Incrementally Refined Acquaintance Model for Consortia Composition

Jan Doubek, Jiří Vokřínek, Michal Pěchouček, Martin Rehák

Agent Technology Group, Gerstner Laboratory
Department of Cybernetics, Czech Technical University in Prague
E-mail: duby.jan@seznam.cz, {vokrinek|pechouc}@labe.felk.cvut.cz

Abstract. This paper presents a specific contracting algorithm that contributes to the process of distributed planning and resource allocation in competitive, semi-trusted environments. The presented contraction algorithm is based on incrementally refined acquaintance models (IRAM) of the actor that provide the right set of approximate knowledge needed for appropriate task decomposition and delegation. This paper reports on empirical evaluation of the IRAM algorithm deployment in consortia formation domain.

1 Introduction

This work focuses on a technique for distributed consortium formation with limited knowledge sharing. The consortia formation is based on a negotiation between independent self-interested providers. The providers can offer several services and the goal is to find the best suitable composition of providers to cover required set of services.

The targeted domain organizes multi-party interaction in the environments that are:

- **non-centralized** and with flat organizational structure [R1] – the existence of a central coordination is minimal and the information about the skills of actors, resource availability, knowledge and goals is distributed,
- **multi-party involvement** [R2] – the final project cannot be implemented in isolation by a single actor, consortium composition can be initiated by several actors simultaneously,
- provides **partial knowledge sharing** [R3] – the actors in the environment are motivated to keep a substantial part of their private planning knowledge and resource availability information undisclosed.

Due to the presented requirements, the consortia cannot be evaluated centrally and the service allocation to the individual providers has to be negotiated. The goal is to minimize the interactions with the providers to the necessary minimum to reduce their private knowledge disclosure and simultaneously ensure the quality of the solution. To fulfill those demands we have designed presented algorithm and acquaintance model representation and provide experimental evaluation.

This algorithm contributes to the process of distributed planning and resource allocation in competitive, semi-trusted environments. The presented algorithm is based on incrementally refined acquaintance models (IRAM) – the model that the actor is maintaining about potential collaborators [1].

2 Problem Statement

The consortium composition can be represented as distributed state-space search through all the potential consortia in the environment with limited information sharing. Let us denote R as a requester agent, A_t as a set of services that R requests to fulfill the task t. Furthermore we have agent P_j, as a provider agent offering a certain set of services A_j^{max}, where $j \in \{1 \ldots n\}$ and

$$F_j(A_j) : \{A_j \subset A_j^{max}\} \tag{1}$$

is pricing function for agent P_j to provide set of services A_j. When asked agent P_j sends back just the price value.
The problem is then to acquire the optimal price

$$c(A_t) = \min \sum_{j=1}^{n} F_j(A_j) \rightleftharpoons \{A_j \subset A_j^{max}; \bigcup_{j \in \{1 \ldots n\}} A_j = A_t\} \tag{2}$$

In this paper we then focus to find optimal vector of sets $(A_1, \ldots, A_n)$ as a decomposition of task t, where the overall price $c(A_t)$ is minimal. Set of all vectors that satisfies task the condition $\bigcup_{j \in \{1 \ldots n\}} A_j = A_t$ will be referenced as a *Deal Space* - DS_t.

In our model we have made several assumptions:

- **Fixed price** – the price of a particular subset of services is fixed during algorithm run.
- **Tasks are independent** – a provider is capable of delivering same services during all negotiation even if he was contracted for some services in the previous tasks.
- **Non-increasing partial price** – a provider constructs a $F_j(A_j)$ as aggregated price from prices for individual services that are hidden to the requester. We assume the individual services price to be non-increasing in reference to increasing $|A_j|$.

