Incrementally Refined Acquaintance Model for Distributed Planning in a Semi-Trusted Environment

Michal Pěchouček[1], **Austin Tate**[2] and **Gerhard Wickler**[2]
[1]Agent Technology Group, Gerstner Laboratory
Department of Cybernetics, Czech Technical University in Prague
[2] Artificial Intelligence Applications Institute
University of Edinburgh, School of Informatics,
E-mail: `pechouc@labe.felk.cvut.cz`

Abstract

This paper analyzes the distributed planning architecture proposed by Durfee (Durfee 1999) and discusses the properties of the suggested architecture in a peer-to-peer, dynamic and semi-trusted environment. The I-Globe integrated architecture linking classical planning algorithms with multi-agent technologies is presented (namely using the Hierarchical Task Network style I-X technology and $\mathcal{A}$**-globe** multi-agent system). As a key technical contribution of the paper we present a specific interaction protocol based on incremental improvement of the social knowledge of agents, the Incrementally Refined Acquaintance Model (IRAM), and provide justification for its deployment in a specific distributed planning scenario.

Introduction

[1]In this paper we will discus the role of a distributed planning architecture and specific planning techniques in a very specific environment that has been formulated in part by the project funding agency. The objective is to design a distributed planning architecture that would support a multi-actor environment that:

– is **non-centralized** and with flat organizational structure [R1] – the existence of a central coordinating and planning process shall be brought to an absolute minimum and the planning knowledge, information about the skills of actors, resource availability, knowledge and goal perception shall be distributed,

– **multi-party involvement** [R2] – the resulting plans cannot be implemented in isolation by a single actor, coordination and sharing of resources is required and the goals are set and the planning processes 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,

– allows **varying interaction availability** [R4] – based on a communication infrastructure featuring partial and temporal inaccessibility due to e.g. ad-hoc networking, unreliability of the communication infrastructure or actors to change off-line/on-line status,

– is **very dynamic** [R5] where both resource availability and goals are expected to be changing between the planning and execution phase, and

– is **opportunistic** [R6] – allowing the actors to reason about potential goal accomplishment opportunities that may arise in the environment and also consider opportunities of the collaborating actors in the environment.

Such a set of requirements is typical for rescue operations, complex humanitarian missions, multi-national coalition operations as well as small size military combat operations. These features are also typical for a completely different set of application domains such as virtual organizations and social networking.

The targeted deployment scenario covers multiple levels of the planning and execution process within a dynamically developing situation, involving a number of:

– humans (command authorities, planners and operators),

– mobile vehicles (such as trucks and multiple unmanned autonomous vehicles (UAV) with appropriate sensor suites),

– a network of unattended ground sensors (UGS),

– software agents (planning systems, sensor processing systems, semantic web resources of various kinds, etc.).

[1]The I-Globe project is sponsored by ERO - European Research Office of US Army under grant number N62558-06-P-0353. The authors' organizations and research sponsors are authorized to reproduce and distribute reprints and on-line copies for their purposes notwithstanding any copyright annotation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of other parties.

Due to the fully distributed nature of the planning problem, the planning architecture is based on multi-agent principles. Each of listed actors is be modeled as either an autonomous agent or a container hosting several intelligent agents. An example of a container could be a flying UAV hosting various software agents implementing the various aspects of its autonomous aspects (such as its planning agent, sensor processing agent, communication and negotiation agent, etc.).

The multi-agent planning system initiates the process of negotiation between humans, autonomous robotic agents, and virtual agents to acquire and maintain geographical and contextual information and to exchange plan-related information in a timely fashion.

The deployment scenario demonstrates automatic retasking of the autonomous agents (e.g. UAVs and other vehicles) through an agent-oriented collaborative process, where the devices with the appropriate capability (e.g., IR sensors, still or streaming imagery, chemical sensors) could be re-tasked based on some mission priority. When re-tasking, other capabilities might need to be reconfigured autonomously to ensure that they can maintain their overall mission tasks in the altered context (e.g., providing communications connectivity over a large geographic area for a mobile force).

