Distributed Planning Algorithm for Coalition Logistics in Semi-trusted Environment

Martin Rehák, Michal Pěchouček and Přemysl Volf
Department of Cybernetics and Center for Applied Cybernetics, Czech Technical University
Technická 2, Prague, 166 27, Czech Republic
{mrehak,pechouc,volf}@labe.felk.cvut.cz

Abstract

We present a collective approach to coalition logistics planning that presents the features crucial for application in an adversarial environment: planning and communication efficiency, well-defined levels of information to be shared, tight integration of trustfulness with the planning and stability with respect to unprecise trustfulness values. To achieve this goal, we combine multi-agent negotiation with efficient fuzzy and flexible linear programming techniques from operation research field. Alternating rounds of global optimization and restricted negotiation split the task into subtasks, create teams, assign them to the tasks and provide a task-resource mapping. Resulting plan execution can be easily verified and verification results can be used to update the trust and social models and potentially to perform re-planning immediately.

1 Introduction

One of the important problems of any coalition operations is a transportation logistics. Each coalition member can manage its own logistics independently, causing significant inefficiencies, resource overbooking and delays in operation. Alternatively, the logistics can be managed by cooperation among the autonomous coalition members.

We study coalition planning problem in the OOTW (Operations other than War) environment, where besides the obvious technical requirements on technical (hardware-level) interoperability and a shared knowledge model (ontology) the cooperative approach requires:

1. methods for building and using *trust models* about coalition members,
2. techniques for handling imprecise information and handling temporary *communication inaccessibility*,
3. knowledge sharing methods preventing unwanted *knowledge disclosure*,
4. *distributed reasoning* (planning) and negotiation algorithms and
5. agents ability to act in *cooperative* while also in *competitive* modes.

In this contribution we intended to address primarily the issue 1, 3, 4 and 5 from the list above. The issues of communication inaccessibility coalition planning has been discussed in [17]. The problem of transportation logistics in OOTW environment can be easily generalised so that the investigated concepts can be used in other semi-trusted, collective activity oriented domains.

The technique we suggest is not only applicable for current coalition operations, where the agents are typically materialized by manned command centers, but even more in the context of the next generation of systems with autonomous vehicles [1], where we may wish to automate not only driving and control, but also the higher level functions – making the vehicles with embedded agents responsible for transportation scheduling and road planning following the requests from the supplied entities.

Unlike classical definitions of competitive and self-interested behaviour, adversariality [16] is understood in this article in the context of a special agents' behaviour that is results in a partial loss of collective welfare of the collective of agents. Adversariality models agent harmful and malicious behaviour in the coalition. While agent's adversariality in the system can be both voluntary or intentional, often caused by coalition member with side interests or involuntary or unintentional, given e.g. by system infiltration.

In the transport logistics domain we have modeled two types of adversarial behaviour: (*i*) adversarial agent stealing the cargo and thus preventing it from being delivered and (*ii*) adversarial agent sharing the transport plan details with a third party, increasing hold-up probability.

In our work, we handle adversariality by: *(i)* **limiting information disclosure** to other agents, respecting each agent's private preferences, and keeping them undisclosed, *(ii)* **integration of trust model** [15] and methods of reasoning about competitive and adversarial agents. In such an environment we were motivated by suggesting a planning solution that would **stable** - we want a stable planning solution even if a trustworthiness or availability of partner agents changes slightly and finally and **efficiency** - we want to be able to find a task decomposition and allocation within reasonable time and with a small number of messages.

The algorithm we present builds on existing multi-agent solutions for the same class of problems. Extended Contract-Net-Protocol as defined in [5] and further extended towards practical application by [12] achieves the same result using the negotiation between coalition leader and perspective members. When the perspective coalition leader wishes to solve the task, it asks other agents to cover the task completely or at least partially. Agents submit their bids, the best ones are selected and provisionally granted the task. The rest of the task is auctioned again and new auctions are organized until the whole task is covered. If the remaining task can not be covered, the algorithm must achieve consistency by backtracking – revocations of provisionally granted tasks and auctioning new ones. Even if we have a unique coalition leader, the planning problem is completely decentralized and requires intensive communication. Consequently, this approach presents performance problems when it prepares the initial plan in large state spaces – even if such planner compares favorably with humans [13, 6], it can be easily beaten by mathematical programming techniques. On the other hand, the agent approach brings more flexibility than mathematical programming as the agents may combine many sources and types of knowledge to prepare the plan, each agent contributing its knowledge, reasoning and resources. Agent's don't need to be aware of each other's mental states, provided that they are syntactically and semantically interoperable.

