Social similarity favors cooperation: the distributed content replication case

Eva Jaho, Merkourios Karaliopoulos, and Ioannis Stavrakakis

Abstract—This paper explores how the degree of similarity within a social group can dictate the behavior of the individual nodes, so as to best trade-off the individual with the social benefit. More specifically, we investigate the impact of social similarity on the effectiveness of content placement and dissemination. We consider three schemes that represent well the spectrum of behavior-shaped content storage strategies: the selfish, the self-aware cooperative, and the optimally altruistic ones. Our study shows that when the social group is tight (high degree of similarity), the optimally altruistic behavior yields the best performance for *both* the entire group (by definition) and the individual nodes (contrary to typical expectations). When the group is made up of members with almost no similarity, altruism or cooperation cannot bring much benefit to *either the group or the individuals* and thus, selfish behavior emerges as the preferable choice due to its simplicity. Notably, from a theoretical point of view, our "similarity favors cooperation" argument is inline with sociological interpretations of human altruistic behavior. On a more practical note, the self-aware cooperative behavior could be adopted as an easy to implement distributed alternative to the optimally altruistic one; it has close to the optimal performance for tight social groups and the additional advantage of *not allowing mistreatment of any node*, *i.e.*, its induced content retrieval cost is always smaller than the cost of the selfish strategy.

Index Terms—Content Replication, Cooperation, Similarity, Social Groups.

1 Introduction

Networks today can be highly personalized, in the sense that their structure and usage are shaped by the personal interests, or behavior in general, of the participating nodes. Nodes in such networks – referred to as social networks – are typically well connected, develop reciprocal trust relations, and share some attributes, such as content interests and locality. Groups of such nodes are called *social groups* [27].

In this paper we consider a group of networked nodes with common preferences for content –or, more generally, data objects such as files and software. The node-members of this social group can store content in their limited local storage and retrieve it when desired at a minimum cost. When a desired object is not stored locally, nodes may either fetch it from the cache of another node-member of the group at low-medium cost or, if the object is not available within the group, from a node outside the group at a higher cost. The low-medium cost associated with fetching an object from within the group may reflect low actual or virtual price such as lower access delay due to locality or high connectivity, or higher level of trust and reliability. We assume that nodes have established trust relationships in order to belong to the same social group, and have access to each others' caches. This may be realized through various schemes (*e.g.*, see [11], [26], [28]).

• *The authors are with the Department of Informatics and Telecommunications, National & Kapodistrian University of Athens, Ilissia, 157 84 Athens, Greece.*
E-mail: {ejaho, mkaralio, ioannis}@di.uoa.gr

Since the local storage is considered to be limited when compared with the plethora of possibly interesting content, an inherently selfish node would opt for storing locally objects of highest interest to it. Our past work in [19] has shown that, in a distributed group with three levels of content access cost as considered here, this is not the best content placement strategy for a node. Instead, a cooperative content placement strategy has been devised on the basis of game-theoretic arguments, whereby nodes exchange information about their content placements and synchronize in adjusting them. This way they avoid storing objects that could be fetched efficiently from other nodes so that the resulting cost for each and every node is at most (and typically much lower than) that induced under the selfish strategy (*mistreatment-free* property). In the present paper, we refer to this placement strategy as the *self-aware cooperative strategy*, to emphasize its cooperative nature and mistreatment-free property.

Mistreatment-free strategies are key to the sustainability of such distributed selfish groups, as they motivate the participation in the group and sharing of objects. The social benefit induced by the self-aware cooperative strategy, *i.e.*, the aggregate benefit of all group nodes, is not in general optimal. The content placement strategy that maximizes the social benefit will be referred to hereafter as the *optimally altruistic strategy*; it can be derived by solving an optimization problem, as done, for instance, in [20]. The practical implementation of the optimally altruistic strategy requires the exchange of richer information among the nodes in the group (complete local demand distributions); whereas, under the self-aware cooperative strategy, nodes need to exchange much less information (indices of locally stored content). Finally, although the optimally altruistic strategy maximizes the aggregate benefit, some of the nodes may end up with high gains and others being mistreated.

Focus of this paper: In view of the above it is evident that a node participating in a distributed group can face a dilemma as to which strategy to follow.

- The *selfish* strategy requires no interaction with and guarantees no mistreatment by the other nodes; on the other hand, both the node itself and the group could benefit more by following another strategy.
- The *self-aware* cooperative strategy outperforms the selfish strategy both at the individual node and group levels, while it ensures no mistreatment of individual nodes; on the other hand, it does not maximize the group benefit, while it introduces complexity that increases with the group size and may outweigh the benefits for individual nodes.
- The *optimally altruistic* strategy yields the maximum possible benefit for the entire group. On the other hand, it can mistreat certain nodes (with the risk of inciting them to leave the group) and requires heavier interaction with the other nodes of the group (increasing complexity); these interactions could be lighter under a centralized derivation of the optimal placement.

In this paper the characteristics of the social group are exploited to address the above dilemma. To this end, we follow an innovative approach to the characterization of the nodes' similarity within a social group and introduce a group *tightness* metric, which explicitly accounts for the level of similarity in their content preferences. Our work highlights the impact of the metric on the induced social and individual node benefits under the three aforementioned content placement strategies, which reflect general patterns of social behavior. It draws important insights as to how rewarding such behaviors can be under given levels of social group tightness.

This study has applications to social networks featuring interactions between computer devices with limited memory resources. These are typically encountered in mobile opportunistic networks that are additionally "socially aware", meaning that either the nodes or their human users are aware of the formation of social groups and the potential benefits from participation in such a group. The underlying assumption is that there are multiple groups, and we focus on the behavior regarding the exchange of information objects between nodes inside a single group. Studying content access patterns, especially between nodes in a social group is quite important to assess the viability of various networking paradigms, *e.g.*, the opportunistic wireless networking and some P2P systems.

The remainder of this paper is organized as follows. In Section 2 we present the three content placement strategies and their main properties. We introduce the tightness metric, a novel metric for capturing the similarity of content preferences within the group based on the Kullback-Leibler divergence between preference distributions, and discuss its appropriateness in Section 3. The evaluation methodology and scenarios are presented in Section 4. In Section 5 we analyze the selfish and self-aware cooperative placement strategies and derive sufficient conditions under which the latter yields the same placements with the selfish and the optimal strategies. The content placement strategies are numerically compared under different *tightness* values in Section 6. We also assess the convergence time of the cooperative algorithm in the more realistic scenario that nodes gradually learn the content preferences of the users behind them and adapt their placements accordingly. Our conclusions about the cooperation gains achievable under different tightness values are further validated in Section 7, where the content preferences of users are extracted by crawling data from an online social bookmarking service. We contrast our contributions against related work in Section 8 and summarize the major conclusions of the paper in Section 9, drawing parallels to outcomes of sociological studies of human behavior.

2 Content Placement Strategies and Related Tradeoffs

Let $\mathcal{N} = \{1, 2, ..., N\}$ denote the set of the nodes in a social group and let $\mathcal{M} = \{1, 2, ..., M\}$ denote the set of objects (or items) these nodes are interested in. Let F_m^n denote the preference probabilities of node n, for object m, and let $F^n = \{F_1^n, F_2^n, ..., F_M^n\}$; F_m^n can be viewed as the normalized rate of requests for object m by node n.

Let P_n denote the *placement* at node n, defined to be the set of objects stored locally at that node with storage capacity C_n. All objects are considered to be of the same size. Throughout the paper, we are interested in placements of cardinality $|P_n| = C_n$ since, under any placement strategy, the node has always motivation to fully utilize its buffer space. More formally, for any placement P_n with $|P_n| < C_n$, we can always find at least one other placement P'_n with $P_n \subseteq P'_n$ and $|P'_n| = C_n$, which (weakly) dominates P_n in terms of cost. Let $\mathcal{P} = \{P_1, P_2, \ldots, P_N\}$ denote the global placement for the social group and $\mathcal{P}_{-n} = \mathcal{P} \setminus P_n$ denote the set of placements for all group nodes but node n.

Let t_l, t_r and t_s denote the cost for accessing an object from the node's local memory, from another remote node within the social group and from nodes in another social group, respectively; $t_l < t_r < t_s$. These costs are assumed to be the same for all nodes in order to simplify the analysis. The most problematic assumption is that the cost of accessing an object from any group node is the same irrespective of the involved nodes, *i.e.*, $t_l < t_r^{ij} = t_r < t_s, \forall i, j \in N$. However it may be reasonable when the group draws on locality or related types of social context.

If the nodes of the group are in proximity the access cost could either represent the additional latency incurred when fetching content or the bandwidth consumed when retrieving content, depending on the scenario of interest. Otherwise, the low-medium cost associated with fetching an object from within the group may reflect low price due to high level of trust.

Given an object placement $\mathcal{P}$, the mean access cost incurred to node n per unit time for accessing its requested objects is given by:

$$\mathcal{C}_n(\mathcal{P}) = \sum_{m \in P_n} F_m^n t_l + \sum_{\substack{m \notin P_n, \\ m \in \mathcal{P}_{-n}}} F_m^n t_r + \sum_{\substack{m \notin P_n, \\ m \notin \mathcal{P}_{-n}}} F_m^n t_s \,. \quad (2.1)$$

The first, second and third addends on the right hand side correspond to the mean cost for accessing objects locally, from other nodes of the group and from external sites if not found within the group, respectively. The object placement strategies considered in this paper are described next.