3 IRAM-based Consortium Formation

We have designed a straightforward decomposition mechanism that finds the optimal decomposition given the right objective function and a complete information about provider's resource availabilities. The decomposition algorithm is polynomial and easy to construct (see [2]). Its behavior, however, worsens strongly with lower quality of information about the provider's prices stored in the requestors' acquaintance models (containing a subset of the deal space). The most efficient approach in fully cooperative communities would be if the requestor queries all the providers and reconstructs the deal space for all services provided by all actors prior to computing the optimal a contract.

As this is not possible in the environment compliant with the requirements R1 and R3, the requestor needs to approximate such knowledge with only partially available information. We are proposing *incrementally refined acquaintance model* (IRAM) algorithm for handling partial knowledge sharing and private knowledge disclosure [1].

This approach has been evaluated and compared with the another method of provider prices estimation - the well-known Chebyschev Polynomials approximation method.

3.1 Acquaintance Model

The acquaintance model can have a number of forms [3], [4]. In this particular application the acquaintance model is understood as function that predicts actor responses to a particular *call-for-proposals* (`CFP`) type of message. We represent the *acquaintance model* (am) as a mapping from a set of $\mathcal{P}(A_j^{max})$ possible subsets asked from the provider P_j to a 1 dimensional real-value space representing cost $\mathcal{C}$.

$$\mathcal{F}_j^{am} : \mathcal{P}(A_j^{max}) \rightarrow \mathcal{C} \tag{3}$$

Let us discuss several properties of an acquaintance model. The *fixed point* is such a mapping among the actor, single service and a particular cost that is based on exact information acquired from the communication with the specific actor. In a fixed point as_x^{σ}

$$\mathcal{F}_j^{am}(as_x^{\sigma}) = f_j(s_x, |A_j|, pc_j) \text{ where } |A_j| = \sigma \tag{4}$$

where $f_j(s_x, |A_j|, pc_j)$ represents the price contribution of presence of service s_x in A_j to the total price of the entire set $F_j(A_j)$.

Provided that the fixed points of the acquaintance model are collected in a set $\Delta(\mathcal{F}_j^{am})$, we define the *size of the acquaintance model* $\delta(\mathcal{F}_j^{am})$ the amount of the fixed points in the acquaintance model as follows:

$$\delta(\mathcal{F}_j^{am}) = |\Delta(\mathcal{F}_j^{am})|. \tag{5}$$

Various approximation functions have been used in the acquaintance models, e.g. [2]. In our model we have selected the pairwise constant approximation. The **unknown** price of service s_y in subset with size $|A_j|$, equals to the closest bigger known fixed point in means of the size of the containing subset$|A'_j| : s_y \in A'_j$

$$\mathcal{F}_j(s_y, |A_j|, pc_j) = \mathcal{F}_j(s_y, |A'_j|, pc_j) \text{ if } |A_j| \lessdot |A'_j|, \tag{6}$$

provided that the symbol $\lessdot$ represent the smallest bigger value.

The *error of the acquaintance model* - $\epsilon(\mathcal{F}_j^{am})$ - represents how well does the acquaintance model capture real capability of the providers. Error of the acquaintance model is a dual quantity to the *quality of the acquaintance model*. There can be a number of ways how the error can be related to the quality. We only require that with a monotonic increase of quality the error decreases and vice versa.

We represent the error of the acquaintance model as a sum of the differences between the real costs and the information on costs provided by the acquaintance model.

$$\epsilon(\mathcal{F}_j^{am}) = \sum_{p,j} |\mathcal{F}_p^{am}(A_{p,j}) - f_p(A_{p,j}, pc_j)| \tag{7}$$

the index p goes through all possible subsets of A_j, and j goes through all partners.