In this contribution, we integrate classical AI and operational research 'heavy-duty' solvers in the context of multi-agent systems. We argue that the abstract models of collaboration in agent systems as they are now used within the multi-agent system community have severe drawbacks – they are well suited for simple reasoning and limited amount of knowledge, while little scalable. Their performance tends to degrade with increasing problem complexity. Therefore, we propose that the AI/OR techniques are a very good fit for agent reasoning due to their high performance and little or no scalability problems. The traditional problems related to their application – restrictive applicability conditions (e.g. linearity) are solved by modern methods [3] and on the other side, acquaintance models [11] provide the necessary knowledge inputs for the model, as well as an efficient mechanisms for its maintenance. As the mechanism we propose is intended to function in adversarial environments, we need to augment the social model with trustfulness information, using trust and reputation models presented in [15] or other [14]. Such trust mechanism must comply with following requirements: (i) trust and reputation must be integrated with the planning mechanism, (ii) model must be robust with respect to environmental noise (natural failure), (iii) its inputs must be compatible with the observed plan outcome.

2 Algorithm Presentation

In the logistics planning problem we consider, we address the transport of goods from initial to terminal location[1] using the resources belonging to self-interested and potentially adversarial agents. Therefore, we must select appropriate routes from the plan base, combine them and allocate resources to the tasks in the plan in order to maximize the expected amount of delivered goods. In the formal problem presentation below, we present the problem from the perspective of the coalition leader – the agent denoted A_0 that organizes a coalition.

2.1 Problem Formalization

Formally, we follow the approach proposed by [18] and instead of decomposing the plan into the action-state graph, we will describe it using actions and objectives (called objects in [18]). Therefore, we define an ***abstract plan*** (e.g. route plan) as a directed bipartite graph, where one side is composed of ***objectives*** (typically corresponding to locations), defined by the set $O = \{o_0(initial), o_1, o_n(terminal)\}$, with each member defined as $o_i = (prer_{o_i}, allows_{o_i})$, where both the $prer_{o_i}$ and $allows_{o_i}$ are subsets from the Ac, while the other graph side contains ***actions*** (transports) linking the objectives, defined in the set $Ac = \{a_1, a_2, ...a_m\}$, where again $a_i = (prer_{a_i}, allows_{a_i})$ and sets $prer_{a_i}$ and $allows_{a_i}$ are subsets of O. By definition, we always start from a single *initial objective* o_0 (with no prerequisites: $prer_{o_0} = \emptyset$) and terminate in a a *terminal objective* that corresponds to the achieved goal state: $allows_{o_n} = \emptyset$. [2] Besides the structural information, we also keep Θ_{a_i} for each action – an estimate of action success likelihood obtained from trust model in a same manner as individual agent trustfulness.

Batches constitute the cargo that is transported. Each batch p_i, from P is defined by its size $size(p_i)$ and *type* (liquid, bulk, etc...) that defines the resources that may carry it. By definition, all batches can be split during transport; we denote $p_i^{a_j}$ the part of the batch allocated to action a_j.

[1] This formal simplification doesn't reduce the generality of our approach - in case of need, we may define formal zero-cost actions between the initial/terminal objective and the real terminal objective for each part of the cargo, provided that we impose appropriate restrictions on these actions.

[2] Therefore, in our graph, the nodes are defined as $Ac \cup O$, while the directed edges describe the relations expressed in $allows$ and $prer$ sets of each action or objective. We may also note that the global state of the system is defined by the state of all objectives.

The transport problem is being solved by ***agents*** from the set $Ag = \{A_0, A_1, ...A_k\}$. Each agent A_i is characterized by its trustfulness $\Theta_{A_j}(A_i)$ as it is perceived by agent A_j, and its complement: *distrustfulness* $\Delta_{A_j}(A_i) = 1 - \Theta_{A_j}(A_i)$. Trustfulness $\Theta_{A_j}(A_i)$ is modelled as a *fuzzy number*, following [15]. This representation allows to correctly represent the uncertainty and can be used in the planning as described in Section 3 Agent is therefore modelled by leader the as a tuple $A_i = (\Theta_{A_0}(A_i), res_{A_0}(A_i))$, where the set $res_{A_0}(A_i)$ models leader's knowledge about agent's resources. Each agent controls one or more ***resources*** as defined in its set res_{A_i}. All resources, regardless of the agent they belong to belong form a set $R_{A_0} = \{r_1^{A_i}, r_2^{A_j}, r_l^{A_j}\}$, where the super index of each resource denotes the agent to which this specific resource belongs. Each resource is described by a tuple $r_i^{A_j} = (A_j, allowed_{r_i}, cap_{r_i})$, where the A_j denotes the owner agent of the resource, $allowed_{r_i}$ is a set of actions (transports) to which the resource can be assigned, and cap_{r_i} its capacity.

