Consiglio Nazionale delle Ricerche

Design and Performance Evaluation of Data Dissemination Systems for Opportunistic Networks Based on Cognitive Heuristics

M.Conti, M. Mordacchini, A. Passarella

IIT TR-15/2012

Technical report

Ottobre 2012

Istituto di Informatica e Telematica

Design and Performance Evaluation of Data Dissemination Systems for Opportunistic Networks Based on Cognitive Heuristics

Marco Conti, Matteo Mordacchini, Andrea Passarella IIT-CNR

ABSTRACT

It is often argued that the Future Internet will be a very large scale content-centric network. Scalability issues will stem even more from the amount of content nodes will generate, share and consume. In order to let users become aware and retrieve the content they really need, these nodes will be required to swiftly react to stimuli and assert the relevance of discovered data under uncertainty and only partial information. The human brain performs the task of information filtering and selection using the so-called cognitive heuristics, i.e. simple, rapid, low-resource demanding, yet very effective schemes that can be modeled using a functional approach. In this paper we propose a solution based on one such heuristics, namely the *recognition* heuristic, for dealing with data dissemination in opportunistic networks. We show how to implement an algorithm that exploits the environmental information in order to implement an effective dissemination of data based on the recognition heuristic, and provide a performance evaluation of such a solution via simulation.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication

General Terms

Algorithms, Design, Performance

Keywords

Opportunistic Networks, Cognitive heuristics, Recognition Heuristic, Data Dissemination

1. INTRODUCTION

In the Future Internet scenario, mobile devices will be part of a vast, dynamic information environment, where data will come from many, disparate sources. More traditional CDNs or P2P networks [Passarella 2012] will be coupled with the data coming and spread by the mobile devices themselves. In fact, the increasing active user participation to the process of data creation and diffusion will create a huge quantity of pervasive information. Moreover, a considerable part of these data will also be very contextualized, i.e. relevant only at specific times and/or geographic areas, and of interest only for specific groups of users. In such a context, it is reasonable to think to data exchange schemes, where data is exchanged directly between users upon physical conctact, rather than rely on fixed infrastructures, both for communication and data sharing. Opportunistic networking techniques will thus become a very important complement to infrastructure- based networks supporting mobile users in order to efficiently disseminate content to interested users [Pelusi et al. 2006].

In this context, each device will be exposed to a massive amount of heterogenous content data coming from disparate sources. In order to avoid to be deluged by such an amount of data, devices moving in this congested information landscape will have to face the challenging task of rapidly react to the discovery of new data and assert the relevance of such content in order to select the most interesting information for both their users and the overall opportunistic exchange of data. This selection should be performed swiftly, since the contextualized nature of information could make it aged or not available anymore before a complex evaluation process has ended. Moreover, not all the variables required to perform a complete evaluation may be known. Finally, nodes - in general - will contribute limited resources to the dissemination process (e.g. in terms of computing and storage capabilities). Thus, the data selection process must be very lightweight and able to perform a sharp distinction between data items, since only a very limited part of them could be stored.

One approach to address the aforementioned problems is to embed autonomic decision-making capabilities into mobile devices. In this paper, we explore a new (to the best of our knowledge) direction in the autonomic networking field, i.e., we exploit results coming from the cognitive psychology area, by using models of how the human brain assesses the relevance of information under partial knowledge. We are not simply proposing another bio-inspired approach to self-*, but we are trying to directly embed in an ICT system the rules and procedures for content selection applied by the final user of the ICT system: the *human brain.*

Within the Future Internet scenario, we envision that mobile devices will act as proxies that allow users to explore the Future Internet information landscape. Mobile devices will then be the *avatars* of their respective users in such a cyber world. As shown on the left side of Fig. 1, a conventional approach is to let these devices apply some (even complex) information filtering task in order to present to their respective human counter-parts some information that the users have to further discriminate. We want to take a different direction with respect to this kind of approaches. As we show in the right side of Fig. 1, since each device is a proxy of its human user, we want to equip it with the *very same* cognitive processes used by human brains to filter information. In fact, brains are able to swiftly contextualize the stimuli they are subject to, identify the relevant features and knowledge to be considered, assert relevance of perceived information and finally select the most useful data, even when only partial information is available. Therefore, we want to go well beyond conventional bio-inspired networking solutions, by letting those devices to apply the *very same* human cognitive functions in order to let them become aware of the environment, and take self-* decisions in order to implement an effective information selection and dissemination strategy in the Future Internet scenario.

When faced with large amounts of data, human brains are able to swiftly react to stimuli and assert relevance of discovered information, even under uncertainty and partial knowledge, with respect to the individualâĂŹs particular needs and context. This ability relies on the so-called *cognitive heuristics*. In computer science, heuristics are computational methods that try to optimize a problem by producing stochastically good results. They are obtained by pruning the search space through an iterative improvement of a candidate solution, with regard to a given measure of quality. On the other hand, cognitive heuristics are fast, frugal and adaptive strategies of the brain that allow humans to face complex situations by addressing simpler problems. Cognitive heuristics are effective, simple rules, requiring little estimation time and working under incomplete knowledge of the problem space. Hence, despite their simplicity, their are indispensable psychological tools, that result to be very effective in solving decision-making problems like information selection and acquisition. The cognitive psychological theory provides a description of the above processes through mathematical models that provide a sort of âĂIJblack boxâĂİ description of the functional behaviour of a cognitive process. Note that this marks a difference with respect to conventional artificial intelligence approaches. As an example of the latter systems, consider a neural network, where each processing element (i.e. each *artificial neuron*) is designed in order to mimic the properties of real, biological neurons. Differently from this kind of approaches, we do not seek to exploit formal descriptions that try to reproduce the physiology of cognition. Rather, we exploit *functional* descriptions of a set of the most relevant processes used by the brain in the decision-making process, i.e. the *cognitive heuristics* (e.g. [Goldstein and Gigerenzer 1996]).

Among all the cognitive heuristics, Goldstein and Gigerenzer [Goldstein and Gigerenzer 1996, Gigerenzer and Goldstein 2002] have studied and modelled one of the simplest and more effective ones: the recognition heuristic. This heuristic assumes that merely recognizing an object is sufficient to take decisions that would theoretically require much more information about the object's properties. A detailed description of the recognition heuristic is provided in Section 3. This kind of heuristic has proved to be not only fast and frugal, but it is also *ecologically rational*, in the sense that it exploits structures of information coming from the environment in order to work.

In this paper, we want to exploit the fast and frugal recognition heuristic to design a data dissemination system in an opportunistic networking scenario. In this scenario, nodes carry some data, are interested in acquiring specific types of content and have the possibility to store some of the data encountered when moving in the environment. We propose an exploitation of the recognition heuristic to let each node rapidly decide which is the utility of taking one data item instead of another upon making direct (i.e. one-hop) contact with other nodes. First of all we define the requisites needed to implement the recognition heuristic in an opportunistic environment by defining the main variables involved in this process. Then, we propose an algorithm inspired by the model of Goldstein and Gigerenzer that exploits the recognition heuristic in order to simplify and limit the complexity of the data selection task. Finally, we evaluate by simulation the data diffusion process when nodes exploit the proposed solution.

The rest of this paper is organized as follows. In Section 2 we briefly survey the state of the art on data dissemination in opportunistic networks. In Section 3 we give a more precise description of the recognition heuristic. In Section 4 we introduce how the recognition heuristic can be implemented by mobile devices, while in Section 5 we define an algorithm that exploits it for the purpose of data dissemination in an opportunistic network. Section 6 presents the experimental results obtained via simulation. Finally, Section 7 concludes the paper.

2. RELATED WORK

The first work that investigated the problem of content dissemination in an opportunistic network scenario was developed in the PodNet Project [Lenders et al. 2008]. In PodNet, items are arranged in channels, based on their content. Each node is subscribed to a channel and thus tries to retrieve all its related items. Part of the cache of each node is devoted to storing the items the node is subscribed to (*private cache*), while another part is made available for a collaborative exchange of information (*public cache*). The public cache will contain the items the node considers most useful for a cooperative item dissemination. Upon meeting, nodes exchange summaries of the items they are carrying. Each node firstly requests to the other the items of its subscribed channel. After that, it evaluates the remaining objects using a function that tries to estimate the utility of storing each particular data item for the effectiveneess of the data dissemination process. The authors propose four different strategies to decide which data items to store, all based on the items channel popularities. All the proposed strategies outperform a scenario where nodes keep only the items of the channel they are subscribed to. PodNet does not exploit any social information about nodes. On the ohter hand, more advanced approaches for data dissemination in opportunistic networks exploit information about users social relationships to drive the data dissemination process [Yoneki et al. 2007, Boldrini et al. 2010].

Specifically, the work in [Yoneki et al. 2007] defines a pub/sub overlay over an opportunistic network. Authors

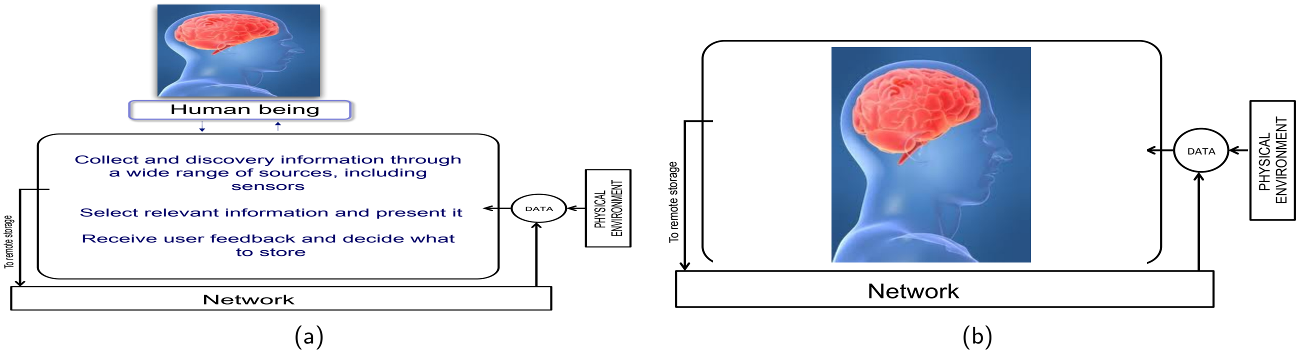

Human being
Collect and discovery information through a wide range of sources, including sensors
Select relevant information and present it
Receive user feedback and decide what to store
DATA
PHYSICAL ENVIRONMENT
To remote storage
Network
(a)
To remote storage
DATA
PHYSICAL ENVIRONMENT
Network
(b)

Figure 1: A traditional (a) and a self-aware (b) node in a content-centric Internet

starts from the observation that data producers and consumers are rarely in the network at the same time. Thus, it is reasonable in an opportunistic network to have some nodes acting as brokers in more traditional pub/sub overlays. They are in charge of dispatching relevant content toward the most interested nodes. More precisely, the authors assumes that nodes in the network can be grouped in communities. Communities are groups of nodes bound by social connections, that spend a significant portion of time together. The authors propose two in-line algorithms that allow nodes to detect their own community. These algorithms are based on a gossip-like exchange of information and have a very good performance compared with off-line community detection algorithms. The very same gossip algorthims are used by nodes to determine their centrality inside their communities, i.e. how easily they can reach any other node of the community. Hence, brokers are the most "socially-connected" nodes, i.e., those nodes that are expected to be most available and easily reachable in the network. Brokers of different communties form a conceptual overlay, using gossiping through encounters. Brokers know the subscriptions of their communties. Data generated in a community is sent to the broker that deliver it to brokers of other interested communities.