The distritbuted planning and coordination task in the presented scenarios can be classified by a hierarchy of planning levels that correspond to very different perspectives:

1. **strategic** - problem analysis, sense-making, high-level strategic task setting and approach selection. This level can be supported by e.g. Compendium (http://www.compendiuminstitute.org) as a basis since this has already been demonstrated and successfully evaluated as useful in support of human military planners. There is the potential to use an issue-based approach to sense-making, option analysis, argumentation and decision support at this level. A combined Compendium and I-Plan tool was demonstrated and evaluated during the Collaborative Operations for Personnel Recovery (Co-OPR) project as part of Experiment B of DARPA's Integrated Battle Command program (Tate *et al.* 2006)[2].
2. **operational** - generation of responses/plans and refinement/repair of these dynamically as needed. This level could use mixed-initiative multi-agent planning approaches and a HTN approach as these have been shown to be at a level that relates well to human planners, who need to maintain and communicate a plan at a suitable level of abstraction. HTN approaches and plan representations also act as a bridge between the levels. This level could be based on, e.g., an I-X Process Panel (Tate *et al.* 1999; Tate, Dalton, & Stader 2002; Wickler, Potter, & Tate 2006). Its HTN planner, I-Plan, will need to be developed further. Refinement of the algorithms to add temporal and consumable/renewable resource constraint management at least as included in the earlier O-Plan planner (Currie & Tate 1991) will be necessary (Drabble & Tate 1994). Multi-level plan execution monitoring (Reece & Tate 1994) and repair algorithms already available for O-Plan should also be incorporated. The aim is to bring I-Plan as a module up to a level of competence to act as a restartable/incremental planner able to refine multiple options for a response and change these as circumstances change.
3. **tactical** - adjustment and further more detailed refinement of the plans to meet the stated objectives within the constraints given, but talking into account local circumstances and context. Here agent-based techniques (Rehak, Pechoucek, & Volf 2006; Pechoucek, Lerch, & Biba 2006) can be used for peer-to-peer negotiation among individual actors aimed at optimal responsibility delegation and resource allocation. ACROSS and $\mathcal{A}$**-globe** (Šišlák *et al.* 2006) provide the basis for this tactical planning layer within the developed application. Interaction within layers is inevitable. Linking the planning service to the tactical agents to assist them in refining and adapting their plans to local circumstances was done e.g. in the O-Plan/WorldSim Operational to Tactical planning and plan execution support techniques (Reece & Tate 1994; Tate 1989).

[2]See also http://www.aiai.ed.ac.uk/project/co-opr/expt/

I-Globe Integrated Architecture

I-Globe, the integrated technology of $\mathcal{A}$**-globe** multi-agent technology and I-X task-support and planning technology has been supported in parts by AIAI and ATG. AIAI is bringing to the project: (*i*) I-X technology including the I-Plan planning agent which is based on a combination of a human-relatable Hierarchal Task Network (HTN) approach coupled with rich constraint representation and satisfaction algorithms and (*ii*) `<I-N-C-A>`(Issues, Nodes, Constraints and Annotations) as a shared ontology suitable for relating the activities of human, vehicle, robots and sensors. ATG is providing to the architecture: (*i*) $\mathcal{A}$**-globe** multi-agent integration platform, (*i*), (*ii*) stand-in agent supporting agents' interaction in the situation of temporal communication inaccessibility and *iii* various agent based tactical planning mechanisms.

The facilities available in the I-X Process Panels include an AI planner (I-Plan) used to provide context sensitive options for the handing of issues (such as the achievements of stated objectives), the performance of activities, and the satisfaction of constraints i.e. to support the underlying `<I-N-C-A>` plan representation.

I-Plan can perform hierarchical partial-order composition of plans from a library of single level plan schemas or "Standard Operating Procedures". This library can be augmented during planning either with a simple "activity details" nterface to add in specific ways to expand a given action (intended for use by users familiar with the application domain but not AI planning techniques) or with a more comprehensive graphical domain editor. Grammars and lexicons for the domain are built automatically during domain editing to assist the user.

Future developments of I-Plan will be able to account for plan repair after partial failures, include handling of calendar time and consumable resource constraints, and account for mutual satisfaction of open variables and other constraints with greater efficiency.

$\mathcal{A}$**-globe** technology provides means for linking heterogenous software applications and reasoning algorithms. In addition to this, $\mathcal{A}$**-globe** facilitates modeling of the environment in which the autonomous actors interact as well as modeling of the behavior of the actors themselves. Therefore $\mathcal{A}$**-globe** is an ideal technology for development of the proof-of-concept prototype of the targeted scenario. $\mathcal{A}$**-globe** also includes basic tools for 2D and 3D visualization support.

there have been developed a number of different $\mathcal{A}$**-globe** integrated planning algorithms and other auxiliary mechanisms that support the distributed planning scenario.

