

Optimal Deployment of Caches in the Plane

Mihaela Mitici*, Jasper Goseling*[‡], Maurits de Graaf*[†] and Richard J. Boucherie*

* Stochastic Operations Research, University of Twente, The Netherlands

[‡] Delft University of Technology, The Netherlands

[†] Thales B.V., The Netherlands

m.a.mitici@utwente.nl, j.goseling@utwente.nl, m.degraaf@utwente.nl, r.j.boucherie@utwente.nl

Abstract—We consider wireless caches placed in the plane according to a homogeneous Poisson process. A data file is stored at the caches, which have limited storage capabilities. Clients can contact the caches to retrieve the data. The caches store the data according to one of the two data allocation strategies: partitioning & coding. We consider the Pareto front of the expected deployment cost of the caches and the expected cost of a client retrieving the data from the caches. We show that there is a strong trade-off between the expected retrieval and the expected deployment cost under the partitioning and the coding strategies. We also show that under coding, it is optimal to deploy a high number of caches, each with low storage capacity.

Index Terms—Wireless communication, networks of caches, coding, cost optimization

I. INTRODUCTION

We consider wireless caches positioned in the plane according to a homogeneous Poisson process. The caches store a large data file. Clients are interested in retrieving the data from the caches. Due to the limited storage capabilities of the caches, a single cache cannot store the entire data.

Since the connection clients-caches is wireless, the cost of retrieving the content of a cache is increasing with the client-cache distance. This cost can represent, for example, the data rate that can be achieved or the delay, and is inversely proportional to the deployment intensity of the caches. One could reduce the expected cost experienced by clients by increasing the intensity of the caches, *i.e.*, the average number of caches per unit area. However, if one also takes into account the cost of deploying the caches, defined to be proportional to the storage capacity and the deployment intensity of the caches, a conflict between minimizing the client's retrieval cost and the caches' deployment cost arises. In the current work, we analyze the trade-off between these conflicting costs.

Depending on the storage strategy, clients have to contact different sets of caches. In previous work [1], several storage strategies were analyzed. The first strategy, partitioning, divides the data into equally-sized fragments. Each cache stores one particular fragment. In order to retrieve the data, a client needs to obtain all the different fragments. The second strategy is based on random linear coding. Each cache stores a random linear combination of the data fragments. Figure 1 illustrates the partitioning and the coding data allocation strategy. It was demonstrated in [1] that under a general cost measure for the data retrieval, coding outperforms the partitioning strategy.

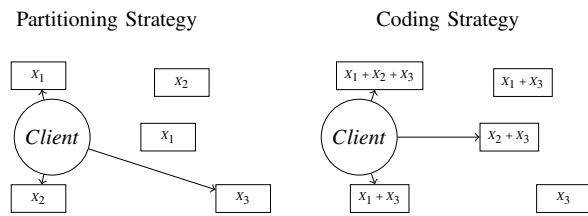


Fig. 1: Data allocation strategies

The use of coding was also explored in [2] where it was shown how to efficiently allocate the data at caches with the aim of ensuring that any sufficiently large subset of caches can provide the complete data. The difference with the current work is that we are taking the geometry of the deployment of the caches into account. In [3] coding strategies for networks of caches are presented, where each user has access to a single cache and a direct link to the source. It is demonstrated how coding helps to reduce the load on the link between the caches and the source. We assume that different transmissions from caches to the clients are orthogonal, for instance by separating them in time or frequency. In [4] the impact of non-orthogonal transmissions is considered and scaling results are derived on the best achievable transmission rates.

Other work on caching in wireless networks is, for instance, [5]–[7]. In [5] the authors analyze the trade-off between energy consumption and the retrieval delay of data from caches. In [6], the authors consider the optimal number of replicas of data such that the distance between a requesting node and the nearest replica is minimized. Data sharing among multiple caches such that the bandwidth consumption and the data retrieval delay are minimal is considered in [7]. None of [5]–[7] are considering coded caching strategies.

Our contribution in this paper is a characterization of the Pareto front of the data retrieval cost and the deployment cost of the caches. The parameters over which we optimize are: i) the intensity of the Poisson process with which caches are deployed and ii) the storage capacity of an individual cache.

