Requests Management for Smartphone-based Applications using a Multi-Agents Model

Gilles Simonin and Barry O’Sullivan
Insight Centre for Data Analytics, University College Cork, Ireland
{gilles.simonin | barry.osullivan}@insight-centre.org

Mots-clés: Multi-agents, optimisation, smartphone-based applications.

1 Introduction

Recently the use of smartphone-based applications has received attention due to the emergence of a number of new online optimisation problems that involve reaching bi-lateral agreements, such as in ride-sharing applications. The majority of such apps require sending a lot of requests or notifications to users. A typical goal is to reduce the overhead for users and to minimise the number of requests sent.

2 A Smart System

One of the qualities of smartphone-based applications is their ability to reduce the demands placed on users and their ability to reach a decision in a timely manner. In many applications (ride-sharing, on-line service between users, games, sales, . . .) users must find a bi-lateral agreement, e.g. a driver is happy to offer a ride to someone who is happy to accept that offer. Each user sends a request to other users and waits for a positive/negative response. In many applications this leads to a proliferation of notifications and requests sent to and by users. For this reason many automated systems that take decisions on behalf of users have emerged. In these systems, users can only initiate the agreement process, but they cannot control how an agreement is reached.

3 From a Graph-based Model to a Multi-Agent Approach

The requests problem can be modelled as a graph : the set of vertices represents users and we define an edge between two users if there exists a potential bi-lateral agreement, a deal, between them. A weight is assigned to each edge in order to quantify the quality of the potential deal. The objective is to minimise the risk of rejection and the number of notifications/requests sent between users. Therefore, the goal is to select a maximum number of edges (deals) that maximises the sum of weights. It is obvious that this selection is constrained by the structure of the problem and by the nature of the smart-phone app in question. For example, in the ride-sharing problem, we want to select the potential riders in the same car. Therefore, we want three edges from a driver, if we want to fill the car, and one edge for a rider. From a deterministic global solution, requests are sent by the system to each user. If a user does not respond quickly, the system has to propose an alternative solution to another potential partner who is waiting to make a deal. The question is when should the system send such a new request?

One can see the requests problem as a multi-agent problem where each user is represented by an agent. Each agent has a personal behaviour based on his preferences and an altruistic behaviour with respect to the links between the agent and his neighbours. Such a behaviour-based
approach informs how to design the agents’ actions over time. This kind of perspective allows us to define agents that are able to evolve in dynamic and partially unknown environments.

3.1 Model

Each active user \( u \) (agent) has a satisfaction function depending on time \( \text{Sat}_u(t) \). This function is bound between \([-1, 1]\) and is decreasing exponentially with each elapsed time period, say every second. An active user \( u \) has a list of potential partners \( P = \{p_1, p_2, \ldots, p_k\} \). For each pair \( \langle u, p_i \rangle \), we have a weight representing the preference \( w_{up_i} \). Thus we can define a list of weights for the potential partners: \( W_{up} = \{w_{up_1}, w_{up_2}, \ldots, w_{up_k}\} \).

According to the satisfaction of \( u \) and the \( p_i \)'s, \( W_{up} \) will change over time. Every second, the variation value for a weight \( w_{up_i} \) decreases (resp. increases) if a request has been already sent to \( p_i \) (resp. not sent). In the next section we present the algorithm that monitors the users satisfiability and weights over time.

3.2 Algorithm

The principle of the algorithm consists of three steps. First we compute and update, after every second, the users’ satisfiability \( \text{Sat}_u \) and the weights for potential partners. Second we check every case where we might send a new request, when a user answers negatively or positively, or when the satisfiability is negative and there exists a partner who has not received a request and has a weight higher than the best weight from partners already requested. Third if we are in one of these cases, we compute a new global solution with the new weights and we send a new request. An example is presented in Table 1.

<table>
<thead>
<tr>
<th>Time</th>
<th>( w_{up1} )</th>
<th>( w_{up2} )</th>
<th>( w_{up3} )</th>
<th>( \text{Sat}_u )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t = 0s )</td>
<td>10</td>
<td>5</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>( t = 300s )</td>
<td>8.536</td>
<td>5.756</td>
<td>3.756</td>
<td>(-0.22)</td>
</tr>
<tr>
<td>( t = 600s )</td>
<td>7.458</td>
<td>6.458</td>
<td>4.458</td>
<td>(-0.49)</td>
</tr>
<tr>
<td>( t = 900s )</td>
<td>6.09</td>
<td>7.091</td>
<td>5.091</td>
<td>(-0.836 \rightarrow 0.164)</td>
</tr>
<tr>
<td>( t = 1200s )</td>
<td>4.53</td>
<td>5.53</td>
<td>5.53</td>
<td>(-0.406)</td>
</tr>
<tr>
<td>( t = 1320s )</td>
<td>NO \rightarrow 0</td>
<td>4.855</td>
<td>5.655</td>
<td>(-0.721 \rightarrow 0.279)</td>
</tr>
<tr>
<td>( t = 1360s )</td>
<td>0</td>
<td>YES</td>
<td>5.342</td>
<td>0.062 \rightarrow 1</td>
</tr>
</tbody>
</table>

4 Conclusions

In this paper we presented a new multi-agent approach to managing the requests made between users of applications for reaching bi-lateral agreements. The objective is to maximise the likelihood of an acceptable match while minimising the burden on the users. The next step will be to empirically study this method with real data for a number of applications, such as car-pooling and online trading.

Acknowledgement. This contribution has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289 and from industry Carma (www.carmacarpool.com).

Références