Interleaving Collaborative Planning and Execution Along with Deliberation in Logistics and Supply Chain

Naeem Janjua, Omar Hussain, Elizabeth Chang

Interleaving collaborative planning and execution along with deliberation in logistics and supply chain

Naeem Khalid Janjua, Omar Khadeer Hussain, Elizabeth Chang

School of Business, Australian Defence Force Academy, University of New South Wales, Canberra.
{n.janjua,o.hussain,e.chang}@adfa.edu.au

Automated planning is a rich technical filed in Artificial Intelligence (AI) and most of the existing research focused on path finding methods in a compact state-transition system where planning is decoupled from execution. The introduction of the Web has led to increasing emphasis in AI on the development of planning algorithms for real-world applications where planning is distributed and plan generation can happen concurrently with plan execution. An example of one such real-world application is logistics and supply chain. In this paper, we envisage a collaborative planning and execution framework for logistics and supply chain operations. The framework supports human planners for a collaborative plan construction. The planning is interleaved with execution where new information collected during execution is used to refine the plan if required. Additionally, planning is defeasible in nature. During planning either conflicting viewpoints may arise among planners and/or the new information collected during execution may result in conflicts among planned tasks (situations). Deliberation module in the proposed framework provides a platform to human planners where they can start an argumentative dialogue to resolve the conflicts by establishing preferences between conflicting tasks. We use situation calculus to model the framework and propose an algorithm to interleave collaborative planning with execution along with deliberation support.

1 Introduction

Planning and decision making processes have been a hot research area in AI for decades with applications in various domains. For example, it is a core activity to the design, organisation and control of logistics and supply chain activities such as procurement, transportation, inventory management, warehousing and materials handling, quality of services, information management and sharing, risk management etc. Collaborative planning is a process of decision making among interdependent parities that involves joint ownership of decisions and collective responsibility of outcomes, and work as coalitions to achieve common goals that otherwise would have been impossible or too expensive to achieve by an individual party. It is a method for solving shared problems by resolving conflicts.

Collaborative planning consists of two important phases, namely; planning and execution. Traditionally, these were separate apart. Most of the planning systems research focused on automated path finding methods in a compact, state-transition system and they ignored the human aspect in decision making process. These systems as categorised as an offline planning system. For example, in logistics and supply chain it is well-known that the strength of transactional enterprise resource planning (ERP) systems is not in the area of planning instead its execution. Hence, Advanced Planning Systems (APS) have been developed to fill this gap. APS are based on the principles of hierarchical planning and make extensive use of solution approaches known as mathematical programming and meta-heuristics.

Multi-agent system planning systems are APS systems applied to independent or loosely-coupled problems to enhance the benefit of distributed planning between autonomous agents as solving this type of problem requires less coordination [19]. Most of automated software agents who engage in collaboration with others, build a model of other agents' mental states and update their own beliefs and goals as the dialogue progresses. This process is known as dialogue understanding [6]. To handle uncertainty, argumentation-driven frameworks have been proposed that allows different users to share their knowledge and resolve conflicts between them to reach the common goal have been proposed in the literature [19]. However, in most MAP systems, planning provides the solution, and on execution part, it is executed as merely traversing the identified path. Such system work well in close world assumption where all of the possible effects of each action are know in advance. However, planning becomes very challenging if the environment is dynamically changing (open world assumptions) and is not pre-engineered to conform to software agent needs. Additionally, automated agents might not able to interpret human planner's intensions and could lack ability of planning and deliberation under uncertainty.

The introduction of WWW has led to increasing emphasis in AI on development of planning algorithms for real-world applications where planning is distributed and plan generation can happen concurrently with plan execution. When and how to interleave planning and execution is well defined complex problem in the literature [18]. Additionally, the challenges for software tools supporting collaborative planning include master data integration, user-specific secure data access and the mutual decision-making process. Systems that enable collaborative planning must support partners during each step of the process [11]. The introduction of Semantic Web technology tools for collaboration has addressed some of the issues of collaborative programming such as information has meaning attached to it that makes it understandable across organisational boundaries. The Collaborative planning and acting model [16] is the first attempt that supports human planners in managing, planning information and facilitates the planning process with automated reasoning. However, currents Semantic Web based Collaborative planning models lack the means of representation incomplete, contradictory information and logical relations that define constraints and axioms of domain begin modeled [14].

