Cooperative Negotiation in a Multi-Agent System for Real-Time Load Balancing of a Mobile Cellular Network

John Bigham, Lin Du
Department of Electronic Engineering
Queen Mary, University of London
London, United Kingdom

ABSTRACT
A cooperative negotiation approach for the real-time control of cellular network coverage is described. The performance of the whole cellular network is improved by contracting and shaping the antenna radiation pattern around traffic “hot spots” and expanding adjacent cell coverage to fill in the coverage loss. The paper shows that the local area real time cooperative negotiation between base stations leads to a near global optimal coverage agreement which is reached in the context of the whole cellular network. Results showing the advantage of this technique are presented. Global optimization using constrained real-coded genetic algorithms provides the benchmark. In our work, instead of using a formal negotiation model of alternating offers, we create certain number of possible local hypotheses and start negotiations based on them. Some negotiation may reach agreements before their deadlines, and the system commits to the best agreement found at the end. This approach is more predictable and controllable than the formal negotiation model. Architecturally the negotiation component described is part of the planning layer of an agent system for resource management in 3G wireless networks. This has a layered architecture with both planning and reactive components. The results are allowing the development of a novel geographic load-balancing scheme for cellular networks that intelligently changes cellular coverage according to the geographic traffic distribution in real time.

Keywords
Cooperative negotiation, Multi-agent systems, Load balancing, Real-time systems

Categories and Subject Descriptors
[Multi-Agent Systems]: Negotiation in Multi-Agent Systems

1. INTRODUCTION
A multi-agent system has been developed to manage the resources of a mobile network in real time. Some agents represent the different service providers in the market (brokers between the user and the network operator) and other kinds of agents represent network operators that manage the radio resource of different regions of the network. Radio resource is scarce and its management is increasingly important as mobile networks become more complex and new applications need more bandwidth. Whilst it is possible to provide improved performance by over-dimensioning networks with extra radio transmitters (base stations), this approach leads to higher prices passed onto the user, and increasingly has environmental issues. Such costs would be a heavy deterrent to the take-up of the new services under development.

Agent technology has been used in the management of telecommunications systems [1, 2, 3] to address the problem of distributed resource control and management, as well as to incorporate the “business dimension” with different service providers [4, 5]. Much of this work has dealt with fixed networks but the concepts can be extended to a wireless network. The extra dimension in a wireless network is the allocation of radio bandwidth to radio cells to avoid local congestion or degradation in QoS. Work by Bodanese [6, 7, 8] has resulted in a distributed resource allocation scheme for first generation mobile networks using intelligent agents: this scheme offers an efficient approach to resource allocation under moderate to heavy loads. Recently the use of intelligent agents for resource management of the radio resource of 3G networks (where the technology is much more complex) has been shown to have several advantages [9]. The autonomous service providers could intelligently select the “best” autonomous network management agent to carry each connection request; a network management agent could dynamically change its connection admission algorithm to preferentially support high value customers; and the network management agents could inter-operate with other network management agents in a decentralised manner without a significant reduction in performance compared to a completely centralised control algorithm.

This paper concentrates on a different aspect: using negotiation to give greater autonomy to individual base stations in the mobile network in deciding the shape of the area that
they cover. This autonomy gives an increase in flexibility to deal with new situations in traffic load, and to decrease the signalling overhead on the network. The results of local negotiation is shown to give network performance very close to that obtained by global assignment of connection requests to the base stations and so local negotiation is a viable approach to finding previously unusable capacity in the network. The gain in capacity depends on the demand; however, simulations of simple hot spot scenarios have shown that this can be 20% [10]. Given that power decreases (at least) as the square of the distance, then decisions regarding mobile connections at locations far away will usually have a negligible impact on decisions based on circumstances in the neighbourhood. So it is intuitively reasonable that local negotiation should be effective. However, it is important that no “holes” are created in the network coverage; no dramatic changes to previous assignments are made lest (too many) connected calls are disconnected; a notion of elasticity in the coverage shape is included lest the shapes become increasingly distorted; and that decisions can be made at the speed that the mobile traffic changes.

The deployment of such a system depends on antenna technology to support the generation of shapes output by the negotiation. This is feasible, at a high cost, by the use of fully adaptive antennas. The work described here has also led to the development of a much cheaper “semi-smart” antenna, which also promises to be incrementally deployable into existing cellular networks [11]. Measurements on experiments in antenna ranges have shown that the desired patterns can be generated with inexpensive components.