$\mathcal{A}$**-globe** provides negotiation and auctioning mechanisms that facilitate distributed resource allocation. In $\mathcal{A}$**-globe** there is also integrated the ECNP auctioning mechanism that provides optimization of multiple rounds in the negotiation process. Performance of ECNP has been already verified on the ground logistics scenario (Rehak, Pechoucek, & Volf 2006). Therefore the requirements for decentralization [R1] of the planning process can be covered by $\mathcal{A}$**-globe** .

$\mathcal{A}$**-globe** also hosts agents with rich social models, the containers of agents mutual awareness. The social models are built from analysis of agents past interaction and their previous behavior. The social models can be used in the situations with partial knowledge availability or temporal communication inaccessibility (as specified by the requirements [R3] and [R4]. The current implementation of the social models is not very expressive and would need to be extended. Preliminary experiences verified the use of social model in semi-trusted logistics environment and the methods of incremental construction of the social models were deployed (Pechoucek, Lerch, & Biba 2006). However we need to build methods that would allow the agents to share their information about the quality of the social models and would alow their run-time integration. $\mathcal{A}$**-globe** integrates the classical A^*-based path planning mechanisms that are used for the $\mathcal{A}$**-globe** -based application for UAV collision avoidance. While these algorithms do not support even the tactical planning level (discussed in Section), they will be critical for some aspects of the final demonstration development.

Key research and technological challenges are in linking the classical planning functionalities of e.g. HTN planning with the very dynamic nature of the multi-agent simulation. This needs to be done for support of not only the [R1] requirement but mainly for the [R5] requirement for dynamics and [R6] requirement for opportunistic planning.

Distributed Planning

The problem of distributed planning (DP) has been often discussed in the AI planning and multi-agent research communities recently (e.g. (Durfee 1999), (DesJardins & Wolverton 1999), (Ephrati & Rosenschein 1997), (de Weerdt *et al.* 2003)). Distributed planning has been viewed as either (*i*) planning for activities and resources allocated among distributed agents, (*ii*) distributed (parallel) computation aimed at plan construction or (*iii*) plan merging activity. The classical work of Durfee (Durfee 1999) divides the planning process into five separate phases: task decomposition (TD), subtask delegation (SD), conflict detection (CD), individual planning (IP) and plan merging (PM). Unfortunately this satisfies that the requirements listed in Section only in parts.

Dynamic Environments: The requirements for planning architecture to support *very dynamic environment* [R5] implies that the computational process needs to be well balanced between the particular phases. Even though the process of planning in the presented environment is not performed in critical-time and very often there is enough time for the use of some of the classical heavy duty planners, for replanning (which may occur frequently) the minimal amount of computation and communication shall be performed. In spite of the fact that such an approach may produce sub-optimal solutions, it is required that the replanning is solved first in the phase of individual planning (IP). If no solution is found in the IP phase, we suggest deployment of the (SD) within a limited subgroup of collaborating actors. That is why, the individual actors are supposed to periodically monitor the activity status of their collaborating actors and be able to process the collected information (referred to as *social knowledge* (Mařík, Pěchouček, & Štěpánková 2001) in the field of multi-agent system). The first conclusion is that the Durfee's distributed planning architecture does not support the process of monitoring and thus replanning capability in very dynamic environment.

Opportunistic Planning: The *requirements for opportunistic planning* [R6] is not covered by the Durfee's distributed planning architecture at all. The planning process controlled by the five listed phases is based on very clear task decomposition (TD) phase without any consideration of the future opportunities of the actors. It is suggested that the presented distributed planning architecture is integrated with *Multiagent Opportunistic Planning* (MAOP) (Lawton & Domshlak 2004). MAOP is a very specific technique for collaborative planning and collaborative plan execution that is making the best use of sharing resources and sharing overlapping goals. The key idea is that each actor creates plans that also include opportunities for the other actors. If the opportunity goal becomes pending it can be achieved by other actors. The goals may became unachievable due to changes in the environment or they are unachieveble from the very start. In (Lawton & Domshlak 2004) there were tested several different strategies for selecting the additional goals for which an opportunity may arise. MAOP claim to be based on minimal knowledge sharing (in MAOP they share information about other agents capabilities and assigned goals). However this amount of knowledge sharing is still far too much in order to satisfy the [R4] requirement. Similarly, the current MAOP implementation does not allow for online replanning in the sense of dropping plans and adopting new plans instead, thus does not support the [R5] requirement.