As said before, the reason why we use the acquaintance models for contracting is that we are motivated by minimizing the unwanted knowledge disclosure during interaction (requirement R3). Each interaction represents disclosure of private information. By `CFP` the actors disclose their inability to perform a task as well as their intention to do so. By a response to `CFP` the agents disclose information about availability of particular resources. It is evident that with rising $\delta(\mathcal{F}_j^{am})$, the acquaintance model is more exact and thus provides better information (i.e. lower $\epsilon(\mathcal{F}_j^{am})$). Better acquaintance model managed to reduce communication (and thus private knowledge disclosure) during the negotiation between the actors. However, bigger $\delta(\mathcal{F}_j^{am})$ (and thus smaller $\epsilon(\mathcal{F}_j^{am})$) required substantial interaction during the acquaintance model construction phase where lots of unwanted information may have been disclosed.

The IRAM algorithm is balancing the size and the quality of the acquaintance models. In order to evaluate performance of the IRAM algorithm we have developed a reference algorithm that is working with a similar acquaintance model, constructed prior negotiation. Both algorithms are based on distributed state-space search using negotiation between actors. As a negotiation protocol, we use the *competitive contract-net protocol* [5], but any protocol that enables iterative contract negotiation can be used.

3.2 IRAM Algorithm

The run of this algorithm for one particular task t is started with the initiation phase. All providers are contacted for every single service and for maximal subset of services from task t. The IRAM model am is constructed using the closest bigger known fix-point approximation (see eq. 6). The model is represented by sets of prices for specific service and pricing function settings (see eq. 1). The price is set blank when is not known, and thus is calculated from other fixed points. The algorithm constructs the Deal Space and evaluates it with the prices from $\mathcal{F}_j^{am}$. The cheapest consortium $Cons^{best}$ is selected and the providers are

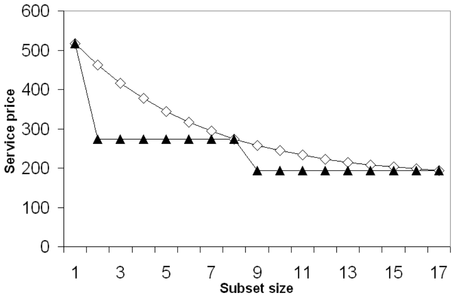


Fig. 1. IRAM model(labeled with triangles) for one service according to s_x, fixed points in 1,8,17. Approximated function is labeled by diamonds

requested for appropriate services. Offered prices are then integrated into the IRAM model. The deal space is then reevaluated and the cheapest consortium is selected. If the new consortium is composed from fixed points (represents real price of the consortium – no price approximation)

, this we understand as optimum. Request for this consortium will lead to the exact same information and due to eq. 6 the algorithm has converge to the optimum. The phases of IRAM can be seen below.

Initialization It is necessary to know at least two fixed points of acquaintance model for specific service from each provider, for proper functionality of IRAM algorithm. So if the algorithm have not these from previous contracts, it obtains them in the first iteration. Due to this fact the amount of communication is considerably higher regarding to following iterations. Preferably we choose the single service data and maximum provider coverage data (A_j^{max}). Single service provides us data needed for proportional price reconstruction necessary due to aggregated price. And the max coverage data gives us the lowest possible prices from provider needed for approximation.

Iteration phases The iteration phases represent processes that follow each other in further negotiation stage.

- Contacting the best known consortium given by acquaintance model
- Updating IRAM model by the received responses
- Reevaluating the acquaintance model
- Sorting the deal space by total consortium price
- Termination condition evaluation

Termination condition The algorithm is iterating (contacting and updating model) till the best evaluated consortium consists of fixed points only (e.g. no part of consortium has estimated evaluation)

The steps of IRAM algorithm can be seen in Figure 2.

