Handling Uncertainties in Distributed Constraint Optimization Problems using Bayesian Inferential Reasoning

Sagir Muhammad Yusuf and Chris Baber*
University of Birmingham, B15 2TT, U.K.

Keywords: Multi-agent Learning, DCOP, Bayesian Learning, Bayesian Inference.

Abstract: In this paper, we propose the use of Bayesian inference and learning to solve DCOP in dynamic and uncertain environments. We categorize the agents Bayesian learning process into local learning or centralized learning. That is, the agents learn individually or collectively to make optimal predictions and share learning data. The agents' mission data is subjected to gradient descent or expectation-maximization algorithms for training purposes. The outcome of the training process is the learned network used by the agents for making predictions, estimations, and conclusions to reduce communication load. Surprisingly, results indicate that the algorithms are capable of producing accurate predictions using uncertain data. Simulation experiment result of a multi-agent mission for wildfire monitoring suggest robust performance by the learning algorithms using uncertain data. We argue that Bayesian learning could reduce the communication load and improve DCOP algorithms scalability.

1 INTRODUCTION

Distributed Constraint Optimization (DCOP) involves the appropriate assignment of variables to agents in order to optimize costs (Fioretto et al., 2018; Fioretto et al., 2015; Fransman et al., 2019; Maheswaran et al., 2004; Yeoh et al., 2011). DCOP exists in different forms based on the agents' environmental evolution and behaviours (Fioretto et al., 2018). Classical DCOPs involves the appropriate assignment of variables by agents under constraints. Multi-objective DCOP is a form of classical DCOP with conflicting cost functions. Probabilistic DCOP follows probabilistic distribution of the agents' environmental behaviours (Fioretto et al., 2018; Stranders et al., 2011). Dynamic DCOP changes overtime, in which the DCOP problem at time t is different from the DCOP problem at time t+1 (Fioretto et al., 2018; Hoang et al., 2017; Yeoh et al., 2011). A current challenge is solving DCOP in a dynamic and uncertain environment (Fioretto et al., 2018; Fioretto et al., 2015; Fioretto et al., 2017; Pujol-Gonzalez, 2011; Fransman et al., 2019; Yeoh et al., 2011), i.e.,a highly changing environment with lots of uncertainties about future events, agents variables, cost functions, and environmental exogenous variables.

*https://www.birmingham.ac.uk/staff/profiles/computer-science/baber-chris.aspx

The DCOP algorithms computation time is important for a highly changing environment; otherwise, the solution will be outdated. This situation can occur as a result of the complexity of the algorithms (communication and computation cost, etc.). To reduce this effect, we propose the use of Situation Awareness (Endsley, 1995; Stanton et al., 2006). That is, allowing the agents to reason about aspect of the current and future situation; therefore, allowing agents to only consider few variables. Another challenging issue to DCOP algorithms is the tolerance of uncertainties and dynamism in environmental variables and cost functions. That is when the agent is not sure of the local cost function or variable to be optimized, having doubt on the given information, missing variables, or the instability of the operating environment (Le et al., 2016; Léauté et al., 2011; Stranders et al., 2011). In this paper, we tackle the problem of uncertainties in DCOP using Bayesian inferential reasoning. The agents made predictions and estimations using the outcome of the learning process. Therefore, agents learn from previous cases and cases from other agents. The potential advantage of this approach is the ability to reduce communication in solving DCOP, reducing the whole complexity of the algorithms by providing an effective way of making estimations, predictions, and decisions in the absence of communication or when trying to utilize sensor use. That is,

Yusuf, S. and Baber, C.
Handling Uncertainties in Distributed Constraint Optimization Problems using Bayesian Inferential Reasoning.
DOI: 10.5220/0009157108810888
In *Proceedings of the 12th International Conference on Agents and Artificial Intelligence (ICAART 2020) - Volume 2*, pages 881-888
ISBN: 978-989-758-395-7; ISSN: 2184-433X

the learned BBN could be used in making predictions, estimations, and decisions instead of direct communication. It could also reduces the use of stochastic variables in most of the DCOP algorithms such as Maximum Gain Message, Distributed Stochastic Algorithm, etc (Fioretto et al., 2018). Instead of random sample, closely correct values will be predicted. Issues arise when there is an absence of data to train the network, in that case, we propose knowledge sharing and rule-based inference to generate training data for the network (Bayesian Belief Network) training using expectation-maximization or gradient descent algorithm (Bottou, 2010; Dempster et al., 1977; Mandt and Hoffman, 2017). The agents learn from other agents during the mission and conduct learned network update on a time-to-time basis.

