On the Joint Content Caching and User Association Problem in Small Cell Networks

M. Karaliopoulos, L. Chatzieleftheriou, G. Darzanos, I. Koutsopoulos

IEEE ICC®

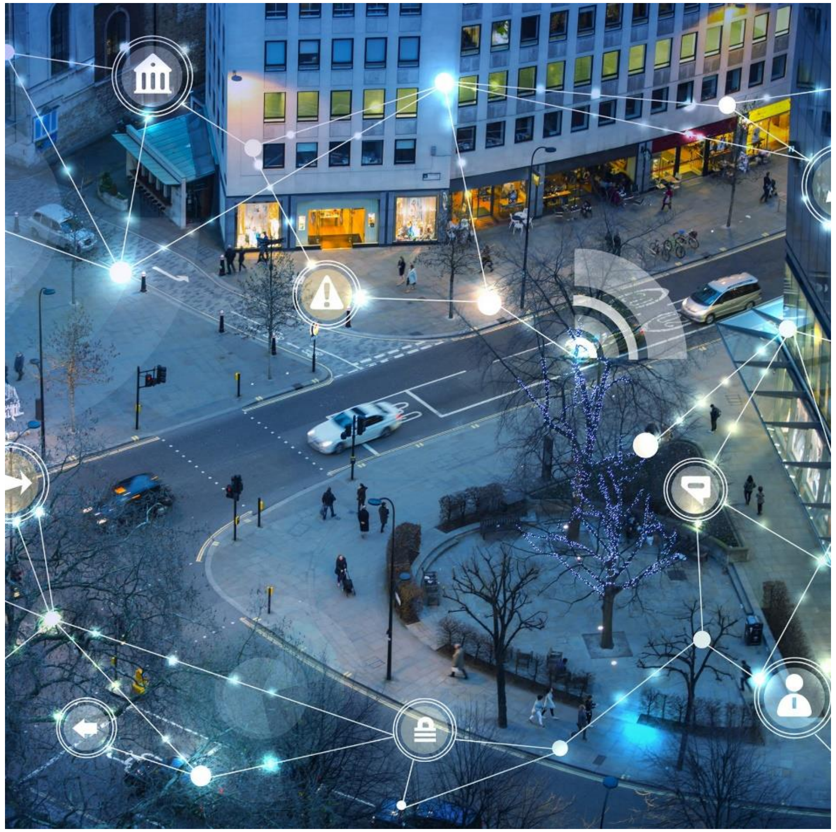
3rd Workshop on Ultra-high speed, Low latency and Massive Communication for Futuristic 6G Networks (ULMC6GN)

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

This research has been funded by the Operational Program "Human Resources Development, Education and Lifelong Learning", co-financed by European Union (EU) and Greek national funds.

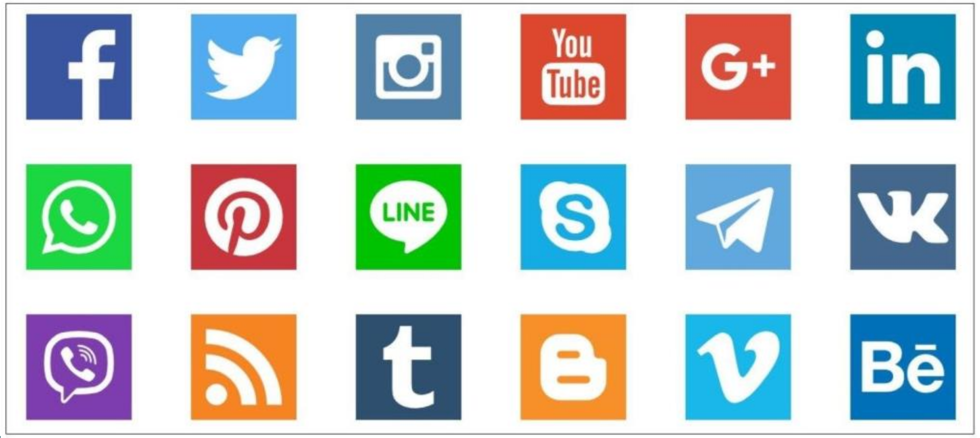
GSRT
GENERAL SECRETARIAT FOR RESEARCH AND TECHNOLOGY

Major persistent trends



- Network densification
  - small cells

- “Beat the clock” race
  - Requirement for faster and faster access, lower and lower latency