Optimally altruistic strategy: the objects are stored in such a way that the total access cost for all nodes in the social group is minimized (*i.e.*, , minimize $\sum_{n=1}^{N} \mathcal{C}_n(\mathcal{P})$). This problem can be transformed into a 0-1 integer programming problem.

Let $X_m^n = \begin{cases} 1, & \text{if } m \in P_n; \\ 0, & \text{otherwise} \end{cases}$ and

$Y_m^n = \begin{cases} 1, & \text{if } m \notin P_n \text{ and } m \in \mathcal{P}_{-n}; \\ 0, & \text{otherwise.} \end{cases}$

The objective is to minimize the total access cost:

$$\sum_{n=1}^{N} \sum_{m=1}^{M} [X_m^n F_m^n t_l + Y_m^n F_m^n t_r + \prod_{j=1}^{N} (1 - X_m^j) F_m^n t_s], \quad (2.2)$$

where

$$Y_m^n = (1 - X_m^n)(1 - \prod_{\substack{j=1 \\ j \neq n}}^{N} (1 - X_m^j)). \quad (2.3)$$

This is a special case of a quadratic programming problem with zero diagonal elements, whose solution is very difficult [22]. It was shown in [20] that this quadratic problem *reduces to* a 0-1 integer linear minimization problem (ILP) with objective function

$$f(X) = \sum_{n=1}^{N+1} \sum_{m=1}^{M} z_m^n X_m^n, \quad (2.4)$$

subject to $\sum_{n=1}^{N+1} X_m^n \geq 1, \ 1 \leq m \leq M$ and $\sum_{m=1}^{M} X_m^n \leq C_n, \ 1 \leq n \leq N$. In (2.4), the terms X_m^n are as above, the additional virtual node $N+1$ represents the ensemble of nodes in other social groups, and the terms z_m^n, $n \in \mathcal{N}$, are defined as $z_m^n = \begin{cases} F_m^n (t_r - t_l), & \text{for } 1 \leq n \leq N; \\ \sum_{j=1}^{N} F_m^j (t_r - t_s), & \text{for } n = N+1, \end{cases}$

In this ILP formulation, there is effectively an implicit reference placement, whereby all nodes can access all objects from the caches of group nodes and aggregate access cost $\sum_{n=1}^{N} \sum_{m=1}^{M} F_m^n t_r$. The aim is then to derive object placements that improve over this reference placement. Hence, the terms z_m^n, $n \in \mathcal{N}$, express the incremental benefit resulting for each node when it stores the object locally instead of retrieving it from the group; whereas, z_m^{N+1} notes the loss all nodes incur if the object is not stored anywhere in the group. The equivalence of the quadratic maximization problem (2.2) to the minimization ILP (2.4) is further explicated in [14].

Selfish strategy: Under the selfish (or Greedy Local, as referred to in [19]) strategy, the nodes store the objects *they* prefer most. Each node n ranks the objects in decreasing order of preference and selects to store the first C_n ones.

Self-aware cooperative strategy: The placement strategy evolves in two rounds. First, each node stores its C_n most preferable items (as with the selfish placement strategy). Then, nodes take turns in adjusting their placements based on the placements of the other nodes in the group. During this second round, each node has the chance to replace some or all its items in order to come up with the most cost-effective placement given the placements of other nodes. Thus, a node may decide to evict an object stored in some other node in the group and insert a new one, if this reduces its access cost according to (2.1). As the nodes amend their placements sequentially, each replacement made by one node affects both the access cost of nodes that have already made their adjustments and the choices made by nodes that follow.

In Table 1 we summarize results for the implementation cost (over all nodes) for the three strategies, considering the *computational complexity* cost. The details of the derivation are provided in [14] and are also available in [19]. The computational complexity refers to the cost for all nodes to decide which objects to store locally.

Table 1: Computational cost

Strategy	Computational
Selfish	$O(NMlogM)$
Self-aware cooperative	$O(NMlogM)$
Optimally altruistic	NP-hard

2.1 Properties of the self-aware cooperative strategy

The self-aware cooperative strategy is a distributed local-search algorithm for solving the distributed content placement game [19]. This is an N-player noncooperative game, whereby the players (nodes) behave as rational selfish agents that aim to minimize their aggregate content access cost. Each node is called to select its own pure strategy(placement) P_n among $\binom{M}{C_n}$ alternatives and the aggregate placement $\mathcal{P}$ resulting from the combination of the nodes' choices presents each node with a payoff $\mathcal{C}_n(\mathcal{P})$, given by (2.1).

The algorithm evolves in two rounds. During the first round, the nodes move asynchronously without sharing any information; whereas, in the second round, the nodes play sequentially following a predefined order. By the time the node playing in the k^{th} position has to determine its own placement, it knows everything that the node who played before it knew, *i.e.*, the placements of the first $k-2$ nodes, plus the placement of the node in position $k-1$. Hence, this second round of the algorithm points to an N-player single-act dynamic game and, irrespective of the particular order of play, can be represented in ladder-nested extensive form [2][1].

In [19], it has been shown that under the self-aware cooperative placement strategy nodes may evict from their caches only objects that are replicated elsewhere in the group (*property P1*); nodes may insert in their caches only objects that are not represented in the group (*property P2*); and the placements do not give rise to node mistreatment phenomena, *i.e.*, for any node n, it holds that $\mathcal{C}_n^C(\mathcal{P}) \leq \mathcal{C}_n^S(\mathcal{P})$, where $\mathcal{C}_n^C(\mathcal{P})$ and $\mathcal{C}_n^S(\mathcal{P})$ denote the mean access cost for node n under the self-aware cooperative and selfish strategy, respectively (*property P3*) (for a discussion of these properties when the within-group access costs are not symmetrical, refer to [14]). More importantly, it has been shown that, irrespective of the order of play, the resulting global placement after all nodes make their amendments in the second round, is a Nash Equilibrium (NE) in that no node has a reason to unilaterally change its own placement. In other words, the self-aware cooperative placement always yields placements that are NE in pure strategies. What changes with the order of play is the resulting placement $\mathcal{P} = \{P_1, P_2, \ldots, P_N\}$, hence the individual and social costs corresponding to the NE. In general, the number of different NE ranges in $[1, D]$, where $D = min(N!, \binom{M}{C}^N)$; the first number reflects the possible permutations in the order nodes play and the second determines all the possible ways the M content objects can be replicated in the caches of N nodes.

The question then becomes how do the placements under the self-aware cooperative strategy compare with the optimal and, secondarily, the selfish ones. From a distributed algorithm viewpoint, the search is for an (ideally, constant and tight) approximation ratio; in game-theoretic terms, we are after the price-of-anarchy (PoA) of the game, *i.e.*, the ratio of the social cost under the worst possible NE over the optimal cost [16]. We elaborate on this later in Section 5.

3 Group tightness metric

Each one of the three placement strategies presented in Section 2 resolves differently the multiple tradeoff among: a) the performance of individual nodes and of the entire group; b) the possibility of individual nodes being mistreated and the respective (lack of) incentives for cooperation; c) the required communication overhead and computational complexity for each strategy realization. The obvious question then for a node-member of the social group is which strategy is the most "appropriate" to follow. In this section, we introduce a metric, which we call *tightness*, for the similarity of interests within the social group that can be of help in reaching a conclusion.

The definition of *tightness* draws on the symmetrized Kullback-Leibler (K-L) divergence [17], a well-known measure of divergence between two distributions. The Kullback-Leibler divergence of distribution Q from S is defined as:

$$D_{S,Q} = \sum_i S(i) log \frac{S(i)}{Q(i)}.$$

and its symmetrized counterpart is $D(S||Q) = D_{S,Q} + D_{Q,S}$.

The average divergence of nodes' preferences within the group can then be written as:

$$\hat{D}_F = \frac{\sum_{(i,j)} D(F^i||F^j)}{N(N-1)/2}, \tag{3.1}$$

where the summation above is carried out over all $N(N-1)/2$ node pairs (i,j). Finally, we define *tightness* T to be the inverse of $\hat{D}_F$:

$$T = \frac{1}{\hat{D}_F}. \tag{3.2}$$

We elaborate on computational aspects of the tightness metric in [14].

3.1 Why tightness as a metric?

In principle, various measures of distributional similarity could quantify the similarity of content preferences across the nodes of a group, such as the Spearman's rank correlation coefficient [24], Kolmogorov-Smirnov distance [31], proportional similarity [30], and total variation distance [10]. Compared to them, the proposed tightness metric has the following advantages:

Sensitivity to rank-preserving dissimilarity: Contrary to metrics such as Spearman's rank correlation coefficient, tightness can capture dissimilarity of interests among nodes that may rank the objects similarly, yet focus with different intensity on the top-k content objects (see rank-preserving dissimilarity case in Section 4).

Account of full preferences' profiles: Contrary to the the Kolmogorov-Smirnov (K-S) distance metric, which considers the supremum of the differences over all elements of a distribution, the K-L divergence accounts for deviations across the whole distribution. Thus, the proposed tightness metric captures more accurately the overall distributional (dis)similarity.