Non-centrality: The requirements for *non-centrality and flat hierarchy* of actors [R1] and the requirements for *multi-party involvement* [R2] are the chief bottlenecks in deployment of the Durfee's distributed planning architecture. Durfee assumes that there is a single actor who is responsible for TD phase of one task. In contrary, these requirements assumes not only that there is no agent who can perform the requested task, but mainly that there is no agent who can plan how the requested task shall be performed. Consequently, the TD and SD phases can be hardly implemented by a centralized algorithms. While TD phase is concerned with availability of the problem solving knowledge for planning, SD phase needs the right amount of social knowledge that is needed for an appropriate resource allocation.

The R1 and R2 requirements in the TD phase are solved simply by agents broadcasting a query for an unknown planning operator. Non-centrality is however more critical in the SD phase. The appropriate choice of the subtask delegation mechanism depends on availability and the quality of social knowledge in the following way:

- Should social knowledge be available in a good quality to the agent, who is charged with the subtask delegation process, the classical centralized scheduling and resource allocation methods shall be used ((Bar-Noy *et al.* 2001), (Sycara *et al.* 1991)).
- Otherwise the subtask delegation problem is to be solved by multi-agent classical techniques such contract-net-protocol, various auctioning and combinatorial auctioning techniques (e.g. (Smith 1980), (Hunsberger & Grosz 2000), (Boutilier, Goldszmidt, & Sabata 1999), (Sandholm 2002))

The request is rejected during the CD phase, provided the it does not match with the agents capabilities, resource availabilities of collaboration preferences. Conflict may arise due to:

- usage of imprecise social knowledge (caused e.g. by confidentiality reasons, resources overestimation, etc.) used during TD and SD phases,
- agents deliberate overconstraining its responsibility during the SD negotiation process, or
- very frequent changes of agents availability (and thus changes of social knowledge) since the TD and SD phases (in case of very dynamic, real-time domains).

We claim that social knowledge availability affects implementation of the proposed DP architecture. Let us investigate the two extreme cases:

- There is high quality of *higher level social knowledge*[3] available, providing very precise information about available resources. In such situations splitting the phases TD and SD is inefficient and both processes will be be implemented by a single algorithm.
- Only *minimal social knowledge*[4] is available which results in the phase SD being implemented by means of negotiation. During this process the agents will avoid conflicting deals. In such situations the phase CD will be embedded within the phase SD.

This argument gives an impression that the DP planning architecture will be based on the separate TD, SD and CD proces in all but the listed two extreme cases. However, if the amount and quality of available social knowledge requires at least small amount of interaction in the phases SD it is unlikely that any sperate phase CD will be required as the conflicts will be avoided during the phase SD. On the other hand, if there is no negotiation and interaction required during the phase SD there is no reason for splitting the phases TD and SD. Consequently splitting the phases is reasonable only in the case where social knowledge availability is different in various situations and within different teams of agents.

Partial Knowledge Sharing: The requirements for *partial knowledge sharing* [R3] and *varying interaction availability* [R4] are linked in sense that the requirement R4 makes knowledge sharing difficult and thus knowledge is shared only partially. The other reasons for partial knowledge sharing may be e.g. limited trust within the multi-actor community or competence among the individual actors. In these cases the planning knowledge in the TD phase, the social knowledge in the SD phase, or even information about activity status needed for the PM phase are shared only partially.

The obvious solution to such problems is to employ the various planning methods that work with incomplete or inexact information (Pěchouček, Mařík, & Bárta 2006). In the environment with varying interaction availability some additional information can be provided by e.g. the *stand-in agent* technology. Stand-in agents, computational copies of the original actors,

[3] knowledge about agent's outer characteristics such as of agents awareness of agents reliability, trust, each other communication, computational and operational load but also information about price and a completion time

[4] information about agents IP physical addresses, port number, their ACL language they use for communication

either becomes online when the owner is off-line or migrates to such a part of the network that retains its connectivity with the other agents. Various distributed methods for optimal placement of the stand-in agents have been designed and investigated (such as forward swarming and backward swarming) stand-ins have been implemented and tested in ad-hoc networking environment [27].

In the communities with limited trust (causing partial knowledge sharing), agents are autonomously building their acquaintance models representing the monitored as well as deduced social knowledge. Autonomously collected and constructed social knowledge provides an important competitive advantages in such communities.

In the following text we will be presenting one specific technology for coping with partial knowledge availability.

Decomposition and Delegation in Semi-Trusted Communities

It is evident that the presented techniques do not satisfactory cover the whole DP architecture in the environment with the requirements R1-R6 listed in Section . In the second part of this paper we present an implemented algorithm for task decomposition and subtask delegation in semi-trusted communities.