In our previous work [8], we proposed a framework for incomplete and conflicting information representation and developed argumentation-based algorithms for reasoning over such information in Semantic Web applications. In this paper, we extend our previous work and apply it in the area of logistics and supply chain where planning for operational risk may need to be addressed at a spoke by considering the information from other multiple global and region specific locations. We propose a conceptual framework and develop algorithms that interleave planning and execution, thus allow in a timely manner enables the human planners to plan and execute the tasks . In case of conflicting viewpoints, planners start an argumentative dialogue to resolve the conflicts, integrate the changes in the plan and proceed with the execution. We use situation calculus to model the the framework and Defeasible Logic Programming (DeLP) for knowledge representation and reasoning in logistics operations.

2 Related work

This work lies at the intersection of Artificial intelligence, in particular classical and distributed planning, Multi agent systems, Argumentation based dialogue systems, Reasoning about actions and change, and the Semantic Web. Automated planning system is a kind of APS system that has been an endeavour of AI leading to numerous technologies, however, their application in real world applications has been unfortunately relatively low [7, 16].

Multi-agent planning system (MAP) involves several autonomous software agents in planning or plan execution activity, share their knowledge and capabilities to provide solutions to loosely-coupled problems with minimum coordination [4]. Collaborative planning is a distributed planning which as been an active area of research in AI for decades and it is still an open challenge. The various existing work on MAP approaches can be classified according to the planning and coordination models they use.

Firstly, **pre-planning distribution of tasks** where a software agent plans and distributes the tasks to other agents. For example, Multi-agent Planning by plan Reuse (MARP) agent allocates the task goals to the participating agents that consider both private and public information. MARP agent then calls each agent iteratively for a solution [2]. In another work [3] agent automatically decomposes tasks into MAP problems, which are then locally solved through a centralized heuristic planner. Emphasis is pre-planning for task distribution and execution by individual agents.

Secondly, **planning followed by merging plan through coordination** where software agents construct independent plans for different subgoals and a centralized algorithm is used to merge those plans. The emphasis is on problem of controlling and coordinating a local plan of agents. For example [5] of the most well known approach for coordination of plans called as a partial global planning framework. Planning First is one of the first planners where agents individually synthesize plans through a state-based planner and the resulting local plans are then coordinated through a distributed Constraint Satisfaction Problem [15].

Thirdly, **interleaving planning and coordination** where software agents propose an iterative refinement of the base plan until a consistent joint plan is obtained that solves the problem. Several advantages of this approach have been discussed in the literature [19].

Although APS systems discussed above provide promising solutions, however, they lack implementations in real world scenarios such as logistics and supply chain. The various reasons reported in literature that leads to low penetration of above mentioned APS systems in real world application areas are as follows [7, 16]:

1. Human planners want to use tool for better visibility of the planning process but want to control the decision making part of planning phase.
2. High level of automation results in reduced situation awareness, complex and skill degradation.
3. The huge amount time and manpower needed to enter all information.
4. Difficulty of converting human concepts into tool supported language.

Therefore, this research is an attempt to overcome the limitations of the APS system identified above. We propose a conceptual framework to support planning interleaving with execution along with deliberation to help the human planners in determining which actions are possible at current stage, helping them in making the best choice by building arguments in favour and against conflicting planning tasks, refine the plan and execute it.

3 Basic action theory

Our formalization model is based on situation calculus [13]. We describe and employ the extended version of Reiter [17] to formalization our model for argumentation-based collaborative planning and acting model.

In our model, each human planner maintains the representation of the domain as basic action theory and it has the following form:

$$\mathcal{D} = \Sigma \cup \mathcal{D}_{ss} \cup \mathcal{D}_{ap} \cup \mathcal{D}_{una} \cup \mathcal{D}_{S_0} \tag{1}$$

Where

- $\sum$ is a set of fundamental domain-independent axioms providing the basic properties of the situation.
- $\mathcal{D}_{ss}$ is a set of successor state axioms represent relational or functional fluents in the domain. Formally, $Poss(a,s) \supset [T(do(a,s)) \equiv \gamma_T^+(a,s) \vee (T(s) \wedge \neg\gamma_T^-(a,s))]$ where γ_T^+ and γ_T^- represent the add and delete conditions of fluent T.
- $\mathcal{D}_{ap}$ is a set of precondition axioms under which action can be performed. Formally, represented as $\Pi_A(s) \equiv Poss(A,S)$.
- $\mathcal{D}_{una}$ is a set of unique names axioms for actions.
- $\mathcal{D}_{S_0}$ is a set of first order sentence that represents initial state of the world.