In ContentPlace [Boldrini et al. 2010], an even more refined and complete approach is used. Specifically, dissemination is driven by the social structure of the network users, such that nodes store data items that are likely of interest to users they have social relationships with (and who, therefore, are expected to be in touch in the near future). Like in PodNet, when two nodes come in contact, they exchange summaries of the content of their caches and decide what to fetch from the encountered node. To this end, ContentPlace proposes a set of social-aware dissemination strategies. Each strategy tries to give an optimal solution to a multicostrained knapsack problem, where the goal is to maximize the social utility of fetching an item and, at the same, taking into account the limited resources of a device, by computing the resource consumption of this action. The main parameter that has to be estimated is the social utility of fetching a given item. This is a sum of the utilities for each single community the node belongs to. Every single community utility is computed by estimating how many community users may be interested in that item (i.e. its *access probability*) and how many nodes in the community already share it (i.e. the item *availability*). Utility is directly proportional to the access probability and inversely proportial to the availability. Both these parameters are estimated as a result of meetings with other nodes. The different strategies proposed give different weights to the utility of each community, thus implementing different dissemination policies. The best results are obtained by strategies that disseminate items on the basis of future encounters.

Rather than exploiting local optimization policies, like in ContentPlace, in [Reich and Chaintreau 2009] the authors define the content dissemination issue as a global optimization problem. They view all the nodes' shared memories as a unique, global cache. The problem of which item to fetch upon contact is defined as a global optimization problem in a similar way as in ContentPlace. Differences are in the fact that the resources to be considerd are not the single node cache, but the global cache is used instead, and the utility function is not computed from a node individual point of view, but it is defined globally. As for the latter point, the authors consider that the utility of a content item for a given node decreases monotonically with the time needed to wait between the issue of the item's request and the moment in which the item is effectively fetched. The global utility function is defined over the whole set of items and the overall set of nodes. It defines the best possible allocation of items to all the nodes. This is done by considering the items' utilities for each single node, weighted with the actual expected rate of requests for every item. Clearly, the global parameters needed to compute the above values cannot be known by each single node. Thus, in practice, each node adopts a local approximation policy. Precisely, a node counts how many contacts are needed from the time it issues a request and the time it gets the required item. If it takes n contacts, then the node will send a copy of the item to the next n encounters.

The Push-and-Track system [Whitbeck et al. 2011] proposes a trade-off between an infrastructure-based dissemination approach and pure opportunistic-based solutions. The authors study the dissemination process in a target area, the size of a compus or even a city. They assume that nodes participating in an opportunistic network are also typically in contact with fixed wireless broadband infrastructures, like the ones of cellular network operators. The authors consider a scenario where content is produced in the Internet and must be delivered to interested users in the area within a given temporal deadline. The central infrastructure sends the content to a small subset of users, which, in turn, start disseminating it with pure opportunistic strategies. Each node that receives the content sends an ack message to the central controller. Using an "ideal dissemination plan", it periodically checks the fraction of users that get the content with respect to the final number of nodes supposed to receive it. If there is a too large gap, the content is re-injected to another subset of nodes. Finally, when the deadline is approaching, the content is sent to all the remaining users that do not have received it yet. The authors propose different strategies for selecting a proper subset of initial users and when content has to be re-injected in the network, i.e. the gap between the actual dissemination status and the "ideal" one is too large.

In addition to the systems reviewed above, other data dissemination algorithms have been proposed for diverse families of mobile networks. The work in [Yin and Cao 2006] is representative of a body of work focused on caching strategies for well-connected MANETs. In this paper we focus on more challenged networking environments, where such policies cannot be applied.

All the above systems use computer-science heuristics. With respect to these approaches, in this paper we take a completely new direction, by borrowing models of human cognitive processes coming from the cognitive psychology domain. Due to the characteristics of these cognitive processes, one of the results we expect to achieve is to build a content dissemination mechanism as efficient as other state-of-the-art solutions (i.e. able to deliver data to all the requesting nodes) while, at the same time, limiting the resources needed to reach this result. As this approach is still totally unexplored, in this paper we limit the set of contextual information that we use to the very minimum, and, for example, we do not exploit information about users social structures. This allows us to obtain initial exploratory results about the feasibility of this novel approach. The work presented in the following sections is an extension of the one we presented in [Conti et al. 2011]. The main extensions that we add in this paper regard a more detailed description of the cognitive concepts that are behind this work, a more complete illustration of the developed algorithms and an extensive set of simulation results, including a comparison with another state-of-the-art solution and tests under various, different scenarios.

3. COGNITIVE HEURISTICS AND THE RECOGNITION HEURISTIC

Heuristics are cognitive strategies that allow the brain to face complex problems where the search of an optimal solution is too complex, requires too much time and information and is too computationally expansive to be computed. In constrast, heuristics are able to deal with difficult problems by answering simpler problems. The cognitive approach behind the study of heuristics is opposed to the study of human behavior as guided by an unbounded rationality. The latter approaches consider that a rational behaviour can be modeled by assuming that a person is able to know all the alternatives and all their consequences (with associated probabilities) related to a given problem. The problem optimal solution can then be computed taking into account all these variables by a complex (and time-consumig) calculation. The *bounded rationality* [Simon 1955, Simon 1990, Goldstein and Gigerenzer 1996] view, on the other hand, argues that in real-world scenarios people act in the environment and take decisions under limits of time, knowledge and computational capabilities. From this perspective, in order to come up with a solution, humans have to rely upon simpler yet effective decision strategies.

Heuristics can be defined as these simple rules used by the brain for facing situations in which people have to act quickly, relying on a partial knowledge of all the problem variables, the final utility evaluation criterion is not known and the problem itself may be ill-defined in such a way that traditional logic and probability theories are prevented to find the optimal solution. Heuristics are *fast and frugal* [Goldstein and Gigerenzer 1996, Gigerenzer 2004]. They are fast because the simplicity of their rules allow them to give a response in a very short time. They are frugal, since they work by ignoring part of the available information. Rather than a limit, exploiting only a fraction of the information translates into an advantage of heuristics when compared to more complex cognitive strategies. The latter, in fact, may *overfit* existing data, i.e. when making predictions they use both "good" data, useful to forecast new events, and irrelevant, noisy information. As a consequence, these methods are good in fitting *all* existing, known information, but become less accurate when have to predict new, unseen data. On the other hand, heuristics rely only on small samples of the whole information. Counting on cognitive limits, such as forgetting, they are more able to keep into consideration relevant data with respect to more sophisticated cognitive models [Gigerenzer 2004].

Critics of the *fast and frugal* framework consider that cognitive heuristics can lead to systematic errors and biases (e.g. [Evans and Over 2010]). Scholars of the *fast and frugal* model reply that they provide formal models that allow to compute *quantitative* results on the number of errors that cognitive heuristics can make. These results show that, in many situations, cognitive heuristics are more accurate, with less effort, than more complex decision-making strategies [Marewski et al. 2010a].

Anyway, each heuristic is not an all-purpose set of rules that can be used to solve almost any problem. Rather, heuristics form a sort of 'adpative toolbox' of the brain [Gigerenzer and Todd 1999, Gigerenzer 2008, Marewski et al. 2010b]. Each heuristic of such a toolbox is shaped to work for solving a single problem under specific enviromental conditions. As often cited in recent cognitive heuristic literature, Simon defined the human rational behaviour as a scissor whose two blades are the structure of the environment and the computational capabilities of the actor [Simon 1990]. In fact, heuristics exploit naturally available evolved capabilities of the mind, like vision and memory in order to derive their simple judgement rules. Moreover, these rules are based on regularities that the actor finds in the environment (both phisical and social). Thus, each heuristic has its own environmental conditions under which is able to give good predictions, allowing to face complex problems. Heuristics are said to be *ecologically rational* when their structure is adapted to the structure of the information in the environment [Gigerenzer and Goldstein 2002, Todd and Gigerenzer 2003]. The mind is able to select the most useful heuristic from its 'adaptive toolbox', given the environment context. One of the main topic of research is to analyze in which environment a heuristic is able to perform well.

In the following, we give a brief description of some examples of cognitive heuristics. *Tallying* [Dawes 1979] uses m out of a total of M cues, in order to discriminate among alternatives using an unknown criterion. In other words, when comparing a set of alternatives, this heuristic makes use of a subset of m cues only, i.e. it typically does not rely on all the available information of M possible cues. For each alternative, it simply counts the number of favorable cues. The heuristic does not give any special weight to any of the m chosen cues. It assumes they all have the same relevance in determining the best option. The alternative with the highest number of positive cues is then selected. In case there is a tie between two or more alternatives, it looks to one more cue. In case no other cues are available, it guesses among the remained alternatives. Tallying has proved to perform the same or even better than multiple regression models.

The *equality heuristic*, or the $1/N$ rule [DeMiguel et al. 2009], is an heuristic used to allocate resources to a set of N possible alternatives. Using this heuristic, resources are allocated uniformely across all alternatives, i.e. all alternatives have the same weight. As an example, consider to have to choose how to allocate money among a set of N possible funds. Using this heuristic, money is equally allocated among all funds. As a matter of fact, the *equality heuristic* has proven to be particularly effective in the financial asset allocation problem, outperforming optimal asset allocation models. More generally, it is effective when the set of alternatives is large, the choice among them is subject to high predictive uncertainty and the learning sample is small.