2 BACKGROUND

2.1 Distributed Constraint Optimization Problems (DCOPs)

The applications of a team of agents to perform together is growing such as in search and rescue missions (Bevacqua et al., 2015), sensor scheduling (Maheswaran et al., 2004), smart homes (Fioretto et al., 2017), traffic lights control (Brys et al., 2014) etc. The agents have limited resources (energy, time, communication, etc.) to accomplish such a mission. Therefore, they need to utilize the available resources by making decisions that will support other co-agents actions (Fioretto et al., 2018; Khan, 2018; Khan, 2018). In a multi-agent system, DCOP can be described as the tuple S:

$$S = \{A, V, D, C, \alpha\} \tag{1}$$

Where A is the set of agents, V is the set of variables for the agents, C is the cost functions to be optimized, D is the domain for the variables, and α is a function for the assignment of the variables. DCOP algorithms arrange agents in a constraint graph or pseudo-tree (Fioretto et al., 2018). In the constraint graph, agents represent the nodes, while the edges are the set of constraints values for the agents. In pseudo-tree structure, agents arrange in a tree-like structure with the hierarchical power of optimizing variables assignment (Fioretto et al., 2018; Ramchurn et al., 2010; Fioretto et al., 2015). That is, agents have local cost functions to optimize and communicate with neighbouring agents(agents with direct communication link) and agree on optimize values (Khan, 2018). In most of the DCOP algorithms, such as Maximum Gain Message (Maheswaran et al., 2004), Distributed Stochastic Algorithms (Ramchurn et al., 2010), Distributed Pseudotree Optimization (Fioretto et al., 2018; Fransman et al., 2019) etc, agents compute their optimal cost and inform other agents for optimizations. In the case of changing environments, agents change their cost functions with time. Uncertainty issues arise when the agent is not sure about its or other co-agents' variables, cost functions, and the outcome of the next steps on the optimization process (Le et al., 2016; Stranders et al., 2011). Uncertainty in DCOP can be defined as the tuple U.

$$U = \{A, V, D, C, \alpha, \lambda\} \tag{2}$$

Where A, V, D, C, and α were defined in equation (1) and λ is the degree of uncertainties agents have on their variables or cost functions. The degree of uncertainty of the agent varies, for instance, whether the agents knows the range of the variables (Romanycia, 2019) or not. Table 1 describes the degree of uncertainty in DCOP algorithms and their example.

Agents develop their knowledge of uncertainty based on their environment adaptability. For example, agents operating in a similar environment can have boundaries for their variables. Complete uncertainty is when the similarities between the agents' environments differ. For example, agents optimizing variables in a very windy and hot environment may have no prior likelihood of operating in a colder environment. No matter how high the rate of the uncertainty is, Bayesian learning algorithms (conjugate gradient descent or expectation-maximization) handle that effectively.

2.2 Bayesian Learning

Bayesian inferential learning allows the agents to familiarize themselves with the environment and make predictions, estimations, and conclusions on the variables using conditional probability of equation 3 (Fransman et al., 2019; Wang and Xu, 2014; Williamson, 2001).

$$P(X_i(t)|Y_i(t)) = \frac{(P(X_i(t)) * P(Y_i(t)|X_i(t)))}{(\Sigma_i^n P(X_i(t)) * P(Y_i(t)|X_i(t)))} \tag{3}$$

$X_1(t), X_2(t), X_3(t), ..., X_n(t)$ is the set of mutually exclusive events at a given time. That is, agents can compute other variables given the conditional probabilities of other mission variables. The agents' sensor information could be modelled using Bayesian Belief Network (BBN). BBN provides a graphical representation of events with their causal relationships (Wang and Xu, 2014; Williamson, 2001; Xiang, 2002). We regard this as a form of Situation Awareness, in which

Table 1: Degree of Agents Uncertainty and their Types.

Uncertainty Type	Meaning	Example
Bounded	When agent has knowledge about the range of the variable, but is not sure about it	Agents receiving services from other agents may use that to utilize their variables, but due to a communication link problem, agents have no options on that. For example, wind speed could range between the usual 1meter per second to 7 meters per second. Agents can perceive that its greater than 4 meters per second or has some likelihood higher than the others
Complete Uncertainty	When agents have no clue or hint about the value of a variable or cost function	For example, a team of rescue UAVs in Sahara deserts, change to the snowy environment, may have complete uncertainty on their variables

agents use sensor data to interpret their local environment and then reason about its likely state now and in the near future.

