number of caches for the cost $c = 4$ is consistently less than 10.

VII. A Randomised Version of the Heuristics

In this section, we present a randomised version of the heuristics. Contrary to the deterministic version, a node n is chosen probabilistically, with a probability which is equal to $\frac{v_n}{\sum_{i=1}^{V} v_i}$, where v_n is the value for node n for the metric that the heuristic uses. The motivation for this randomised version is to give the chance to a heuristic to correct a potentially wrong choice of its deterministic version.

We evaluate the performance of the randomised version focusing on the optimisation problem B. We study exactly the same setup with the deterministic version of the heuristics. Initially, we present the boxplot based on the normalised metric y. As we can see from Fig. 5a, the randomised version of BC is again the best metric; even the unique outlier for VtlWavenet2011 is pretty close to the optimal metric, implying that there is an improvement for the relative performance of the randomised BC for this topology. DC is ranked again as the 2^{nd} best metric (with higher variance than the deterministic DC but still small). Then, CC and CM are quite close in the performance, following by EC which has a higher median value but lower variance than CC and CM. Moreover, again LC presents the worst performance, which however is closer to the optimal performance than its deterministic version. Finally, regarding the number of caches, as Fig. 5b reveals, the trend is quite similar with the deterministic version. For almost all cases, less than 10 caches are needed in order to optimise the objective of problem B.



(a) Ranking of the heuristics. (b) Number of caches.

Fig. 5: Relative performance of the randomised version of the heuristics and statistics on the number of caches.

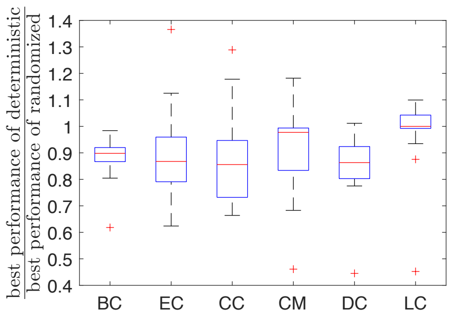


Fig. 6: Statistics on the best performance of the deterministic version of each heuristic vs. the best performance of its randomised version.

Finally, we compare the performance of each deterministic heuristic with its randomised version. For this reason, we compute the ratio of the best performance of a deterministic heuristic over the best performance of the randomised heuristic for each topology and present the outcome using boxplots in Fig. 6. It is clear that, excluding LC, the deterministic heuristic outperforms the randomised heuristic; only some whiskers and outliers are better for the randomised version. Even for the case of LC, the benefit from the randomised metric is small and not guaranteed. We conclude that, in general, it is better to stick to the choices of the deterministic metrics.

VIII. Conclusions

Motivated by the stringent need for extremely fast cache placement decisions in cache-enabled wireless networks where even end-users (through device-to-device communications) and other edge devices (through mobile edge computing) could potentially host caches, we introduce a family of lightweight content placement heuristics. Based on graph-theoretic metrics that measure instantly the importance of the nodes in the network, our heuristics are directly applicable for a number of network optimisation objectives. For the two objectives that we explicitly analysed their performance through real network datasets, our experiments revealed that betweenness centrality and degree centrality are ranked consistently higher than the other heuristics. Moreover, the deterministic version of the heuristics perform better than the randomised version.

Potential future directions include the performance evaluation with different objective functions, the analysis of their scalability with synthetic topologies, and machine learning extensions to predict content popularity.

IX. Acknowledgment

The work presented in this paper was supported by the EU funded H2020 ICT project POINT, under contract 643990.

References

[1] G. Paschos, E. Bastug, I. Land, G. Caire, and M. Debbah, "Wireless caching: Technical misconceptions and business barriers," *IEEE Communications Magazine*, vol. 54, no. 8, pp. 16–22, 2016.

[2] W. Jiang, S. Ioannidis, L. Massoulié, and F. Picconi, "Orchestrating massively distributed CDNs," in *ACM International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, 2012, pp. 133–144.

[3] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A survey of information-centric networking research," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1024–1049, 2014.

[4] G. Zhang, Y. Li, and T. Lin, "Caching in information centric networking: A survey," *Computer Networks*, vol. 57, no. 16, pp. 3128–3141, 2013.

[5] M. O. Jackson, *Social and economic networks*. Princeton University Press, 2010.

[6] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[7] M. Mangili, F. Martignon, and A. Capone, "Optimal design of information centric networks," *Computer Networks*, vol. 91, pp. 638–653, 2015.

[8] Y. Wang, Z. Li, G. Tyson, S. Uhlig, and G. Xie, "Optimal cache allocation for content-centric networking," in *IEEE International Conference on Network Protocols (ICNP)*, 2013.

[9] ——, "Design and evaluation of the optimal cache allocation for content-centric networking," *IEEE Transactions on Computers*, vol. 65, no. 1, pp. 95–107, 2016.

[10] J. Sahoo and R. Glitho, "Greedy heuristic for replica server placement in cloud based content delivery networks," in *IEEE Symposium on Computers and Communication (ISCC)*, 2016, pp. 302–309.

[11] C. Papagianni, A. Leivadeas, and S. Papavassiliou, "A cloud-oriented content delivery network paradigm: Modeling and assessment," *IEEE Transactions on Dependable and Secure Computing*, vol. 10, no. 5, pp. 287–300, 2013.

[12] J. Khan, C. Westphal, and Y. Ghamri-Doudane, "A popularity-aware centrality metric for content placement in information centric networks," in *International Conference on Computing, Networking and Communication (ICNC)*, 2018.

[13] C. Jarray and A. Giovanidis, "The effects of mobility on the hit performance of cached D2D networks," in *International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 2016.

[14] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2012, pp. 1107–1115.

[15] K. Poularakis, G. Iosifidis, V. Sourlas, and L. Tassiulas, "Exploiting caching and multicast for 5G wireless networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 4, pp. 2995–3007, 2016.