# Chapter 13
# OMAC: A Discrete Wavelet Transformation Based Negotiation Agent

**Siqi Chen and Gerhard Weiss**

**Abstract** This work describes an automated negotiation agent called OMAC which was awarded the joint third place in the 2012 Automated Negotiating Agent Competition (ANAC 2012). OMAC, standing for "Opponent Modeling and Adaptive Concession," combines efficient OMAC making. Opponent modeling is achieved through standard wavelet decomposition and cubic smoothing spline; concession-making is made through setting the best possible concession rate on the basis of the expected utilities of forthcoming counter-offers.

**Keywords** Automated multi-issue negotiation • Discrete wavelet transformation • Opponent modeling

## 13.1 Introduction

Negotiation provides a mechanism for coordinating interaction among computational autonomous agents which represent respective parties of different or even conflicting interest. As automated negotiation can be applied to fields as diverse as electronic commerce and electronic markets, supply chain management, task and service allocation, etc, it has become a core topic of multi-agent systems [6]. This paper introduces a novel negotiation agent called OMAC ("Opponent Modeling and Adaptive Concession") for complex scenarios, where agents have no useful information about their opponents, and in addition they are under

---

This is a shortened version of our OMAC description provided in [3].

S. Chen (✉) • G. Weiss
Department of Knowledge Engineering, Maastricht University, Maastricht, The Netherlands
e-mail: siqi.chen@maastrichtuniversity.nl; gerhard.weiss@maastrichtuniversity.nl

I. Marsa-Maestre et al. (eds.), *Novel Insights in Agent-based Complex Automated Negotiation*, Studies in Computational Intelligence 535,
DOI 10.1007/978-4-431-54758-7_13, 

real-time constraints. The negotiation strategy of OMAC integrates two key aspects of successful negotiation: efficient OMAC making. Opponent modeling realized by OMAC aims at predicting the utilities of an opponent's future counter-offers and is achieved through two standard mathematical techniques, namely, wavelet decomposition and cubic smoothing spline. Adaptive concession making is achieved through dynamically adapting the concession rate (i.e., the degree at which an agent is willing to make concessions in its offers) on the basis of the utilities of future counter-offers it expects according to its opponent model.

The remainder of this paper is structured as follows. Section 13.2 describes the standard negotiation environment underlying our research. Section 13.3 overviews OMAC. Sections 13.4–13.6 describe OMAC in detail. Finally, Sect. 13.7 identifies some important research lines induced by the work.

## 13.2 Negotiation Environment

We adopt a basic bilateral multi-issue negotiation setting which is widely used in the agents field (e.g., [2, 3]). The negotiation protocol is based on a variant of the alternating offers protocol proposed in [5]. Let $I = \{a, b\}$ be a pair of negotiating agents, $i$ represent a specific agent ($i \in I$), $J$ be the set of issues under negotiation, and $j$ be a particular issue ($j \in \{1, \ldots, n\}$ where $n$ is the number of issues). The goal of $a$ and $b$ is to establish a contract for a product or service. Thereby a contract consists of a package of issues such as price, quality and quantity. Each agent has a lowest expectation for the outcome of a negotiation; this expectation is called reserved utility $u_{res}$. $w^i_j$ ($j \in \{1, \ldots, n\}$) denotes the weighting preference which agent $i$ assigns to issue $j$, where the weights of an agent are normalized (i.e., $\sum_{j=1}^{n}(w^i_j) = 1$ for each agent $i$). During negotiation agents $a$ and $b$ act in conflictive roles which are specified by their preference profiles. In order to reach an agreement they exchange offers $O$ in each round to express their demands. Thereby an offer is a vector of values, with one value for each issue. The utility of an offer for agent $i$ is obtained by the utility function defined as:

$$U^i(O) = \sum_{j=1}^{n}(w^i_j \cdot V^i_j(O_j)) \tag{13.1}$$

where $w^i_j$ and $O$ are as defined above and $V^i_j$ is the evaluation function for $i$, mapping every possible value of issue $j$ (i.e., $O_j$) to a real number.