The *fluency heuristic* [Schooler and Hertwig 2005, Jacoby and Brooks 1984, Whittlesea 1993] assumes that, among two alternatives, the one that is recognized *faster* than the other has a higher value with respect to the (unknown) evaluation criterion. This heuristic is useful when the actor is able to retrieve (recognize) both alternatives from memory, but one is retrieved faster. Hence, people rely more easily on the fluency heuristic when knowledge about alternatives is poor, since differences in retrieval times tend to be more relevant in this case.

The *default heuristic* [Johnson and Goldstein 2003] deduces that, if a default exists, and the adherence of the actor to it implies no actions, then the actor should do nothing to change her status. This heuristic proves to be particular relevant in the definition of policies for specific problems. In particular, it has proved to be relevant in the organ donation policies, where an opt-out policy (non-donors have to explicitly declare their status) comes out to be more effective than opt-in (donors have to register as such) strategies.

Among all the heuristics, one of the simplest, and one that attracted a broad attention in the last decade of research, is the *recognition heuristic* [Goldstein and Gigerenzer 1999, Gigerenzer and Goldstein 2002]. The recognition heuristic is based on a very simple rule. When evaluating a couple of objects, and one is *recognized* (i.e. the actor is able to recall from memory that she has already "heard" about that object) and the other is not, the recognition heuristic inferes that the recognized object has an higher value with respect to a given evaluation criterion. It can also be used with sets of more than two objects, in order to draw out the subset of the most significant objects [Marewski et al. 2010]. People tend to rely on this heuristic when the real criterion value is not available, not known or requires further, more complex (and longer and expensive) reasoning to be computed. If the criterion is available, otherF kinds of processes can be applied. The recognition heuristic is said to be *ecologically rational*, i.e. effective, when the recognition of objects is highly correlated with the final evaluation criterion (to be inferred). This heuristic adaptively derives this correlation from the surrounding environment.

In order to better explain how the recognition heuristic works and which are the main elements that are taken into account, Gigenrenzer and Goldstein [Gigerenzer and Goldstein 2002] use, as an example, the estimation of the university endowments. It the following example we refer also to Fig. 2, which depicts the general elements involved in the recognition heuristic, and the relationships between them. In this example, a person is asked to determine which university has the biggest endowment, choosing between a couple of university names. Hence, the evaluation criterion to be used is the value of the endowment. Anyway, such information is generally not publicly available. Nevertheless, it is argued that newspapers could act as mediators, since they periodically publish news related to the most important universities. Thus, the number of times a university appears on the newspapers could be a strong indicator that it has larger endowments than universities that do not, or rarely, appear on the media. From the perspective of the recognition heuristic, the role of newspapers is that of environmental *mediators*. In fact, the heuristic exploits the presence in the environment of some *mediators* that carry information used by the heuristic itself to approximate the value of objects with respect to the criterion. Mediators spread this information in the environment, thus determining which objects are recognized. In other words, the more often information about an object is encountered in the environment (carried by the mediators), the more probable the object will be recognized. The correlation between mediators and the evaluation criterion is called *ecological correlation*. In the example, newspapers play the role of mediators and the mediator variable related to the criterion is the number of citations. In fact, newspapers influence the recognition of university names, since the more they cite an institution, the more likely that institution name will be remembered and, thus, recognized. When a person has to choose which university has the biggest endowments between a couple of institution names, she uses the recognition heuristic and chooses a recognized name against an unknown one. Since the brain evaluates options exploiting the citations on newspapers instead of the real, unknown criterion, the relation between the recognition and the mediators is called *surrogate correlation*. From this example it is straightforward to notice that the effectiveness of the recognition heuristic, i.e. the *recognition validity*, is continuously reinforced by the stimuli received from the environment.

Critiques have been addressed to this model of the recognition heuristic. Critics point out that the experiments used to validate the recognition heuristic could imply other cognitive processes in addition to recognition [Oppenheimer 2003], supporting this conjecture with additional sets of experiments. Anyway, recognition heuristic scholars say that this latter experiments were done under different conditions as the orginal ones and in such a way that participants could either recognize items from a knowledge independent from the final cirterion (i.e. there is no ecological correlation), thus invalidating the heuristic, or because they had direct knowledge about the criterion [Gigerenzer 2008, Gigerenzer and Goldstein 2011].

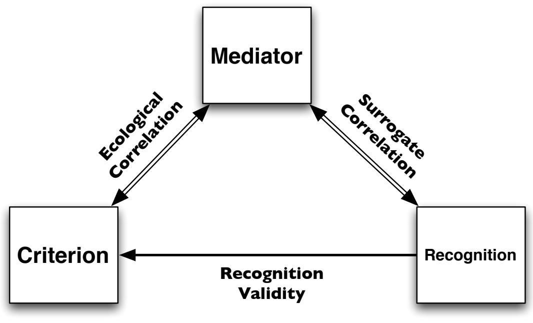


Figure 2: Ecological Rationality of the recognition heuristic

While in the previous description the recognition heuristic is used for choosing among pairs of options, it is now becoming to be considered as one of the cognitive strategies for the creation of so-called *consideration set* when dealing with multi-alternative choices. The general notion of *consideration sets* comes from the marketing literature [Shocker et al. 1991, Alba and Chattopadhyay 1985, Hauser and Wernerfelt 1990, Laroche et al. 2003]. Whitin this filed, a *consideration set* can be defined as the subset of brands that consumers evaluate when making a purchase decision. Since products of many brands can be on display, each having similar features and potentially subject to various price promotions, the brain has to rely on strategies that tries to minimize the cost of information search and limit the attention only to a (small) subset of the available brands. This limited subset of all the available products is termed as *consideration set.* The final pruchase decision will be sorted out from this set. More broadly, a consideration set can be regarded as a smaller subset of all the available information, where only the most relavant data is kept, which contains what will be the final result of the evaluation process. If needed, items in a consideration set can be further ranked using other strategies, like other heuristics. Anyway, the recognition heuristic permits a sensible reduction of the number of alternatives, and to exploit consideration sets to make the decision-making process easier.

The simple description of the recognition heuristic made it a powerful tool for making predictions about a given criterion. The recognition heuristic can be exploited as a support in decision-making processes. As such, it has been successfully used in various fields [Marewski et al. 2010b], like financial decision-making processes [Monti et al. 2009], forecasting future purchase activities [Goldstein and Gigerenzer 2009] or even sport events results [Serwe and Frings 2006] or political election outcomes [Marewski et al. 2010].

4. THE RECOGNITION HEURISTIC FOR DATA DISSEMINATION IN OPPORTUNISTIC NETWORKS

4.1 Overall Concepts

In this section we describe and define how the recognition heuristic can be exploited for solving the data dissemination problem for mobile nodes of an opportunistic network.

More precisely, the scenario we consider is made up of a number of mobile and autonomous nodes which generate data items and other peers[1] can be interested in them. The system is completely decentralized and the device owners are interested in data *channels*, i.e. the high-level topics to which the data items belong. Items generated by each node may pertain to one or more channels. The goal is to bring all the data items of a given channel to all the nodes that are interested in it. To this end, nodes collaboratively contribute to the diffusion of information. In fact, each peer contributes a limited amount of storage space to help the dissemination process, since *contacts* between users are the *only way* to disseminate data items.

More in detail, a node internal storage space is orgazined as depicted in Fig. 3. With respect to this figure, we have:

Data caches:

- LI is the cache containg the *Local Items*, i.e. the items generated by the node itself

- SC is the *Subscribed Channel* cache, i.e. the cache containing the items belonging to the channel the node is subscribed to and obtained by encounters with other peers

- OC is the *Opportunistic Cache*, i.e. the cache containing the objects obtained by exchanges with other nodes and beloging to channels the node is not subscribed to. They are the items the node believes to be the most "useful" for a collaborative information dissemination process. They are selected using the values contained in the *Recognition* caches

Recognition caches:

- CC is the *Channel Cache*: whenever the node meets another peer subscribed to a given channel, the channel ID is put in this cache, along with a counter. It exploits a *recognition threshold* R_c

- IC is the *Item Cache*: similarly to the previous cache, when a new data item is seen in exchanges with other nodes, its ID is put in this cache, along with a counter. It exploits a *recognition threshold* R

In the following, we show how these caches are used to obtain a cognitive heuristic-based information dissemination scheme. When two nodes come in contact, they exchange summaries of data items in all of their *data* caches. Firstly, each node fetches the items of the channel it is subscribed to and add them to its SC. Then, ideally, the node should evaluate which of remaining data items of the encountered peer should be fectched on the basis of their utility in the global information diffusion process. Clearly, this is a hard (or impossible) target criterion to evaluate for a single node. The application of a fast, frugal and effective strategy like the recognition heuristic can significantly reduce the complexity of this evaluation process. Since we wish to select the subset of the most relevant items to fetch, among a bigger set of data items possibly available during an encounter, we want

[1] hereafter the terms nodes and peers are used interchangeably

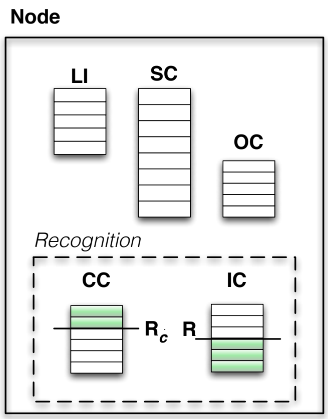


Figure 3: Organization of a node's internal memories

to exploit the recognition heuristic to select a data consideration set. To this end, in this section we characterize the elements that permit to use the recognition heuristic in this environment and describe, in Alg. 1, how it is implemeted, using the CC and IC caches. In the next section, we define an algorithm that, starting from the recognition heuristic, effectively filters the information, with the aim of maximizing the utility of the exchange of objects among nodes, and it is used to select the items to fetch and keep in the OC cache.

In order to exploit the recognition heuristic in such a way, the first step we have to take is to define the elements upon which recognition will be made in an ecologically rational way. With reference to Fig. 2 and the description of the recognition heuristic given in the previous section, we have to identify the elements that define the ecological rationality of this heuristc, in order to use it in our scenario. Specifically, we have to identify:

- the features (like the name of cities or universities in the examples of Goldstein and Gigerenzer) that are highly correlated with the selection criterion and that are thus spread by the mediators;
- the environmental mediators;
- the way by which nodes implement the heuristic based on the information collected from mediators

As for the first point, it is of particular importance to decide which are the elements that contribute to determine the utility of a data item. We consider that an item utlity is driven by two simple factors: the popularity of its channel, and its availability. These factors have always been considered as fundamental in the data management literature, starting from the area of web caching [Balamash and Krunz 2004] and are considered also in the opportunistic network literature (e.g. [Boldrini et al. 2010]). Specifically, the utility of a data items is positively correlated with the popularity of its channel (how many users are interested in that item), and negatively correlated with its availability (how many times that item is already replicated).