Tasks are a result of the planning process. They form a set $T = t_{a_1}, t_{a_2}, ..., t_{a_m}$, and each task corresponds to one action. Task is defined as $t_{a_i} = (batch_{t_{a_i}}, com_{t_{a_i}})$, where $batch_{t_{a_i}}$ is a set of batches transported in the task and $com_{t_{a_i}}$ is a set of ***commitments*** – each commitment[3] $c = (a_i, A_j, r_k^{A_j}, p_l^{a_i}, cap)$ is an assignment of a specific resource r_k (and consecutively its owner A_j) to one partial batch $p_l^{a_i}$ from the set $batch_{t_{a_i}}$ and cap determines the capacity that is to be assigned. If the r_k capacity allows it, one resource can be committed to more than one batch/action and a single partial batch $p_l^{a_i}$ can be covered by several commitments – in such case, we denote $cap(r_k^{a_i})$ the aggregate size of all commitments from the task t_{a_i} to which the resource r_k is committed. Commitments of resources relative to a single task define a ***team*** from the set $E = e_{a_1}, e_{a_2}, ..., e_{a_m}$. Each team $e_{a_i} \subset Ag$ contains all the agents contributing their resources to the task t_{a_i}. ***Coalition*** Co is defined as a union of all teams from the set E.

2.2 Public, Semi-Private and Private Information

Sharing the information about resources, plans, goals and intentions is significantly different from the cooperative agent systems. In adversarial environments, agents must seriously consider the possibility of information misuse and try to find the equilibrium between minimum information disclosure and cooperation and planning efficiency. Therefore, following [10], we define three types of information:

Public information is accessible to any agent in the system. It includes information about agent identity, existence, location and basic annotation of provided services - *type* of the resources $res_{A_0}(A_i)$ it offers, but without any information concerning their capacity, number or restrictions.

Semi-private information facilitates the planning process. It is mutually shared within groups of trusted cooperators that collaborate frequently and enables them to prepare the plans easier than by negotiating through all possible options [10]. For each agent A_i, it includes the information about its resources *aggregated* by type and including the restrictions regarding their use on the set Ac. Such compromise provides enough knowledge for the first stage of the planning process, and detailed task allocation is then finalized in course of negotiation without exposing more data than necessary[4].

Private information is reserved only to the owner agent and never shared with anyone else - it contains the detailed information about its resources, including their individual capacity, restrictions, locations and other information.

2.3 Algorithm Overview

This section provides an overview of the planning algorithm we suggest, combining the social model and linear programming planner with focused and well-targeted negotiations in the later stages of the process. The planning process proceeds as follows (see also Fig 1):

1. **Initial Planning**: Team leader uses its social knowledge and planning capabilities in order to prepare initial plan. This happens in two phases: *(i)* abstract plan construction and *(ii)* task allocation to the agents.

2. **Local Plan Evaluation**: Initial plan is evaluated by the respective agents: *(i)* the members evaluate the plan and make an attempt to trade the commitments within teams and *(ii)* proposals are sent by members back to the leader.

3. **Coherence & Verification**: Proposals are included in the detailed planning that ensures the plan coherence.

4. **Plan Execution**: Final commitments are received by members, may be swapped and the plan is executed.

2.3.1 Initial Planning

In the first phase of the plan, we assume that the coalition leader A_0 has a goal to accomplish and is obliged to form a coalition with other agents. It uses its social knowledge to draft a preliminary plan in the following steps.

Constructing the Abstract Plan. The first step is a preparation of the abstract plan – an action-objectives bipartite graph capturing the relationship between initial and terminal objectives (states) – typically covering alternative solutions. The graph must contain at least one path connecting the initial and terminal objective – if such path can't be identified, agent A_0 is unable to solve the planning problem.

[3] Formally, until being evaluated and updated by bidding agents, commitments must be regarded to as mere *commitment opportunities*.

[4] Note that the sets R as perceived by various agents are not identical due to the fact that they don't have the access to the same information.