- Growing demand for content
  - Internet platformisation

Caching at the edge as enabler

- New waveforms alone do not suffice to fulfil the networks ambitious objectives
  - support is needed “beyond-the-radio layers”

- Bringing caching functionality at the mobile network edge has been discussed for quite some time
  - Different alternatives have been analyzed as to how close to the user these caches can reach
  - Several tradeoffs have emerged involving performance, communication encryption, adaptability to user access patterns
  - Possibilities to combine caching with resource management functions

This work

Focuses on the **joint content caching and user association problem (JCAP)**

- Users may be associated with a single cell plus a macro cell at the same time; content is replicated at multiple caches; each content request is directed towards the cache of the small cell the user is associated with.
- Implies the capability to reiterate upon cached content and existing user associations each time a new user emerges and needs to associate with the network.

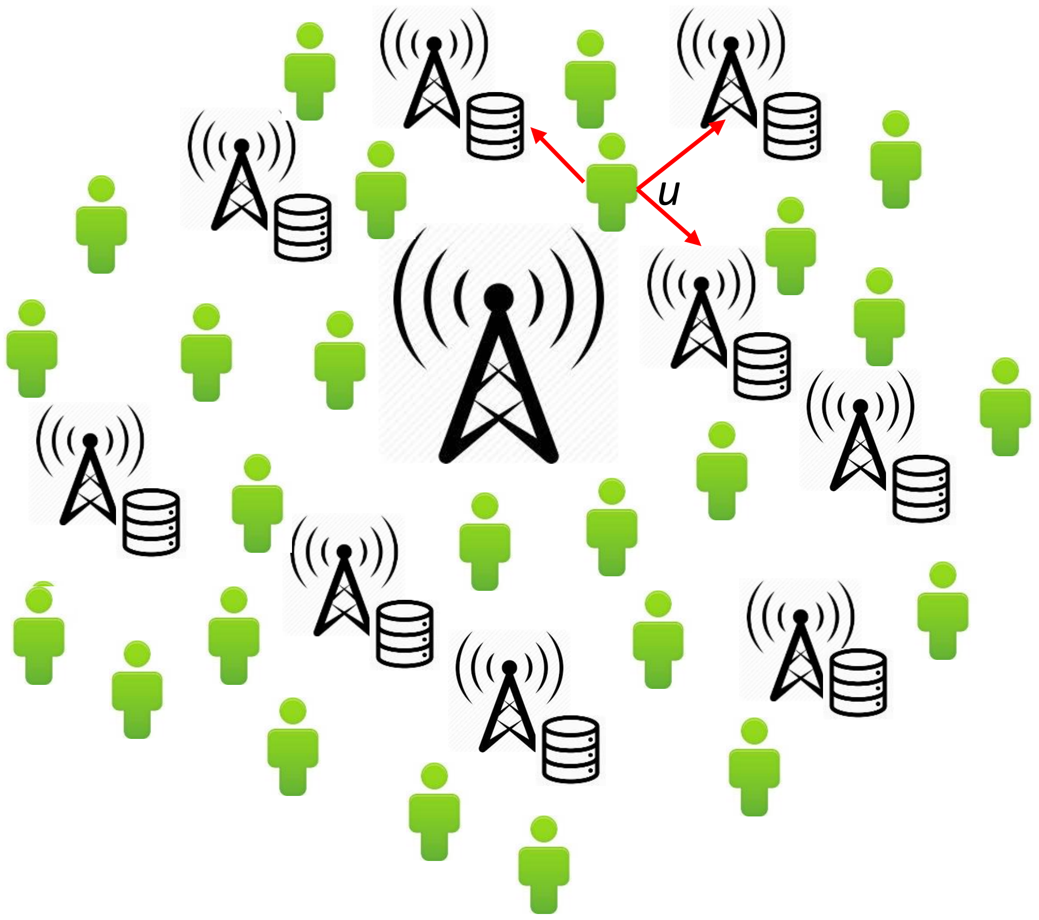
Contribution in a sentence

We propose, analyze, and assess a computationally efficient heuristic algorithm for the joint problem of content caching and user associations (JCAP) in dense small cell networks

System model - assumptions

- *I* : set of items
- *U* : set of users
- *C* : set of cache-enriched small cells (SBSs), besides the macro-cell
  - each cache with finite storage space L_c
  - each cell with finite capacity B_c