We show that the retrieval and the deployment costs exhibit a strong trade-off both under the partitioning and the coding strategy, *i.e.* a small decrease in one of the costs is achieved at the expense of a significant increase in the other cost. For the

partitioning strategy, we show that the optimal cost depends only on the ratio of the deployment intensity and the storage capacity of the caches. Hence, when deploying caches, one has some flexibility in either increasing the cache intensity or the cache capacity. For the coding strategy, however, we show that the cost is minimized by maximizing the cache intensity while minimizing the storage capacity. Hence, while coding offers lower costs, it reduces deployment flexibility. Finally, we show that for a fixed deployment cost of the caches, the coding strategy results in a lower retrieval cost than partitioning.

II. MODEL AND PROBLEM STATEMENT

A data file consisting of n symbols, n fixed, is stored at the caches. The symbols are elements from a finite field \mathbb{F}_q .

Caches are positioned in the plane according to a homogeneous Poisson process with intensity λ , where $0 < \lambda \leq \lambda_{max}$ is a parameter over which we optimize. The caches have limited capacity that we express in terms of an integer k , which is the second optimization parameter. A cache can store $\lceil \frac{n}{k} \rceil$ symbols. To ensure that caches store at least one symbol and at most all n symbols, we assume that $1 \leq k \leq n$.

A client placed at a random location in the plane is interested in retrieving the data from the caches. We assume that the client has complete knowledge about the content and the location of the caches in the plane. The client requests data from a set of k caches that ensures the retrieval of the data.

Which caches are contacted depends on the way the data is stored at the caches. We consider two data allocation strategies: partitioning (P) and coding (C).

Partitioning is a data allocation strategy according to which the data is divided into k different fragments, each of $\frac{n}{k}$ symbols. Each cache selects uniformly at random a fragment to store. The client requests data from the k nearest caches such that each contacted cache has a different fragment than the other contacted ones.

Coding assumes that each cache stores a random linear combinations of the k fragments. We assume that the k nearest caches provide independent linear combinations from which the original data can be retrieved. While there is a probability that these linear combinations are not independent, it was shown in [1] that this has negligible influence on the cost. Hence, under the coding strategy, we analyze the cost of contacting the k nearest caches.

Next, we define the relevant cost measures. These cost measures are a function of the model parameters k and λ , over which we optimize. Firstly, we consider the cost of retrieving the data under allocation strategy $A \in \{P, C\}$. Let the cost of retrieving data from caches located at distances $\delta_1, \dots, \delta_l$ be

$$\tilde{C}_1(\delta_1, \dots, \delta_k) = \frac{n}{k} \sum_{i=1}^l \delta_i^a, \quad (1)$$

where $a \geq 1$ is an arbitrary, but fixed, parameter. In (1), δ_i^a is the cost of retrieving one symbol at distance δ_i and $\frac{n}{k}$ approximates the cost of doing so for $\lceil \frac{n}{k} \rceil$ symbols. We are interested in the expected cost $C_1^A(k, \lambda)$, where the expectation

is taken over the Poisson process according to which the caches are placed in the plane.

As discussed in [1], the above cost function captures several relevant performance measures. For instance, if $a > 2$ denotes the path loss exponent in the wireless medium, then the cost reflects the maximum achievable throughput, following from the channel capacity. For $a = 1$, the cost captures the delay under the TCP protocol.

Secondly, we consider the expected deployment cost of the caches in the plane per unit area. The cost of deploying a single cache is proportional to its storage capacity, which we approximate by $\frac{n}{k}$. Therefore, the expected cost per unit of area of deploying caches with density λ is defined as:

$$C_2(k, \lambda) = \frac{n}{k} \lambda, \quad (2)$$

where the expectation is taken over the randomness in the spatial Poisson process.

We consider the multi-objective optimization problem which aims at minimizing the deployment cost $C_2(k, \lambda)$ and the retrieval cost $C_1^A(k, \lambda)$ under the data allocation strategy A ,

$$\begin{cases} \min C_1^A(k, \lambda) \\ \text{and} \\ \min C_2(k, \lambda) \end{cases}, \text{ where } k \in \{1, 2, \dots, n\}, 0 \leq \lambda \leq \lambda_{max}. \quad (3)$$

In general, a single point simultaneously minimizing both objectives does not exist, in which case the multi-objective problem (3) does not have a unique optimal solution. Hence, we characterize the Pareto front as follows [8]: A feasible solution (k_1, λ_1) *dominates* a feasible solution (k_2, λ_2) if and only if $C_i(k_1, \lambda_1) \leq C_i(k_2, \lambda_2)$, $i \in \{1, 2\}$ with strict inequality for at least one of the inequalities. A feasible solution is a *Pareto optimal point* (k_{PO}, λ_{PO}) if there is no point (k, λ) such that (k, λ) dominates (k_{PO}, λ_{PO}) . All Pareto optimal solutions of a multi-objective function is called the *Pareto front*.