Whoever is in charge of the TD phase (either an individual agent or a collective of agents), the quality of decomposition strictly depends on the quality of the social knowledge – in our case the information about agent capabilities and their availabilities. The difficult part is that while the agent capabilities are expected to be independent from the the suggested decomposition, agents availability (and perhaps cost) will differ for various suggested decompositions. Thus **decomposition depends on agents availability and agents availability depends on decomposition**. This problem is very important in the situations where the various decompositions differ in the amounts of requested services from the individual agents.

> **Example:** Let us have two providers who can provide any amount of service x, while for different amounts they each have different delivery times (see e.g. Figure 1). Provided that a requestor requests 1000 pieces of the service, we solve the problem how to decompose this amount into two batches for each of the provider. We assume that the delivery times for the various amounts of the service is not known a'priori and is regarded as private. Besides each of the two providers is motivated to disclose only the minimal information about its availability (thus the delivery time for the service).

The key idea behind the presented approach is as follows: Based on rather imprecise social knowledge, the agent suggests a preliminary decomposition. This allocation based on the suggested decomposition is sent to the the potential providers, who either accept or reject such an allocation. The rejection is supposed to be complemented by a counterproposal, that is used by the decomposition making agent for further improvement of its social knowledge, that is again used for yet another decomposition attempt. This process is repeated iteratively, until the decomposition is based on very accurate social knowledge. An important added value is in the fact that the agents are working only with the most relevant pieces of social knowledge.

Brief Problem formalization

Let us denote R as a requester agent, S as a type of service that requester R request in order to complete its task, a_t as a total amount of service S that requester R requests, n as a number of providers offering the service type S, P_j as a provider agent offering service S, where $j = \{1, ..., n\}$ and $d_j(a)$ as a duration for the agent P_j to deliver the amount a of service S.

The decomposition of the service S in amount a_t to agents $P_1, \ldots, P_n$ is an arbitrary vector of non-negative integers $(a_1, \ldots, a_n)$, where

$$a_j \in \mathbb{Z}^+, \quad \sum_{j=1}^{n} a_j = a_t. \tag{1}$$

If a_j equals 0, then the provider P_j is not contracted at all. The optimization problems is to find such vector $(a_1, \ldots, a_n)$ so that the overall duration is minimized as

$$(a_1^{min}, \ldots, a_n^{min}) = \arg\min_{a_1,\ldots,a_n} \max_j d_j(a_j) \tag{2}$$

for $j \in \{1, \ldots, n\}, a_j > 0$.

Solution

We have designed a straightforward decomposition mechanism that finds the most 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 (Pechoucek, Lerch, & Biba 2006)). Its behavior, however, worsens strongly with lower quality of information about the provider's resource availabilities stored in the requestors' acquaintance models. The most efficient approach in fully cooperative communities would be if the requestor queries all the providers and constructs the entire acquaintance model for all services provided by all agents prior computing the optimal a contract.

As this is not possible in the environment compliant with the requirements R3 and R4, the requestor needs to approximate such knowledge with only partially available information. Two algorithms were constructed for constructing the approximate, inexact acquaintance models:

1. *uniform acquaintance model* (UAM) and
2. *incrementally refined acquaintance model* (IRAM).

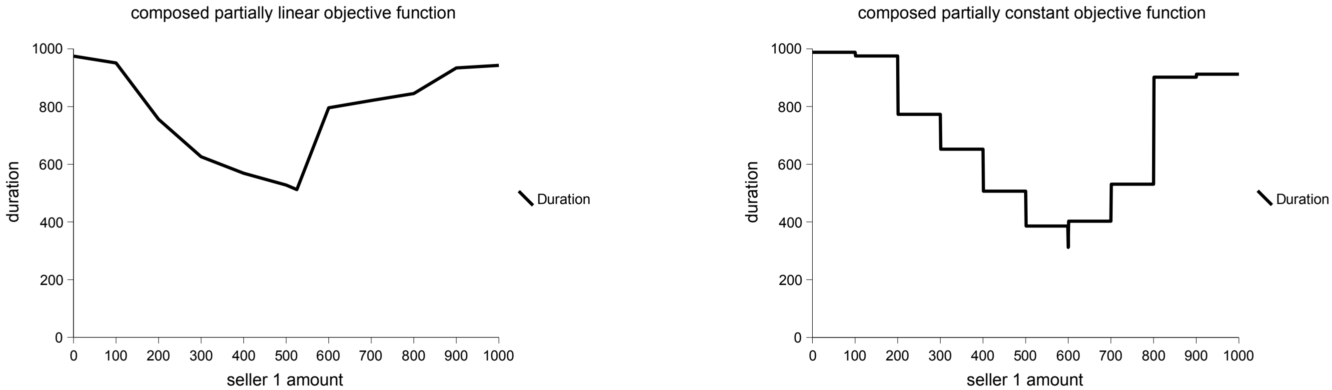

Figure 1: (a) Composed piecewise *linear* objective function — (b) Composed piecewise *constant* objective function