- *N(u)* : cells within range of user *u*
  - b_{uc} : association cost of user *u* to cell *c*
    - aggregate per-cell association cost is an additive function of individual association costs
  - p_{ui} : probability user *u* requests item *i*
    - perfect knowledge



The Joint content Caching and user Association Problem (JCAP)

- Two types of binary decision variables

$$x_{ic} = \begin{cases} 1 & if\ item\ i\ is\ stored\ at\ SBS\ cache\ c \\ 0 & otherwise \end{cases} \qquad y_{uc} = \begin{cases} 1 & if user\ u\ is\ associated\ with\ SBS\ c \\ 0 & otherwise \end{cases}$$

- The optimization problem becomes

$\max_{x,y}$	$\sum_{u \in U} \sum_{c \in N_u} \sum_{i \in I} p_{ui} x_{ic} y_{uc}$	(P1)	Aggregate cache hit ratio
s.t	$\sum_{i \in I} l_i\, x_{ic} \leq L_c,\ c \in C$	(1)	cache storage constraints
	$\sum_{u \in U} b_{uc}\, y_{uc} \leq B_c,\ c \in C$	(2)	cell capacity constraints
	$\sum_{c \in N(u)} y_{uc} \leq 1,\ u \in U$	(3)	each user can be associated with up to one SBS within her range
	$x_{ic}, y_{uc} \in \{0,1\},\ u \in U,\ c \in C,\ i \in I$		

JCAP characterization

- JCAP is an instance of bilinear programming
  - class of non-convex quadratic programming

- It is trivial to show that JCAP is NP-hard (by generalization)
  - Fixing variables $\{x_{ic}\}$, the problem reduces to an instance of the Maximum Generalized Assignment Problem
    - cells → bins, users → items, bin-specific item profits → the user demand satisfied by the content stored at each SBS cache
    - sometimes referred to as LEGAP in literature (e.g. Martello and Toth, *Knapsack problems*, pp. 190-191)
  - LEGAP is NP-hard and so is its generalization

Towards solving JCAP: linearization

- The products of binary variables in the JCAP objective can be linearized

- For each pair of variables (x_{ic}, y_{uc}), $u \in U$, $c \in N_u$, $i \in I$, a new binary variable $z_{iuc} = x_{ic}\, y_{uc}$ can be defined, subject to the additional constraints:
  - $z_{iuc} \leq x_{ic}$
  - $z_{iuc} \leq y_{uc}$
  - $z_{iuc} \geq x_{ic} + y_{uc} - 1$

- Plugging z_{iuc} in the JCAP objective function and adding these constraints to the (P1) formulation, we get an Integer Linear Program (ILP)
  - with $O(CIU)$ additional decision variables and $O(3CIU)$ additional constraints with respect to (P1)
  - solvable with generic ILP solvers for adequately small (C, I, U) values to get the optimal solution OPT_{JCAP}

An iterative heuristic solution to JCAP (1/4)

Initialization phase

- Determine the cache placement at each SBS cache assuming that all users within range of a given cell are associated with it
  - Set $y_{uc} = 1$ for each SBS $\in N_u$ → equivalent of solving (P1) relaxing the cell capacity and user association constraints
  - Each item $i \in I$ can satisfy demand $f_{ic} = \sum_{u:y_{uc}=1} p_{ui}$ when stored at cache $c \in C$

- Work independently with each cell cache $c \in C$

$$\max_{x} \quad \sum_{i \in I} f_{ic}\, x_{ic} \qquad \text{(P3a)}$$

$$\text{s.t} \quad \sum_{i \in I} l_i\, x_{ic} \leq L_c \qquad \text{cache storage constraints}$$

$$x_{ic} \in \{0,1\},\ i \in I$$

and end up solving C instances of the 0-1 Knapsack Problem (KSP) to determine the cache placements x_{ic}, $i \in I$, $c \in C$

An iterative heuristic solution to JCAP (2/4)

Iterative phase – user association step

For given cache placements $x_{ic}, i \in I, c \in C$ determine/update the user associations

- each user bears cell-specific association cost b_{uc} and cache-specific profit $f_{uc} = \sum_{i:x_{ic}=1} p_{ui}$

Then solve one instance of the Generalized Assignment Problem over the whole network

$$\max_{y} \quad \sum_{u \in U} \sum_{c \in \mathrm{N}_u} f_{uc} y_{uc} \qquad \text{(P3b)}$$

$$\text{s.t} \quad \sum_{u \in U} b_{uc} \, y_{uc} \le B_c, \, c \in C$$

$$\sum_{c \in N(u)} y_{uc} \le 1, \, u \in U$$

$$y_{uc} \in \{0,1\}, \, u \in U, \, c \in C$$

to determine the user associations to cells, $y_{uc}, u \in U, c \in C,$ and yield the first feasible solution of the problem