A basic action theory for logistics application specifies a plan and the tasks of the domain of concern and the contextual settings in which the dialogue operates. A plan in situation calculus is treated as an executable situation that satisfies a goal statement. We assume that the sets Fluents, NonFluents and Actions are shared among the planners. Additionally, they share a common goal, knowledge about the fundamental axioms, unique name axioms for actions and the names of object in the domain. We use the definition of plan and planning problem defined in [1] as follows:

Definition 1. *Given a basic action theory $\mathcal{D}$ and a Goal g with single free variable s, a plan π is a variable-free situation term s_π iff $\mathcal{D}| = executable(s_\pi) \wedge g(s_\pi)$ where $executable(s_\pi) \stackrel{\text{def}}{=} (\forall a, s^*).do(a, s^*) \sqsubseteq s_\Pi \supset Poss(a, s^*)$.*

It is important to note here is that the term s_π represents the history for the execution of the actions of a plan in sequence.

Definition 2. *A planning problem $\mathcal{P}$ is a tuple $< \mathcal{D}, g >$ where $\mathcal{D}$ is a basic action theory denoting the planning domain and g is a fluent sentence specifying the goal.*

As a result of above definition of plan and foundational axioms for situations we can identify that $excutable(do(a, s)) \equiv executable(s) \wedge Poss(a, s)$. This enables the transformation of plan definition as follows:

Definition 3. *A plan $\pi = A_1, A_2; .....; A_n$ is a solution to a planning problem p iif $\mathcal{D} \models \ Poss(A_1, S_0) \wedge do(A_1, S_0) = S_1 \wedge Poss(A2, S_1) \wedge do(A_1, S_1) = S_2 \wedge .... \wedge Poss(A_n, S_{n-1}) \wedge do(A_n, S_{n-1}) = S_n \wedge G(S_n)$.*

This definition asserts that the actions in the plan can be performed in sequence eventually performing the final action results in goal sentence $\mathcal{G}$ be true.

A basic action theory is necessary to define a domain for reasoning. However, in classical AI reasoning is performed under certain assumption such as follows:

1. The given problem can be fully addressed with available information (solution to the problem lies within the available situation tree). In order to elucidate it, let us consider an example. A planner wants to improve a decision making process and he believes that all the information he holds is sufficient to identify the issues and address them.
2. The domain knowledge is consistent. In other words, they assume that there will be no conflicting events and situation during the decision-making process.
3. New information is consistent with the already available information or specifications.
4. New information does not lead to retraction of previous conclusions.

Because of these limitations discussed above, AI failed to provide a solution to many real world scenarios where some of the information or actions in a plan may result in conflicting situations. To overcome this, we employ a model based on an extended version of action theory, $\mathcal{D}_{ext}$, that consider representation of conflicts and provide support for conflict resolution dialogue methodology.

Definition 4. *Extended action theory is defined as follows:*

$\mathcal{D}_{ext}= \mathcal{D} \cup \Omega_c \cup \Omega_p$

where

- $\mathcal{D}$ is a basic action theory.
- Ω_c is a set of axioms for representing conflicting information. For example argument(X,p), argument(Y,$\neg p$), counterArgument(Y, X, do(a,s)), underCut(Z,C, do(a,l)) etc.
- Ω_p is an set of axioms used in deliberation module such as speech acts for communication and dialogue movies for establishing preference between conflicting situations. For example, propose(A), reject(A), Argue(A=>P), Why(P), Support(P) etc.