Following Rubinstein's alternating bargaining model [5], each agent makes, in turn, an offer in form of a contract proposal. Negotiation is time-limited instead of being restricted by a fixed number of exchanged offers; specifically, each negotiator has a hard deadline by when it must have completed or withdraw the negotiation. The negotiation deadline of agents is denoted by $t_{max}$. In this form of real-time constraints, the number of remaining rounds are not known and the outcome of a negotiation depends crucially on the time sensitivity of the agents' negotiation strategies. This holds, in particular, for discounting domains, that is, domains in which the utility is discounted with time. As usual for discounting domains, we define a so-called discounting factor $\delta$ ($\delta \in [0, 1]$) and use this factor to calculate the discounted utility as follows:

$$D(U, t) = U \cdot \delta^t \tag{13.2}$$

where $U$ is the (original) utility and $t$ is the standardized time. As an effect, the longer it takes for agents to come to an agreement the lower is the utility they can achieve.

After receiving an offer from the opponent, $O_{opp}$, an agent decides on acceptance and rejection according to its interpretation $I(t, O_{opp})$ of the current negotiation situation. For instance, this decision can be made in dependence on a certain threshold $Thres^i$: agent $i$ accepts if $U^i(O_{opp}) \geq Thres^i$, and rejects otherwise. As another example, the decision can be based on utility differences. Negotiation continues until one of the negotiating agents accepts or withdraws due to timeout.

## 13.3 Overview of OMAC

An overview of OMAC is given in *Algorithm 5*. In more detail, OMAC includes two core stages—opponent modeling and concession rate adaptation—as described in detail in Sects. 13.4 and 13.5, respectively. A third important stage of OMAC, its response mechanism to counter-offers, is described in Sect. 13.6.

## 13.4 Opponent Modeling

According to OMAC, the aim of opponent modeling realized by a negotiating agent is to estimate the utilities of future counter-offers it will receive from its opponent. This corresponds to the lines 3 to 8 in *Algorithm 5*. Opponent modeling is done through a combination of wavelets analysis and cubic smoothing spline. When receiving a new bid from the opponent at the time $t_c$, the agent records the time stamp $t_c$ and the utility $U(O_{opp})$ this bid has according to the agent's utility function. The maximum utilities in consecutive equal time intervals and the corresponding time stamps are used periodically as basis for predicting the opponent's behavior (line 5 and 6). The reasons for a periodical updating are twofold as discussed in [2]. Firstly, this degrades the computation complexity so that the agent's response time is kept low. Assume that all observed counter-offers were taken as inputs, then the agent might have to deal with thousands of data points in every single session. This computational load would have a clear negative impact on the quality of negotiation in a real-time constraint setting. Secondly, the effect of noise can be reduced.

**Algorithm 5:** The strategy of OMAC. $t_c$ refers to the current time, $\delta$ the time discounting factor, $\lambda$ the layer of wavelet decomposition, $\psi$ the wavelet function, and $t_{max}$ the deadline of negotiation. $O_{opp}$ is the latest offer of the opponent, and $O_{own}$ the offer to be proposed by OMAC. $\chi$ represents the time series comprised of the maximum utilities over intervals. Let $\upsilon$ be the smooth component of $\lambda$-th order wavelet decomposition based on $\psi$, and $\alpha$ the predicted main tendency of $\chi$. $t_l$ is the time we preform prediction process and $u_l$ is the utility of our most recent offer. $u'$ is the target utility at time $t_c$. $R$ is the reserved utility function

1: $Require : t_{max}, \delta, \lambda, \psi, R$<br>
2: **while** $t_c <= t_{max}$ **do**<br>
3:     $O_{opp} \Leftarrow$ receiveMessage();<br>
4:     recordBids($t_c$, $O_{opp}$);<br>
5:     **if** needUpdate($t_c$) **then**<br>
6:         $\chi \Leftarrow$ preprocessData($t_c$)<br>
7:         $(\alpha, t_l, u_l) \Leftarrow$ predict($\chi, \lambda, \psi$);<br>
8:     **end if**<br>
9:     $u'$ = getTarUtility($t_c, t_l, u_l, \delta, \alpha, R$);<br>
10:     **if** getOwnUtility($O_{opp}, t_c, \delta$) $\geq u'$ **then**<br>
11:         $accept(O_{opp})$;<br>
12:     **else**<br>
13:         $O_{own} \Leftarrow$ constructOffer($u'$);<br>
14:         proposeBid($O_{own}$);<br>
15:     **end if**<br>
16: **end while**

In multi-issue negotiation a small change in utility of the opponent can result in a large utility change for the negotiator and this can easily result in a misinterpretation of opponent's behavior.