An iterative heuristic solution to JCAP (3/4)

Iterative phase – cache placement step

- For given user associations $y_{uc}, u \in U, c \in C,$ determine/update the cache placements
  - each item $i \in I$ can satisfy demand $f_{ic} = \sum_{u:y_{uc}=1} p_{ui}$ when stored at cache $c \in C$

- Then use these updated values of f_{ic} to solve anew the C instances of the 0-1 (KSP)

$$\max_{x} \quad \sum_{i \in I} f_{ic}\, x_{ic} \qquad \text{(P3c)}$$

$$\text{s.t} \quad \sum_{i \in I} l_i\, x_{ic} \leq L_c \qquad \text{cache storage constraints}$$

$$x_{ic} \in \{0,1\}, \; i \in I$$

and determine the cache placements $x_{ic}, i \in I, c \in C$-

An iterative heuristic solution to JCAP (4/4)

- Overall, the heuristic proceeds iterating between the two steps of the iterative phase, the cache placement step and the user association step.
- The solution produced in each step is checked against the current one and replaces it as far as it improves upon it in terms of achievable cache hit ratio.

Properties

- The algorithm is correct and terminates in a finite number of steps
  - Its achieved solution is upper bounded by the OPT_{JCAP} value
  - In general, it is a local maximum that may deviate from OPT_{JCAP}
    - the evaluation of the algorithm (see later slides) shows tight match
- The time complexity of the algorithm is $O(kCIL_c)$, k: number of iterations (no more than 10 in all experiments reported later)

Evaluation – set up

The evaluation process evolves in two steps:

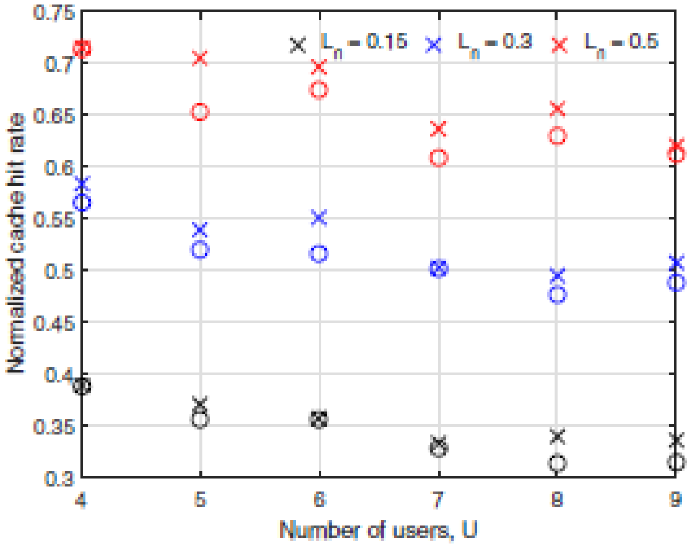
- Comparison of the heuristic solution with the optimal one
  - "Small" problem instances → the ILP solver can compute the optimal solution
  - Evidence about the accuracy of the algorithm – how well does it approximate the optimal solution

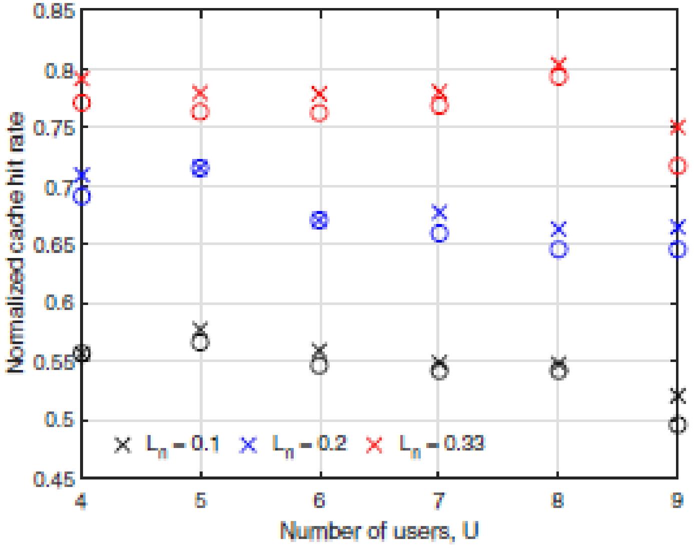
- Comparison of the heuristic solution with two alternative computationally feasible solutions
  - a **Greedy** algorithm and one that first determines the user associations and then the cache placements (**Decoupled**)
  - Realistic problem instances, amenable to sensitivity analysis and what-if scenarios