The negotiation mechanism described is a software component that belongs in the planning layer of the network provider resource agent within the architecture described. The negotiation updates the shapes every minute. The reactive components of the network provider resource agent are triggered at each call request. There can be many hundreds per minute. Switching to a new shape creates a new power distribution over the mobile network, which affects the decisions made by the components at the reactive layer. While each agent has its own interests, conflicts may arise when agents are trying to collaborate with each other to build a shared plan as they are using the same underlying radio resource. Negotiation is to be seen as a key technique to resolve any conflicts in both competitive and cooperative Multi-Agent Systems and this paper concentrates on a negotiation approach that exploits the constraints of a wireless network.

Section 2 describes the relevant parts of the agent system architecture. Section 3 describes the problem domain, load balancing in cellular networks. Section 4 describes the negotiation mechanism for resolving the problem. Section 5 gives the results for the local negotiation and the benchmark global optimisation and gives a comparison.

2. FUNCTIONAL ARCHITECTURE

Each agent has a multi-layered architecture sharing features of INTERRAP [12], Touring Machines [13] and the architecture described by Bodanese [8], who used reactive, local planning and cooperative layers. The local planning and cooperative layers are often considered in combination as “the planning component” of an agent. It is vital in managing individual connections that decisions are made in real time, so the reactive layer is designed for very fast response. It dumbly executes one or more actions, in response to incoming events from the environment, and each of these actions is associated with a policy that specifies it. These policies are formulated by the planning layers of the agents, and may be altered by parameterisation (tuning) or outright substitution of one policy for another.

When a user wishes to make a connection for a particular type of service (a voice conversation, for example) the User Agent (UA), which has a client interface in the user’s mobile terminal, contacts the Resource Agent of some chosen service provider (SPRA). The service provider may have been chosen manually by the user, via the terminal’s interface, or by gate-keeping intelligence within the UA. The SPRA then chooses an appropriate network provider resource agent (NPRA) to check if the call can be carried in the network. It should be noted that all of this activity takes place within the reactive layers of the agents mentioned, completely separate from negotiation or planning activity. The NPRA is responsible for controlling and managing the radio resource of its network provider, at the level of a group of cells controlled by a Radio Network Controller. Figure 1 depicts the internal architecture of the NPRA, and the information it receives when a connection is requested.

Figure 1: NPRA internal architecture

To handle incoming connection requests, the NPRA must perform at least two actions: assign a user to a base station to carry the connection (assignment), and check if the base station can carry the call (connection admission control - CAC). In addition, the NPRA may provide a fallback mechanism to permit connection of some services that might otherwise be blocked, probably because the QoS they require (in terms of bit rate) is not deliverable (exception handling). The exception handler considers whether the request should still be rejected or modified according to its current policy (perhaps reducing the QoS) and then resubmits.

Each of these actions (assignment, CAC, exception handling) is determined or affected by a separate policy formulated in the planning component of the NPRA. In the case of coverage shapes the directional (rather than uniform) distribution of power requires mobiles to use less power to transmit and hence create less interference than would otherwise be the case. This affects both the assignment and CAC components at the reactive layer.

If assignment and CAC are successful, the connection is
4. COOPERATIVE NEGOTIATION IN A MULTI-AGENT SYSTEM

Negotiation can be defined as a process of improving agreement on common viewpoints or plans through the structured exchange of relevant information [14]. There is considerable research on negotiation that has been carried out in the domains of Economics and Social Science as well as Artificial Intelligence [15, 16, 17, 18], but the time taken to negotiate is still considered as a major problem [19], especially if it is to be used in real-time systems.

In our work, instead of using a formal negotiation model of alternating offers [16], we create many possible local hypotheses by sampling from a probability distribution that generates high-utility requests first. Negotiation then starts on all of these hypotheses and some may reach agreement within the given time. The negotiation commits to the best agreement found at the end of the time period. Messages are used as the communication means between agents, and they are sent and received asynchronously. An extended version of a flexible algorithm described in [20] is used as the scheduling algorithm for the negotiation and commitment process in order to obtain high system efficiency and robustness at the same time. This approach is more predictable and controllable than the formal negotiation model, as described in detail later.