Behavior prediction is mainly done by applying discrete wavelet transformation (DWT) to the time series $\chi$; this is captured by line 7. We decided to use DWT because wavelet analysis is known to be an efficient multi-scaling tool for exploring features in data sets. With DTW a signal can be decomposed into two parts, an approximation and a detail part. The former is smooth and reveals the trend of the original signal, and the latter is rough and corresponds to noise (resulting e.g. from seasonal fluctuations). OMAC focuses on the approximation part and intentionally ignores the detail part for three reasons. First, the approximation part represents the trend of the opponent concession in terms of utility and indicates how the concession of opponent will develop in the future. Second, it is smooth enough (compared to the original signals, i.e. $\chi$) to allow for quality prediction performance. Third, the detail part contains information which is of little value in a negotiation setting. As we saw in various empirical investigations, the ratio between the main tendency term and the original signal tends to be about 0.98 with a small standard deviation. Precise extension of those detailed components can improve effectiveness of our model slightly, it is however very costly for a medium-range lead time in real-time negotiation.

Given the discrete wavelet function $\psi_{j,k}(t)$ transformed by a mother wavelet $\psi(t)$,

$$\psi_{j,k}(t) = a_0^{-j/2}\psi(a_0^{-j}t - kb_0), \quad j,k \in Z \tag{13.3}$$

DWT corresponds to a mapping from the signal $f(t)$ to coefficients $C_{j,k}$ which are related to particular scales, where these coefficients are defined as follows:

$$C_{j,k} = \int_{-\infty}^{+\infty} f(t)\overline{\psi_{j,k}(t)}dt, \quad j,k \in Z \tag{13.4}$$

The $\psi(t)$ is required to be an orthogonal wavelet, the set $\{\psi_{j,k}(t)|j,k \in Z\}$ is then an orthogonal wavelet basis such that the signal $f(t)$ can be reconstructed.

With recursive application of DWT to the signal $f(t)$, the approximation (low frequency) and detail (high frequency) components are recovered, respectively. For instance, $f$ can first be decomposed into $a_1 + d_1$ and the resulting part $a_1$ can then be decomposed in finer components, that is, $a_1 = a_2 + d_2$, and so on. Based upon this recursive process, the signal can be expressed as $f = a_1 + a_2 + \ldots + a_n + d_n$ (further details on wavelets are given in e.g. [4]). The results reported in this paper are achieved through wavelet decomposition using the Daubechies' wavelets of order 10.

We use the following notation:

$$\chi = \upsilon + \sum_{n=1}^{\lambda} d_n \tag{13.5}$$

where $\upsilon$ represents the approximation component of $\chi$ and $d_n$ is $n$-layer detail part ($n$ is determined by the decomposition level $\lambda$). An example can be found in Fig. 13.1 which shows $\chi$ and its corresponding approximation part $\upsilon$ along with the estimated upper and lower bounds of $\chi$. The two bounds are represented by $v \pm \sigma$, where $\sigma$ is the standard deviation of the ratio between $\chi$ and $\upsilon$.

In order to forecast the opponent's future behavior, cubic smoothing spline is used to extend the smooth component $\upsilon$. Cubic spline is widely used as a tool for prediction, see [7]. For equally spaced time series, a cubic spline is a smoothing piecewise function, denoted as the function $g\hat{(t)}$ which minimizes:

$$p\sum_{t=1}^{n} w(t)(f(t) - \hat{g}(t))^2 + (1-p)\int(\hat{g}(u)'')^2du \tag{13.6}$$

where $p$ is the smoothing parameter controlling the rate of exchange between the residual error described by the sum of squared residuals and local variation represented by the square integral of the second derivative of $g$ and $w$ is the weight vector (for further details, refer to [1]).