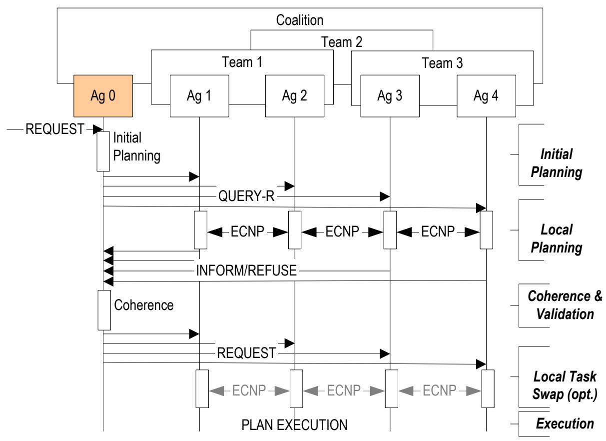

Figure 1. Overview of the protocol phases: Agent A_0 is a coalition leader and has decomposed the global task into three tasks.

Constructing the abstract plan is a computationally exponential problem in complex domains. Recent advancements in the field of AI planning provided very efficient techniques for constructing the plans in reasonable amount of time such as GraphPlan [8] or SAT-Plan [2]. These techniques implement a sophisticated breadth-first search based on expansion of the bipartite graph or iterative propositionalization of the planning problem.

Task Allocation. Once an acceptable abstract plan is established, leader proceeds with the **(ii)** allocation of batches and resources to individual actions in the plan, while respecting the constraints defined in the objectives. Note that for sake of computational efficiency, some actions and objectives from the abstract plan can be removed during this phase if there are no resources or batches to allocate to them. Then, we use a fuzzy linear programming (FLP) that either provides an acceptable initial task allocation T, or identifies a constraint preventing the solution.

The constraints we define for the problem are the following. The first equation expresses the node equilibria - conservation of goods in each node.

$$\forall o_i \in O \setminus \{o_0, o_n\}, \forall p_j \in P : \quad (1)$$
$$\sum_{a_k \in prer(o_i)} size(p_j^{a_k}) \cdot \Theta_{a_k} = \sum_{a_l \in allows(o_i)} size(p_j^{a_l})$$

where the Θ_{a_k} represents the estimated action trustfulness (probability of completion) taken from the trust model (e.g. delivery ratio in our case) - it allows us to model the probable losses in the actions from the set $prer(o_i)$. It may range from 0 – no hope of delivery – to 1 , resulting in the same amount of resources allocated for outgoing cargo.

The initial node has a simpler relation, declaring that we can't take away more cargo than available:

$$\forall p_j \in P : size(p_j) \geq \sum_{a_l \in allows(o_0)} size(p_j^{a_l}) \quad (2)$$

while the terminal node doesn't introduce any constraint.

Furthermore, for each action a_i (elementary transport) and each batch p_j, we must ensure that the commitments cover the whole partial batch $p_j^{a_i}$ ($size(p_j^{a_i} \leq p_j)$ due to the possible parallelism):

$$\forall a_i \in Ac, \forall p_j \in P : p_l^{a_i} = \sum_{c \in com_{t_{a_i}} : batch(c) = p_l^{a_i}} cap(c) \quad (3)$$

then, we must also make sure that no resource is used beyond its capacity:

$$\forall r_i \in R : cap(r_i) \geq \sum_{a_j \in Ac} cap(r_i^{a_j}) \quad (4)$$

besides these restrictions, we need to set-up the **utility function** for which we optimize:

$$U_m = \alpha \cdot \sum_{p_i \in P} size(p_i^{o_n}) - \beta \cdot \sum_{c_j \in C} size(c_j) \Delta_{A_j}(ag(c_j)) \quad (5)$$

, where $p_i^{o_n}$ denotes the part of the batch p_i delivered to the terminal objective and $ag(c)$ the agent committing to c.

We minimize the expected amount of the cargo lost (second sum) and we balance the cost of losses and value of delivery (first sum) by setting the constants α and β to domain-appropriate values.The ultimate goal is to allocate the resources of the coalition members to cover most of the delivery, while minimizing the risk of the attack.

Once the solution of the above FLP problem is identified (see Section 3), leader determines all perspective coalition members (owners of resources assigned to various tasks) and queries each perspective member whether it is capable and willing to participate. Therefore, each perspective coalition member A_i is sent a structure: $cma_{A_i} = (A_0, coalmem, assign)$, where A_0 is a coalition leader, set $coalmem$ lists all coalition members and set $assign$ lists the relevant information about tasks the agent's resources are assigned to, defined as $(e_{a_j}, com_{t_{a_j}}(A_i))$, where j is an action (task) index and $com_{t_{a_j}}(A_i)$ are commitments suggested to agent A_i on task t_{a_j}.