Fig. 2. IRAM algorithm steps

```
1 Construct deal space $DS_t$.
2 Send CFP($s_i$) for all $s_i \in t$ to all providers $P$.
3 Update $am$ according to received responses.
4 Send CFP($A_p^{max}$) to all providers $P$.
5 Update $am$ according to received set of responses $A_p$.
6 Select $Cons^{best}$ from $DS_t$ evaluated by $am$.
7 If $Cons^{best} \subset \bigcup_{j \in P}(\Delta\mathcal{F}_j^{am})$ then terminate algorithm.
8 Send CFP($A_{p,j}$), where $\bigcup_{j=1 \ldots n} A_{p,j} = Cons^{best}$ to providers $1 \ldots n$.
9 Goto 5.
```

3.3 Properties of IRAM

The presented IRAM algorithm is sound and complete. The proof of completeness of the algorithm is made through conversion of whole idea to A^* algorithm [6], where the nodes of searched space are individual consortia from DS, and edges represent inclusion (or exclusion) of one provider to a consortium. This representation corresponds to *Coalition Structure graph* [7].

The heuristics of A^* is then based on acquaintance model approximation, all of the nodes are priced particularly by real prices (fixed-points) and by computed prices given by acquaintance model. The price of the consortium $Cons$ is represented by

$$c(Cons) = g(Cons) + h(Cons),$$

$$\text{where } g(Cons) = \sum_{A_j \subset \Delta(\mathcal{F}_j^{am})} (\mathcal{F}_j^{am}(A_j)) = \sum_{A_j \subset \Delta(\mathcal{F}_j^{am})} F_j(A_j),$$

$$\text{and } h(Cons) = \sum_{A_j \subset \mathcal{P}(A_j^{max})/\mathcal{P}(\Delta(\mathcal{F}_j^{am}))} \mathcal{F}^{am}(A_j). \tag{8}$$

The $g(Cons)$ represents price of the of subsets from $Cons$ that is known from previous negotiations (the fixed-points) and $h(Cons)$ is the unknown price of the subsets from $Cons$ estimated by acquaintance model.

The non-increasing individual pricing function assumption causes

$$s_x \in A_{j,1}; s_y \in A_{j,2}; s_y = s_x; |A_{j,1}| \leq |A_{j,2}|$$

$$\Rightarrow f_j(s_x, |A_{j,1}|, pc_j) \geq f_j(s_y, |A_{j,2}|, pc_j) \tag{9}$$

According to eq. 6 and

9 the $h(Cons)$ is always equal or lower then the real price of this subset, so $h(Cons) \leq h^*(Cons)$ and the eq. 8 is admissible heuristics of A^* algorithm.

Since the IRAM is based on exploration of the best candidates evaluated by eq. 8 the algorithm provides the features of A^* algorithm [6].

3.4 Reference Algorithm

The presented approach has been empirically validated by comparison with the state-of-the-art algorithm with behavior similar to IRAM algorithm. Generally the IRAM method is used to approximate unknown pricing functions of partners and determine the direction of future negotiation. For the benchmarks purposes we have implemented a reference algorithm based on deployment of known Chebyshev polynomials [8], previously used for modeling of sellers production pricing function.

The *Chebyshev polynomials* exactly Chebyshev polynomials of the first kind are defined as

$$T_n(x) = \cos n \arccos x \tag{10}$$

Due to its orthogonality with respect to the weight $w(x) = (1 - x^2)^{-1/2}$ in the interval [-1,1] are widely used for approximation, in most cases they are more effective than Taylor's. The Chebyshev polynomial state can be represented by the recursion formula: $T_{n+1} = 2xT_n(x) - T_{n-1}(x)$, where $T_0 = 1; T_1 = x$. Further information can be obtained in [9].

The approximation itself is made through computing weights (c_k). They represent the contribution of every Chebyshev polynomial to the resulting function. The complete approximation formula is

$$f(x) \approx \sum_{k=0}^{N-1} c_k T_k(x) - \frac{1}{2} c_0 \tag{11}$$

The defining weights are reconstructed from approximated known points $(x_k, f(x_k)), k = 1..M$ with formula

$$c_j = \frac{2}{M} \sum_{k=1}^{M} f(x_k) T_j(x_k) \tag{12}$$

The values of x_k should be mapped to $[-1, 1]$.