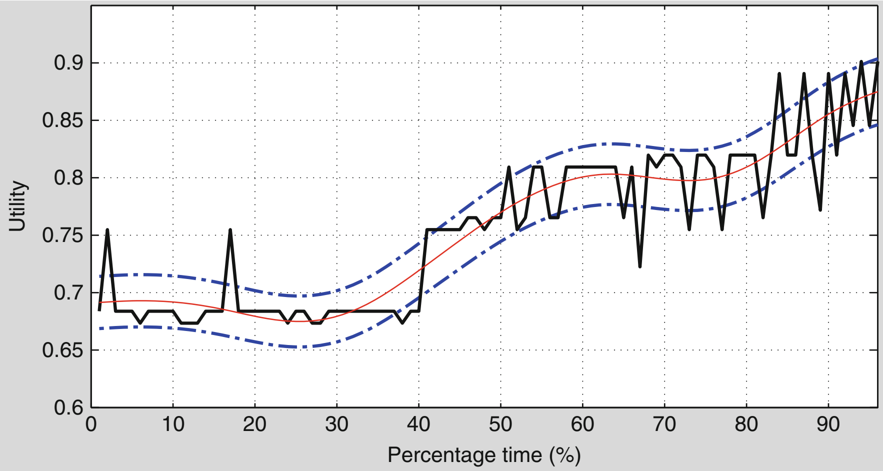


**Fig. 13.1** Illustrating the opponent's concession (given by $\chi$, the *thick solid line*) and the corresponding approximation part $\upsilon$ (the *thin solid line*) when negotiating with Agent_K2 in the $Camera$ domain (this agent and domain are taken from ANAC 2011). The *two dash-dot lines* represent the estimated upper and lower bounds of $\chi$

Figure 13.2 shows the actual and the predicted smooth parts of opponent concession at different time points for the opponent "Iamhaggler2011": as this figure illustrates, cubic spline is able to forecast the given signal within a medium range very well. Since OMAC applies a periodical updating mechanism, it is not necessary and not wise to forecast globally (i.e., from the current moment to the end point of negotiation), because this probably brings too much noise into the prediction. OMAC limits the range of forecasting to $\zeta$ intervals and in this way achieves efficiency and noise reduction.

## 13.5 Adaptive Adjustment of Concession Rate

Given the extended version of the smooth part—$\alpha$, we now discuss how to use it for adaptively setting the concession rate of our expected utility (see line 9 in *Algorithm 5*). A possibility is to maximize the expected utility merely according to the predicted opponent move. This is quite straightforward but may be not so effective. Suppose the negotiation partners are "tough" and always avoid making any concession in bargaining. In this case the result of prediction could indicate a very low expectation about the utility offered by the opponent and this, in turn, would result in an adverse concession. In OMAC a simple function $R$, called reserved utility function, is used to realize concession adaptation. This function guarantees the minimum utility at each given time step. This is because the function values