Broader range of values: In contrast with the proportional similarity and total variation distance metrics [30], which yield values in $[0, 1]$, tightness values vary in $(0, +\infty)$. Therefore it can resolve easier finer levels of

1. The game has also connections with leader-follower (Stackelberg) games [2]. However, the nodes do not play in Stackelberg mode, *i.e.*, assuming a priori that each subsequent node will play best-response to the future states of the game (node placements); they rather play themselves best-response to the *current* state of the game.

distributional divergence. In recent work [13], we have shown how this property of the tightness metric can benefit a different task, that of community detection, by modulating its resolution.

Finally, we should note that when some element values of one of the distributions are zero while the corresponding elements of the other distribution are not (*i.e.*, the request rate of a node for an object is zero), the K-L distance value approaches infinity. In order to avoid such problems, smoothing methods such as interpolation and backing-off schemes can be used for providing reliable probability estimates. These methods have been studied in statistical language modelling in order to estimate the distribution of natural language elements as accurately as possible. In our case, non-zero request rates for objects can be discounted with different discounting methods (see [23]), whereas all other non-requested objects can be given a minimal ϵ probability. In this paper, we will consider that all nodes have probability mass (*i.e.*, positive request rate) for all content objects, so that we do not need to apply any smoothing method.

4 Evaluation methodology and scenarios

Tightness expresses the similarity of preferences among the nodes of the social group and is always greater than or equal to zero. In fact, $T \to \infty$ when the group nodes have very similar preferences and $T \to 0$ when they have very diverse preferences. As T is an average metric over all node pairs of a social group, it is clear that a given value of T may arise under different combinations of node-level content preference distributions.

The content preferences of nodes are modeled by Zipf distributions with variable shape parameter s, *i.e.*, the normalized interest of node n for its k^{th} most interesting object is $(1/k)^s / \sum_{l=1}^{M} 1/l^s$. Zipf distributions combine modeling simplicity with flexibility in that proper manipulation of their shape parameter s, gives rise to a wide set of distributions ranging from uniform ($s = 0$) to the highly skewed ones with power-law characteristics ($s >> 0$)[2].

In order to draw more insightful conclusions in the current study, we distinguish between the following two broad patterns of dissimilarity in the preference distributions.

4.1 Rank-preserving dissimilarity

The rank of the objects remains the same for all group member nodes, *i.e.*, the i^{th} most popular object for all nodes is the same, $i \in [1, M]$. However, the mass of the distributions concentrates more towards the highly-ranked objects as the shape parameter s increases.

More specifically, the preferences F_m^n for the object m, $m \in [1, M]$, are drawn from Zipf distributions with different exponent s_n for each node. We let $s_1 = 0$ for the first node (uniform interest distribution) and $s_n = p(n-1)$ for node n, $n \in [2, N]$, where $p \in \mathbf{R}$ is the increment parameter. As shown in Table 2(a), under the (object-)rank-reserving dissimilarity scenario, *tightness* is a monotonically decreasing function of p. As p increases, the content preference distributions of nodes diverge more strongly, resulting in higher pairwise K-L divergence values between any two node distributions.

4.2 Shape-preserving dissimilarity

The preference distributions are identical in shape for all nodes, yet the ranking of a given object differs from node to node, *i.e.*, the $k^{th}, 1 \leq k \leq M$, most popular object for each node is different. The dissimilarity of nodes can be more dramatic in this case and lower tightness values are expected on average.

Contrary to the rank-preserving dissimilarity scenario, the request rates F_m^n are drawn from a Zipf distribution with the same exponent s for all nodes. The object preference rank for first node is $[1, 2..., M]$ and is shifted to the right by $k(n-1)$ positions for node n, $n = 1, \ldots, N$, where $k \in [0, M-1]$ is the *shift parameter*. For example, the most preferable object for node n is the one with index $u = mod(k(n-1), M)$ and its object preference rank is $[u, u+1, ..., M, 1, ...u-1]$. Table 2(b) lists the values of *tightness* for various values of k. Notably, *tightness* is a monotonically decreasing function of the shift parameter k for k values satisfying $(N-1)k + C < M$. This inequality is satisfied throughout our performance evaluation of the placement strategies. As k increases in this interval, the divergence in the content preferences between any two nodes increases. Likewise, tightness is monotonically decreasing in s; only the manipulation of s yields larger changes in the tightness values. The two parameters allow high flexibility in tuning the tightness value, whereby s provides for larger change steps and k the finer control of the value.

The two dissimilarity patterns reflect groups with very different breadth in their members' preferences. The rank-preserving dissimilarity pattern points to groups with quite similar and focused interests. This could be the result of a membership in some thematic-oriented online association. It could also be due to trusting the same popular websites for getting informed about content of interest (*e.g.*, music or sport). In social networks and blogs most popular users tend to influence the majority of passive users, effectively amortizing variations in interests and preferences [6]. On the other hand, shape-preserving similarity loosely points to groups, whose members' interests are spread over a wider range of content, with the relative intensities being about the same. For example, someone who likes listening to music in her leisure time might download music-related content with the same normalized intensity as

2. Zipf distributions have also been shown to be good models of content popularity both within and across different Internet Autonomous Systems [12], and have been used widely in the literature in this respect.

Table 2: Example tightness values when $M = 50$ and $N = 5$

(a) rank-preserving		(b) shape-preserving s=1		s=0.5		s=0.1	
p	T	k	T	k	T	k	T
0.0	∞	0	∞	0	∞	0	∞
0.05	45.74	1	0.37	1	2.24	1	83.42
0.1	10.17	2	0.27	2	1.49	2	52.09
0.2	2.09	3	0.23	3	1.22	3	40.93
0.4	0.46	4	0.21	4	1.07	4	35.15
0.6	0.24	5	0.20	5	0.98	5	31.65
0.8	0.17	6	0.19	6	0.93	6	29.38
1.0	0.14	7	0.18	7	0.89	7	27.86

someone interested in sports would download sports-related content.

Nevertheless, these two patterns have been chosen primarily because they let us *control* the level of preferences' dissimilarity in our experimentation rather than thanks to their representative power. The parameters p, for the rank-preserving dissimilarity, and $\{k,s\}$, for the shape-preserving dissimilarity, serve as tuning knobs with predictable effect. Tuning these parameters, we can synthesize a broad range of possible dissimilarity patterns across the social group. Last but not least, these special similarity patterns let us gain further insights as to how well the self-aware cooperative placement strategy approximates the optimal one in special cases. We exercise this flexibility in the analysis that follows in Section 5 and the numerical evaluation in Section 6; later, in Section 7, we experiment with real-world content preference profiles drawn from an online social bookmarking application.

5 Placement cost under the selfish and self-aware cooperative strategies

The *aggregate* content access cost under the altruistic strategy can be numerically computed, at least for small values of M, N, by solving the ILP problem (2.4) in Section 2. Herein, we derive analytical expressions for the *per-node* access cost under the selfish and self-aware cooperative content placement strategies for the two content preference-dissimilarity patterns described in Section 4. Apparently, the costs under the altruistic and selfish strategies constitute lower and upper bounds, respectively, for the overall cost of the self-aware cooperative strategy.

In our analysis, the group nodes are indexed in order of increasing Zipf distribution exponent s (for the rank-preserving dissimilarity) and distribution-shift k (for the shape-preserving dissimilarity). For content items, on the other hand, two kinds of item indexing become relevant: the "global" one, enumerating all objects in decreasing preference order of node 1; and, the local node-specific ones, which index objects in decreasing preference order of the respective nodes. The two types of indexing coincide under rank-preserving dissimilarity; whereas there are N different local indexings, one per node, under shape-preserving dissimilarity. The indexing type of relevance in each case should be apparent from the context.

5.1 Selfish placements

When the nodes behave selfishly, it is possible to analytically compute the amount of content they access from the three levels of data storage, *i.e.*, locally C_l, remotely from the caches of the group member nodes C_r, and externally from server(s) or nodes in other social groups C_s, as well as the resulting access costs. We treat separately the two generic dissimilarity patterns described in Section 4.