3.5 Implementation

As mentioned above, searched approximation method was selected according to initial environmental conditions similar or equal to IRAM's. In case of Chebyshev polynomials the conditions matched exactly. The implementation of the IRAM algorithm was just slightly modified in the field of approximation and all the other interfaces (like communication, consortium construction and evaluation) were left untouched. For a proper explanation of its deployment, let us explain the adaptation of Chebyshev polynomials to the environment. The first two samples (fixed points), needed in general for every approximation, are gathered from the initiation phase of negotiation (same as IRAM). The deal space is then evaluated from Chebyshev approximation of the pricing functions. The final condition is also same as in IRAM. If the same solution is evaluated as the best in two following iterations it is returned as result.

As mentioned previously, the only alteration (from the IRAM algorithm) in the optimal consortium search was the approximation process. Just like in IRAM there is a pricing function model for every particular service from every provider, this model consists of set of weights (eq. 11), the number of weights depends on the number of Chebyshev polynomials used. The number is also correlated with approximation quality (will be further described). The origin of the polynomials is the same for all approximations thus is stored as a constant. The exploiting of the incoming information takes place during the weight computation, which is performed as a result of collecting each new price information.

4 Experiments

The key contribution of the presented paper is in empirical evaluation of the presented algorithm. We will be analyzing the behavior of IRAM in relation to the reference algorithm presented above.

For presenting the contribution of IRAM we construct a market model that contains a set of four providers P, one simple *requester* that requests providers for set of 17 tasks $S = t_1 \dots t_{17}$ using IRAM and reference algorithm for comparison.

In our experiments we randomly generated 4 providers, where everyone of them was capable of delivering 14 services from total 18 services, this setting was chosen due to computation requirements. The max size of *acquaintance model* is $\delta_{max}(\mathcal{F}_j^{am}) = 65532$ possible proposals i.e. *fixed points*, and average *deal space* size in one task is $\langle |DS_t| \rangle = 8777$. The individual pricing functions were randomly generated as follows. We generate uniform distribution set of prices $UP = (bp_1 \dots bp_s)$ in defined range $(200, 600)$ for base price bp of every service from U. Then we create one random value in range $d_j \in (0.6, 1)$ for particular provider p_j that represent the discount in price according to the number of total services asked $|A_j|$. From discount is then computed margin value m_j

$$m_j = 1 + 0.05 * ((1 - d_j)/0.8) * ((1 - d_j)/0.8 + 1)/2. \tag{13}$$

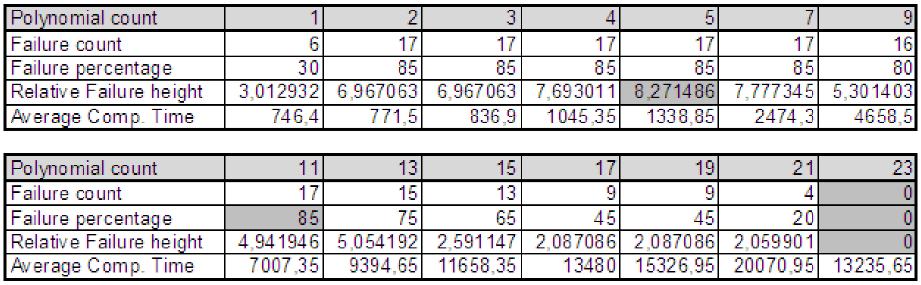
The price $f_{i,j}$ of single service s_i is then derived from base price bp_i and total service asked $|A_j|$ as follows.

$$f_j(s_i) = m_j bp_i d^{|A_j|-1} + 0.5 m_j bp_i \tag{14}$$

Then the pricing function corresponds to eq. 1 Every provider then responds only with this price when is asked for some services.