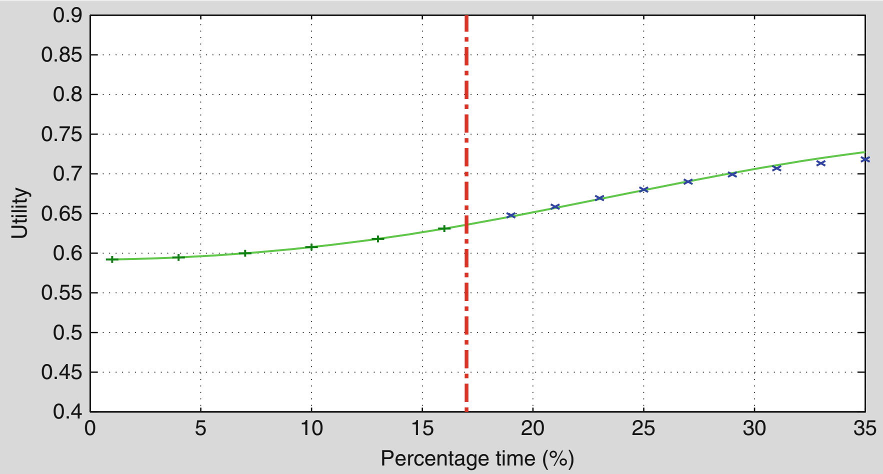


**Fig. 13.2** Illustration of the predictive power of OMAC in two consecutive ranges. The *dash line* indicates the time point $t_c$ at which the current prediction is made. The *plus signs* on left of the *dash line* are the actual points of $\upsilon$ before $t_c$. The *crosses* to the right of the *dash line* show the actual points of $\upsilon$ after $t_c$. The extended version of $\upsilon - \alpha$ (i.e., the prediction of $\upsilon$) is shown by the *solid line*. These results are achieved against the agent Iamhaggler2011 in the domain $Amsterdam\ party$ (the agent and domain are taken from ANAC 2011)

are set as the lower bound of our expected utilities. Moreover, in principle it makes concession over time, thereby taking into account the impact of the discounting factor. Specifically, the reserved utility function is given by:

$$R(t) = u_{res} + (1 - t^{1/\beta})(maxUtility(p) \cdot \delta^{\eta} - u_{res}) \tag{13.7}$$

where $u_{res}$ is the minimum utility the agent would accept, $\beta$ is a parameter which has a direct impact on the concession rate, *maxUtility(p)* is the function specifying the maximum utility given by the preference profile $p$ of a negotiation domain, and $\eta$ is a parameter called risk factor which reflects the agent's expectation about the maximum utility it can achieve.

We define the estimated received utility $E_{ru}(t)$, which gives our agent the expectation of opponent's future concession, as follows:

$$E_{ru}(t) = D(\alpha(t)(1 + Stdev(ratio_{[t_b,t_c]})), t), \quad t \in [t_c, t_s] \tag{13.8}$$

where $Stdev(ratio_{[t_b,t_c]})$ is the standard deviation of ratio between the smooth part $\upsilon$ and the original signal $\chi$ from the beginning of negotiation($t_b$) till now and $t_s$ is the end of $\alpha$.

Suppose the future expectation the agent has obtained from $E_{ru}(t)$ is optimistic, in other words, there exists an interval $\{T|T \neq \emptyset, T \subseteq [t_c, t_s]\}$, so that

$$E_{ru}(t) \geq D(R(t), t), \quad t \in T \tag{13.9}$$

OMAC then sets the time $\hat{t}$ at which the optimal estimated utility $\hat{u}$ is reached as:

$$\hat{t} = argmax_{t \in T}(E_{ru}(t) - D(R(t), t)) \tag{13.10}$$

and $\hat{u}$ is simply assigned by:

$$\hat{u} = E_{ru}(\hat{t}) \tag{13.11}$$

When the opponent's future concession is estimated to be below the agent's expectations according to $R(t)$ (i.e., there is no such interval $T$ described above), OMAC investigates whether the best possible outcome under that "pessimistic" expectation of opponent concession should be accepted given the threshold $\rho$. This outcome is denoted as $\xi$ and is given by:

$$\xi = \rho^{-1} \cdot E_{ru}(t_\xi)/D(R(t_\xi), t_\xi), \quad t_\xi \in [t_c, t_s] \tag{13.12}$$

where $\rho$ is the tolerance threshold to accept $E_{ru}(t_\xi)$ as target utility and $t_\xi$ is given by:

$$t_\xi = argmin_{t \in [t_c, t_s]}(|E_{ru}(t) - D(R(t), t)|) \tag{13.13}$$

The rationality behind it is that if the agent rejects the "locally optimal" counter-offer, the agent will probably loose the opportunity to reach a "globally good" agreement (especially in discounting domains). If $\xi > 1$, $\hat{u}$ and $\hat{t}$ are assigned to $E_{ru}(t_\xi)$ and $t_\xi$, respectively. Moreover, the agent records the utility and time of its last bid as $u_l$ and $t_l$, respectively. Otherwise, the estimated utility is set to $-1$, meaning it does not take effect anymore, and $D(R(t_c), t_c)$ is used to set the target utility $u'$.