When the requestor builds the UAM, it divides the space of possible providers availabilities (for each resource) in N parts and requests the availability information for each part. With higher N the quality of the acquaintance model is better but it is usually more difficult to construct such model (e.g. due to providers intension to undisclose the information or when requestor needs to pay for the information).

The above presented algorithm has proved to be efficient and pragmatic solution for our decomposition/delegation problem. The key difficulty is that its behavior is parameterized by the constant N determining the amount of messages sent in the acquaintance model construction phase. As there is no a priori knowledge about the distribution of the providers' services, the appropriate granularity (and hence the parameter N) is unknown before the requests are broad-casted. This property makes the algorithm rather inflexible.

If N is reasonably high the model is very precise, while it also represents substantial amount of unneeded information. If the implementation does not allow parallel querying, this algorithm may also increase the communication traffic substantially. Not only communication traffic matters. We may also assume that every piece of information that provider discloses has to be payed for by the requester (this assumption reflects the fact that a high amount of information disclosure is likely to be just unwanted by the provider).

IRAM is based on flexible approximation of the available information. Only one specific amount of the requested service is queried before contracting is initiated. This value is used for very imprecise approximation that is encoded in the acquaintance model. Based on this imprecise model decomposition is computed and appropriate providers are queried for availability their resources. If the bids provided by the providers are close enough to the values approximated by the acquaintance model, the providers are contracted accordingly. If the bids are different then expected, the information provided in the bid is used for refinement of the acquaintance model. New decomposition is computed again and all the process is repeated until the bids are close to the values in the acquaintance model. See below for more detailed specification of the incremental acquaintance model refinement algorithm:

1. Let us set N equal to 2 and query all providers $P_1, \ldots, P_n$ for $a = \{1, a_t\}$, where a_t is desired total amount. We therefore estimate the real distributions by linear distributions which match with $d_j(a)$ for $a = 1, a_t$.
2. For those approximations we find optimal decomposition $a_1^{min}, \ldots, a_n^{min}$ and set of best suited providers.
3. For each specific amounts the requester broadcasts appropriate queries to the providers and collects the replies with the amount values suggested by the providers.
4. Provided that difference between computed amount and the amount actually offered by the provider is smaller than some specific Δ[5], the acquaintance model is regarded as precise enough for the specific contract and the decomposed amounts can be contracted. The algorithm terminates here.
5. If the algorithm did not terminate in the step 4, the requester inserts new value corresponding to the reply provided by the provider into its acquaintance model and carries out new linear approximation and return to step 2.

Experiments

Properties and efficiency of the presented mechanism have been tested in a simple experimental setting according to the example in Section . We have tested *piecewise linear* and *piecewise constant* distribution of $d_j(a)$ – real provider's resource availability in time. While the former distribution may represent e.g. duration of processing a certain amount of data in a gridthe latter can represent e.g. total delivery time of a specific commodity that may be transportees in a number of batches.

[5] an a priori defined error of the admissible acquaintance model precision.