- In both steps
  - The item sizes and the user association costs vary randomly in $\{1, l_{max}\}$ and $\{1, b_{max}\}$, respectively
  - Two scenarios are considered for the content demand probabilities $\{p_{ui}\}$
    - Random → permutations of Zipf distributions are randomly assigned to users
    - Spatial Locality → users are clustered into N_{cl} clusters according to their physical location and identical distributions are assigned to each cluster

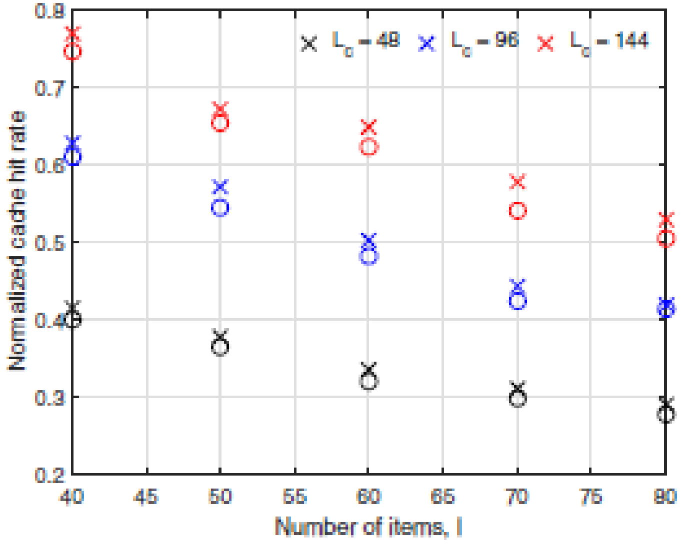
Small problem instances : heuristic vs. optimal