We will make use of the gamma function, which for $x > 0$ is represented as $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$ and the digamma function, $\psi(x) = \frac{d}{dx} \ln \Gamma(x) = \int_0^\infty \left(\frac{e^{-t}}{t} - \frac{e^{-xt}}{1-e^{-t}} \right) dt$ (see, for instance, [9]).

III. RESULTS

Multi-objective methods such as scalarization or goal programming can be used to compute the Pareto front. We refer to [10] for an extensive survey on methods to compute Pareto solutions. In this paper, the specific structure of the two objective functions allows us to directly determine the Pareto front. In our analysis we provide the Pareto front of optimization problem (3) with the relaxed non-integer constraint $1 \leq k \leq n$. By doing so, we identify the general form of the Pareto front for problem (3), which includes the Pareto optimal points that can be achieved with k integer.

A. Partitioning Strategy

We start with the following known result.

Theorem 1 ([1]). *The expected cost of retrieving the complete data file under the partitioning (P) strategy is:*

$$C_1^P(k, \lambda) = \frac{n}{k} k \left(\frac{k}{\lambda \pi} \right)^{\frac{a}{2}} \Gamma\left(\frac{a}{2} + 1\right). \quad (4)$$

Our first contribution is the following result:

Theorem 2. *The Pareto front for the partitioning (P) strategy consists of the following points:*

$$[C_1^P(k, \lambda), C_2^P(k, \lambda)] = \left[\frac{n^{(1+a/2)}}{(\pi\alpha)^{a/2}} \Gamma\left(\frac{a}{2} + 1\right), \alpha \right], \quad \alpha \in (0, n\lambda_{max}).$$

Proof. The two objectives (2) and (4) are related as follows,

$$C_1^P(k, \lambda) = \frac{n^{(1+a/2)}}{\pi^{a/2}} \frac{\Gamma(\frac{a}{2} + 1)}{C_2(k, \lambda)^{a/2}}.$$

If we consider all pairs (k, λ) with $1 \leq k \leq n$, $0 < \lambda \leq \lambda_{max}$ such that $C_2(k, \lambda) = \alpha$, $0 < \alpha \leq n\lambda_{max}$, then

$$C_1^P(k, \lambda) = \frac{n^{(1+a/2)}}{\pi^{a/2}} \frac{\Gamma(\frac{a}{2} + 1)}{\alpha^{a/2}}.$$

□

The above results demonstrate that the Pareto optimal costs $C_1^P(k, \lambda)$ and $C_2^P(k, \lambda)$ depend on λ and k only through the ratio $\frac{\lambda}{k}$. Also note that $C_2^P(k, \lambda)$ takes all values in $(0, \lambda_{max})$. Hence, the Pareto front is a continuous curve.

B. Coding Strategy

Theorem 3 ([1]). *The expected cost of retrieving the complete data file under the coding (C) strategy is:*

$$C_1^C(k, \lambda) = \frac{n}{k} \left(\frac{1}{\lambda \pi} \right)^{\frac{a}{2}} \frac{\Gamma(\frac{a}{2} + 1 + k)}{(\frac{a}{2} + 1)\Gamma(k)}. \quad (5)$$

Before the main result, we state the following lemma, which is proved in the appendix:

Lemma 1. *Given $k \in [1, n]$ and $x > 1$, $f(k, x) = k\psi(k+x) - k\psi(k) - x < 0$.*

Theorem 4. *The Pareto front for the coding strategy consists of the following points:*

$$[C_1^C(k, \lambda), C_2^C(k, \lambda)] = \begin{cases} \left[\frac{\Gamma(\frac{a}{2} + 1 + n)}{(\pi\alpha)^{a/2} (\frac{a}{2} + 1)\Gamma(n)}, \alpha \right], & \alpha < \lambda_{max} \\ \left[\frac{\alpha \Gamma(\frac{a}{2} + 1 + \frac{n\lambda_{max}}{\alpha})}{\pi^{\frac{a}{2}} (\frac{a}{2} + 1) \lambda_{max}^{(1+\frac{a}{2})} \Gamma(\frac{n\lambda_{max}}{\alpha})}, \alpha \right], & \lambda_{max} \leq \alpha \end{cases}$$

where $0 < \alpha \leq n\lambda_{max}$.