Rank-preserving dissimilarity, $p \neq 0$: All nodes store locally the *same* C content items that commonly rank top at their preferences and access the remaining $M - C$ items from external sources. What changes with p is the preference amount that is concentrated in the C items each node stores, which is controlled by the exponent $s_n = p\cdot(n-1)$ in the Zipf content preference distribution. Therefore, $C_l = C$, $C_r = 0$, and $C_s = M - C$ for all nodes and the overall access cost for node n, is given by

$$\begin{aligned}\mathcal{C}_n^S(\mathcal{P}) &= \frac{\sum_{j=1}^{C} j^{-s_n}}{\sum_{j=1}^{M} j^{-s_n}} t_l + \frac{\sum_{j=C+1}^{M} j^{-s_n}}{\sum_{j=1}^{M} j^{-s_n}} t_s \qquad (5.1.1)\\ &= \frac{\sum_{j=1}^{C} j^{-p\cdot(n-1)}}{\sum_{j=1}^{M} j^{-p\cdot(n-1)}} t_l + \frac{\sum_{j=C+1}^{M} j^{-p\cdot(n-1)}}{\sum_{j=1}^{M} j^{-p\cdot(n-1)}} t_s.\end{aligned}$$

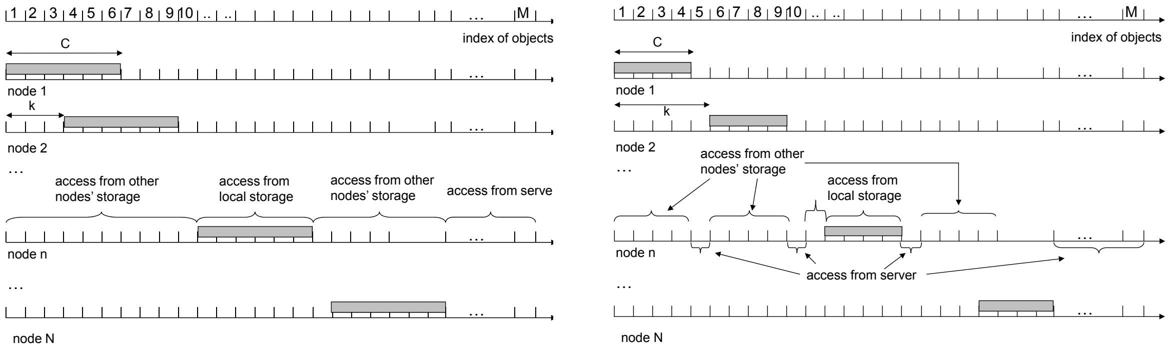
Shape-preserving dissimilarity, $k \neq 0$: Now, the C items each node stores locally are, generally, different than those other nodes store. Assuming that the number of content items exceeds the cumulative group storage capacity, $M \gg N\cdot C$, we can distinguish two possibilities:

a) $k < C$: There is partial overlapping in the preferences of two (or more) nodes (Fig. 1(a)). Each node accesses $C_r = (N-1)k$ items from the storage of the other group nodes and $C_s = M - C - (N-1)k$ items from the server(s). As the shift parameter k increases, the nodes of the social group cumulatively store and can access from each others' storage more content. Yet, nodes with higher index n have to access more items ranking higher at their preferences from external sources since they are not stored by any other group member. Worst of all, the node N has to fetch the content items ranking at positions $[(C+1), (M-(N-1)k)]$ in its own preference distribution at cost t_s, whereas he can get access to content objects in positions $[M-(N-1)k+1, M]$ through the storage of the other group nodes.

The content access cost for node n can be written

$$\begin{aligned}\mathcal{C}_n^S(\mathcal{P}) &= \frac{\sum_{j=1}^{C} j^{-s}}{\sum_{j=1}^{M} j^{-s}} t_l + \frac{\sum_{j=C+(N-n)k+1}^{M-(n-1)k} j^{-s}}{\sum_{j=1}^{M} j^{-s}} t_s \qquad (5.1.2)\\ &+ \left[\frac{\sum_{j=C+1}^{C+(N-n)k} j^{-s}}{\sum_{j=1}^{M} j^{-s}} + \frac{\sum_{j=M-(n-1)k+1}^{M} j^{-s}}{\sum_{j=1}^{M} j^{-s}}\right] t_r.\end{aligned}$$

b) $k > C$: There is no overlapping in the items each node stores locally in its cache (Fig. 1(b)) so that $N \cdot C$ different content items are stored within the group. The rest of the $C_s = M - N\cdot C$ objects have to be fetched from external sources. As long as $N \cdot k + C \leq M$, the access cost increases with higher k and node index n values.



(a) $k < C$, partial overlapping in local storage

(b) $k \geq C$, each nodes stores different items

Figure 1: Objects stored at different group nodes under shape-preserving dissimilarity.

$$\mathcal{C}_n^S(\mathcal{P}) = \frac{\sum_{j=1}^{C} j^{-s}}{\sum_{j=1}^{M} j^{-s}} t_l + \left[\sum_{i=1}^{N-n} \frac{\sum_{j=i(C+k)+1}^{i(C+k)+C} i^{-s}}{\sum_{j=1}^{M} j^{-s}} + \sum_{i=1}^{n-1} \frac{\sum_{j=M-ik-(i-1)C+1}^{M-i(k+C)+1} j^{-s}}{\sum_{j=1}^{M} j^{-s}} \right] t_r$$
$$+ \left[\sum_{i=1}^{N-n} \frac{\sum_{j=iC+(i-1)k+1}^{i(C+k)} j^{-s}}{\sum_{j=1}^{M} j^{-s}} + \sum_{i=1}^{n-1} \frac{\sum_{j=M-(i-1)(k+C)}^{M-ik-(i-1)C} j^{-s}}{\sum_{j=1}^{M} j^{-s}} + \frac{\sum_{(N-n+1)(C+k)+1}^{M-(n-1)(C+k)} j^{-s}}{\sum_{j=1}^{M} j^{-s}} \right] t_s. \quad (5.1.3)$$

The resulting content access cost for node n is given by (5.1.3).

Identical uniform content preferences: This represents an extreme case of zero dissimilarity in the preferences of group nodes. It results from the general rank-preserving dissimilarity scenarios when $p = 0$, and from the shape-preserving dissimilarity scenarios, when $s = 0$.

Contrary to the general dissimilarity scenarios, the distribution of content objects at the three storage levels and the corresponding access cost are no longer deterministic. The C objects stored locally at each node are randomly chosen out of the full set of M objects so that the total number of *different* objects collectively stored at the caches of the N group nodes is a random variable X, $C \leq X \leq M$. Each node now accesses C objects from its own cache, $X - C$ objects from the caches of the other N-1 group nodes and the remaining $M - X$ content objects from the server and the *expected* per-node content access cost is given by

$$\mathcal{C}_n^S(\mathcal{P}) = \frac{1}{M}[C \cdot t_l + (E[X] - C) \cdot t_r + (M - E[X]) \cdot t_s] \quad (5.1.4)$$

Whereas the probability distribution of X is more involved and given in [14], to compute the expected value $E[X]$, it suffices to remark that the selection or not of each object by a single node is a Bernoulli trial with success probability $p_s = \binom{M-1}{C-1} / \binom{M}{C} = C/M$. Therefore, the selection of each object by *at least* one out of the N nodes equals $1 - (1 - p_s)^N$ and

$$E[X] = M(1 - (1 - C/M)^N) \quad (5.1.5)$$

5.2 Self-aware cooperative placements

The starting point for the self-aware cooperative placements are the selfish placements of the first step. In the second step of the strategy, nodes take turn in adjusting the selfish placements of the first step evicting objects that are replicated elsewhere in the group and inserting new ones, not yet stored anywhere in the group, inline with the properties (P1)-(P3) listed in Section 2.1 and proven in [19]. In general, by the end of the second step, each node n has retained the first j_n objects of the initial selfish placement and inserted $C - j_n$ new ones, which are not replicated anywhere else in the group, according to property (P2).

5.2.1 Rank-preserving dissimilarity

By the end of the first step, the selfish strategy gives rise to full replication of the same C objects in the group so that candidates for insertion are objects with global indices in $[C + 1, M]$. Let s_i be the exponent in the Zipf preference distribution for node i and consider the first node just before executing the second step (placement adjustment). Depending on its distribution skewness, node 1 will retain j_1 objects in its cache, $1 \leq j_1 \leq C$, and remove the rest, where

$$\begin{aligned} j_1 &= min(max\{u : \frac{t_r - t_l}{u^{s_1}} > \frac{t_s - t_l}{(2C - u + 1)^{s_1}}\}, C) \\ &= min(\lfloor \frac{2C + 1}{1 + (\frac{t_s - t_l}{t_r - t_l})^{1/s_1}} \rfloor, C) \end{aligned} \quad (5.2.1)$$

namely, j_1 equals the maximum item index of those stored locally in the first step, for which the benefit of retaining it in the local storage exceeds the insertion benefit from the next most preferred item among those not yet stored elsewhere in the group (Fig. 2).

Since nodes only insert non-represented objects in this step (P2), candidates for insertion by the second node will be the objects $[2C-j_1+1, 3C-j_1]$, and generalizing, by the n^{th} node, the objects $[nC-\sum_{l=1}^{n-1} j_l+1, (n+1)C-\sum_{l=1}^{n-1} j_l]$. The number of items retained locally by node n is

$$\begin{aligned} j_n &= min(max\{u : \frac{t_r - t_l}{u^{s_n}} \\ &> \frac{t_s - t_l}{[(n+1)C - \sum_{i=1}^{n-1} j_i - u + 1]^{s_n}}\}, C) \\ &= min(\lfloor \frac{(n+1)C - \sum_{i=1}^{n-1} j_i + 1}{1 + (\frac{t_s - t_l}{t_r - t_l})^{1/s_n}} \rfloor, C) \quad (5.2.2) \end{aligned}$$

Moreover, the index of the last inserted object in the group, after all N nodes have finished with their second step, is $in_{max} = NC - \sum_{i=1}^{N-1} j_i$. If we denote $j_{min} = min_i\{j_i\}$, we can state the following property for the structure of the content placement at the caches of the group nodes.