In this work, negotiation is used as an intelligent control technique to adjust the cell size and shapes dynamically and balance the traffic load over the whole cellular network. The cooperative coverage negotiation is triggered when the local traffic in any cell exceeds a certain threshold, as the heavily loaded base station has to shrink its coverage to reduce its utilization and negotiate with others for covering its coverage loss. By communicating with its own antenna agent and negotiating with relevant adjacent base station agents according to the current call traffic distribution, a base station agent can find an optimum local coverage in the context of the whole network. As an agreement is defined by the current traffic condition and other agents’ decisions, its validity has to be checked and mutual commitment needs to be performed.

In this section, the negotiation system structure is presented first. The messages involved in both negotiation and commitment are listed next. In order to give an overview of the negotiation and commitment process an example is presented. In the end of this section, the detailed negotiation and commitment process is described.

4.1 Negotiation System Structure

The negotiation system structure is shown in Figure 3.
Figure 4: (a) Cell numbering for a negotiation example; (b) The sequence of messages

performs the negotiations to satisfy the traffic changes and also the commitment to the mutually agreed pattern; the antenna agent performs coverage generation. The process of negotiation and the commitment to an agreed pattern will be explained after the message list below.

4.2 Message Definition

There are nine types of messages involved in the negotiation and commitment process, listed as follows.

Request For Help (RFH): The areas that need some help from adjacent base stations are stored in this message; it is the message that initiates a coverage negotiation.

Return From Base Station Agent (RFB): Message returned from adjacent base station agent, with the price to cover the areas in the previous RFH message. This message is used to end a coverage negotiation.

Request For Commitment (RFC): Message used to start the commitment process after negotiations reach some agreements.

Acknowledgment Of Commitment (AOC): Message answering the RFC message if the required hypothesis is still valid and the base station agent is not involved in other transaction.

Change Your Coverage (CYC): When the initial base station agent receives all the AOC messages, it sends CYC messages to relevant base station agents to change their coverage.

Change Your Pattern (CYP): When a base station agent receives a CYC message, it sends a CYP message to its associated antenna agent to change the coverage pattern physically.

Scheme Expires (SE): If the hypothesis has expired or the base station agent is currently in other transaction, a SE message is sent back to the RFC requester.

Cancel Transaction (CT): Message used to cancel an ongoing transaction.

Transaction Cancelled (TC): Message used to acknowledge the cancellation of the specific transaction.

4.3 Negotiation and Commitment Example

Suppose the traffic load in cell 45 in Figure 4(a) becomes too high; the intelligent geographic load balancing control will be triggered and the negotiation and commitment steps described below are executed, as shown in Figure 4(b).

When base station agent (BS45) detects its utilization exceeding the threshold $T$, it initiates a pattern negotiation. Its local optimizer first proposes a set of local coverage hypotheses and sorts them according to the same evaluation function. The number of the set is chosen large enough to give a good solution and small enough to allow a fast convergence. A value of ten has been found to be a good compromise. The negotiator creates RFH messages according to the coverage losses in each coverage hypothesis, and sends these RFH messages to relevant adjacent base station agents. Suppose that there is some coverage loss in the area near base station 46 and 36 (region A in Figure 4(a), the BS45 agent then sends two RFH messages to both BS46 and BS36, e.g. RFH(1). They have to propose some local coverage hypotheses that cover the losses of BS45 and if they have enough free resource (including free capacity and transmitting power), RFB messages will be returned immediately, like RFB(4) and RFB(5). However, if their free resources are not enough to help BS45, this process will involve further negotiation with other base station agents, like the RFH(11,12) from BS36 to BS27. After BS36 gets the required RFB, replies from other base station agents involved (if any), it chooses the best one and returns BS45 agent the scheme ID with an accumulatively calculated price by an RFB message.

Once the BS45 agent gets enough RFB replies from other base station agents involved, its executer (the component responsible for the commitment process) chooses the cheapest one, and starts the commitment by sending RFC messages to all the base station agents involved in this negotiation to commit their promised hypotheses. These base station agents, like BS46, check the validity of schemes and reply
to the original base station agent BS45 with the confirmation message, namely AOC. If BS45 receives all the AOC replies, it sends CYC messages to all the base station agents involved. Finally, base station agents send CYP messages to their own antenna agents, and the executers in antenna agents send the synthesized parameters to the antenna in order to change the real patterns.

After commitment, all the messages relative to a negotiation are discarded, and the states for involved agents are reset to “idle”. They are then ready for the further negotiations and commitments.