Definition 5. *A situation p conflicts with situation $\neg p$ in a plan iff $\neg p$ executes action a after p has executed an internal action $\neg a$ that conflict with a in the system.*

$Conflict(a, a) = counterArgument(p, \neg p, do(a, s)) =_{def} p \neq \neg p \wedge Poss(\neg p, a, s) \wedge (\exists \neg a, s^{'})[Poss(p, a^{'}, s) \wedge do(a^{'}, s^{'}) \subset s \wedge Poss(a^{'}, a, >, s)]$

A set of possible conflicts set $Con\hat{f}lict$ contains situations that can be used to generate counter-arguments. Note that no situation weights are used in both plan construction and conflict set. Therefore, the attack between arguments are symmetric i.e. they are equally acceptable. Therefore, planners need to perform meta-argumentation to establish a preference between conflicting arguments. Therefore, $Preference(a^{'}, a) =_{def} Assign(Conflict(a^{'}, a)))$, where assign is a primitive action that triggers Deliberation Dialogue.

The deliberation dialogue is a meta-argumentation system consists of union of arguments that are constructed by the proponent and the opponents i.e. $\mathcal{A} = \mathcal{A}_{pro} \cup \mathcal{A}_{opps}$. As a result of the dialogue process, argumentation lines are constructed and acceptability of arguments is computed to establish priority between conflicting situations. Once the priority is established, the preference is included into the plan along with reasoning/argumentation line supporting it. We reuse the syntax and semantics for argumentation system defined in [10] and extend it for dialogue based system using semantics defined in [12]. We explain the working of argumentation based dialogue system in section 4.

Definition 6. *Given extended action theory $\mathcal{D}_{ext}$, a collaborative plan Π for a common Goal G is a variable-free situation term s_Π iff $\mathcal{D}_{ext} \models executable(s_\Pi) \wedge G(s_\pi)$ where $executable(s_\Pi) \overset{\text{def}}{=} (\forall a, s^*).do(a, s^*) \sqsubseteq s_\Pi \supset Poss(a, s^*)$ and if conflict exists, there exists a preference relationship, such that $Preference(a^{'}, a) \equiv Support(a^{'}, a) \models do(a^{'}, s)$*

Definition 7. *A collaborative planning solution $\mathcal{CP}_s$ is a tuple $< \mathcal{D}_{ext}, \mathcal{G}, \hat{p} >$ where $\mathcal{D}_{ext}$ is extended action theory for representing the sequence of actions and $\mathcal{G}$ is a fluent sentence specifying the goal and $\hat{p}$ represents the priority relationships over conflicting situations.*

4 Conceptual framework

The conceptual framework shown in figure 1 provides the infrastructure for collaborative planning to synthesis plan whose success is suggested by the evidence provided by the planners and evaluate the acceptability of the plan by comparing the evidence supporting them against possible objections (conflicts). The advantage of deliberation (argumentative dialogue) within the domain of decision making is its ability to manage conflicts in the knowledge, preferences and the rules by which a decision is made. Therefore, the notion of acceptability is embedded within the planning problem and the plans are supported with arguments. In following sub-sections we explain working of the framework in detail.

4.1 Planning module

A planner needs to reason about its actions. Therefore, they need models for choosing, organising, and revisiting their actions and plans [7]. During collaborative planning it might be possible that planner may need to retract from planning task defined earlier. Therefore, the knowledge representation model should be expressive enough to represent planning task that may be retracted later. Therefore the planning is defeasible in nature.

DeLP is a general-purpose defeasible argumentation formalism based on logic programming, intended to model inconsistent and potentially contradictory knowledge (both strong and weak negation). A defeasible logic program has the form $\psi = (\Pi, \Delta)$, where Π and Δ stand for strict knowledge and defeasible knowledge, respectively. We extended DeLP for knowledge representation and reasoning in semantic web application [8]. We defined syntax and semantics for strict and defeasible rule representation. We reuse our work here for defining the planning tasks. In the rule base, a planning task (rule) takes the following form *[rule identifier] [rule body] [type of rule] [head]*. The rule body represents precondition and rule head represent the effects. Planners have objective and to achieve them, they define tasks using a web-based interface. A human planner who need to collaborate with other human planners to achieve some common goals, define their planning tasks in the