As for the second point, we have to determine which are the actors that are present in the environment and that can carry useful information, with respect to the above-mentioned features. We use nodes themselves as mediators. The variables they spread are, respectively, the channel they are interested into, and the set of items they are currently storing in their shared storage space. The communication of such information by any other peer, is used by a node as a stimuli from the environment it is interacting with. Upon such stimuli, it is then possible to build a recognition process.

As for the third point, since we have now defined which are the relevant features and who is spreading them, we need to determine the process upon which the recognition heuristic can be implemented. The bottomline idea is to use two recognition heuristics to separately recognize channels and data items. Intuitively, a node recognizes a channel as soon as it becomes "enough popular". It means that a node considers a channel popular as soon as it encounters enough other nodes that are interested in that channel. Furthermore, a node recognizes that a data item is "spread enough" as soon as that item is encountered at least a given number on other nodes. In a parallel with the cognitive recognition heuristic, being "enough popular" or "spread enough" mean that a node was subject to a suffcient number of stimuli from the environment about a channel or an item. In other words, subscriptions to a channel or the presence of an item in another node's cache were communicated so many times that the corresponding channel or item becomes recognized. Thus, a channel or an item are marked as *recognized* once the stimuli associated to them has been reiterated a number of times greater than a given "recognition threshold". This behaviour is based on the cognitive science research on how the recognition memory works in the brain. As all the cognitive heuristics, the recognition heuristic is based on advanced capabilities of the human brain. Memory is among them. The recognition heuristic prefers items that the brain is able to recall about against unknown ones. When happens that the brain is able to recall that it already "saw" an item and, thus, recognizes it? Although the question is still open and debated in the cognitive science community, results reported in the cognitive psychology literature show that the recognition memory works on a threshold-based principle. Therefore, when the brain is exposed to an information over a certain number of times, that information can be recognized. Findings on this topic [Schooler and Hertwig 2005] describe this behaviour as founded over a single recognition threshold (i.e items are recognized only when they are "seen" more than a given number of times) , while a more recent work [Erdfelder et al. 2011] argues that the recognition memory response could be based on two thresholds, one over which information is surely recognized and one under which items are certainly not recognized, with a more fuzzy conduct inbetween the two thresholds. In order to limit the number of parameters involved in this first attempt to translate the recognition heuristic in an ICT scenario, we adhere to a single-threshold model of the recognition memory, leaving space for further extensions and studies on different models, based on very recent cognitive science results on this matter.

4.2 Detailed algorithm

Hereafter we describe how the points described above can be pracatically implemented in order to exploit the recognition heuristic strategy in an opportunistic network scenario. The complete recognition algorithm is shown in Algorithm 1, to which we refer in the following description. As shown in Fig. 3, each node of the network maintains a separate *recognition cache* for channels and data items, i.e. the CC and IC caches. Entries of each of those caches correspond to channels of interest for or data items carried by encountered nodes, respectively. Each entry contains a counter and a TTL associated with the channel or data item. Since it is proved that forgetting could help the recognition heuristic [Schooler and Hertwig 2005], we consider that each element can remain in memory for a limited time only. After that time has elapsed, the element is "forgotten", i.e. it is dropped from memory. Anyway, every new stimulus about an element (channel or data item) reinforces its presence in the caches. Thus, whenever a node interested in a channel (or storing a data item) is encountered, the associated counter is incremented and the TTL reset, prolonging its permance in memory (line 11 of the algorithm). Whenever a channel subscription (data item) is found during an exchange, its associated counter is increased. When the counter reaches a certain threshold (R_c for channels, R for items, as in Fig. 3) , the corresponding channel or data item is deemed as *recognized* (lines 4–10). Since the space in the caches is limited, when a cache becomes full and new elements are encountered, a replacement might occur (line 13). In this case, the entry with the oldest TTL is selected for replacement. Since we believe that recognized elements are of extreme relevance, if the selected element corresponds to a recognized channel (or data item), this entry is stored and preserved in a Bloom filter. Otherwise, it is dropped (lines 14–18). A Bloom filter allows to keep track of old recognized elements exploiting only a very limited memory.

Algorithm 1 Recognition algorithm

```
 1: Let $i$ be an observed channel/item;
 2: Let $H$ be a hashed index of removed channels/items
 3: Let $R_\theta$ be the recognition threshold
 4: if Cache.contains( $i$ ) then
 5:     if $i$ is not recognized then
 6:         Increment $i$ counter
 7:         if $i$.counter = $R_\theta$ then
 8:             Mark $i$ as recognized
 9:         end if
10:     end if
11:     reset $i$.TTL
12: else
13:     if Cache is full then
14:         Select the item $o$ with the oldest TTL
15:         if $o$.counter $\geq R_\theta$ then
16:             Move $o$ to $H$
17:         end if
18:         Drop $o$
19:     end if
20:     Put $i$ in the Cache
21:     $i$.counter = 1
22:     Set $i$.TTL
23: end if
```

Bloom filters allow nodes to distinguish, among entries that are not in the cache, those that correspond to recognized items (stored in the Bloom filter), and not recognized items. This is important in case such items are encountered again, as, if they are in the Bloom filter, they can be immediately marked as recognized. Note that this algorithm mimics the way in which the human brain refreshes, flushes and recalls "items" in memory.

5. A MODIFIED *TAKE-THE-BEST* ALGORITHM FOR OPPORTUNISTIC NETWORKS

Having described how to implement the recognition heuristic, we now present an algorithm that exploits it in the data dissemination process. Also in this case, we take inspiration from the cognitive psychology literature. The *Take the Best* algorithm, defined in [Goldstein and Gigerenzer 1996], mimics a fast and frugal way of reasoning for choosing among two alternatives.

The goal of the algorithm is comparing two objects and infer which one has the higher value. To this end, objects are tested against an ordered set of cues, stopping at the first (**best**) cue that discriminates among them. Cues are tested in order of validity. A cue validity is defined with respect to the evaluation criterion. The first cues to be looked at are the ones that give a more discriminatory power with respect to the final evaluation criterion. When none of the cues can discriminate, the algorithm chooses by some additional discriminating criterion, which usually requires much more complex information to be evaluated with respect to the cues.

As typical for this kind of cognitive processes, the advantage of *Take-the-best* is that it needs only a few information in order to provide a decision. It relies on the first discriminating cue only, discarding all the other available information. Nonetheless, it is able to be very effective, since, like the recognition heuristic, it does not overfit existing data. Czerlinski et al. [Czerlinski et al. 1999] proved that *Take-the-best* is able to outperform a multiple regression model in predicting new events in 20 real-world problems, using an average of only 2.4 cues, in contrast with 7.7 cues of the other model. Moreover, Goldstein and Gigerenzer give an algorithmic description of *Take-the-best* [Goldstein and Gigerenzer 1996], making it an ideal candidate for defining an information selection strategy in an ICT context.

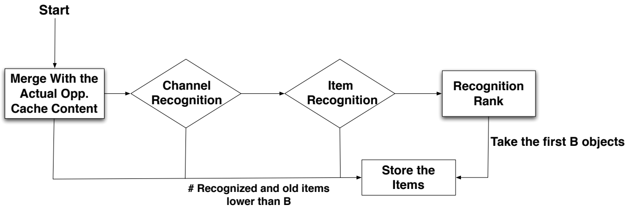


Figure 4: Modified Take The Best Algorithm

We propose to adapt this algorithm in the scenario of an opportunistic dissemination of information. In this scenario, each peer is not dealing with a selection between two alternatives only. Rather, it is presented a set of resources (*data items*), coming from an encountered node, that as to be stored in a limited memory space. As already detailed in the previous section, the goal is to maximize the utility of stored items, with respect to the information diffusion process. Thus, we want to exploit the *Take-the-best* algorithm in order to recursively create, by means of different cues, increasingly refined *consideration sets* of data items, proceeding till the first, best cue that is able to sort out a set of the required cardinality, i.e., small enough to be stored in the node's opportunistic cache.

Precisely, we detail how the proposed solution works, following its description in Algorithm 2. When a node meets another peer, they exchange summaries of the items they are carrying in their data caches. Items belonging to the node's subscribed channel are fetched and stored in the node's SC (lines 2–6). After that, each node gather the information about the remaining objects with the one of the content of its OC (lines 7–9) and rank this new set using an adaptation of the *Take-the-best* algorithm, as depicted in Fig. 4. In particular, since cues of *Take-the-best* are looked in order of validity, the first two cues we consider consist of the recognition of channels and items, in this order. Their recognition is based on the algorithm presented in Section 4. In this context, we exploit the recognition heuristic for creating consideration sets, similarly to the description given in Sec. 3.

Algorithm 2 Modified *Take-the-best* Algorithm

```
1: Let $S$ be a set of items received from another node;
2: for each $s \in S$ do
3:     if $s$.channel = subscribedChannel then
4:         $\mathsf{SC} \cup= s$
5:     end if
6: end for
7: Let $S' = S - \mathsf{SC}$
8: Let $B$ be the OC storage capacity limit
9: Let $I = S' \cup \mathsf{OC}$
10: Let $recChItems = \emptyset$
11: for each $i \in I$ do
12:     if $i$.channel is recognized then
13:         $recChItems \cup= i$
14:     end if
15: end for
16: Let $notSpreadItems = \emptyset$
17: if $recChItems$.size $> B$ then
18:     for each $r \in recChItems$ do
19:         if r is not recognized then
20:             $notSpreadItems \cup= r$
21:         end if
22:     end for
23:     if $notSpreadItems$.size $> B$ then
24:         Rank $notSpreadItems$ in ascending order w.r.t the counters of its items
25:         Select and keep in OC the first $B$ objects of $notSpreadItems$
26:     else
27:         $\mathsf{OC} \cup= notSpreadItems$
28:     end if
29: else
30:     $\mathsf{OC} \cup= recChItems$
31: end if
```

The first cue is the channel recognition. The node looks at the channels of the items to be selected. The ones belonging to recognized channels are ranked higher than the others and selected for the next steps (lines 10–15). Using the channel recognition for building the first consideration set permits to easily throw out entire classes of items, thus potentially being a first, strong pruning rule. If the total size of the set of remaining items (considering both the node and the peer shared storage spaces) is greater than B (the size of the node's opportunistic cache), items are further discriminated using the second cue (line 17). This is represented by the item recognition. As pointed out in the previous section, in this case the recognition assumes a *negative* meaning. Recognized items are ranked lower than the others, since they are considered to be already very spread in the network. Hence, they are not considered anymore. The second consideration set is then made of the items that are *not* recognized at this step (lines 18–22). Even in this case, the algorithm stops if there is enough space to store the old items already in OC and the new ones, contained in the last consideration set.