Property 5.1. *Under the self-aware cooperative placement strategy and rank-preserving dissimilarity across nodes' preferences, the number of replicas within the group is: N for objects with (global) indices in $[1, j_{min}]$ (full replication); $r, 1 \le r < N$, for objects with indices in $[j_{min}+1, C]$; one for objects with indices in $[C+1, in_{max}]$; and, zero for objects with indices in $[in_{max}+1, M]$.*

Drawing on this property, we can derive conditions for two particular instances of the placements.

Proposition 5.1. *The original selfish placements of all nodes remain intact during the second step when*

$$(\frac{t_s - t_l}{t_r - t_l})^{1/s_{min}} < 1 + 1/C \quad (5.2.3)$$

where $s_{min} = min_j\{s_j\}$.

Proof: As can be seen in (5.2.4) and (5.2.2), the number of objects retained by node n is a monotonically increasing function of s. In the same time, it depends on its order of play, *i.e.*, which nodes have preceded him in adjusting their placements. For two nodes k and l, with k playing before l and $s_k \le s_l$, (5.2.2) suggests that

$$j_k = \frac{(k+1)C - \sum_{i=1}^{k-1} j_i + 1}{1 + (\frac{t_s - t_l}{t_r - t_l})^{1/s_k}} \ge \frac{(l+1)C - \sum_{i=1}^{l-1} j_i + 1}{1 + (\frac{t_s - t_l}{t_r - t_l})^{1/s_l}} = j_l$$

The self-aware cooperative placements will coincide with the selfish ones only if $j_n = C, \forall n \in N$. The necessary and sufficient condition for this is that $j_{min} = C$, or equivalently that the node with the least skewed distribution of preferences does not evict any item given that nodes playing before it have not done so either. Hence, it must hold that

$$min(\lfloor \frac{2C+1}{1 + (\frac{t_s - t_l}{t_r - t_l})^{1/s_{min}}} \rfloor, C) = C \quad (5.2.4)$$

which directly yields (5.2.3). □

Likewise, we can derive the condition for inserting $(N-1)C$ items during the second step so that eventually NC different items be stored within the group.

Proposition 5.2. *Under the self-aware cooperative placement strategy and rank-preserving dissimilarity, $N \cdot C$ different content objects will be stored in the caches of the group nodes, each represented only once, if*

$$s_{max} < \frac{log(\frac{t_s - t_l}{t_r - t_l})}{log(NC)} \quad (5.2.5)$$

Proof: To have all objects only once replicated within the group, nodes [1,N-1] should replace *all* C items of their original selfish placements with non-represented ones. Therefore, objects $[nC+1, (n+1)C], 1 \le n \le (N-2)$ should be inserted at node n, whereas node N will always retain its original selfish placement since these most preferred objects will not be replicated anywhere else in the group(P1).

The worst-case setting is that the node playing in position $N-1$ is the node with the most skewed distribution, s_{max}. Therefore, for a placement with NC different items in the group caches, irrespective of the order of play, it suffices that the object NC replaces the first most preferred object of node $N-1$,

$$\frac{t_s - t_l}{(NC)^{s_{max}}} > \frac{t_r - t_l}{1^{s_{max}}}. \quad (5.2.6)$$

Solving for s_{max} yields (5.2.5). □

With these results at hand, we can compute the per-node access cost as

$$\begin{aligned} \mathcal{C}_n^C(\mathcal{P}) &= \frac{\sum_{i=1}^{j_n} i^{-s_n}}{\sum_{i=1}^{M} i^{-s_n}} t_l + \frac{\sum_{i=in_{max}}^{M} i^{-s_n}}{\sum_{i=1}^{M} i^{-s_n}} t_s \quad (5.2.7) \\ &+ \frac{\sum_{i=j_n+1}^{in_{max}} i^{-s_n} - \sum_{i=nC-\sum_{l=1}^{n-1} j_l+1}^{(n+1)C-\sum_{l=1}^{n} j_l} i^{-s_n}}{\sum_{i=1}^{M} i^{-s_n}} t_r \end{aligned}$$

whereby the objects that a node eventually accesses from the other group nodes' caches are the full set of non-represented objects that are inserted at the second step of the strategy(algorithm) *minus* those locally stored at that node.

5.2.2 Shape-preserving dissimilarity

As with selfish placements, we need to distinguish between two cases:

a) $k > C$: assuming that $N(C+k) \le M$, the selfish placements in the first step result in the placement of NC different objects in the caches of the group nodes, each one represented only once. According to (P1), nodes do not evict objects from their caches; hence, the placements under the self-aware cooperative placement coincide with those under the selfish strategy.

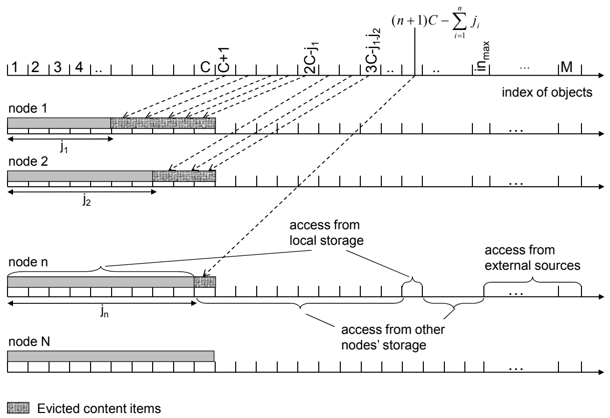


Figure 2: Self-aware cooperative strategy, rank-preserving dissimilarity: evicted and inserted content items per node.

Proposition 5.3. *Under shape-preserving dissimilarity and $k > C$ the self-aware cooperative placement coincides with the optimal.*

Proof: For $k > C$, not only the self-aware cooperative but also the optimal placement coincides with the selfish one. Since the selfish placement comprises NC discrete content objects, the optimal placement could, in principle, differentiate in three ways: a) replicate one of the content objects more than once; b) mutually exchange two content objects stored in different nodes; c) replace one of the objects in the selfish placement with one of those not represented in the placement, *i.e.*, objects with indices in $[nC + 1, (n + 1)C - k]$. It is straightforward to see that the global utility generated in each one of three cases is lower than that accruing through the selfish placement. Therefore, the two placements coincide. □

b) $k < C$: the selfish placements give rise to overlaps in the contents of nodes' caches. The number of different objects that are placed in the whole group are $m_d = (n-1)k+C$ and their replication count varies in $[1, \lfloor \frac{C}{k} \rfloor]$. Moreover, for $k < \lfloor C/2 \rfloor$, the $m_d - 2k$ objects stored by the group nodes feature at least two replicas and could be evicted by one or more group nodes in the second step.

Whereas, the exact computation of the per-node access cost in this case is cumbersome, it is easier to prove the following result.

Proposition 5.4. *The placements under the self-aware cooperative strategy and shape-preserving dissimilarity across nodes' preferences coincide with the selfish placements when*

$$k/C > (\frac{t_s - t_l}{t_r - t_l})^{1/s} - (1 + 1/C) \qquad (5.2.8)$$

Proof: The two placements will coincide as long as no object evictions/insertions are made during the second step. Given that the local indices of objects $[m_d + 1, M]$, hence their preference rank, increases for higher node indices, two conditions should be met so that nodes' original placements do not change.

- the $(N-1)^{th}$ node should not (find it profitable to) evict its least preferable object currently in its cache.
- the last node should not (find it profitable to) evict its least preferable object that is replicated in the group.

The first condition translates to

$$\frac{t_r - t_l}{C^s} > \frac{t_s - t_l}{(C + k + 1)^s} \Rightarrow \frac{C + k + 1}{C} > (\frac{t_s - t_l}{t_r - t_l})^{1/s}$$

whereas the second condition can be expressed as

$$\frac{t_r - t_l}{(C - k)^s} > \frac{t_s - t_l}{(C + 1)^s} \Rightarrow \frac{C + 1}{C - k} > (\frac{t_s - t_l}{t_r - t_l})^{1/s} \qquad (5.2.9)$$

since $(C + 1)/(C - k) > (C + k + 1)/C, \ \forall k > 0$, the first inequality is the active constraint and (5.2.8) results trivially. □

Therefore, equations (5.2.3) and (5.2.8) already suggest that the placements emerging under the self-aware cooperative and selfish strategies tend to coincide as the exponents of the Zipf preference distributions ($s_1 \propto p$ for rank-preserving dissimilarity) and shift parameter k increase. In other words, since tightness decreases with s and k (see Table 2), the gain under cooperation fades out as the content preferences of nodes diverge. We elaborate on this result in the next section.

Identical uniform content preferences: In this extreme case of demand distributions, the self-aware cooperative strategy will generate a placement of NC different content objects, each one represented only once in the union of the group's caches.