Proof. Consider all pairs (k, λ) with $1 \leq k \leq n$, $0 < \lambda \leq \lambda_{max}$ such that $C_2^C(k, \lambda) = \alpha$, $0 < \alpha \leq n\lambda_{max}$. Then (5) becomes,

$$C_1^C(k, \lambda) = \frac{n^{(1+a/2)} \Gamma(\frac{a}{2} + 1 + k)}{\pi^{\frac{a}{2}} (\frac{a}{2} + 1) \Gamma(k) k^{(1+a/2)}} \cdot \frac{1}{\alpha^{\frac{a}{2}}} = g(k, \alpha) \frac{1}{\alpha^{\frac{a}{2}}}, \quad (6)$$

where in the last equality $g(k, \alpha) = \frac{n^{(1+a/2)} \Gamma(\frac{a}{2} + 1 + k)}{\pi^{\frac{a}{2}} (\frac{a}{2} + 1) \Gamma(k) k^{(1+a/2)}}$.

W.l.o.g. assume (k_0, λ_0) dominates all points (k, λ) for which $C_2^C(k_0, \lambda_0) = C_2^C(k, \lambda) = \alpha$. Then $C_1^C(k_0, \lambda_0) < C_1^C(k, \lambda)$.

Clearly, for all pairs (k, λ) with $1 \leq k \leq n$, $0 < \lambda \leq \lambda_{max}$ such that $C_2^C(k, \lambda) = \alpha$, $C_1^C(k, \lambda) = g(k, \alpha) \left(\frac{1}{\alpha}\right)^{\frac{a}{2}}$ is minimized when $g(k, \alpha)$ is minimal. Taking the derivative with respect to k , we have that

$$\frac{\partial g(k, \alpha)}{\partial k} = \frac{n^{(1+a/2)} \Gamma(\frac{a}{2} + 1 + k)}{\pi^{\frac{a}{2}} (\frac{a}{2} + 1) \Gamma(k) k^{(2+a/2)}} f\left(k, 1 + \frac{a}{2}\right) < 0$$

since $f(k, x) = k\psi(k+x) - k\psi(k) - x < 0$, $1 \leq k \leq n$, $x = 1 + \frac{a}{2} > 1$, (see Lemma 1). Hence, $g(k, \alpha)$ is a decreasing function. Therefore, $C_1^C(k, \lambda)$ is minimized when k is maximal.

Now consider the following cases:

Case 1: $0 < \alpha < \lambda_{max}$

We look for the largest possible k such that $C_2^C(k, \lambda) = \alpha = n \frac{\lambda}{k}$ with $1 \leq k \leq n$ and $0 < \lambda < \lambda_{max}$. By choosing $k = n$, the largest value for k , $\lambda = \alpha$ and $0 \leq \lambda < \lambda_{max}$.

Case 2: $\lambda_{max} \leq \alpha \leq n\lambda_{max}$

We look for the largest possible k given that $C_2^C(k, \lambda) = \alpha = n \frac{\lambda}{k}$. To have the highest possible k , we increase λ as much as possible. Hence, we consider $\lambda = \lambda_{max}$. Then $k = \frac{n\lambda_{max}}{\alpha}$ and $1 \leq k \leq n$.

We now show that the front obtained for Cases 1 and 2 does not contain dominated points, i.e., if $C_2^C(k_0, \lambda_0) = \alpha_0 < C_2^C(k_1, \lambda_1) = \alpha_1$, then $C_1^C(k_0, \lambda_0) > C_1^C(k_1, \lambda_1)$.