Although powerful, recognition-based rules could not be enough to obtain a sufficiently small consideration set. If further discrimination have to be carried out, in order to comply with a node's storage space constrain, the precise value of the estimated availability of items is considered. In other words, new items and the old ones in OC are considered together. The node looks at values of each item diffusion stored in its IC cache. Less available items are ranked higher and stored, while the others are dropped (lines 23–26). Note that estimated availability values are the very same used by the item recognition process. Thus, these values are already stored by the node and do not required the maintenance of any additional information. Moreover, since they come from the recognition process, they are derived from stimuli coming from the environment. Hence, they are also part of the ecological process of information gathering carried on by a node. As for the original *Take-the-best* Algorithm, not all the steps are required, and the last (and more costly) one is run only on a subset of the items. Clearly, this result depends from the recognition thresholds setting: a low R_c value and an high R allow channels to be recognized very rapidly (i.e. the channel recognition is effective only initially) while items become recognized later (i.e. the item recognition is effective only when a significant amount of time is already passed). A more strict filter is obtained the other way round (high R_c and low R). In the experiments that we show in Sec. 6, we found that the first two steps are sufficient for filtering the information in a proportion that goes from a lowest point with less restrictive thresholds of nearly 53% of the cases on a network of 200 nodes, up to 99% of the cases with more restrictive thresholds, for any network size.

As an example, consider the situation presented in Fig. 5. Two nodes, A and B, exchange information upon meeting. In particular, the left side of the figure shows the summary of data that A is passing to B. This summary includes all the shared information carried by A, i.e. the one stored in its IC, LI and OC caches. On the right side of the figure, node B is shown with the internal status of its CC and IC recognition caches and the content of its OC cache. In this example, we suppose that OC has a total of 3 total available slots. The dotted lines in both CC and IC mark the separation between recognized and unrecognized channel and items, respectively.

Starting to evaluate the received data summary, B applies the modified *Take-the-best* algorithm. It sees that the only recognized channels are Channel 3 and 4. The items of the other channels (4 out of 7) are then discarded.The first consideration set is formed by Items 3, 6 and 7. After that, the

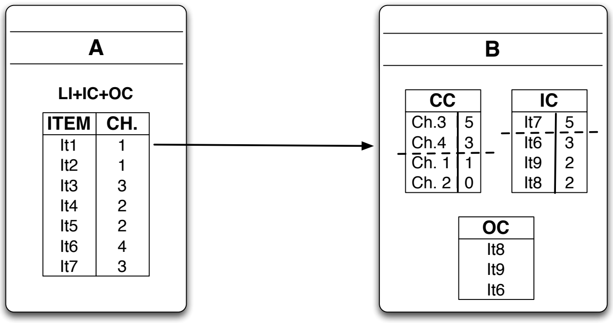


Figure 5: Example of information exchange with the modified *Take-the-best* Algorithm

node looks into its IC cache and find that Item 7 is already recognized as being too spread. After throwing it away, B has a second consideration set made of Items 3 and 6. After the two recognition steps, the majority of items (5 over 7) contained in the summary given by A has been pruned. The remaining items are merged with the content of OC, where Item 6 is already present. They are all ranked according to the diffusion values contained in IC. Items 3, 8 and 9 are ranked higher than 6, that is dropped. B can then ask to fetch Item 3 (the only missing one) to node A.

The algorithm allowed B to work on the data summary only, letting it to swiftly decide which were the more relevant items to keep into consideration. The final result is that only one item, considered as relevant, has to be passed from A to B, limiting also the load due to the exchange of real data items.

6. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed solution, we conducted a series of experiments in a simulatated scenario. In order to simulate real user movement patterns, nodes move in a 6 x 6 grid (1000 m wide), according to the HCMM model [Boldrini and Passarella 2010]. The HCMM model is a mobility model that integrates temporal, social and spatial notions in order to obtain an accurate representation of real user movements. In the simulation scenario, groups represent set of users that have social and spatial relationships. Groups are initially assigned to different home cells and any physical contact among groups is avoided. The only way to exchange and obtain data among groups is through node mobility. Nodes can move in the cell of their group only, with the only exception of a set of few nodes in each group, named *travellers*. Each traveller is allowed to visit just one of the other groups. Hence, travellers are the bridge that allow the flow of information between different communities. This model well represents social communities, in which people typically stay, with a few people commuting between different communities due to different social relationships [Boldrini and Passarella 2010]. Specifically, in our simulations setting, each group has one traveller for each of the other groups.

In order to assert the validity of the proposed solution, we first give a comparison of the system we designed with another, more tradinational data dissemination scheme for opportunistic networks. In particular, we tested our system against ContentPlace [Boldrini et al. 2010]. ContentPlace (described in more details in Sec. 2) approaches the problem of content dissemination in opportunistic network trying to achieve local optimal soultions for a distributed knap-sack problem. In order to compare these two systems, we tested our solution under the very same conditions proposed in the original ContentPlace paper, summarized in Table 1.

Parameter	*Value*
Node speed	Uniform in [1,1.86] m/s
Transmission range	20 m
Simulation Area	1000 x 1000 m
Number of cells	4x4
Number of Nodes	45
Number of Channels	3
Number of Items	297 (99 per channel)
Number of Groups	3
Number of travellers	6 (2 per group)
Simulation Time	50000s

Table 1: Comparison experimental parameters

In the ContentPlace simulation scenario nodes are grouped into 3 different communities, each containing the same fraction of the global number of peers. The data items available in the network are assigned to channels. There are 3 channels (as many as groups) with 99 items each (297 items in total). Items are uniformely distributed among nodes in the network and are all generated at the start of the simulation. Each node subscribes to one channel only at the beginning of the experiment. NodesâĂŹ subscriptions are distributed according to a Zipf law (with parameter 1) within each group. Moreover, interests are rotated, so that the most popular channel in a group is the second in another and the third one in the other, and so on. In Fig. 6 the results of this comparison are reported. We computed the performance of the two systems at various time instants after the simulation starts, using a performance figure, the *Hit Rate.* It is defined as the mean ratio between the number of retrieved objects of the subscribed channel and the total amount of objects of the channel. Results in the figure are obtained as the average result of 10 repeated simulations.

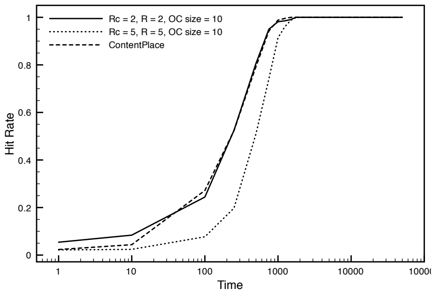


Figure 6: Recognition-based data dissemination vs. ContentPlace

From these results we can observe that the simulation setting with $R_c = 2$ and $R = 2$ allows our recognition-based solution to perform almost the same as ContentPlace. Note that the values of the recognition thresholds ($R_c = 2, R = 2$) have contrasting effects. The lower a channel recognition threshold, the faster the associated channel is recognized and its items start to circulate. On the other hand, the lower an item recognition threshold, the faster items start to be recognized and, thus, excluded from further replication in the network.

Having seen that a solution based on embedding cognitive heuristic schemes in mobile devices is able to perform exactly like one, more tradtional and complex solution in this field, we can check whether a recognition-based approach satisfies the basic principles of heuristics: fast and frugal. To this end, we show in Fig. 7 the number of messages exchanged in all the network during the simulation by ContentPlace and the best recognition solution presented in the previous figure.

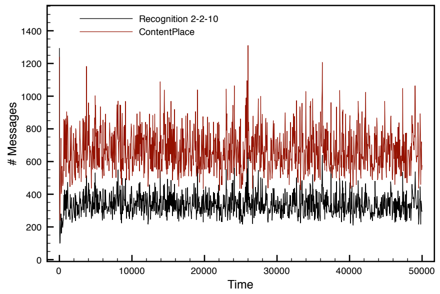


Figure 7: Number of messages exchanged by Recognition-based data dissemination and ContentPlace

It easy to see that, generally, ContentPlace exchanges twice the number of messages required by the system we designed. In addition, we also checked the number of items exchanged with the above messages. Fig. 8 plots the evolution of the total number of exchanged items over time. Exchanged items include both data summaries and data items.

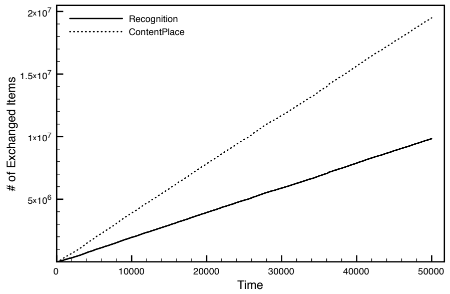


Figure 8: Number of items exchanged by Recognition-based data dissemination and ContentPlace

Again, the number of items exchanged by all the nodes using the recognition heuristic is lower than half the items exchanged by ContentPlace.

These preliminary results indicate that a data dissemination scheme based upon the recongition heuristic and the *Take-the-Best* algorithm can perform equally well (in terms of Hit Rate) like another state-of-the-art solution on this topic, while requiring less than half the messages and the number of items to be exchanged. This demand less effort, and thus less resource in order to achieve the very same result in term of Hit Rate.

Starting from these findings, we now want to test the behaviour of the recognition-based system in more challenging conditions. Since data items can reach interested users in communities other than those where they are generated only through nodes mobility, in order to highlight the effectiveness of the data dissemination algorithm we want to study its performance in a more complex scenario, with more nodes and channels. In the following experimental settings, summarized in Table 2, there are 8 channels with 25 items each (200 items in total). As in the previous setting, the items of each channel are uniformly generated inside all groups at the start of the simulation. Also in this case, node interests follow a Zipf law with parameter 1 inside each group, with the channel popularities rotated among all groups.

In the following graphs average values and 95% confidence intervals are computed by conducting 10 simulations of each scenario with different random seeds. Each simulation runs for 25,000 seconds.