2.3.2 Local Plan Evaluation

When the coalition members A_i (selected by the leader in the previous step) receive the coalition proposals from the leader, they must use their private knowledge to create the bid reflecting their preferences and local situation. At first, the agents must decide whether they trust the coalition leader and members sufficiently to cooperate with them, typically putting emphasis on their trust in the leader ($\Theta_{A_i}(A_0)$) and agents within the same teams ($\forall e_k : A_i \in e_k \forall A_j \in e_k \Theta_{A_i}(A_j)$). If the agent is confident enough with the coalition and proposed commitments, it will try to assign its resources to its commitments.

At this level, we handle several issues that are ignored by the leader's first-level planning – resource granularity (unknown to the planning agent due to the privacy issues) and relations between the resources assigned to different tasks. In the first round, each agent assigns its resources to the commitments that are the best fit for available resources, trying to cover all commitments. Then, it will offer the excess capacity of the resources assigned to the task t_{a_j} to all members of the team e_{a_j} using the multi-phase auction mechanism described in [5]. This step is designed to eliminate the resource allocation inefficiencies that are due to the possible leader's lack of knowledge about actual resources or a side effect of selected planning method. More formally (see also Fig. 2), to start the the negotiations, each agent A_i working on task t_{a_j} broadcasts a CFP message containing its free capacity to all team $e_{t_{a_j}}$. If the other team members are interested in using this capacity for the task they were allocated, they submit their bid. Agent A_i selects one or more bids and answers them with a temporary grant, making them binding *for the bidders*; other are temporarily refused. When the agent A_i participates in several teams, it can now reshuffle its resources between the tasks to use them in an optimal manner. Once the resource reallocation is terminated, all compatible temporary grants are confirmed, while the others may be refused (In case we the agent has replaced the original resource with a lower-capacity one.). If appropriate, agent can now offer the new free capacity for trading using the same protocol.

Note that the auctioning and negotiation takes place only within the single task team, therefore minimizing the knowledge dispersion and communication load. On the other hand, agents may therefore miss a better task allocation. Once the negotiation is finished, all team members send their answers to the coalition leader. The answer is a list of commitments that are actually *binding* for each agent, but may differ from those originally assigned to the agent as: **(i)** the agent is not always able to cover the whole assigned commitment and commits only to a part of the original commitment or **(ii)** it notifies the coalition leader about the transfer of the whole commitment or its part to other coalition member (this member lists this commitment in its turn as covered). When the agents submit their binding commitments to the coalition leader, they have an alternative to offer the free capacity of the resources they've

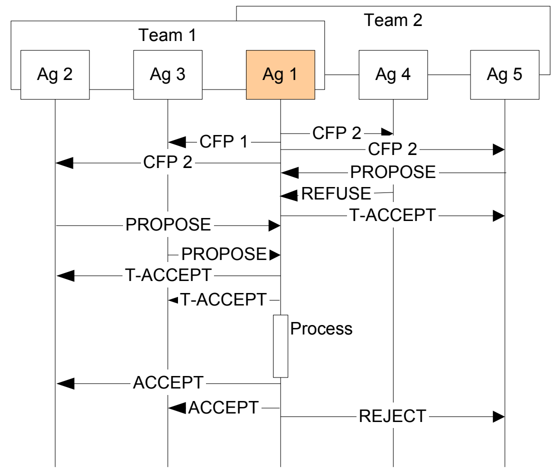

Figure 2. Use of the ECNP to allocate agent's resources across two different teams. Agent A_1 first temporarily accepts the offer from A_5, but later on finds a better resource allocation and prefers to commit larger resource to *team 1*. Therefore, it rejects the bid from A_5.

allocated to the task to the coalition leader - the leader may include use it to cover other batches from the same task, as specified by relation 6. While this remains an attractive optimization feature, this approach has two major drawbacks – the leader can easily guess the capacity of agent's resources and the free resources can not be used on another task.

2.3.3 Coherence & Verification Phase

In this phase, coalition leader receives the answers from the coalition members and must re-combine them into a globally coherent plan. As the initial planning has produced a coherent plan, the plan is coherent when all proposed commitments were covered by members. If not, the leader must add all updated commitments/refusals from the agents to the initial plan and perform the new calculation to make sure that the condition 1 is valid for the final plan.