4.1 Quality of a model

As shown in [8], the quality of a approximation rises with the number of polynomials used. It can be shown that for a approximation of a polynomial function of a certain degree d the number of Chebyshev polynomials needed for exact approximation is also d. However in our case the pricing functions (eq. 1) are little bit different to be specified exactly like a polynomial. Therefore we run quality tests to find the right Chebyshev polynomial count. The results are shown in figure 4.1.



Polynomial count	1	2	3	4	5	7	9
Failure count	6	17	17	17	17	17	16
Failure percentage	30	85	85	85	85	85	80
Relative Failure height	3,012932	6,967063	6,967063	7,693011	8,271486	7,777345	5,301403
Average Comp. Time	746,4	771,5	836,9	1045,35	1338,85	2474,3	4658,5

Polynomial count	11	13	15	17	19	21	23
Failure count	17	15	13	9	9	4	0
Failure percentage	85	75	65	45	45	20	0
Relative Failure height	4,941946	5,054192	2,591147	2,087086	2,087086	2,059901	0
Average Comp. Time	7007,35	9394,65	11658,35	13480	15326,95	20070,95	13235,65

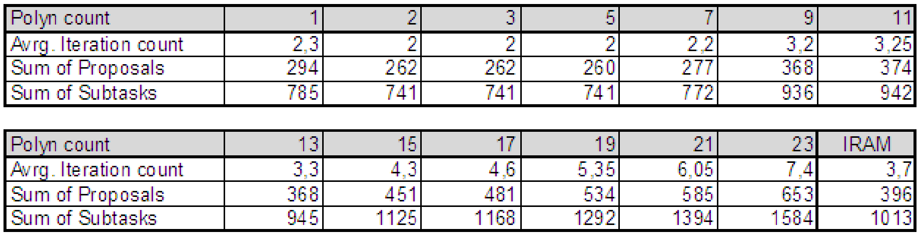
Fig. 3. Chebyshev approximation performance with different polynomial count

The scenario of the test is performed by 6 partners with 17 services each from 34 possible services. They are gradually asked for 20 tasks composed of 9 services. The pricing functions specifications are the same as usual.The size of the deal space is then according to the setting 19683. There is a set of tests on the same setting just with different count of Chebyshev polynomials used. Measured variables was *failure count* (finding the wrong optimum), *failure percentage*, *average relative failure* height (to the price of optimum) and *computational time* (on the same hardware setting in ms). The table shows us that the approximation has interesting numbers in one polynomial case, which represents the linear approximation with one straight line. The other results worsens with rising polynomial number until the peak at 11 polynomial for failure count and 5 for average relative failure height. The computational time is rising due to computing of higher and higher powers. By the polynomial count of 23 we can se ideal approximation of our pricing functions represented by 0 failures. So this setting we can use as a proper benchmark for IRAM in this particular environment.

4.2 Benchmarking IRAM vs. Chebyshev

Finally we can unveil the key comparison of Chebyshev method and IRAM. For the first set of tests we simply use the setting described in previous section in order to illustrate major differences. The measured variables are *average Iteration count*, *sum of asked proposals* and *sum of asked services.* The Table 4.2 providing measured variables with IRAM's results in the last column, clearly shows that with the same level of quality (zero failures) the shared information (gathered and paid) from partners is 50 % bigger in negotiation than using Chebyshev approximation. To be equal in information requirements we have to accept 85 %

chance to have 5 % failure with Chebyshev of the order 11. Just for clear comparison of the *average computational time* for one IRAM negotiation was 859,8 ms comparing to 13235,65 ms of Chebyshev of the order 23. The time results have just relative information value because the implementation of Chebyshev polynomial computation can be slightly upgraded, which is not the key object of this paper. For the next experiments we will use the Chebyshev polynomial of the order 11 and Chebyshev polynomial of the order 23 for benchmarking with IRAM's results.