Parameter	*Value*
Node speed	Uniform in [1,1.86] m/s
Transmission range	20 m
Simulation Area	1000 x 1000 m
Number of cells	6x6
Number of Nodes	from 200 to 600
Number of Channels	8
Number of Items	200 (25 per channel)
Number of Groups	8
Number of travellers	56 (7 per group)
Simulation Time	25000s

Table 2: Experimental parameters

6.1 Homogeneous scenario

In order to have a deep understanding of the interplay of the various parameters and of the conditions under which the proposed solution is able to perform at best, we show here various experimental results. They are obtained with different values of the main variables involved in the model and with different simulation settings, like the presence of churning nodes, or the sudden insertion of a new channel, with all its related data items.

We start by presenting the results coming from the experiments conducted with all the simulation setting described at the beginning of this section. Unless otherwise stated, figures are presented with a log scale in the x axis.

Fig. 9 and 10 show the temporal evolution of the Hit Rate with different values of the network size N, while, for the same experiments, Table 3 presents the final *convergence time* for the different network sizes. The convergence time is defined as the time at which the Hit Rate reaches a value above 0.995. Fig. 9 presents the result obtained by setting the maximum available space in the OC cache to 10 slots, while Fig. 10 presents the same figures with a OC cache of 50 slots. Thresholds for channel and item recognition, in

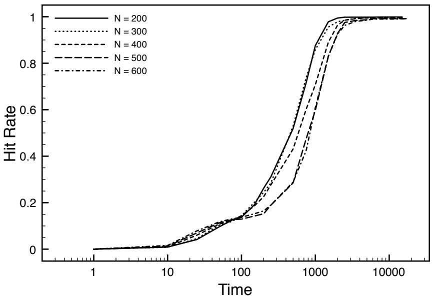


Figure 9: Hit Rate with Variable Network Sizes, OC size = 10

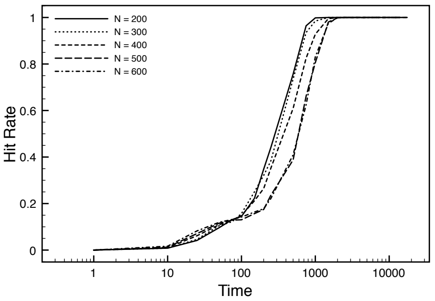


Figure 10: Hit Rate with Variable Network Sizes, OC size = 50

these cases, are both fixed to a value of 2. Both the evolution of the Hit Rate and the convergence time are influeced by the time that elapsed between two successive encounters and the avarage time needed by a traveller to get from one group to another. The first quantity influences the spreading of information within each community, while the second value impacts the flowing of data from one community to another. The first value depends on the community size, and varies from an avarage of 4.43 sec., with 200 nodes, to 1.73 sec. with 600 nodes. The second value depends only on the placement of groups within the simulation area. Since this data does not change, the mean time needed by a traveller to go from one community to another is about 113 sec. for any configuration.

Net. size	*Convergence Time*	
	OC size = 10	OC size = 50
200	2500	1000
300	3500	1500
400	3500	1500
500	6000	2000
600	6500	2000

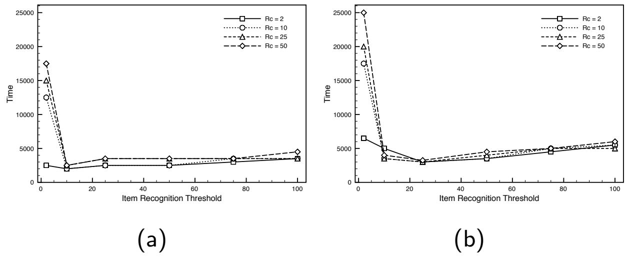
Table 3: Hit Rate convergence time for diffenrent OC and network sizes

The first thing to note is that, with all the network sizes, the Hit Rate reaches the maximum very rapidly. Almost all the items are deliverd to their interested users within the first 6500 seconds (with an opportunistic cache with 10 slots) and within 2000 seconds for an OC with 50 slots. The other relevant fact to observe is that this "convergence speed" is incremented by incrementing the Opportunistic Cache size. This is an expected result, since a greater cache give the chance to more items to be circulated among nodes.

These results highlight the impact of the OC and network sizes to the dissemination process. We wish now to give a more detailed view of the impact of the recognition threshold to data diffusion process. In particular, the convergence time is faster in smaller networks. In fact, the smaller the network (and thus, each community), the higher the time that elapses from one encounter to another. Since each node in a community of a large network makes a high number of meetings, items become recognized more rapidly. Thus, they are not considered for inclusion in the OCs of that groups and, as a consequence, it is more difficult for travellers to fetch and take them to other communities, fovoring the information dissemination process.

Fig. 11 presents the variations of the convergence time as a function of the value of R, by fixing R_c and an OC with 10 slots, while Fig. 12 presents the results obtained with an OC with 50 slots. Results related to a network with 200 nodes are reported in Fig. 11(a) and 12(a), while Fig. 11(b) and Fig. 14(c) show the results obtained with 600 nodes. The numerical results of these subfigures are reported in Table 4 (OC size=10) and Table 5 (OC size=50), respectively.

Looking at all these reults, it is easy to note that the convergence time curves have a point of minimum. For all the tested values of R_c, when $R = 10$, for 200 nodes, and $R = 25$, for 600 nodes, the information diffusion process is generally faster than with other values of R. On the other hand, with $R = 2$ the system usually has its worst convergence time. We can deduce that with the lowest value of R, items are recognized too rapidly and, hence, have less chances to be exchanged between nodes using the opportunistic mechanism. As a result, the final convergence time is higher.



(a)

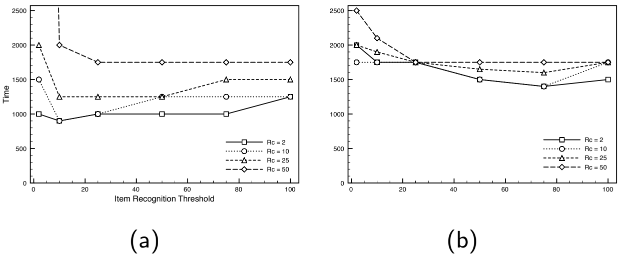
(b)

Figure 11: Hit Rate convergence time as a function of R for OC size=10 and for 200 (a) and 600 (b) nodes

On the other hand, a proper value of R (10 for 200 nodes, 25 for 600) allows the system to achieve an optimal tradeoff between the need to let items circulate and the necessity of limiting the diffusion of already spread items. We can say that the results show the existence of a value of R that maximizes the *recognition validity* of the item recognition heuristic applied by the system. We can also deduce that this optimal value varies according to the network size.

One other thing that intuitively we could expect is that, with higher values of R_c, the information diffusion is slower.

In fact, channels are recognized later and, as a consequence, their items could be spred more slowly. Generally, the results we have seem to go in this direction. Anyway, the interplay between the channel and the item recognition thresholds could lead to different results than expected. As highlight previously, when items start to circulate very early, they become recognized as too widespread faster. As a confirmation of this intuition, the results for a network with 600 nodes show, for $R = 10$, that the lowest value of R_c lead to a slower convergence time than the other values of this parameter.



(a) (b)

Figure 12: Hit Rate convergence time as a function of R for OC size=10 and for 200 (a) and 600 (b) nodes

6.2 Channels with different popularities

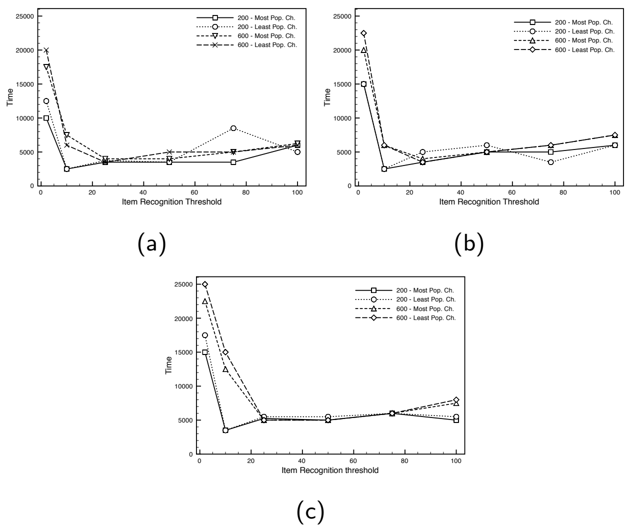
All the results presented so far are realized in a somewhat “homogeneous” scenario. In fact, note that despite the subscriptions inside each group are assigned according to a Zipf distribution, popularities are rotated in such a way that the most popular channel in a group is the second one in another, and so on. As a result, all the channels have the same overall number of subscriptions. Moreover, since each group has a traveller to each of the others, all the channels are in tha same conditions for achieving the final convergence.

In the following set of experiments, we want to study the behaviour of the proposed approach in a more challenging, unbalanced scenario. By keeping the other parameters unchanged, in this context, the *global* channel subscriptions are assigned at random to nodes according to a Zipf distribution of parameter 1. Thus, the first channel is the *most popular* one, while the eighth channel is the one with the least number of subscribed nodes. We want to study how the final convergence times of channels with different popularities are affected by the parameters of the cognitive-based solution we propose.

These results are of particular interest, since we wish to avoid that the system is saturated only with the content of the most popular channels, risking that the diffusion of less popular channels get stuck.

Results reported in Fig. 13 show the converge time as function of R for three different values of R_c: 2, 10 and 50. respectively. The figures show the convergence times for the most and least popular channels with networks made up of 200 and 600 nodes. The OC size is fixed to 10 slots. Fig. 14 shows the results obtained with using an OC with 50 slots and keeping the other parameters unchenged, with respect to the previous experiment.