For Case 1, it is obvious that $C_2^C(k, \lambda)$ and $C_1^C(k, \lambda)$ are inversely proportional. We further analyze Case 2, when $k = \frac{n\lambda_{max}}{\alpha}$. Using relation (6), we have the following:

$$\frac{C_1^C(k_0, \lambda_0)}{C_1^C(k_1, \lambda_1)} = \frac{g(k_0, \alpha_0)}{g(k_1, \alpha_0)} \left(\frac{\alpha_1}{\alpha_0} \right)^{\frac{a}{2}} = \frac{\Gamma(\frac{a}{2} + 1 + k_0)}{\Gamma(k_0) \cdot k_0} \cdot \frac{\Gamma(k_1) \cdot k_1}{\Gamma(\frac{a}{2} + 1 + k_1)}.$$

Since $\alpha_0 < \alpha_1$, it follows that $k_1 < k_0$. It only remains to show that $\frac{k\Gamma(k)}{\Gamma(x+k)}$, $x = 1 + \frac{a}{2}$ is a decreasing function in k . Taking the first derivative with respect to k ,

$$\frac{\partial}{\partial k} \frac{k\Gamma(k)}{\Gamma(x+k)} = -\frac{\Gamma(k)}{\Gamma(x+k)} [k\psi(x+k) - k\psi(k) - 1] < 0,$$

where in the last inequality we used that $k\psi(x+k) - k\psi(k) - 1 > 0$, $1 \leq k \leq n$, $x > 1$, which follows after similar computations as for the proof of Lemma 1. Hence, $\frac{k\Gamma(k)}{\Gamma(x+k)}$, $x = 1 + \frac{a}{2}$ is decreasing in k . Therefore, $\frac{C_1^C(k_0, \lambda_0)}{C_1^C(k_1, \lambda_1)} > 1$ for $k_1 < k_0$ or, equivalently, for $\alpha_0 < \alpha_1$. □

The above results show that the Pareto optimal cost $C_1^C(k, \lambda)$ is based on either taking the maximum possible k , i.e. it is optimal to fragment the data as much as possible given a deployment cost of $C_2^C(k, \lambda) < \lambda_{max}$, or considering the maximum deployment density λ_{max} , i.e. deploy as densely as possible the caches in the plane.

Theorem 5. *Each of the Pareto points under partitioning are dominated by a Pareto point under coding, i.e. $\frac{C_1^P}{C_1^C} > 1$, $n \geq 1$.*

Proof. For $n = 1$, the entire data is stored at each cache. In this case, the coding and partitioning strategies coincide. We now analyze the case when $n > 1$, $C_2^P = C_2^C = \alpha$, $0 < \alpha \leq n\lambda_{max}$.

$$\frac{C_1^P}{C_1^C} = \begin{cases} \frac{n^x \Gamma(x) \Gamma(n)}{\Gamma(x+n)} = \omega(x, n), & 0 < \alpha \leq \lambda_{max}, \\ \frac{z^x \Gamma(x) \Gamma(z)}{\Gamma(x+z)} = \omega(x, z), & \lambda_{max} \leq \alpha \leq n\lambda_{max}, \end{cases}$$

where $x = \frac{a}{2} + 1$, $\frac{\lambda_{max}}{\alpha} = y$, $\frac{1}{n} \leq y \leq 1$, $1 \leq z = ny \leq n$.

Taking the first derivative

$$\frac{\partial \omega(x, n)}{\partial n} = \frac{xn^{x-1}\Gamma(n)\Gamma(x)[n\psi(n) - n\psi(n+x) + x]}{\Gamma(x+n)} > 0,$$

where $[n\psi(n) - n\psi(n+x) + x] > 0$ (from Lemma 1).

Hence, $\omega(x, n)$ is increasing in n and $\lim_{n \rightarrow 1} \omega(x, n) = 1$.

Therefore, $\omega(x, n) = \frac{C_1^p}{C_1^c} > 1$, $n > 1$, $0 < \alpha \leq \lambda_{max}$. Similarly,

$\omega(x, z) = \frac{C_1^p}{C_1^c} > 1$, $n > 1$, $\lambda_{max} \leq \alpha \leq n\lambda_{max}$. \square

IV. DISCUSSION AND CONCLUSIONS

This paper has provided insight into the trade-off between the expected deployment cost of caches in the plane and the expected cost of a client retrieving a large data file from the caches. For the partitioning strategy, we derived a simple relation for the Pareto points. For the coding strategy, we showed that it is optimal to have small-sized caches and to densely deploy the caches in the plane.