Updated commitments are included as follows (refusals or previously unassigned commitments are considered as commitments with 0 capacity):

$$\forall a_i \in Ac, \forall r_j \in R : cap(r_j^{a_i})^{prop} \geq cap(r_j^{a_i})^{final} \quad (6)$$

It is at this stage of planning process when we also detect the failure to execute the plan altogether – the proposals (or actually refusals) from the members may be mutually incompatible. If the coalition leader manages to find

Figure 3. Start of the operation as shown in 3D simulation – real batch positions and operations are shown for the selected plan.

an acceptable planning outcome, it prepares the final commitments (with the quantities assigned that are less or equal to the binding ones proposed by members) and re-submits them to the coalition members.

2.3.4 Plan Execution

As the proposals by the agents were binding, coalition members shall be all able to start performing the assigned tasks immediately. Alternatively, when the final commitments are lower than the ones they have proposed, they may change their resource allocation or trade the assignments with their peers in the team in the same way as in the Local Plan Evaluation phase, provided that they manage to honor their commitments.

3 Algorithm properties

In this section, we will analyze the above-described algorithm and discuss several interesting properties it presents: computational efficiency, preservation of private information and stability of the solution with respect to environmental perturbations.

Reduced Communication. The present algorithm represent a special approach to distributed planning that in parts uses the classical AI planning algorithms in combination with multi-agent, negotiation based approach to plan generation. Unlike state-of-the-art approaches such as *Partial Global Planning* [4] the *initial planning phase* of the presented approach substantially constrains the space of possible negotiation and thus makes the planning process substantially more scalable. On the hand it still allows the agents to influence the plan that is to be imposed on their operation by task delegation enabled by ECNP protocol during the *local plan evaluation phase*.

Obviously, the desired situation is that the amounts of decision making and computation is somewhat balanced between these two phases of the planning algorithm. Finding this equilibrium affects the efficiency, stability and the amount of private knowledge disclosure during the planning process (as explained in Section 1). This is why the right amount social knowledge stored by the perspective coalition leader (depends on the amount of knowledge the agents are happy to share) affects the right fitting the operation of the algorithm. With an increasing amount of knowledge shared within the community (therefore higher knowledge disclosure), the initial planning phase is getting more precise resulting in substantially less communication traffic (and thus further knowledge disclosure) in the local plan evaluation phase. It has been studied recently how the amount and structure of a priori shared social knowledge affects the coalition formation process [9].

Operation-research integration with negotiation and cognitive methods is natural and seamless: output of most trust models in existence today can be transformed into the fuzzy number-form and this representation fits well into the context of modern FLP methods as shown below. On the other hand, the individual team members don't need any notion of FLP techniques - they only reason about the coalition, their teams and negotiate within their teams to achieve optimal resource allocation.

Contraction of the solution space is another key feature – each step of the planning, centralized or distributed, reduces the solution space. Initial planning performs the greatest reduction, as the actions/tasks are selected, resources pre-allocated and agent teams created. Local planning phase then further clarifies resource allocation and team composition and the results of this phase are incorporated as additional restrictions for the FLP planning problem solved in the coherence and validation phase – we effectively ensure that any overall solution will respect the commitments received from coalition members and can be executed. Optional re-allocation step doesn't break our assumptions, it only permits team members to trade resources in a situation where the batch sizes may have been reduced. If the plan can not be implemented due to the member refusal or resource incompatibility, the situation is detected in the coherence planning step. LP method used identifies the interfering restriction and can direct the coalition leader towards plan reconfiguration. Therefore, in the global algorithm as suggested, we don't allow any backtracking (except the team-scale negotiation). On the other hand, the algorithm as presented doesn't guarantee that the result it returns will be the optimal plan. We don't consider this as a serious drawback, because none of the comparably efficient algorithms currently in use can guarantee such result.

Stability of Flexible & Fuzzy Linear Programming One of the important properties of the trustfulness $\Theta_{A_0}(A_i)$ (and distrustfulness $\Delta_{A_0}(A_i)$) values is their uncertainty, emphasized by the fact that they are modelled as fuzzy numbers. There are two approaches how to use these values in the LP algorithms: either to solve a *flexible linear programming* problem, or to *defuzzyfy* the values and solve a classical LP problem.