4.4 Negotiation Process

As seen in the previous example, whenever a base station agent observes its utilization exceeding the threshold $T$, it has to lose some coverage to reduce its utilization, and negotiations between adjacent base station agents are performed. If the helping base station agents in the immediate neighbours do not have enough free resource (capacity or transmitting power) the coverage negotiation may spawn further negotiations. Message loops are checked to eliminate potential deadlocks, and hop counters are used to avoid chained coverage negotiations going too far. The current hop limit is two. Since radio power decreases very quickly with distance, base station agents managing cells that are far apart are reasonably independent. This is also a factor to achieve timely responses of negotiations, so that most of the negotiation tasks can reach good agreements before their deadline.

As the coverage is no longer fixed, it is inadequate to model the area of coverage simply by a hexagonal cell. In order to realize the negotiation, coarse granularity of coverage patterns has to be used. We define the frontier of a base station as the maximum outreach of any of its antenna patterns (the envelope of all the possible antenna patterns). Any demand outside the frontier is of no direct interest to the base station, as it cannot service any of it. Within the frontier the area is divided into locations using polar coordinates, as shown in Figure 5(a), in order to avoid producing unrealizable patterns. Each location, named as Quantization Cell (QC), can contain many registered and connected mobiles. The hexagonal region represents the default, baseline pattern of coverage, and the shadowed part in the middle represents the forbidden zone (no mobiles in it).

Figure 5: (a) Grid in polar coordinates representing coverage patterns; (b) Two coverage hypotheses proposed by base station agent

Before the start of negotiations, the local optimizer proposes several sets of QCs to remove from current local coverage. Each set of QCs is called a removal hypothesis. A hypothesis is generated by randomly selecting some worst QCs as the candidates of loss from the current coverage pattern, with the probability proportional to a measure described below, until the prospective utilization is below the threshold $T$. Figure 5(b) shows two removal hypotheses A and B. A consists three QCs, and B of two.

Given that $QC_i$ is currently covered by base station $BS_j$, the probability of losing its coverage is proportional to how bad it is to the $BS_j$. If far away, $QC_i$ is less important to the $BS_j$ hence more likely to be selected as a removal candidate. If prospective helper $BS_k$ is lightly loaded, it has less difficulty to take over $QC_i$, and $QC_i$ is also more likely to be selected. More clearly, this can be formulated as the utility function $U(i, j)$,

$$U(i, j) = w_0 \cdot I(i, j) + w_1 \cdot D(i, k)$$

where $I(i, j)$ represents the importance of serving $QC_i$, and $D(i, k)$ represents the difficulty of the helper $BS_k$. $w_0$ and $w_1$ are weights. $I(i, j)$ and $D(i, k)$ can be expressed as

$$I(i, j) = (1/\text{Distance})^2$$

$$D(i, k) = 1/(\text{Capacity}_{BS_k} - \text{Load}_{BS_k} + 0.0001)$$

Where Distance is the distance from $QC_i$ to $BS_j$, $\text{Capacity}_{BS_k}$ is the capacity of the adjacent base station $k$ that is most likely to take over, $\text{Load}_{BS_k}$ is the current load of that base station. To avoid dividing by zero, a small value is added in denominator.

Therefore, we get the probability function, $P(i, j)$, representing the probability value of $QC_i$ to be selected as a removal by $BS_j$. It is the normalized form of $U(i, j)$ with exponential re-scaling of $\alpha$. The probability of $QC_i$ being selected as a removal candidate is inversely proportional to $U(i, j)$, hence proportional to $P(i, j)$,

$$P(i, j) = e^{\alpha \cdot (1 - U(i, j)/U_{\text{max}})}$$

As the traffic does not change drastically between control cycles, it is not necessary to evaluate all the locations within the coverage. Currently the local optimizer only looks at the 15% of QCs from the boundary. This avoids getting irregular patterns and makes the negotiation output more stable.

The local optimizer keeps proposing new coverage hypotheses up to an upper limit (currently ten as explained before) or until the maximum deformation is reached for the control interval. After this, it sorts the hypotheses by the control interval. After this, it sorts the hypotheses by $P(i, j)$, hence proportional to $P(i, j)$.

$$P_1(A, j) = \sum_{i=0}^{\text{all the losses}} [w_2 N(i, j) + w_3 R(i)]$$

where $N(i, j)$ is the number of calls being served by $BS_j$ in $QC_i$, $R(i)$ is the number of registered but silent mobiles, and $w_2, w_3$ are different weights. This is essentially the weighted number of switching users required by Hypothesis $A$, and should be kept at the minimum to avoid too much disturbance to the system. This is achieved by processing the hypothesis with the lowest local price first in this work.