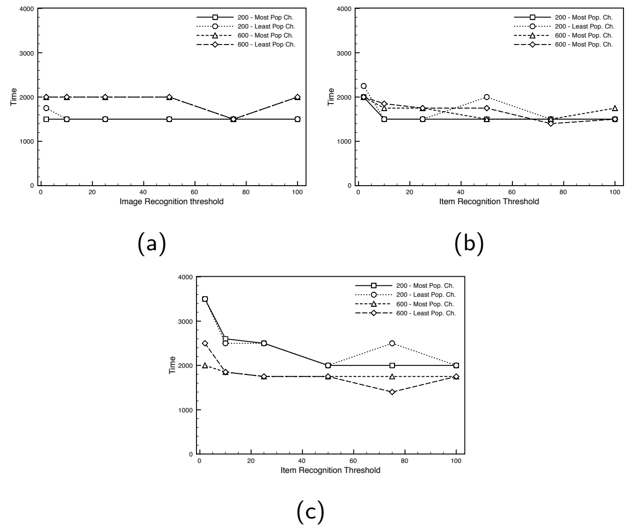
The first relevant thing to note is that convergence times for the most and least popular channels tend to be very similar, or even the same, in all the reported results. Note



(a) (b)

(c)

Figure 13: Hit Rate convergence time as a function of R for OC size =10 and $R_c = 2$ (a), 10 (b), and 50 (c)



(a) (b)

(c)

Figure 14: Hit Rate convergence time as a function of R for OC size =10 and $R_c = 2$ (a), 10 (b), and 50 (c)

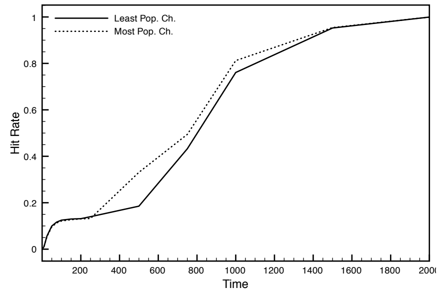


Figure 15: Hit Rate with an Item threshold = 10

that having the same convergence time does not imply that the slopes of the corresponding Hit Rate curves are the same. As an example, Fig. 15 shows the Hit Rate curves for the most and least popular channels with $R_c = 10.R = 10$ and with 10 slots in each OC (x axis has a linear scale).

The most popular channel has an initial faster diffusion and the gap with the least popular channel initially tend to increase. Anyway, when items of the most popular channel begin to be sufficiently spread in the network, the system adaptively change the priorities for fetching items in each node's OC. As a consequence, the dissemination of the least popular channel is speeded up. As time passes, the gap between the two channels decreases and, at the end, they converge at the same time. This adaptive behaviour of the system can be seen with all the parameters we used.

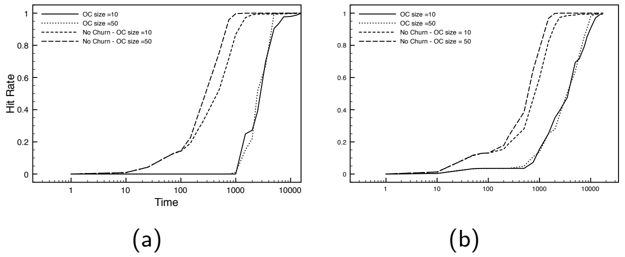
Another thing to note is that the system maintains a behaviour that is similar to the one showed in the "homogenous" scenario. In fact, the convergence times generally have a point of minimum. This behaviour is more evident with the lowest OC size, where the minimum is reached at the same values of R of the homogenous case. A biggest OC size give to the system the possibility to speed up the convergence time for both the most and least popular channel. As for the convergence time, this last size of OC is big enough to flatten the differences between channel popularities and the impact of the item recognition threshold.

It is interesting to observe that, apart for the lowest values of R, convergence times of a network with 200 nodes and one with 600 peers are very close, or the same. With an opportunistic cache size of 50 slots and R_c, they are even always better. We can infer that the combined effect of the delay introduced by such a high value of R_c and the high dimension of the OCs make the information diffusion process easier when the node density is higher.

6.3 Churning nodes

We now explore how the system behaves under other, more challenging conditions. In the next set of experiments we present the outcomings of the system performance in a scenario of churning nodes. The overall settings of this scenario are homogenous, i.e. channels popularities are rotated among groups and items are initially uniformely distributed among communities. In this environment, every 5 seconds, each node has a probability to deactivate itself. This means that, altough it continues to move inside the simulation area, it neighter distributes nor receives any information to/from the other nodes. Anyway, it does not delete the information collected so far. Deactivated nodes have a probability to reactivate and re-join the information dissemination process, starting from the situation they had before deactivating.

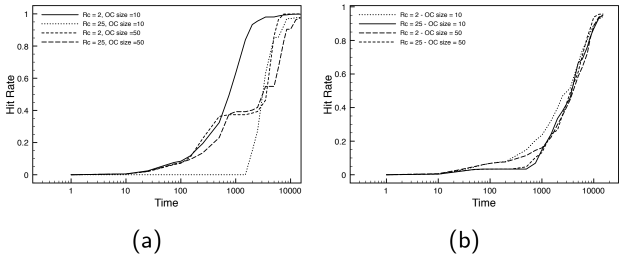
In the following, we show results obtained with a deactivation probability of 0.5 and a re-activation probability of 0.5. Thus, on average, only half of the nodes are active.



(a) (b)

Figure 16: Hit Rate trends with a deactivation probability=0.5 and various OC sizes for 200 (a) and 500 (b) nodes

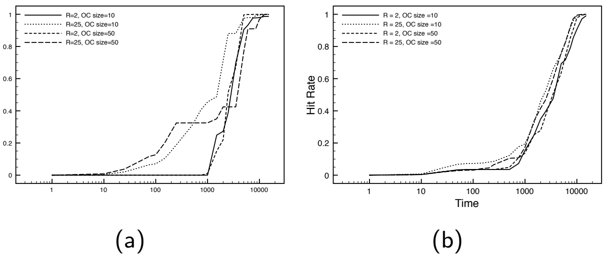
In Fig. 16 we show the results regarding the impact of the opportunistic cache size in this scenario. The values of R_c and R are both set to 2. We can see that convergence times are greatly delayed. This is expected, since only half of the nodes on avarage are active. One thing to note is that the start of the diffusion process is delayed with respect to a scenario with non-churning nodes. The impact of churning nodes is more relevant when items are still replicated on few nodes of the network. As a result, the initial diffusion of items is generally delayed. In accordance with the nodes' behaviour, the Hit Rate trends are more irregular. The size of the opportunistic cache seems to give little advantage in the item diffusion process. Anyway, bigger OCs still perform slightly better.



(a) (b)

Figure 17: Hit Rate trends with a deactivation probability=0.5 and varying R_c for 200 (a) and 500 (b) nodes

With respect to the values of the channel recognition threshold, Fig. 17 shows the Hit Rate progression over time when changing this parameter. Note that with 200 nodes, the results exhibit more instability. This can be ascribed to the fact that a single deactivating node counts proportionally more in a smaller network rather than in a bigger one. As expected, by fixing the value of R, lower values of the channel threshold favor a more rapid information diffusion. Differences are more evident with 200 nodes, while for 500 nodes the advantage given by the channel recognition threshold holds only initially, while, when approaching the convergence point, different settings lead to no particular differences.



(a)

(b)

Figure 18: Hit Rate trends with a deactivation probability=0.5 and varying R for 200 (a) and 500 (b) nodes

Similarly to the results of the previous figure, experiments on the item recognition threshold reported in Fig. 18 show more instability in the trends of the Hit Rate for 200 nodes, while for 500 nodes all parameters have fewer effects on the final result. For both the network sizes, the system preserves the same behaviour of the homogeneous case. In fact, with a value of $R = 2$ performances are always worse than those obtained wiht $R = 25$.

6.4 Insertion of a new channel

In the last set of experiments, we study the impact of suddenly introducing new items associated to a new channel in the network. In order to perform this set of experiments we assume that, at a given instant in time, a new channel appears in the network. It has exactly the same number of objects of all the other channels and it takes a randomly chosen degree of popularity. Anyway, it has the same popularity in all the groups. Accordingly, a required number of (randomly chosen) nodes un-subscribe from their previous channels and subscribe to the new one. Thus, they remove older items from their SC caches, since they no more correspond to the nodes' subscriptions. At this time, nodes checks whether those items can enter the OC cache instead. Then, the usual recognition-based information dissemination process starts to be apllied also to the new channel and its items. In the experiments, the new channel is inserted when all the other have reached or are very close to reach the convergence.

		Hit Rate						
R	R_c	1500s	2000s	2500s	3500s	5000s	7500s	10000s
2	2	0	0.396	0.856	0.970	0.979	0.983	1.0
	25	0	0.476	0.886	0.955	0.994	0.999	1.0
25	2	0	0.489	0.834	0.920	0.949	0.958	1.0
	25	0	0.359	0.805	0.876	0.949	0.962	1.0

Table 6: Hit Rate trends for a newly injected channel – 200 nodes, OC size=10

		Hit Rate						
R	R_c	5000s	5500s	6000s	7000s	7500s	8500s	10000s
2	2	0	0.424	0.825	0.983	0.989	0.992	1.0
	25	0	0.403	0.791	0.985	0.987	0.995	1.0
25	2	0	0.428	0.898	0.997	0.999	1.0	1.0
	25	0	0.245	0.660	0.976	0.991	1.0	1.0

Table 7: Hit Rate trends for a newly injected channel – 600 nodes, OC size=10

Tables 6 and 7 show the numerical results for 200 and 600 nodes respectively. The new channel is inserted at sec. 1500 for 200 nodes and at sec. 5000 for 600 nodes. The results are obtained using an OC size of 10 slots, in association with two different values for both the channel the item recognition thresholds. Since all the other channels are near their convergence, the diffusion of the items of the new one can exploit almost all the available OCs, thus obtaining a very quik diffusion. Then, the system shows a very good reactivity to the sudden injection in it of new, previously unseen items, associated with a newly created topic of interest.

7. CONCLUSIONS

In the Future Internet scenario, devices will act as proxies of their users in a very crowded information landscape. These devices will need efficient mechanisms to select the most relevant information for their users and for a collaborative exchange of information. In this paper, we proposed a new model for trying to directly embed in an ICT system the rules and procedures for content selection applied by the human brain. These rules are known as *cognitive heuristics* . They are models of how the human brain assesses the relevance of information using only partial knowledge of the problem space and very limited resources. In this paper we present how to exploit these models (already established and coded in the cognitive psychology field) to drive data dissemination processes in an opportunistic networking environment.

In particular, we focus on one of the most simple and efficient cognitive heuristic, i.e. the *recognition* heuristic. The recognition heuristic discriminates objects with respect to a given criterion, without requiring to collect all the information needed to exactly compute the criterion. It assumes that recognized objects have higher value (with respect to the criterion) than non-recognized objects, and discriminates among them accordingly. We have shown how the recognition heuristic can be implemented by nodes of an opportunistic network. Then, we show how nodes can efficiently combine multiple instances of the recognition heuristic to assess the relevance of available data objects, thus deciding what to store and what to drop. This selection is based based on a variation of the cognitive science *Take the best* algorithm, also originally proposed in the cognitive psychology literature.

Simulation results show that an information dissemination system based on cognitive heuristics is able to achieve the same performance as a more complex, state-of-the-art algorithm, while needing less then half of the resources. Moreover, we tested our solution in other complex scenarios, with an increasing number of nodes in the network. In a network where subscriptions to topics of interest are distributed unevenly among nodes, the system is able to balance the diffusion of the most and least popular channels, leading them to converge at the same time. Other results show the ability to adapt its behaviour and promptly react to the presence of churning nodes and the sudden insertion of new channels. Results show that a correct tuning of the heuristic parameters has to be evaluated in order to let the system achieve its best performance.