Flexible linear programming techniques [3] that work with fuzzy coefficients provide us with a unique feature - a stability of the solution with respect to small changes of coefficient (e.g. trustfulness) values defined as symmetrical triangular fuzzy numbers. Problem formulation remains the same, but we must solve a non-linear optimization problem in order to obtain the solution – a major disadvantage of the approach. On the the other hand, once we have an appropriate solver, we may effectively adjust the stability of the solution by varying the width of the trustfulness values – by restricting their width, we approach the unstable classical linear programming problem, while the widening of trustfulness representation ensures the stability with respect to bigger perturbations. This ability is a very desirable feature when the agents encounter an intelligent adversary in an unknown environment – agents can adjust their planning to be robust when they still gather the information about the environment and reduce the predictability of their behavior in later phases. The shape representing the trustfulness $\Theta_{A_0}(A_i)$ supports this adaptation, as it "narrows" with the increasing number of data.

In the alternative approach, $\Theta_{A_0}(A_i)$ and $\Delta_{A_0}(A_i)$ must be defuzzyfied before they are inserted into the planning constraints of the normal LP problem. During the transformation, we replace the fuzzy numbers in the constraints and utility function by the center of the core of the trustfulness values and solve the resulting problem. This approach is more sensitive to the noise, especially with limited number of data, but as the agent gathers more experience, it converges to the center of gravity method due to the shape of the trustfulness function $\Theta_{A_0}(A_i)$ as defined in [15].

4 Prototype Deployment

Presented protocol is currently being integrated with ACROSS scenario (see Fig. 3) to solve humanitarian aid logistics problem on a simplified simulated version of Java island in Indonesia. We have taken existing system based on generic multi-agent approach to coalition planning and introduced a new agent with planning capability. This agent simulates a humanitarian organization that organizes a delivery of goods from the initial location (Jakarta) to several

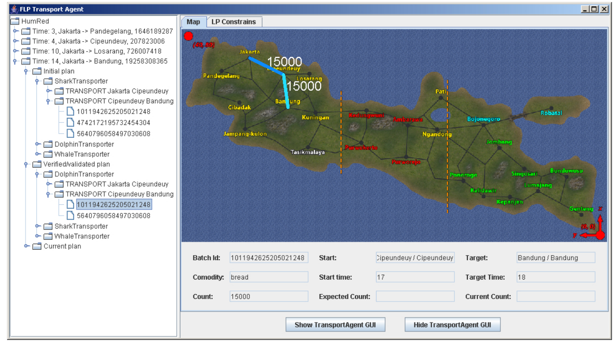

Figure 4. Internal planning process of the coalition leader – initial plan and coherent plan based on input from coalition members.

cities throughout the island.

This agent (denoted HumRed) assumes the role of a coalition leader (A_0). Other agents, who actually provide their vehicles (resources) for the transportation are coalition members and don't use any FLP solvers, they merely assign their resources to tasks suggested by coalition leader. Therefore, coalition leader implements *initial planning* and *coherence and validation phase* (Fig 4).

While the overall algorithm is already integrated with ACROSS environment, we are currently proceeding with implementation of its more advanced features (most notably ECNP negotiations within teams and more advanced FLP solvers) to improve the quality of the resulting plan. Advancing the implementation state of these features would also allow us to formally verify the protocol and judge the impact of its various features and phases on plan quality.

5 Conclusions and Future Work

In our contribution, we have presented a combined planning algorithm that can be used to efficiently create a shared plan in an adversarial environment, featuring only a limited and controlled information disclosure by self-interested agents. Adversarial behavior of agents and environmental reasons of failure (actions with low action trustfulness) can be detected and provide an input for the embedded trust model, that in its turn provides an input for further planning.

One of the important open issues of this research is the concept of *plan diagnosis*. Plan diagnosis [18] is important to achieve long-term efficiency by elimination of untrustful cooperators and bad actions (paths). We can not assume that the state of the problem is observable – implicitly, we assume that only the initial and terminal objective status are known by the coalition leader and that the state of some of the intermediary objectives **may** be known. Integration of the latest results from the monitoring selectivity problem [7] into a trust model update is a subject for future research.

Another key challenge for the future research in this area is to investigate the plan execution phase and allow *intelligent replanning*. Re-planning can occur as a result of a coalition leader detects a failure in completion of one or more commitments or upon a request from the coalition members. If these commitments fully or partially precondition other tasks (and their commitments), it may be advantageous to re-plan the plan in order to eliminate inefficient future commitments. Such operation is analogous to coherence phase (see Section 2.3.3), but for each commitment with known outcome, we can fix the commitment size to the real delivered value, or limit it by using the information about partial deliveries/losses. In the same manner, the agent can react when some task was more successful than expected and more resources are necessary for subsequent transport.