Variable users, C = 2, I = 100





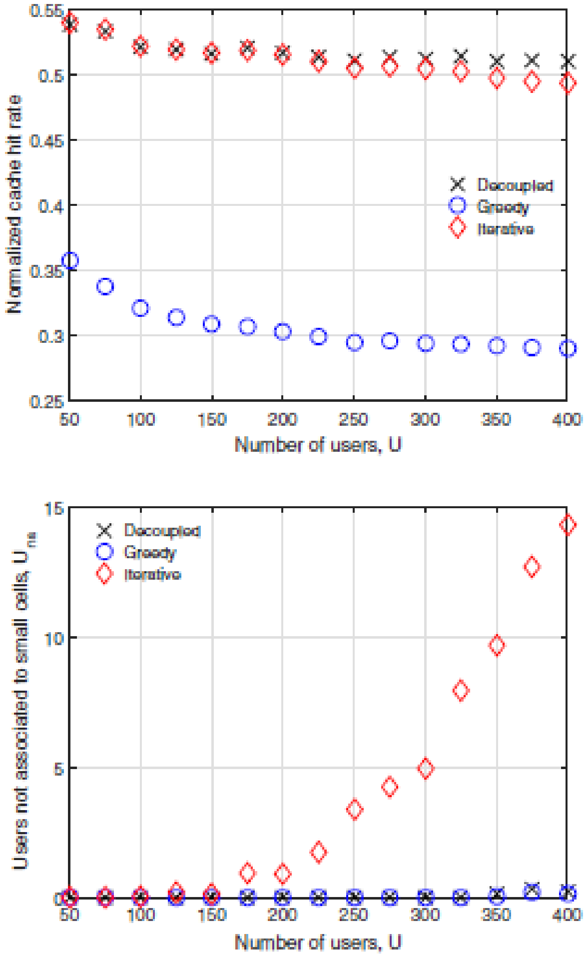
Variable items, C = 3, U = 8



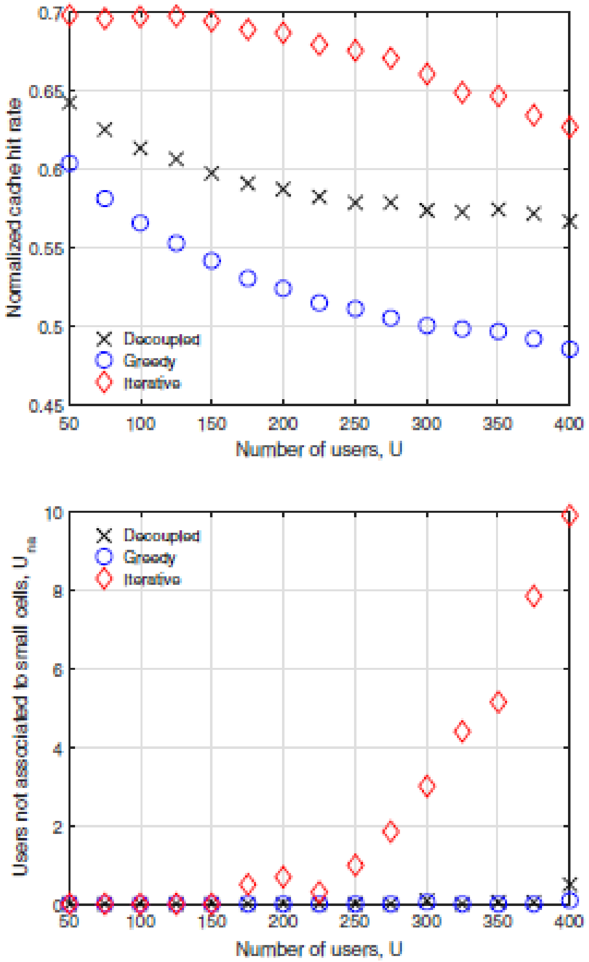
- HR_{heur} : cache hit ratio under the heuristic algorithm
- HR_{opt} : optimal cache hit ratio
- $\Delta H = HR_{opt} - HR_{heur}$
- $G = \frac{\Delta H}{HR_{opt}} 100\%$

Comparison scenario	var Users rand. demand	var Users clust. demand	var Items rand. demand
median $\Delta H(G)$	0.001(0.2%)	0(0%)	0.004(0.81%)
95^{th} perc. $\Delta H(G)$	0.097(17.7%)	0.1(14.2%)	0.081(13.3%)
max $\Delta H(G)$	0.161(29.2%)	0.21(25%)	0.189(35.2%)

Realistic problem instances : results



Random demand, $L_n = 0.3$



Spatially local demand, $N_{cl} = 10$, $L_n = 0.1$

General performance trends

- Under random demand, the Decoupled heuristic competes with our iterative heuristic
  - and even outperforms it at high user load, managing to associate all users with some cell within range, whereas our iterative heuristic directs some users to the macro cell
- Under spatially local demand, our heuristic achieves up to 12% higher cache hit rates than the Decoupled heuristic
  - in those cases it matters more *which* users are grouped in each cell rather than only *how many*
  - this advantage pertains over a broad scenario of cache sizes and cell capacities (refer to the paper)
- In all experiments, the greedy algorithm ranks last

Conclusions – steps forward

- We have proposed a computationally efficient and performance-wise effective iterative heuristic algorithm for the joint problem of content caching and user associations (JCAP) in dense small cell networks
  - The algorithm iteratively solves multiple instances of the 0-1 KSP to determine cache placements and an instance of maximum GAP to derive the user associations
  - As a side contribution, we have defined two more heuristics for the JCAP – these serve as comparison references in our work

- The algorithm exhibits very good performance, in particular for the more realistic scenarios of spatial locality in the user demand for content
  - the measured gains in our experimentation vary from 3-15% over the second best option
  - moreover, in experiments with small problem instances, the algorithm matches closely the optimal solution

- Open questions and paths forward
  - algorithmic front : approximability properties of the algorithm
  - evaluation front : use of real data, with more realistic footprints of spatial locality, to confirm the good performance of the algorithm

On the Joint Content Caching and User Association Problem in Small Cell Networks

M. Karaliopoulos, L. Chatzieleftheriou, G. Darzanos, I. Koutsopoulos

IEEE ICC

3rd Workshop on Ultra-high speed, Low latency and Massive Communication for Futuristic 6G Networks (ULMC6GN)

Send your comments/questions to: mkaralio@aueb.gr

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

This research has been funded by the Operational Program "Human Resources Development, Education and Lifelong Learning", co-financed by European Union (EU) and Greek national funds.

GSRT
GENERAL SECRETARIAT FOR RESEARCH AND TECHNOLOGY