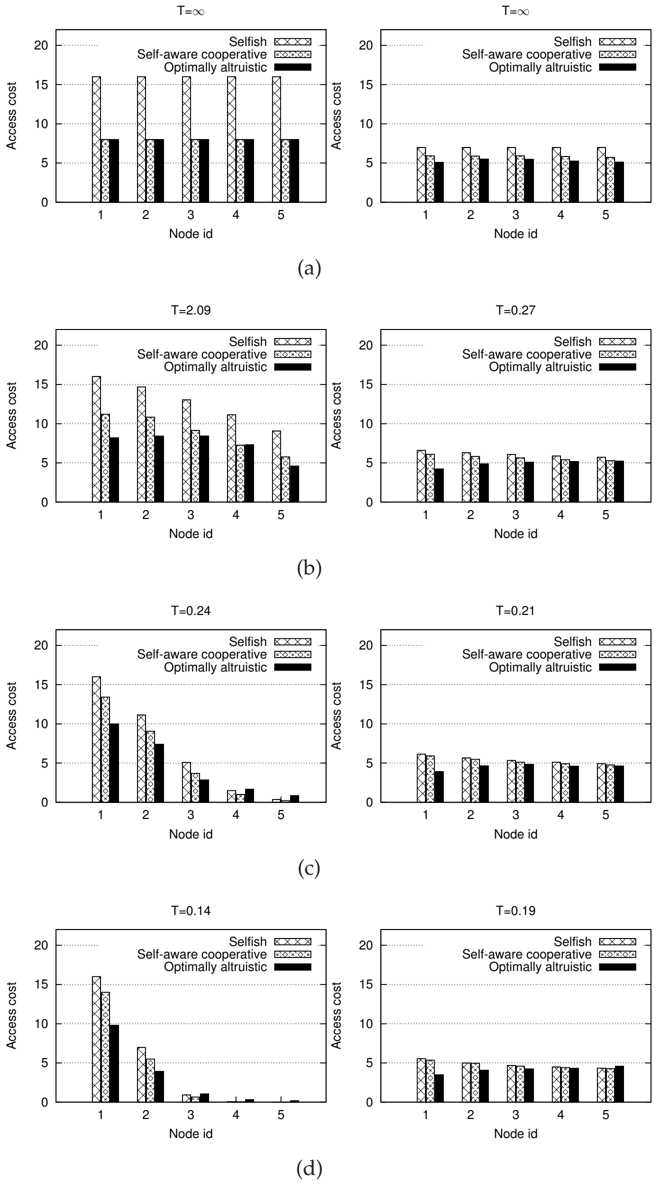
Proposition 5.5. *The cost of the self-aware cooperative placement under identical uniform content preferences equals the optimal one.*

Proof: Irrespective of the order of play, nodes evict objects that are elsewhere represented in the group and replace them with equally wanted objects that were not selected by any group node in the first round of selfish placements. There are $\prod_{i=0}^{N-1} \binom{M-iC}{C} = \frac{M!}{C!^N (M-NC)!}$ different possible placements with the same social cost, which coincides with the best possible. □

6 Results and Discussion

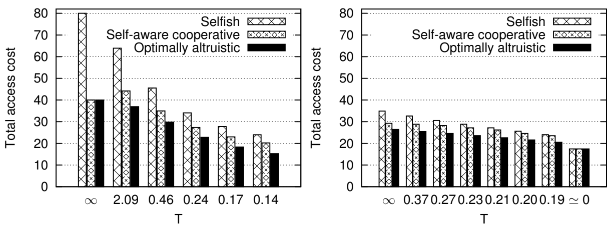
The numerical examples in this section illustrate how group similarity, aka *tightness*, shapes the tradeoffs induced by the three behavior-based content placement strategies. Therefore, they help establish guidelines as to which behavior (strategy) would be beneficial to individual nodes and/or the entire group, under given similarity levels in the preferences of the nodes in the social group.

In the numerical examples in this paper initially $N = 5$ nodes; a small number of nodes helps us better illustrate and discuss the results regarding the content access cost for each node. The default value for node storage capacity is $C = 10$ objects, for object population $M = 50$ objects and for the costs $t_l = 0$, $t_r = 10$ and $t_s = 20$ cost units.



(a) (b) (c) (d)

Figure 3: Individual access cost under the three content placement strategies for different values of *tightness* T, under rank-preserving (figures on the left) and shape-preserving (figures on the right) dissimilarity.



(a) Rank-preserving (b) Shape-preserving

Figure 4: Total access cost *vs.* *tightness* T.

6.1 Placement strategy comparison

Figures 3 and 4 consider separately the two scenarios for preferences' dissimilarity in Section 4. They plot the per-node and aggregate (*i.e.*, for the entire group) access cost under the three content placement strategies and for different values of *tightness*, T. We discuss these two viewpoints for very high and very low tightness values.

6.1.1 *Social groups with infinite or very high* tightness

The first important remark out of the two plots is that the optimally altruistic strategy outperforms the other two regarding not only the cost for the entire group (by definition), but also the cost for individual group nodes (Fig. 3(a)). This relation holds irrespectively of the dissimilarity scenario in question and suggests that the optimally altruistic behavior is the clear winner-behavior for every node in a very tight social group.

The second noteworthy outcome is that the access costs for both individual nodes and the entire group under the self-aware cooperative strategy are very close to (slightly higher than) those under the optimally altruistic strategy. Since its implementation is significantly simpler than the optimal altruistic one, it emerges as the favorite alternative for node-members of tight social groups, achieving a good tradeoff between performance and complexity.

Looking closer into the rank-preserving dissimilarity plots, when tightness approaches infinity (Fig. 3(a) on the left), the access cost for all group nodes under the self-aware cooperative and optimally altruistic strategies tend to become equal. Infinite tightness in the general case ($p = 0$, $s \neq 0$ in Section 4) implies that a given object is requested with the same intensity by all nodes. Both the self-aware cooperative (in line with Proposition 2) and the optimally altruistic strategies end up with the same placements $\mathcal{P}$, which insert many items and spread them across the storage capacity of the group nodes; contrary to the selfish placements, which blindly replicate the same C objects N times.

Summarizing, the tighter the social group, the more incentives the nodes have to behave in cooperative or, even, altruistic rather than selfish manner.

6.1.2 *Social groups with low* tightness

The first conclusion out of Fig. 3 concerns the way the total access cost is spread across the group nodes, regardless of the placement strategy, under the two types of dissimilarity and for low T values. The content access cost split under *rank-preserving dissimilarity* is uneven. Nodes with higher indices "pay" much less than nodes with smaller indices, irrespective of the adopted content placement strategy. This unfair cost distribution becomes more pronounced as tightness decreases. The reason behind this unfairness has to do with the demand distributions of the five nodes. Remember from Section 4 that the higher the node index the more skewed the node preference distribution. Hence, the preference of nodes is more concentrated around fewer top-ranked objects and there is less demand for the remaining objects that have to be accessed from remote storage, whether from the $N-1$ group nodes at cost t_r or outside the group at cost t_s. On the contrary, under the *shape-preserving dissimilarity* scenarios, the total access cost is more uniformly split across the group nodes and smaller in absolute values. There is far more diversity in the content that different nodes are interested in.

Contrary to high tightness scenarios, Fig. 3 and 4 suggest that neither cooperation nor altruism are attractive options when the node preferences are highly diverse. Firstly, the performance of the self-aware cooperative and altruistic strategies approaches that of the selfish one as tightness decreases. Nodes' interests are focused on fewer objects and, thus, they do not gain much by adjusting their placements through eviction and insertion of other items.

Secondly, the altruistic strategy cannot avoid mistreatment of individual nodes; for example, this is the case with Node 5 in Fig. 3(d), for both preference dissimilarity patterns we consider. Two are the straightforward remarks when looking closer at Fig. 3: a) the number of mistreated nodes increases as the value of T drops; b) it is mainly the nodes with higher indices that are mistreated (this is more apparent for the rank-preserving dissimilarity scenarios). The optimally altruistic strategy shuffles the objects in a more radical way so that some nodes may end up replacing their C top-ranked items and raise significantly their own access cost in favor of the aggregate access cost minimization.

Note that in the previous results, the nodes play in specific order. Whereas the order of play affects significantly the access cost experienced by individual nodes [14] [19], it has a much milder impact on the aggregate access cost. Table 3 reports the minimum and maximum values for the total access cost over all 120 possible permutations in the order the five nodes update their caches in the second round of the self-aware cooperative strategy. The variance of this cost with the order of play is negligible compared to the change with the Zipf distribution parameter and the shift parameter k. Relevant results can be found in [14].

6.2 Responsiveness of the self-aware cooperative strategy to changes in user preferences

In this set of simulation experiments we study how fast the self-aware cooperative strategy responds to changes in the content preferences of users. Initially the nodes' caches are empty. Nodes learn the content preferences of users over time and every time they receive a number of requests, say R, they run one iteration of the self-aware cooperative strategy; namely, each node n updates its cache with the C_n most requested content items (selfish first step) and then takes turn into adjusting its placement through evictions and insertions of new items, based on its running estimates for the user's interest in them. We let $N = 10$, $M = 10000$ items and $C_n = C = 10$ for all nodes.

In the course of the simulation, we change the content preference distributions of users twice. In the beginning, all nodes receive requests for content following the Zipf distribution with exponent $s = 1$, *i.e.*, there

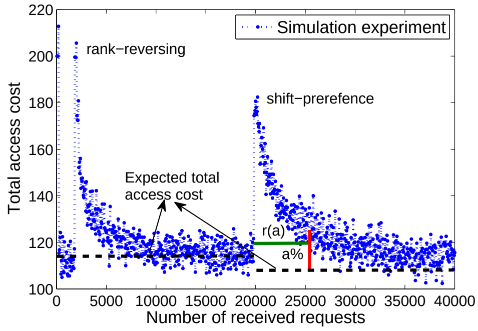


Figure 5: Performance of self-aware cooperative as function of number of received requests.