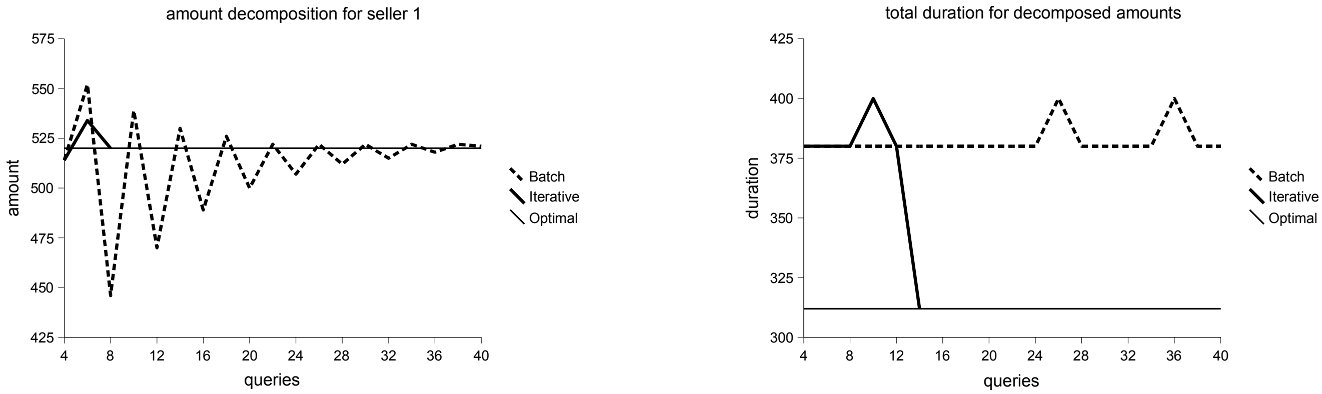

Figure 2: Approximation of piecewise *linear* (a) and *constant* (b) distribution in acquaintance model

Figure 1 illustrates the combined delivery distribution for 1000 pieces delivered by two agents with piecewise linear (a) and piecewise constant (b) distribution function. Minimum of these distributions need to be found during the process of decomposition (i.e. 520/480 for piecewise linear and 600/400 for piecewise constant distribution).

In Figure 2a the straight solid line shows the optimum decomposition. The thick dashed line gives the solution suggested by the acquaintance model construction mechanism based on uniform partitioning of the delivery distributional in to N (horizontal axis) parts and further linear approximation (carried out before any negotiation). It is seen that this mechanism provides results close to optimum with the space partitioned into 40 parts (and thus requiring 40 pieces of exact data from the provider) The thick solid line gives the solution suggested by the incremental AM refinement mechanismproviding optimal solution after 8 iteration of the algorithm.

The optimal delivery time of 512 time units is estimated by the UAM constructed model after sending 40 queries (i.e. 20 iterations – in each iteration queries are sent to both the providers) while IRAM construction mechanism requires only 8 queries.

An interesting result has been obtained when working with the piecewise constant functionswhere the optimal solution lies on the specific pike.

Figure 2b show that uniform partitioning algorithms have not find an optimal decomposition even after 40 queries and delivery due date has been estimated 380, while the optimal delivery is 312 units. The incremental AM refinement algorithm managed to find the optimal solution by means of 14 iterations.

It needs to be noted that besides obvious advantages, there also several disadvantages of the IRAM algorithm. Mainly, the IRAM algorithm is for the same quantity of message exchange generally slower. It is caused by the fact that in the case of UAM algorithm, messages are sent to every provider in parallel (or perhaps in a single message). In the case of IRAM algorithm, new queries are generated on the ground of previously received proposals. Therefore, we think that the IRAM algorithm is particularly suitable in the domains, where:

- (i) the service amount granularity is very fine and it is technically impossible to enumerate all the amounts and (ii) the N parameter is not known a priori
- where the providers are motivated to minimize the amount of disclosed information (or the requester needs to pay for every information it receives when building the acquaintance model)
- getting the right $d_j(a_j)$ values takes the providers specific amount of time (e.g. given by measurement or non-trivial computation)

In real-life the $d_j(a_j)$ values in competitive environments also often depend on various other aspects such as past contracting track record, other providers providing to the same requester, providers not providing the respective requester, trust, etc.

A major disadvantage of the presented decomposition method is the fact that its results substantially depend on the chosen accuracy of the acquaintance model (i.e. the choice Δ for the IRAM construction mechanism). Both the underestimation and overestimation of the objective functions may cause a deviation of the achieved decomposition from the optimum that possibly result in an increase of the real delivery time with respect to the expected $d_j(a_j)$.

Conclusions

This paper challenges the classical architecture of distributed planning and explains why it is not sufficient for highly decentralized, dynamic, semi-trusted, opportunistic environments. It briefly introduces the key elements of the I-Globe integrated architecture that is linked to classical AI planning algorithms with multi-agent interaction and planning approaches as embodied in the I-X technology and related I-Plan planner.

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 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.

References

Bar-Noy, A.; Bar-Yehuda, R.; Freund, A.; Naor, J. S.; and Schieber, B. 2001. A unified approach to approximating resource allocation and scheduling. *J. ACM* 48(5):1069–1090.

Boutilier, C.; Goldszmidt, M.; and Sabata, B. 1999. Sequential auctions for the allocation of resources with complementarities. In *IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 527–523. Morgan Kaufmann Publishers Inc.