Polyn count	1	2	3	5	7	9	11
Avrg. Iteration count	2,3	2	2	2	2,2	3,2	3,25
Sum of Proposals	294	262	262	260	277	368	374
Sum of Subtasks	785	741	741	741	772	936	942

Polyn count	13	15	17	19	21	23	IRAM
Avrg. Iteration count	3,3	4,3	4,6	5,35	6,05	7,4	3,7
Sum of Proposals	368	451	481	534	585	653	396
Sum of Subtasks	945	1125	1168	1292	1394	1584	1013

Fig. 4. IRAM vs. Chebyshev comparison

Another key comparison which can be shown on multiple task solving scenarios, is the progress of the harvested data utilization. Due to incrementally better and larger model sizes we can obtain the solution with lower negotiation requirements . Results of this experiments are shown in figures 4.2 and 4.2. In the first one there is the number of fixed points requested from the partners in every negotiation. This value has the meaning of shared information among the partners. We can see a notable computational overhead needed for Chebyshev of the order of 23 in the beginning of the negotiations. This trend then gets significantly better than IRAM numbers with increasing number of task solved. This is caused by bigger (thus better) Chebyshev of the order 23 model. The Chebyshev of the order 11 has almost the same data. It was chosen because of its equal data requirements as IRAM. On this particular data we can see progressive IRAM's character in the single task formation problem, i.e. the model is build from scratch. In the first step of the graph IRAM outperforms Chebyshev by 50 %. In unknown environments or in highly dynamic cases this capability become very useful. In the second figure we shown the sizes of each model (see eq. 5) and how this size continually grows with the number of negotiations. Again, we can see the overhead needed by Chebyshev of the order 23. It comes with the robustness of 23 Chebyshev polynomials and its requirements for initial data to construct the proper approximation. Trends for all of the algorithms are almost the same. With an increasing number of negotiations they tend to converge to certain bound where the models have mapped all interesting areas of partner's pricing functions. In those regions IRAM shows significantly lower need for collected information to provide the best solution.

As we can see, the deployment of IRAM algorithm in this type of domains brings significant improvement in performance, not just in information sharing field but with its simple implementation even in computational requirements.

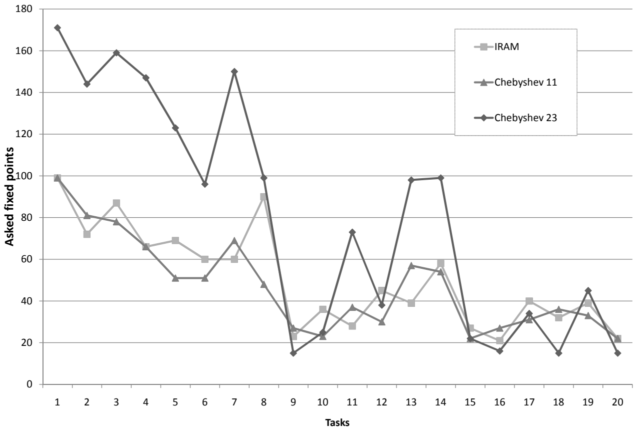


Fig. 5. IRAM vs. Chebyshev 11 and Chebyshev 23 asked fixed points in particular negotiations

5 Conclusion

The paper also presents a specific algorithm for distributed task delegation and resource allocation in semi-trusted multi-actor communities - the Incrementally Refined Acquaintance Model (IRAM). This algorithm is based on incrementally maintained social knowledge of the service requestor about the service providers. The novelty of the presented approach is in the fact that social knowledge is used even if very imprecise and it is gradually refined by means of unsuccessful attempts to contract.

The presented IRAM algorithm allows consortia composition with respect to defined requirements, mainly non-centralized approach and minimization of private knowledge disclosure. In environments with a certain degree of dynamics, IRAM's decent information requirement in initiative phase of building the model brings it a significant benefit in comparison with classical approximation approaches (Chebyshev). It can be outperformed in longer sets of negotiations due to more voluminous models representing the opponents.