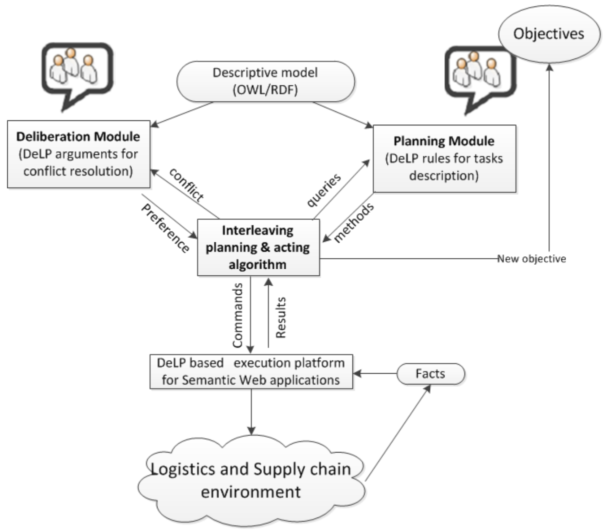

Fig. 1. Interleaving collaborative planning and execution framework along with deliberation in logistics and supply chain

form of strict and defeasible rules. Each planner defines its tasks and as a result the system produces a process map using forward chain reasoning as described in [9]. In such collaborative problem solving model, forward chain reasoning is used to digitize the plans and make them alive for the human planners. For more information about forward chain reasoning using rete algorithm, readers are referred to [10]. It is important to note is that the planning task are initiated with domain knowledge defined in descriptive model. We used ontologies to define domain knowledge and used rules defined in Table 1 to transform ontology implicit knowledge to become explicit for rule based reasoning.

Rule 1	type(X,C)→C(X)	Class
Rule 2	subClassof(Sc, C), Sc(X)→C(X)	Subclass
Rule 3	objectProperty(X), domain(X, Y), range(X,Z) →X (Y, Z)	Object Property
Rule 4	objectProperty(X), X(Z, V), subProperty(X, Y)→Y(Z, X)	subProperty
Rule 5	dataProperty(X), domain(X, Y), range(X, Z) →X(Y, Z)	Data Property
Rule 6	dataproperty(X), X(Z, V), subProperty(X, Y)→Y(Z, X)	SubProperty

Table 1. Translation rules for OWL/RDF predicates to DeLP facts

4.2 Deliberation module

Time pressure and the distributed nature of logistics and supply chain, force the planners to execute an action. In non-deterministic planning domain, execution of plan tasks is the best source of collecting real observations as an effect of the plan. In most of real world applications, planning is defeasible in nature and execution of plan tasks may result in conflicting tasks or retraction of previous planning results. We build deliberation module on the work done by [12] and it helps planners firstly; to put forward their arguments that may be incomplete statements and offer them ways of advancing well-formed arguments as well as to reuse arguments that often appear in discussions, secondly; with the help of algorithms to compute the acceptability of arguments at any stage of the discussion. The deliberation module is defined by:

- Topic Language: DeLP as a logical language.
- Argumentation Logic: as defined in [10]. The only difference is that in our previous work it was assumed that system has collated all the relevant information and hybrid reasoning engine reasoning over it. Here, we replace automated algorithm with human planners and conflict resolution process is a dialogue driven activity. We reuse the definition of argument, sub-argument, attack, static defeat , dynamic defeat.
- Communication Language: Set of Locutions S and two binary relation Ra and Rs of attacking and surrendering reply on S.
- Dialogue Moves and Termination: as defined in [12].

It might be possible that task defined as defeasible tasks may get in conflict with other task. If both tasks are defined by a single planner, he can define a preference between conflicting tasks during planning phase, Otherwise, during execution, deliberation module is used to establish preference between conflicting tasks.

4.3 Inter-leaving planning, execution and deliberation

Various approaches in the literature [32, 33], including our previous work [12], have developed argumentation-based algorithms for closed-loop systems, but no work has been done to identify and model the interaction steps of argumentation when various stakeholders are involved in the risk management of an open-loop system such as logistics and supply chain that needs interleaving planning and execution. The proposed framework address this drawback by by interleaving planning and execution.