Currie, K., and Tate, A. 1991. O-plan: the open planning architecture. *Artificial Intelligence* 52:49–86.

de Weerdt, M. M.; Bos, A.; Tonino, J.; and Witteveen, C. 2003. A resource logic for multi-agent plan merging. *Annals of Mathematics and Artificial Intelligence, special issue on Computational Logic in Multi-Agent Systems* 37(1–2):93–130.

DesJardins, M. E., and Wolverton, M. J. 1999. Coordinating a distributed planning system. *AI Magazine* 20(4):45–53.

Drabble, B., and Tate, A. 1994. The use of optimistic and pessimistic resource profiles to inform search in an activity based planner. In *Artificial Intelligence Planning Systems*, 243–248.

Durfee, E. H. 1999. Distributed problem solving and planning. In Weiß, G., ed., *A Modern Approach to Distributed Artificial Intelligence*. San Francisco, CA: The MIT Press. chapter 3.

Ephrati, E., and Rosenschein, J. S. 1997. A heuristic technique for multiagent planning. *Annals of Mathematics and Artificial Intelligence* 20(1–4):13–67.

Hunsberger, L., and Grosz, B. J. 2000. A combinatorial auction for collaborative planning. In *ICMAS '00: Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS-2000)*, 151–158. IEEE Computer Society.

Lawton, J. H., and Domshlak, C. 2004. Multi-agent opportunistic planning and plan execution. In *ICTAI '04: Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'04)*, 408–415. Washington, DC, USA: IEEE Computer Society.

Mařík, V.; Pěchouček, M.; and Štěpánková, O. 2001. Social knowledge in multi-agent systems. In Luck, M.; Mařík, V.; and Štěpánková, O., eds., *Multi-Agent Systems and Applications*, LNAI. Springer-Verlag, Heidelberg.

Pechoucek, M.; Lerch, O.; and Biba, J. 2006. Iterative query-based approach to efficient task decomposition and resource allocation. In Klusch, M.; Rovatsos, M.; and Payne, T., eds., *Cooperative Information Agents X*, volume 4149 of *Lecture Notes in Computer Science*, 258–272. CIA Workshop.

Pěchouček, M.; Mařík, V.; and Bárta, J. 2006. Role of acquaintance models in agent's private and semi-knowledge disclosure. *Knowledge-Based Systems* (19):259–271.

Reece, G. A., and Tate, A. 1994. Synthesizing protection monitors from causal structure. In *Artificial Intelligence Planning Systems*, 146–151.

Rehak, M.; Pechoucek, M.; and Volf, P. 2006. Multilevel approach to agent-based task allocation in transportation. In Klusch, M.; Rovatsos, M.; and Payne, T., eds., *Cooperative Information Agents X*, volume 4149 of *Lecture Notes in Computer Science*, 273–287. Springer.

Sandholm, T. 2002. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence* 135(1-2):1–54.

Smith, R. G. 1980. The contract net protocol: High level communication and control in a distributed problem solver. *In IEEE Transactions on Computers* C-29(12):1104–1113.

Sycara, K.; Roth, S.; Sadeh, N.; and Fox, M. 1991. Resource allocation in distributed factory scheduling. *IEEE Intelligent Systems and Their Applications* 6(1):29–40.

Tate, A.; Levine, J.; Dalton, J.; and Aitken, S. 1999. Supporting the planning process using open planning process panels.

Tate, A.; Buckingham Shum, S.; Dalton, J.; Mancini, C.; and Selvin, A. 2006. Co-opr: Design and evaluation of collaborative sensemaking and planning tools for personnel recovery.

Tate, A.; Dalton, J.; and Stader, J. 2002. I-p2 - intelligent process panels to support coalition operations. In *Proceedings of the Second International Conference on Knowledge Systems for Coalition Operations*.

Tate, A. 1989. Coordinating the activities of a planner and an execution agent. In Rodriguez, G., ed., *Proc. NASA conf. on space telerobotics*. NASA JPL: JPL Publications.

Šišlák, D.; Rehák, M.; Pěchouček, M.; and Pavlíček, D. 2006. Deployment of a-globe multi-agent platform. In *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, 1447–1448. New York, NY, USA: ACM Press.

Wickler, G.; Potter, S.; and Tate, A. 2006. Using i-x process panels as intelligent to-do lists for agent coordination in emergency response. *International Journal of Intelligent Control and Systems (IJICS), Special Issue on Emergency Management Systems*.