Figure 2 shows the general form of the Pareto front of the two conflicting objectives under the partitioning and coding strategy. The two objectives have a trade-off curve with a steep slope, which indicates that a small decrease in the expected retrieval cost is achieved at the expense of a large increase in the deployment intensity of the caches and/or their storage size, *i.e.*, the expected deployment cost. It was

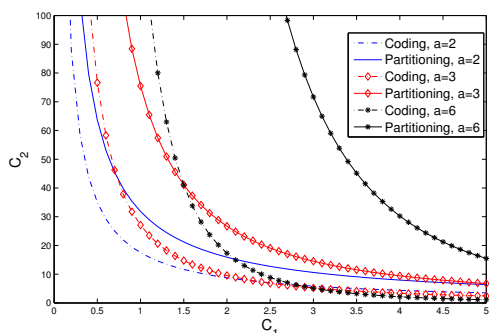


Fig. 2: General form of the Pareto front, $n = 10$.

proved in [1] that coding incurs a lower expected retrieval cost than partitioning for fixed cache deployment intensity λ and cache capacity $\frac{n}{k}$. This property is maintained for the trade-off of the expected deployment and retrieval cost. This is a generalization of the results in [1], where the number of caches k to be contacted and the deployment intensity λ are fixed. In this paper we also showed that optimizing over k and λ results in lower retrieval costs under coding than under partitioning.

The Pareto front provides a valuable insight into the overall cost of the cache storage system. Since the cache deployment and the data retrieval costs are dependent, optimizing for one of the objectives necessarily influences the other. The Pareto front shows to what extent one of the objectives can be improved at the expense of the other. This allows to optimize the overall cost of the system.

ACKNOWLEDGMENTS

This work was performed within the project RRR (Realisation of Reliable and Secure Residential Sensor Platforms) of the Dutch program *IOP Generieke Communicatie*, IGC1020, supported by the *Subsidieregeling Sterktes in Innovatie*.

APPENDIX

PROOF OF LEMMA 1

Let $f(k, x) = k\psi(k+x) - k\psi(k) - x$, $x > 1$ and $1 \leq k \leq n$. We show that $f(k, x) < 0$.

$$\begin{aligned} f(k, x) &= k\psi(k+x) - k\psi(k) - x \\ &= \int_0^\infty \frac{ke^{-kt}(1-e^{-xt})}{(1-e^{-t})x} x dt - x \\ &= \int_0^\infty h(x, t)(ke^{-kt})x dt - x, \end{aligned}$$

where we defined $h(x, t) = \frac{(1-e^{-xt})}{x(1-e^{-t})}$.

First, we show that $h(x, t) < 1$ for $x > 1, t > 0$. This follows from

$$\frac{\partial h(x, t)}{\partial x} = \frac{e^{-xt}(1+tx) - 1}{x^2(1-e^{-t})} < \frac{e^{-xt} \cdot e^{xt} - 1}{x^2(1-e^{-t})} = 0,$$

where, for the last inequality we used $1+tx < e^{xt}$. Thus, $h(x, t)$ is a decreasing function in x and $\lim_{x \rightarrow 1, t \rightarrow 0} h(x, t) = 1$. Consequently, $h(x, t) < 1$.

Now, since $h(x, t) < 1$ and $\int_0^\infty ke^{-kt} = 1$,

$$f(k, x) = \int_0^\infty h(x, t) \cdot ke^{-kt} \cdot x dt - x < 0.$$

REFERENCES

- [1] E. Altman, K. Avrachenkov, and J. Goseling, "Coding for caches in the plane," *arXiv preprint:1309.0604*, 2013.
- [2] A. G. Dimakis, V. Prabhakaran, and K. Ramchandran, "Ubiquitous access to distributed data in large-scale sensor networks through decentralized erasure codes," in *Proceedings of the International Symposium on Information Processing in Sensor Networks*, 2005, p. 15.
- [3] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *arXiv preprint arXiv:1209.5807*, 2012.
- [4] U. Niesen, D. Shah, and G. W. Wornell, "Caching in wireless networks," *IEEE Transactions on Information Theory*, vol. 58, no. 10, pp. 6524–6540, 2012.
- [5] P. Nuggehalli, V. Srinivasan, and C.-F. Chiasserini, "Energy-efficient caching strategies in ad hoc wireless networks," in *Proceedings of the International Symposium on Mobile Ad hoc Networking & Computing*, 2003, pp. 25–34.
- [6] S. Jin and L. Wang, "Content and service replication strategies in multi-hop wireless mesh networks," in *Proceedings of the International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2005, pp. 79–86.
- [7] L. Yin and G. Cao, "Supporting cooperative caching in ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 1, pp. 77–89, 2006.
- [8] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [9] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables*.
- [10] G. P. Liu, J. F. Whidborne, and J.-B. Yang, *Multiojective Optimisation and Control*. Research Studies Press, 2003.