In order to further explore the potential of this solution, key topics for future research include the developement of analytical models that allow to formally understand the impact and the interplay of the parameters. Moreover, we wish to investigate how the proposed data dissemination works when additional context information (such as social relationships between users) is exploited. Furthermore, it will also be interesting to understand whether it is possible, in this context, to define an equivalent of the cognitive "adaptive toolbox". In particular, it could be interesting to know whether other heuristics (beyond recognition) can be effectively applied to the data dissemination and how they can be exploited in conjuction with the recognition heuristic.

8. ACKNOWLEDGEMENTS

This work is funded by the EC under the FET-AWARENESS RECOGNITION Project, grant 257756.

9. REFERENCES

[Alba and Chattopadhyay 1985] Alba, J. and Chattopadhyay, A. 1985. Effects of context and part-category cues on recall of competing brands. *Journal of Marketing Research*, 340–349.

[Balamash and Krunz 2004] Balamash, A. and Krunz, M. 2004. An overview of web caching replacement algorithms. *IEEE Communications Surveys and Tutorials 6,* 1-4, 44–56.

[Boldrini et al. 2010] Boldrini, C., Conti, M., and Passarella, A. 2010. Design and performance evaluation of contentplace, a social-aware data dissemination system for opportunistic networks. *Comput. Netw. 54*, 589–604.

[Boldrini and Passarella 2010] Boldrini, C. and Passarella, A. 2010. Hcmm: Modelling spatial and temporal properties of human mobility driven by users' social relationships. *Comput. Commun. 33*, 1056–1074.

[Conti et al. 2011] Conti, M., Mordacchini, M., and Passarella, A. 2011. Data dissemination in opportunistic networks using cognitive heuristics. In *WOWMOM.* IEEE, 1–6.

[Czerlinski et al. 1999] Czerlinski, J., Gigerenzer, G., and Goldstein, D. 1999. How good are simple heuristics. *Simple heuristics that make us smart*, 97–118.

[Dawes 1979] Dawes, R. 1979. The robust beauty of improper linear models in decision making. *American psychologist 34,* 7, 571.

[DeMiguel et al. 2009] DeMiguel, V., Garlappi, L., and Uppal, R. 2009. Optimal versus naive diversification: How inefficient is the 1/n portfolio strategy? *Review of Financial Studies 22,* 5, 1915–1953.

[Erdfelder et al. 2011] Erdfelder, E., Küpper-Tetzel, C., and Mattern, S. 2011. Threshold models of recognition and the recognition heuristic. *Judgment and Decision Making 6,* 1, 7–22.

[Evans and Over 2010] Evans, J. and Over, D. 2010. Heuristic thinking and human intelligence: a commentary on Marewski, Gaissmaier and Gigerenzer. *Cognitive Processing 11*, 171–175.

[Gigerenzer 2004] Gigerenzer, G. 2004. Fast and frugal heuristics: The tools of bounded rationality. *Blackwell handbook of judgment and decision making*, 62–88.

[Gigerenzer 2008] Gigerenzer, G. 2008. Why heuristics work. *Perspectives on Psychological Science 3,* 1, 20–29.

[Gigerenzer and Goldstein 2011] Gigerenzer, G. and Goldstein, D. 2011. The recognition heuristic: A decade of research. *Judgment and Decision Making 6,* 1, 100–121.

[Gigerenzer and Goldstein 2002] Gigerenzer, G. and Goldstein, D. G. 2002. Models of ecological rationality: The recognition heuristic. *Psychological Review 109,* 1, 75–90.

[Gigerenzer and Todd 1999] Gigerenzer, G. and Todd, P. 1999. Fast and frugal heuristics: The adaptive toolbox.

[Goldstein and Gigerenzer 1996] Goldstein, D. G. and Gigerenzer, G. 1996. Reasoning the fast and frugal way: Models of bounded rationality. *Psychological Review 103,* 4, 650–669.

[Goldstein and Gigerenzer 1999] Goldstein, D. G. and Gigerenzer, G. 1999. The recognition heuristic: How ignorance makes us smart. In *Simple Heuristics That Make Us Smart*, G. Gigerenzer and P. M. Todd, Eds. Oxford University Press, 37■–58.

[Goldstein and Gigerenzer 2009] Goldstein, D. G. and Gigerenzer, G. 2009. Fast and frugal forecasting. *Int. Journal of Forecasting 25*, 760–772.

[Hauser and Wernerfelt 1990] Hauser, J. and Wernerfelt, B. 1990. An evaluation cost model of consideration sets. *Journal of Consumer Research*, 393–408.

[Jacoby and Brooks 1984] Jacoby, L. and Brooks, L. 1984. Nonanalytic cognition: Memory, perception, and concept learning. *The psychology of learning and motivation: Advances in research and theory 18*, 1–47.

[Johnson and Goldstein 2003] Johnson, E. and Goldstein, D. 2003. Do defaults save lives? *science 302,* 5649, 1338.

[Laroche et al. 2003] Laroche, M., Kim, C., and Matsui, T. 2003. Which decision heuristics are used in consideration set formation? *Journal of Consumer Marketing 20,* 3, 192–209.

[Lenders et al. 2008] Lenders, V., May, M., Karlsson, G., and Wacha, C. 2008. Wireless ad hoc podcasting. *SIGMOBILE Mob. Comput. Commun. Rev. 12*, 65–67.

[Marewski et al. 2010a] Marewski, J., Gaissmaier, W., and Gigerenzer, G. 2010a. We favor formal models of heuristics rather than lists of loose dichotomies: a reply to Evans and Over. *Cognitive Processing 11*, 177–179.

[Marewski et al. 2010b] Marewski, J. N., Gaissmaier, W., and Gigerenzer, G. 2010b. Good judgments do not require complex cognition. *Cogn. Process 11*, 103–121.

[Marewski et al. 2010] Marewski, J. N., Gaissmaier, W., Schooler, L. J., Goldstein, D. G., and Gigerenzer, G. 2010. From recognition to decisions: Extending and testing recognition-based models for multialternative inference. *Psychonomic Bulletin & Review 17,* 3, 287–309.

[Monti et al. 2009] Monti, M., Martignon, L., Gigerenzer, G., and Berg, N. 2009. The impact of simplicity on financial decision-making. In *Proc. of CogSci 2009, July 29 - August 1 2009, Amsterdam, the Netherlands.* The Cognitive Science Society, Inc., 1846–1851.

[Oppenheimer 2003] Oppenheimer, D. 2003. Not so fast!(and not so frugal!): Rethinking the recognition heuristic. *Cognition 90,* 1, B1–B9.

[Passarella 2012] Passarella, A. 2012. A survey on content-centric technologies for the current internet: Cdn and p2p solutions. *Computer Communications 35,* 1, 1 – 32.

[Pelusi et al. 2006] Pelusi, L., Passarella, A., and Conti, M. 2006. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *Communications Magazine, IEEE 44,* 11, 134 –141.

[Reich and Chaintreau 2009] Reich, J. and Chaintreau, A. 2009. The age of impatience: optimal replication schemes for opportunistic networks. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies.* CoNEXT '09. ACM, New York, NY, USA, 85–96.

[Schooler and Hertwig 2005] Schooler, L. and Hertwig, R. 2005. How forgetting aids heuristic inference. *Psychological Review 112,* 3, 610.

[Serwe and Frings 2006] Serwe, S. and Frings, C. 2006. Who will win wimbledon? the recognition heuristic in predicting sports events. *J. Behav. Dec. Making 19,* 4, 321–332.

[Shocker et al. 1991] Shocker, A., Ben-Akiva, M., Boccara, B., and Nedungadi, P. 1991. Consideration set influences on consumer decision-making and choice: Issues, models, and suggestions. *Marketing letters 2,* 3, 181–197.

[Simon 1955] Simon, H. 1955. A behavioral model of rational choice. *The quarterly journal of economics 69,* 1, 99.

[Simon 1990] Simon, H. 1990. Invariants of human behavior. *Annual review of psychology 41,* 1, 1–20.

[Todd and Gigerenzer 2003] Todd, P. and Gigerenzer, G. 2003. Bounding rationality to the world. *Journal of Economic Psychology 24,* 2, 143–165.

[Whitbeck et al. 2011] Whitbeck, J., Amorim, M., Lopez, Y., Leguay, J., and Conan, V. 2011. Relieving the wireless infrastructure: When opportunistic networks meet guaranteed delays. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a.* IEEE, 1–10.

[Whittlesea 1993] Whittlesea, B. 1993. Illusions of familiarity. *Journal of Experimental Psychology: Learning, Memory, and Cognition 19,* 6, 1235.

[Yin and Cao 2006] Yin, L. and Cao, G. 2006. Supporting cooperative caching in ad hoc networks. *IEEE Trans. Mob. Comput. 5,* 1, 77–89.

[Yoneki et al. 2007] Yoneki, E., Hui, P., Chan, S., and Crowcroft, J. 2007. A socio-aware overlay for publish/subscribe communication in delay tolerant networks. In *MSWiM.* 225–234.

		Convergence Time					
Net. size	R_c	$R = 2$	$R = 10$	$R = 25$	$R = 50$	$R = 75$	$R = 100$
200	2	2500	2000	2500	3000	3000	3500
	10	12250	2000	2500	2500	3500	3500
	25	15000	2500	3500	3500	3500	3500
	50	17500	2500	3500	3500	3500	4500
600	2	6500	5000	3000	3500	4500	5500
	10	17500	3500	3000	3500	5000	5500
	25	19750	3500	3000	4000	5000	5000
	50	25000	4000	3250	4500	5000	6000

Table 4: Hit Rate convergence time for OC size=10 and for 200 and 600 nodes

		Convergence Time					
Net. size	R_c	$R = 2$	$R = 10$	$R = 25$	$R = 50$	$R = 75$	$R = 100$
200	2	1000	900	1000	1000	1000	1250
	10	1500	900	1000	1250	1250	1250
	25	2000	1250	1250	1250	1500	1500
	50	15000	2000	1750	1750	1750	1750
600	2	2000	1750	1750	1500	1400	1500
	10	1750	1750	1750	1500	1400	1750
	25	2000	1900	1750	1650	1600	1750
	50	2500	2100	1750	1750	1750	1750

Table 5: Hit Rate convergence time for OC size=50 and for 200 and 600 nodes