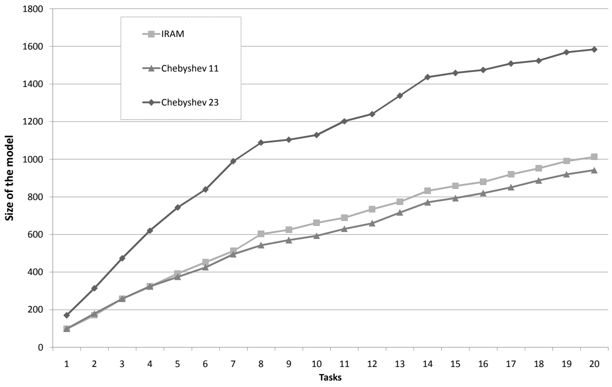


Fig. 6. IRAM vs. Chebyshev 11 and Chebyshev 23 in size of the models

High dynamics of the environment causes devaluation of acquaintance model and it can lead to incorrect solution. In such dynamic environment, the IRAM algorithm has to start building the model from scratch for every new task. In one deal scenario, the presented IRAM algorithm still provides quick convergence to the optimum with low communication and thus low private knowledge disclosure. Another option is limit the validity of the information obtained and reconstruct part of the model only.

Acknowledgment

The work is funded by the European Commission's FP6 programme within the projects Collaborative Process Automation Support using Service Level Agreements and Intelligent Dynamic Agents in SME clusters (PANDA), IST-No.027169. The research is also partly funded by the Ministry of Education, Youth and Sports of the Czech Republic grant No. MSM 6840770038 and No. 1M0567.

Any opinions expressed in this paper are those of the author(s)/organization and do not necessarily reflect the views of the European Community. The Community is not liable for any use that may be made of the information contained herein.

References

1. M. Pěchouček, V. Mařík, and J. Bárta, "Role of acquaintance models in agent's private and semi-knowledge disclosure," *Knowledge-Based Systems*, no. 19, pp. 259–271, 2006.
2. M. Pěchouček, O. Lerch, and J. Bíba, "Iterative query-based approach to efficient task decomposition and resource allocation," in *Cooperative Information Agents X*, ser. Lecture Notes in Computer Science, M. Klusch, M. Rovatsos, and T. Payne, Eds., vol. 4149, CIA Workshop. Springer, 2006, pp. 258–272.
3. W. Cao, C.-G. Bian, and G. Hartvigsen, "Achieving efficient cooperation in a multi-agent system: The twin-base modeling," in *Cooperative Information Agents*, ser. LNAI, P. Kandzia and M. Klusch, Eds., no. 1202. Springer-Verlag, Heidelberg, 1997, pp. 210–221.
4. V. Mařík, M. Pěchouček, and O. Štěpánková, "Social knowledge in multi-agent systems," in *Multi-Agent Systems and Applications*, ser. LNAI, M. Luck, V. Mařík, and O. Štěpánková, Eds. Springer-Verlag, Heidelberg, 2001.
5. J. Vokřínek, J. Hodík, J. Bíba, J. Vybíhal, and M. Pěchouček, "Competitive contract net protocol," in *LNCS 4362 – SOFSEM 2007: Theory and Practice of Computer Science*. Berlin: Springer-Verlag, 2007, pp. 656–668.
6. R. Dechter and J. Pearl, "Generalized best-first search strategies and the optimality of A*," *J. ACM*, vol. 32, no. 3, pp. 505–536, 1985.
7. T. Sandholm, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. Cambridge, MA.: MIT Press, 1999, ch. Distributed Rational Decision Making, pp. 201–258.
8. S. Saha, A. Biswas, and S. Sen, "Modeling opponent decision in repeated one-shot negotiations," in *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*. New York, NY, USA: ACM, 2005, pp. 397–403.
9. T. Rivlin, *Chebyshev Polynomials: From Approximation Theory to Algebra and Number Theory*. Wiley-Interscience, 1974.