The evaluator checks the coverage losses in the hypotheses, and the negotiator creates up to six 2 RFH messages

There are six immediate neighbours for each base station, which is the most common topology of cellular networks, as shown in Figure 4(a).
for possible losses and sends them to adjacent base station agents. Each RFH message may contain one or more coverage losses. To simplify the system, only the closest base station agent is selected as the helper for each request. The process of coverage negotiation is shown as Figure 6.

When the helper receives RFH messages, its local optimiser also proposes some local coverage hypotheses that arbitrarily cover the QCs stored in the RFH messages. If there is enough free resource (capacity and transmitting power) at the helper, RFH messages with zero prices will be returned back to the requester to finish this coverage negotiation. However, if there have to be coverage losses, further negotiations with the second level helpers (except the requester) will be initiated, and the prices are accumulatively calculated. Here, any loops in helper requests have to be detected so as to eliminate potential deadlocks, and a maximum hop limit (currently of two) is used to prevent the chained coverage negotiation from going too far. The price that the helper BS_h asks for the requester BS_j is defined as (5),

$$P_2(j,k) = \sum_{i=0}^{\text{all the losses}} [w_1 N(i,k) + w_5 R(i)] \quad (5)$$

where $N(i,k)$ is the number of calls being served by base station $k$ in location $i$, $R(i)$ is the number of registered but silent mobiles, and $w_1, w_5$ are different weights.

If any helpers need further help, it has to wait until it gets enough RFB replies (currently four out of six) from chained helpers; after that, it calculates the prices accumulatively and returns the total price to the initiator with another RFB message.

Finally, when the initiator receives enough RFB replies (currently six out of ten, including direct and chained replies), the negotiator picks up the cheapest solution according to the local price and helpers’ prices, and passes it to the executor for commitment. The reason for not waiting for all RFB replies, or starting commitment after getting one RFB reply, is a compromise between the best solution and the fastest reaction. This is more predictable and controllable than altering offers, as the total process time for each negotiation is easier to calculate than that. Therefore, we can choose proper parameters to let most negotiation tasks complete before their deadlines.

4.5 Commitment Process

In distributed environments, a two-phase commitment protocol [21] has been widely used to synchronise the commitment for distributed transactions. A modified version of the two-phase commitment is used as the protocol for committing the agreement in this work, as shown in Figure 7.

When the negotiation initiator finds a solution, it starts a distributed transaction to actualise this solution. Its executor checks which of the adjacent base station agents are involved with this transaction first. It then creates and sends RFC messages for each of the helpers, which contain the ID of the helpers’ hypotheses. It also sets the agent’s state as “in transaction”. If any helpers need further help, it has to wait until it gets enough RFC replies, or starting commitment after getting one RFC reply, is a compromise between the best solution and the fastest reaction. This is more predictable and controllable than altering offers, as the total process time for each negotiation is easier to calculate than that. Therefore, we can choose proper parameters to let most negotiation tasks complete before their deadlines.

4.5 Commitment Process

In distributed environments, a two-phase commitment protocol [21] has been widely used to synchronise the commitment for distributed transactions. A modified version of the two-phase commitment is used as the protocol for committing the agreement in this work, as shown in Figure 7.

When the negotiation initiator finds a solution, it starts a distributed transaction to actualise this solution. Its executor checks which of the adjacent base station agents are involved with this transaction first. It then creates and sends RFC messages for each of the helpers, which contain the ID of the helpers’ hypotheses. It also sets the agent’s state as “in transaction”. If any helpers need further help, it has to wait until it gets enough RFC replies, or starting commitment after getting one RFC reply, is a compromise between the best solution and the fastest reaction. This is more predictable and controllable than altering offers, as the total process time for each negotiation is easier to calculate than that. Therefore, we can choose proper parameters to let most negotiation tasks complete before their deadlines.