The collaborative planning algorithm interleaving execution, integrated with argumentation-driven dialogue system works as follows:

1. Terminate the collaborative planning activity if the objective statement is satisfied in current planning phase. Return a list of tasks (situations) indicating which have been executed.
2. Check if there is an executable plan task i.e., a step which has been planned but not yet executed. if there is none, go to step 4
3. For every executable plan task, executed the task
   – Incorporate new found information from execution into planning stage.
   – if conflict arise, initiate call to deliberation module for establishment of preference over conflicting situations.
   – Go to step 2
4. Plan:
   – Either
   –– identify a objective that needs to be achieved.
   –– Add a new tasks to the existing plan step to achieve the objective.
   – Or
   –– Apply (simulate execution of) tasks previously selected to be in the plan.
   – Go to step 1

5 Conclusion

In this paper, we propose a collaborative planning and execution framework. MAP based approaches discussed in literature takes planning and execution as separate steps and as a result, most of their implementations lack real-world applications. Our work is first of its kind that provides a collaborating environment to human planners where planning interleave with the execution along with deliberation support in an open loop system like logistics and supply chain. We plan to develop a prototype system and extend the system's functionality with conflict blocking and conflict propagation based reasoning models.

References

1. Belesiotis, A., Rovatsos, M., Rahwan, I.: A generative dialogue system for arguing about plans in situation calculus. In: McBurney, P., Rahwan, I., Parsons, S., Maudet, N. (eds.) Argumentation in Multi-Agent Systems, Lecture Notes in Computer Science, vol. 6057, pp. 23–41. Springer Berlin Heidelberg (2010)

2. Borrajo, D.: Multi-agent planning by plan reuse. In: Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems. pp. 1141–1142. International Foundation for Autonomous Agents and Multiagent Systems (2013)
3. Crosby, M., Rovatsos, M., Petrick, R.P.: Automated agent decomposition for classical planning. In: ICAPS (2013)
4. Durfee, E.H.: Distributed problem solving and planning. In: Multi-agent systems and applications, pp. 118–149. Springer (2001)
5. Durfee, E.H., Lesser, V.R.: Partial global planning: A coordination framework for distributed hypothesis formation. Systems, Man and Cybernetics, IEEE Transactions on 21(5), 1167–1183 (1991)
6. Gabaldon, A., Langley, P.: Dialogue understanding in a logic of action and belief (2015)
7. Ghallab, M., Nau, D., Traverso, P.: The actors view of automated planning and acting: A position paper. Artificial Intelligence 208(0), 1 – 17 (2014), http://www.sciencedirect.com/science/article/pii/S0004370213001173
8. Janjua, N.K.: A Defeasible Logic Programming-Based Framework to Support Argumentation in Semantic Web Applications. Springer Theses, Springer International Publishing (2014)
9. Janjua, N.K.: Process map discovery from business policies: A knowledge representation approach with argumentative reasoning (kr@ pmd). In: A Defeasible Logic Programming-Based Framework to Support Argumentation in Semantic Web Applications, pp. 201–233. Springer (2014)
10. Janjua, N.K., Hussain, F.K.: Web@ idss–argumentation-enabled web-based idss for reasoning over incomplete and conflicting information. Knowledge-Based Systems 32, 9–27 (2012)
11. Kilger, C., Reuter, B., Stadtler, H.: Collaborative planning. In: Supply Chain Management and Advanced Planning, pp. 257–277. Springer (2015)
12. Kok, E.M., Meyer, J.J.C., Prakken, H., Vreeswijk, G.A.: A formal argumentation framework for deliberation dialogues. In: Argumentation in Multi-Agent Systems, pp. 31–48. Springer (2011)
13. McCarthy, J.: Situations, actions, and causal laws. Tech. rep., DTIC Document (1963)
14. Mott, D., Harwood, W.: Representing logic and rationale in semantic web technologies. In: The Fourth Annual Conference of the International Technology Alliance. London, England. pp. 13–17 (2010)
15. Nissim, R., Brafman, R.I.: Multi-agent a* for parallel and distributed systems. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 3. pp. 1265–1266. Richland, SC (2012)
16. Patel, J., Dorneich, M., Mott, D., Bahrami, A., Giammanco, C.: Improving coalition planning by making plans alive. Intelligent Systems, IEEE 28(1), 17–25 (Jan 2013)
17. Reiter, R.: On knowledge-based programming with sensing in the situation calculus. ACM Trans. Comput. Logic 2(4), 433–457 (Oct 2001), http://doi.acm.org/10.1145/383779.383780
18. Stone, P., Veloso, M.: User-guided interleaving of planning and execution. In: Proceedings of the European Workshop on Planning. vol. 12 (1995)
19. Torreño, A., Onaindia, E., Sapena, Ó.: An approach to multi-agent planning with incomplete information. In: ECAI. pp. 762–767 (2012)