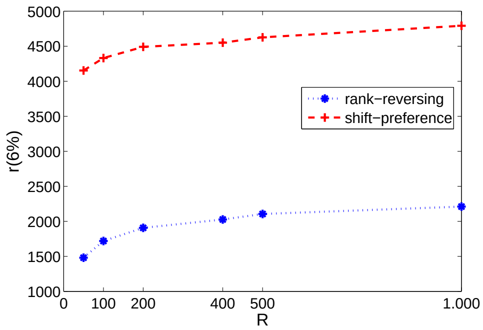


Figure 6: Access cost convergence time $vs.$period of placement algorithm execution, measured in number of requests.

is high similarity in the content preference across the group nodes and the cooperation benefits are maximized. After 2000 requests are received by all nodes, we reverse the content preference distributions so that $F'^{n}_{m} = F^{n}_{M-m+1}, \forall m \in \mathcal{M}$ and $n \in \mathcal{N}$ (*rank-reversing*). In Fig. 5, we see that temporarily, till nodes gradually inform their caches and converge to the right placements, the nodes spend much more than the average expected cost for accessing content. Each plotted point is the cost averaged over the last 50 requests received by all nodes and 100 simulation runs. The system gradually improves its performance and drops the running average access cost to within $a = 6\%$ of the expected access cost (113.93) after $r(a) = 1719$ requests. Note that the total expected access cost remains the same in this case since the distributions are symmetric. After further 18000 requests, the preference distributions change again, this time following the shape-preserving dissimilarity pattern with shift value $k = C/2$ (*shift-preference*). This time the overshoot of the running aggregate access cost is milder than the first change; yet it takes the algorithm 4232 requests before the cost converges to $a = 6\%$ of the steady-state expected aggregate access cost.

The frequency of placement adjustments (*i.e.*, iterations of the self-aware cooperative algorithm) induces a tradeoff between the excess access cost in the transient phase, during which the nodes fill their caches in response to their estimates about the nodes' preferences, and the overhead of the algorithm execution, both in terms of computations and network resources.

Table 3: Maximum and minimum values of total access cost over the 120 permutations in the order nodes update their caches in the 2nd round of the the self-aware cooperative strategy: variable T, $M = 50$, and $N = 5$.

(a) rank-preserving			(b) shape-preserving		
Tightness (T)	Max total access cost	Min total access cost	Tightness (T)	Max total access cost	Min total access cost
∞	40.0000	40.0000	∞	29.2582	29.2582
45.74	40.4886	39.4035	0.37	29.2925	28.2860
10.17	42.8774	39.2597	0.27	28.7365	27.6922
2.09	42.9292	40.1894	0.23	27.1734	27.1091
0.46	34.8047	33.1579	0.21	26.1813	25.6366
0.24	27.4510	26.1728	0.20	24.6086	24.6086
0.17	22.8114	22.3532	0.19	23.5663	23.5663

Figure 6 plots the access cost convergence time $r(6\%)$ against the period of algorithm execution R, both measured in number of requests, for the *rank-reversing* and *shift-preference* types of content preference changes. Although frequent invocations of the algorithm can accelerate the convergence of the algorithm upon disruptive changes in the users' preferences, the performance gap between lower and higher R values is rather moderate. Considering that these scenarios represent extreme cases of content preference changes across the social group population, we conclude that the self-aware cooperative strategy can respond rather fast to stochastic changes of users' content preferences even at moderate execution frequencies.

7 Application to a real network

We apply the selfish and the self-aware cooperative strategy to data traces extracted from the Delicious website (www.delicious.com). Delicious is a social bookmarking application where users can save all their web bookmarks (annotated with tags) online, share them with other users, and track what other users are bookmarking themselves. Each Delicious user together with other users, who have subscribed to see her/his bookmarked web pages, effectively forms a network. We draw on the organization of users into networks and their interests into tags to generate user interest distributions and set the content item population size, M, equal to the number of different tags in each user-network. The purpose of this example is to assess the relation between interest similarity of user groups and achievable cooperation gains under "real world" interest dissimilarity patterns beyond the synthetic ones we introduced in Section 4 and used as references for our analysis and experimentation in Sections 5 and 6.

From user interest profiles to interest distributions. Let M be the set of most popular tags used by each Delicious user. Let B_m^n be the number of bookmarks tagged with m $(1 \leq m \leq M)$ by user n $(1 \leq n \leq N)$. Then the (normalized) interest of node n in tag m is given by the ratio of the number of bookmarks tagged with m by node n over the total number of bookmarks of this user:

$$F_m^n = \frac{B_m^n}{\sum_{m=1}^{M} B_m^n}. \quad (7.1)$$

Experimentation set-up. The Delicious network is crawled in two ways. As we shall see, different ways of crawling the network can derive user groups with different tightnesses. The first method starts from four Delicious accounts (root users) chosen randomly from the website. From each root user 29 users, who follow the root user, are extracted using a breadth-first exploration of the graph formed by these links. We consider that each of these users has a capacity of 10 bookmarks in their cache. To avoid the long tail of infrequently used tags, only bookmarks that contain the 99 most popular tags (or objects) are considered for each user. The interest profiles of 120 in total users are derived from (7.1). The tightness of this group of nodes is 0.0956, which shows that common interests are not the primary reason why nodes choose to follow other nodes. Running the selfish and self-aware cooperative strategy for this group results in gain under cooperation 1.0459. This gain is similar to that obtained for low similarity under the rank- and shape-preserving dissimilarity patterns in [14], Fig. 1 and 2.

The second procedure is similar to the first one; only now the four root users are selected *among those having placed recent bookmarks on the website* and we retain the 30 highest preference tags. The tightness and gain under cooperation for this case are computed to be 0.1420 and 1.5240, respectively. As expected, they are higher since many users are interested in the same tags.

Overall, the relation between interest similarity and cooperation gain, as analyzed in Section 6, pertains also under the dissimilarity patterns that emerge from the Delicious user-networks. On a secondary note, the tightness values of these *sample* networks are low, implying that they do not avail strong interest similarity structure. This is a subject worth investigating further, our first results being reported in [13].

8 Related work

Algorithms for file sharing between computers have mostly been studied in the context of data replication [21]. Data replication refers to the storage of files or, more generally, information objects, in specific points in a network, so that they can be retrieved by requested nodes at smaller access costs. Earlier research in this area has mostly considered centralized implementations [7], [20] of file placements. However, in modern networks (*e.g.*, ad-hoc, p2p, opportunistic networks) as both the number of nodes increases and they become more autonomous, distributed algorithms become more relevant and important [19], [32]. The distributed selfish replication game is introduced and studied in [19], where the authors propose an algorithm for its solution and analyze its main properties. In [19] the assumption is that all nodes within a group can communicate and cooperate with each other. More recently, Pacifici and Dan in [25] relax this assumption and consider replication games over arbitrary social graphs, which capture possible topological constraints on the possible interaction between the players. They derive sufficient conditions for letting the players reach an equilibrium of the game and propose a distributed algorithm in this respect. On the other hand, Borst *et al.* in [4] assume altruistic players making placements that maximize the aggregate benefit over the whole network rather than theirs. The performance of their greedy algorithm is within a constant factor of two from the globally optimal performance under arbitrary demands and, even closer, within 1.33 of the optimal under identical content preferences and uniform cache capacities.

The self-aware cooperative strategy we consider coincides with the distributed algorithm in [19]. The group nodes are rationally selfish and perfectly connected and seek to maximize their own benefit from the objects they select to store in their caches. Contrary to prior work, our focus is set on the behavioral aspects of the algorithms, *i.e.*, selfish, cooperative, altruistic. We introduce a measure, *tightness*, which captures the degree of similarity in the preferences of nodes within a social group, and assess its impact on the relative performance of these algorithms(behaviors) and the resulting tradeoffs among them.

Data replication has also been studied in the context of mobile social networks, with social characteristics being embedded into data replication algorithms. In [3], the authors construct a dynamic learning algorithm where nodes from various social communities opt for a utility-maximizing content placement strategy based on their encounters with other nodes. The content utility is related to the availability of content in different communities, as well as the ties a user has with each community. In [5] the authors study how content is distributed in an opportunistic network considering both technical constraints (*e.g.*, battery/proccesing power and wireless bandwidth) and user preferences. In [15] the authors propose an approach that can enhance content dissemination by associating both interest- and locality-based dynamics of social groups. Finally, one of the main contributions of [18] is the development of a model for assessing the impact of users' characteristics (*e.g.*, interest in content and willingness to share it) on the potential gains achievable through opportunistic contacts.

9 Conclusions

In this paper we looked closer into three different strategies for content placement within a network, herein called selfish, self-aware cooperative and optimally altruistic, respectively. As their names suggest, these strategies reflect three fundamental behavioral paradigms in networked communications.