Acknowledgment

Effort sponsored by the Air Force Office of Scientific Research, Air Force Material Command, USAF, under grant number FA8655-04-1-3044. The U.S. Government is authorized to reproduce and distribute reprints for Government purpose notwithstanding any copyright notation thereon[5].

References

[1] D. G. C. 2005. Darpa grand challenge 2005. `http://www.darpa.mil/GRANDCHALLENGE/`, 2005.

[2] M. Baioletti, S. Marcugini, and A. Milani. C-satplan: a satplan-based tool for planning with constraints. In *Proceedings of AIPS-98 Workshop on Planning as Combinatorial Search*, 1998.

[3] C. Carlsson and R. Fullér. *Fuzzy Reasoning in Decision Making and Optimization*. Physica Verlag, Springer, Heidelberg, 2002.

[4] E. H. Durfee and T. A. Montgomery. Coordination as distributed search in a hierarchi- cal behavior space. *IEEE Transactions on Systems, Man, and Cybernetics – Special Issue on Distributed Artifcial Intel- ligence*, 21(6):1363 – 1378, November 1991.

[5] K. Fischer, J. P. Muller, M. Pischel, and D. Schier. A model for cooperative transportation scheduling. In *Proceedings of the First International Conference on Multiagent Systems.*, pages 109–116, Menlo park, California, June 1995. AAAI Press / MIT Press.

[6] J. Himoff, P. Skobelev, and M. Wooldridge. Magenta technology: multi-agent systems for industrial logistics. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 60–66, New York, NY, USA, 2005. ACM Press.

[7] G. A. Kaminka and M. Tambe. Robust multi-agent teams via socially-attentive monitoring. *Journal of Artificial Intelligence Research*, 12:105–147, 2000.

[8] I. Miguel, P. Jarvis, and Q. Shen. Flexible graphplan. In W. Horn, editor, *Proceedings of the Fourteenth European Conference on Artificial Intelligence*, pages 506–510, 2000.

[9] M. Pechoucek, V. Marik, and J. Barta. Role of acquaintance models in agents private and semi-private. *Knowledge Based Systems*, accepted to publication, to appear Spring 2006.

[10] M. Pěchouček, V. Mařík, and J. Bárta. A knowledge-based approach to coalition formation. *IEEE Intelligent Systems*, 17(3):17–25, 2002.

[11] M. Pěchouček, V. Mařík, and O. Štěpánková. Role of acquaintance models in agent-based production planning systems. In M. Klusch and L. Kerschberg, editors, *Cooperative Infromation Agents IV - LNAI No. 1860*, pages 179–190, Heidelberg, July 2000. Springer-Verlag, Heidelberg.

[12] D. Perugini, D. Lambert, L. Sterling, and A. Pearce. A distributed agent approach to global transportation scheduling. In *The 2003 IEEE/WIC International Conference on Intelligent Agent Technology (IAT 2003)*, pages 18–24, Halifax, Canada, 2003.

[13] D. Perugini, D. Lambert, L. Sterling, and A. Pearce. Agent-based global transportation scheduling in military logistics. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1278–1279, Washington, DC, USA, 2004. IEEE Computer Society.

[14] S. Ramchurn, D. Huynh, and N. R. Jennings. Trust in multi-agent systems. *The Knowledge Engineering Review*, 19(1), 2004.

[15] M. Rehák, Lukáš Foltýn, M. Pěchouček, and P. Benda. Trust model for open ubiquitous agent systems. In *Intelligent Agent Technology, 2005 IEEE/WIC/ACM International Conference*, number PR2416 in IEEE, 2005.

[16] M. Rehák, M. Pěchouček, and J. Tožička. Adversarial behavior in multi-agent systems. In M. Pechoucek, P. Petta, and L. Z. Varga, editors, *Multi-Agent Systems and Applications IV: 4th International Central and Eastern European Conference on Multi-Agent Systems, CEEMAS 2005*, number 3690 in LNCS, LNAI, 2005.

[17] D. Šišlák, M. Pěchouček, M. Rehák, J. Tožička, and P. Benda. Solving inaccessibility in multi-agent systems by mobile middle-agents. *Multiagent and Grid Systems*, 1(2):73–87, 2005.

[18] C. Witteveen, N. Roos, R. van der Krogt, and M. de Weerdt. Diagnosis of single and multi-agent plans. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 805–812, New York, NY, USA, 2005. ACM Press.

[5] The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the U.S. Government.