4.5 Commitment Process

In distributed environments, a two-phase commitment protocol [21] has been widely used to synchronise the commitment for distributed transactions. A modified version of the two-phase commitment is used as the protocol for committing the agreement in this work, as shown in Figure 7.
When the initiator receives all the AOC replies, it creates and sends CYC messages to involved helpers to ask them to actually change their coverage. Similarly, further CYC messages may be needed for more helpers. When those helpers receive CYC messages, they send CYP messages to their antenna agents, and the antenna agents start to synthesize the pattern and finally make the changes of antenna patterns happen.

However, if the RFC message arrives at a helper that is currently in another transaction, or the hypothesis being asked for has expired, the whole transaction has to be canceled. This helper sends SE and CT messages to the requester and possible helpers respectively. Again, further SE or CT messages may be needed to cancel this distributed transaction. Those agents reset their states and acknowledge the cancelation with TC messages. After the initiator receives all the TC messages, it starts another commitment process with the next best solution.

After all the agents commit their agreements, all the messages relative to this negotiation are discarded, and the states for all involved agents are reset to idle. They are ready for the next negotiations and commitments.

5. SIMULATION AND RESULTS

System-level simulations for a cellular network were performed to test the efficacy of the cooperative negotiation. In order to have enough cooperative participators in the negotiation and reduce the boundary effect of cellular network simulation, a 100 diamond-mesh cellular network model is used, as shown in Figure 8.

![Figure 8: A cellular network with 4 traffic hotspots (shadow area)](image)

5.1 Simulation configuration

The approach is being evaluated using a sequence of 200 traffic snapshots taking from an imaginary cellular network where four hotspots are forming (shadowed area in Figure 8). The time interval between two traffic snapshots is 60 seconds. The configuration for traffic snapshots is:

- Each traffic snapshot contains 50,000 traffic units;
- 80% of traffic is uniformly distributed in the area;
- 20% of traffic gathering to 4 hotspots from the first snapshot to the last one. The locations of the traffic in hot spots yield normal distribution (the mean value, \( \mu \), is uniformly distributed over the whole area, and the standard deviation, \( \sigma = 0.5R \));
- The talking and call arrival time for each traffic unit has a negative exponential distribution. The mean values are 120 sec and 720 sec respectively;
- Base stations are situated in the centre of cell, and the average load is 70% of their capacity.

5.2 Simulation Results

Simulations have been performed for 200 traffic snapshots, and the results are presented in Figure 9. The load-balancing scheme is evaluated with three 4-element array antennas and ideal patterns (generated by cubic spline interpolation [22] from discrete shapes). The results for 40 traffic snapshots using the global optimization technique are also plotted in Figure 9 as a performance benchmark. The real coded genetic algorithm and the associated constraint handling method used for the global optimization are described in [10].

![Figure 9: Simulation results for cooperative negotiations](image)

![Figure 10: Some discrete and synthesised patterns from negotiation results](image)

Note that the performance is measured using the antenna patterns generated by pattern synthesizers in antenna agents, which are physically feasible, and not the idealized patterns.
created by base station agents. Both patterns are shown in Figure 10. The average computation time for the intelligent control algorithm for 100 base stations is about 30 seconds at a PC computer (Pentium III 750MHz CPU with 256MB RAM) for each traffic snapshot. In practice, this control algorithm will be running at base station controllers, which usually manages seven base stations each. Then it only takes about 2 seconds for each control interval on average given the similar processing capability.

As hot spots form, the call-blocking rate increases. However, the scheme using intelligent geographic load balancing shows much lower blocking than the conventional one, especially when there are some hotspots. This demonstrates that by the use of intelligent geographic load balancing, the system capacity can be improved significantly for non-uniformly distributed traffic.

6. CONCLUSIONS

This paper has proposed and investigated an intelligent traffic load balance scheme for mobile cellular network, which dynamically changes the cell size and shape according to geographic traffic distribution in the real time. The results from computer simulations show that it does increase the system capacity significantly.

We evaluate a cooperative negotiation approach for the real-time intelligent control of cellular coverage. A modified version of two-phase commitment protocol is used for coordinating the mutual commitment of negotiation agreements.

A key aspect of the negotiation is that while many hypotheses (and hence requests) are created to explore the search space this is counterbalanced by the assumption that each agent only helps one requester at a time. The latter ensures timely response because it reduces the complexity of the negotiation. To ensure real time behaviour respondents have a deadline to generate offers to satisfy the requests. The best of what is offered by the deadline is chosen and the system commits to this. If none are offered the pattern does not change.

7. REFERENCES