Our results suggest that the level of similarity in nodes' preferences across a social group is key to deciding which content placement strategy (*i.e.*, what kind of behavior) to follow. Altruism emerges as a win-win behavior only in tight social groups as long as the implementation cost is not an issue: it minimizes the content access cost not only collectively for the whole group (by definition) but also for each individual node. As *tightness* decreases, the collective group gain under the altruistic and self-aware cooperative strategies fades out, while certain nodes may be mistreated when behaving altruistically. Therefore, and considering also its low complexity, the selfish strategy becomes more attractive. In summary, our evaluation shows that *the benefits of cooperation increase with the group tightness*. Therefore, an on a more practical note, tightness should be used as a decision criterion: a) when choosing content placement strategies under given group membership; or, more broadly, b) for carrying out performance-driven group management operations such as group formation/merging/splitting.

As a final note, it is worth mentioning that the positive correlation between similarity and cooperation/altruism is reported in studies of human social behavior [9]. Research results in literature suggest that cooperation between individuals with similar characteristics evolves over time to a stable strategy (behaviour) [1]. Further, among the explanations evolutionary theorists have provided about the emergence of altruism in human behavior is reciprocal altruism, where individuals obtain mutual benefits through their exchanges [8], [29]. In our application, the similarity in the interests of nodes (*i.e.*, high tightness) could be seen as a catalyst for reciprocal altruism since it increases the chances of mutually beneficial interactions.

10 Acknowledgments

We thank Walter Colombo and Martin Chorley from Cardiff University for kindly providing us with traces they collected from the Delicious network. This work has been supported by the European Commission IST-FET project SOCIALNETS (FP7-IST-217141) and the the Marie Curie grant RETUNE (FP7-PEOPLE-2009-IEF-255409).

References

[1] T. Antala, H. Ohtsuki, J. Wakeley, P. D. Taylor, and M. A. Nowak. Evolution of cooperation by phenotypic similarity. *Proceedings of the National Academy of Sciences*, 106(21):8597–8600, May 2009.

[2] T. Basar and G. Olsder. *Dynamic Noncooperative Game Theory. 2nd edition*. New York, NY, USA, 1999.

[3] C. Boldrini, M. Conti, and A. Passarella. Design and performance evaluation of ContentPlace, a social-aware data dissemination system for opportunistic networks. *Computer Networks*, 54(4):589 – 604, March 2010.

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

[5] I. Carreras, D. Tacconi, and A. Bassoli. Social opportunistic computing: Design for autonomic user-centric systems. In *Autonomic Communication*, pages 211–229. 2009.

[6] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi. Measuring User Influence in Twitter: The Million Follower Fallacy. In *In Proc. 4th International AAAI Conference on Weblogs and Social Media (ICWSM)*.

[7] W. Chu. Optimal file allocation in a multiple computer system. *IEEE Transactions on Computers*, 18(10):885–889, October 1969.

[8] L. Cosmides and J. Tooby. Neurocognitive adaptations designed for social exchange. In *D. M. Buss (Ed.), Handbook of evolutionary psychology*, pages 584–627, 2005.

[9] O. Curry and R. I. Dunbar. Why birds of a feather flock together: the effects of similarity on altruism. In *under submission*.

[10] M. Denuit and S. Van Bellegem. On the stop-loss and total variation distances between random sums. *Statistics & Probability Letters*, 53(2):153–165, June 2001.

[11] Y. Gil and V. Ratnakar. Trusting information sources one citizen at a time. In *ISWC '02: Proceedings of the First International Semantic Web Conference on The Semantic Web*, pages 162–176, London, UK, 2002.

[12] M. Hefeeda and O. Saleh. Traffic modeling and proportional partial caching for peer-to-peer systems. *IEEE/ACM Trans. Netw.*, 16:1447–1460, December 2008.

[13] E. Jaho, M. Karaliopoulos, and I. Stavrakakis. ISCoDe: a framework for interest similarity-based community detction in social networks. In *NetSciCom 11': Proceedings of the Third International Workshop on Network Science for Communication Networks*, Shanghai, China, 2011.

[14] E. Jaho, M. Karaliopoulos, and I. Stavrakakis. Supplemental material for Social similarity favors cooperation: the distributed content placement case. 2012.

[15] E. Jaho and I. Stavrakakis. Joint interest- and locality-aware content dissemination in social networks. In *WONS'09: Proceedings of the Sixth international conference on Wireless On-Demand Network Systems and Services*, pages 161–168, Piscataway, NJ, USA, 2009.

[16] E. Koutsoupias and C. H. Papadimitriou. Worst-case equilibria. *Computer Science Review*, 3(2):65–69, 2009.

[17] S. Kullback. *Information Theory and Statistics*. New York, 1959.

[18] K.-W. Kwong, A. Chaintreau, and R. Guerin. Quantifying content consistency improvements through opportunistic contacts. In *CHANTS '09: Proceedings of the 4th ACM workshop on Challenged networks*, pages 43–50, New York, New York, USA, September 2009.

[19] N. Laoutaris, O. Telelis, V. Zissimopoulos, and I. Stavrakakis. Distributed selfish replication. *IEEE Trans. Par. Distr. Systems*, 17(12):1401–1413, December 2006.

[20] A. Leff, J. Wolff, and P. Yu. Replication algorithms in a remote caching architecture. *IEEE Trans. Par. and Distr. Systems*, 4(11):1185–1204, November 1993.

[21] T. Loukopoulos, I. Ahmad, and D. Papadias. An overview of data replication on the internet. In *ISPAN '02: Proceedings of the 2002 International Symposium on Parallel Architectures, Algorithms and Networks*, pages 31–36, Washington, DC, USA, 2002.

[22] D. Luenberger and Ye. *Linear and Nonlinear Programming*. third edition, 2008.

[23] R. D. Mori. *Spoken Dialogues with Computers*. Orlando, FL, USA, 1997.

[24] J. L. Myers and A. D. Well. *Research Design and Statistical Analysis (2nd edition)*. November 2003.

[25] V. Pacifici and G. Dán. Selfish content replication on graphs. In *Proceedings of the 23rd International Teletraffic Congress*, ITC '11, pages 119–126, 2011.

[26] A. A. Rahman and S. Hailes. A distributed trust model. In *NSPW '97: Proceedings of the 1997 workshop on New security paradigms*, pages 48–60, New York, NY, USA, 1997.

[27] J. P. Scott. *Social Network Analysis: A Handbook*. January 2000.

[28] A. Shikfa, M. Onen, and R. Molva. Privacy in content-based opportunistic networks. In *WAINA '09: Proceedings of the 2009 International Conference on Advanced Information Networking and Applications Workshops*, pages 832–837, Washington, DC, USA, 2009.

[29] R. L. Trivers. The evolution of reciprocal altruism. *The Quarterly Review of Biology*, 46(1):35–57, 1971.

[30] J. Vegelius, S. Janson, and F. Johansson. Measures of similarity between distributions. *Quality and Quantity*, 20(4):437–441, 1986.

[31] J. Wang, W. W. Tsang, and G. Marsaglia. Evaluating Kolmogorov's distribution. *Journal of Statistical Software*, 8(18):1–4, November 2003.

[32] O. Wolfson and S. Jajodia. Distributed algorithms for dynamic replication of data. In *PODS '92: Proceedings of the eleventh ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 149–163, New York, NY, USA, 1992.

Eva Jaho is currently working as an IT consultant at Athens Technology Center, Greece (www.atc.gr). She received the PhD and MsC in Networking from the Department of Informatics and Telecommunications of the National & Kapodistrian University of Athens, Greece, in 2011 and 2007 respectively. She received the Diploma degree from the same university department in 2005. She has participated in several European research projects in the field of networking and telecommunications. Her main research interests lie in the analysis of content networks and data dissemination, as well as social networking applications.

Merkouris Karaliopoulos holds a Marie Curie Fellowship with the Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Greece. He was awarded his diploma in Electrical and Computer Engineering from Aristotelian University of Thessaloniki, Greece, in 1998, and his PhD in the same field from the University of Surrey, UK, in 2004. He spent one year (2006) as postdoctoral researcher in the Computer Science Dept. of University of North Carolina at Chapel Hill, NC, USA and three years (2007-2010) as Senior Researcher and Lecturer in the Swiss Federal Institute of Technology (ETHZ), Zurich, Switzerland. His research interests lie in the general area of wireless networking, currently focusing on aspects of network and service resilience to imperfect node cooperation.

Prof. Ioannis Stavrakakis received the Diploma in Electrical Engineering from the Aristotelian University of Thessaloniki, Greece (1983), and the Ph.D. in EE from the University of Virginia (1988). He was Assist. Professor of CSEE, Univ. of Vermont, during 1988-1994, Assoc. Professor of ECE, Northeastern Univ., Boston, during 1994-1999, Assoc. Professor of Informatics and Telecommunications, University of Athens, Greece, during 1999-2002, and Professor since 2002. His research interests are on resource allocation protocols and traffic management for communication networks, with recent emphasis on: peer-to-peer, mobile, ad hoc, autonomic, delay tolerant and future Internet networking. His past research has been published in more than 180 scientific journals and conference proceedings. He has organized several conferences, has been a Chairman of IFIP WG6.3, Assoc. Editor for the IEEE/ACM Trans. on Networking, ACM/Springer Wireless Networks, Computer Networks, and currently of Computer Communications Journals. He is currently Assoc. Dept Chair and Director of Graduate Studies and an IEEE Fellow.