When the agent expects to achieve better outcomes (see Eq. (13.9)), the optimal estimated utility $\hat{u}$ is chosen as the target utility for our agent's future bids. Obviously, it is not rational to concede immediately to $\hat{u}$ when $u_l \geq \hat{u}$, nor should it shift to $\hat{u}$ without delay given $u_l < \hat{u}$, especially because the predication may be not absolutely accurate. To simplify the negotiation strategy, OMAC applies a linear concession making and the concession rate is dynamically adjusted to grasp every chance to maximize its profit. Overall, the target utility $u'$ is given as follows:

$$u' = \begin{cases} D(R(t), t) & if \ \ \hat{u} = -1 \\ \hat{u} + (u_l - \hat{u})\frac{t - \hat{t}}{t_l - \hat{t}} & otherwise \end{cases} \tag{13.14}$$

## 13.6 Response Mechanism

The response stage corresponds to lines 10 to 15 in *Algorithm* 5. With the target utility $u'$ known (Eq. 13.14), the agent then needs to examine the counter-offer to see if the utility of that offer $U(O_{opp})$ is higher than the target utility. If so, it accepts this counter-offer and, with that, terminates the negotiation session. Otherwise, the agent construct a bid to be proposed next round whose utility is indicated by $u'$.

In multi-issue negotiation, offers with exactly the same utility for one side can have different values for the other party. Moreover, in time-limited negotiation scenarios no explicit limitation is imposed on the number of negotiation rounds and it is possible to generate many offers having a utility close to $u'$. OMAC takes advantage of this and aims at generating many offers in order to explore the space of possible outcomes and to increase the acceptance chance of own bids. Specifically, offers are constructed in such a way that the agent randomly selects an offer whose utility is in the range $[0.99u', 1.01u']$. If no such solution is found, the latest offer made by the agent is used again in the subsequent round. Moreover, in view of negotiation efficiency, if $u'$ drops below the utility of the best counter-offer according to the agent's utility function, this best counter-offer is proposed by the agent as its next offer. This makes sense because the counter-offer tends to satisfy the expectation of opponent and is thus likely to be accepted by the opponent.

## 13.7 Conclusions and Future work

This paper introduced an effective negotiation agent called OMAC for automated negotiation in complex—bilateral multi-issue, time-constrained, no prior knowledge, low computational load, etc.—scenarios. This agent, based on its efficient decision-making mechanism, achieved the joint third place in ANAC 2012.

We think the experimental results justify to invest further research efforts into this strategy and we see several interesting research questions. First, are there opponent modeling techniques which are even more efficient than wavelet decomposition and cubic smoothing spline? Second, are there techniques for concession rate adaptation which are more accurate than the basic technique currently used? And third, can opponent modeling of OMAC, which currently focuses on modeling the opponent's strategies, be extended toward modeling the opponent's preferences as well?

## References

1. de Boor, C.: A Practical Guide to Splines. Springer, New York (1978)
2. Chen, S., Ammar, H.B., Tuyls, K., Weiss, G.: Optimizing complex automated negotiation using sparse pseudo-input Gaussian processes. In: Proceedings of the 12th International Joint Conference on Autonomous Agents and Multi-Agent Systems, pp. 707–714. ACM, Saint Paul, Minnesota (2013)

3. Chen, S., Weiss, G.: An efficient and adaptive approach to negotiation in complex environments. In: Proceedings of the 20th European Conference on Artificial Intelligence, pp. 228–233. IOS, Montpellier, France (2012)
4. Daubechies, I.: Ten Lectures on Wavelets. Society for Industrial and Applied Mathematics, Philadelphia, PA (2006)
5. Rubinstein, A.: Perfect equilibrium in a bargaining model. Econometrica **50**(1), 97–109 (1982)
6. Weiss, G. (ed.): Multiagent Systems, 2nd edn. MIT, Cambridge, MA (2013)
7. Yousefi, S., Weinreich, I., Reinarz, D.: Wavelet-based prediction of oil prices. Chaos Solitons Fract. **25**(2), 265–275 (2005)