In multi-agent systems, agents familiarise themselves with their operating environment based on the previous mission data. These recorded data can be used to obtain a well-trained network (BBN) for making predictions and estimations of the agent's current and future variables with their uncertainties. The conjugate gradient descent and expectation-maximization algorithms could be used in training the networks (Romanycia, 2019). Expectation-maximization algorithms compute optimal predictions in two steps (i) computes conditional values by using (1) and (ii) iterates towards optimal predictions (Dempster et al., 1977). Gradient descent algorithm finds optimal predictions by following the steepness direction of the likelihood of the objective variables (Bottou, 2010; Mandt and Hoffman, 2017; Romanycia, 2019). We categorize the learning processes into two:

- Intra-agent Learning Process (local learning)
- Inter-agent Learning Process (central learning)

Intra-agent learning: agents learn from their previous actions and interactions with other agents and be able to learn, make predictions, estimations, and conclusion. For example, agents learn and monitor how they interact with other agents and the effect on their cost functions. In the absence of communication, they can use that learned network to make predictions. In order to avoid continuous learning and optimize resources, agents could learn on check-pointing bases (that is on a time to time basis).

Inter-agent learning process involves the sharing of information between the groups of agents and learned collectively. It may causes the updates of the local networks in order to reduce communications and uncertainties handling (figure 7). It occurs when agents are within a communication range or connected in a centralized passion.

3 RELATED WORK

Different algorithms were developed in solving DCOP for dynamic, probabilistic, and classical forms (Fioretto et al., 2018). For instance, in Maximum Gain Message Algorithm (Maheswaran et al., 2004), agents start with random allocation to their variables and inform neighbouring agents about those variables in order to have an optimal decision by adjusting the randomly selected variables. In Distributed Stochastic Algorithms (Zhang et al., 2005), agents do not communicate the selected random variables with other agents rather they keep adjusting the random variables until these fit the situation. Pectu and Faltings (Petcu and Faltings, 2005) describe a Distributed Pseudotree Optimization (DPOP) algorithm for solving DCOP based on arranging tree-like structures. The child nodes of the agents forward their variables to parents for optimization. In Fransman et al (Fransman et al., 2019) applied Bayesian inferential reasoning is used in solving the DPOP; that is, when agents arrange themselves in tree-like structures, they will optimize their variables using Bayesian inference before forwarding to their parents. Many algorithms were developed to tackle the environment dynamism such as Proactive DCOP algorithm (Billiau et al., 2012; Hoang et al., 2016; Hoang et al., 2017) in which agents react to environment changes instantly. Predictive dynamism handling, uncertainty tolerance, situation-awareness, and scalability remain the great challenge bedevilling the aforementioned algorithms.

In Léauté et al (Léauté et al., 2011) uncertain DCOP was defined and solved using heuristic-based algorithms with rewards forecasting using probability distributions, agents joint decision making, and risk assessment. They assume that the agents variables are stochastic and beyond the control of the agents. The model was tested on Vehicle Routing Problem (VRP) and shows the possibility of obtaining an optiml solution in an incomplete DCOP algorithms. A similar approach was used by Stranders et al (Stranders et al., 2011) to solve uncertain DCOP whereby the cost functions is independent of the agents' variables. The propose algorithms operate on acyclic graph and uses a concept of first-order stochastic dominance (Fioretto et al., 2018).

In this paper, we apply Bayesian learning to tackle the agents' uncertainty and environmental dynamism in DCOP. Agents learn individually as well as from other agents to know how to make an effective prediction using uncertain data. The learning algorithms used are conjugate gradient descent, and expectation-maximization (Bottou, 2010; Dempster et al., 1977; Mandt and Hoffman, 2017; Romanycia, 2019) which handle uncertainties and provides very good predictions and variables estimations. In the case of a highly changing environment, we propose a time-base learning algorithm such as gradient descent algorithm of (Bottou, 2010) to produce the learned BBN.

The learned BBN could be used in making optimal predictions, estimation, and conclusions in the absence of available data or communication link. It could also reduce the random variables allocations in the Maximum Gain Message (Maheswaran et al., 2004), Distributed Stochastic Algorithms (Hale and Zhou, 2015; Zhang et al., 2005), etc. Therefore, agents could make a perfect prediction and estimate optimal variables. This approach reduce communication, computation cost, and improve the scalability of DCOP algorithms.

4 THE MODEL

We subject the agents' uncertainties in solving DCOP problem to Bayesian learning algorithms in order to obtain an effective prediction tools. During the agents' operations (forest fire monitoring simulated on AMASE (https://github.com/afrl-rq/OpenAMASE, 2019) figure 1), the agents (UAVs) record their variables and the uncertainties in those variables due to missing values, delay in delivery, unreliable source, or error in data delivery to their Bayesian Belief Network to be updated using the sensor data.

Figure 1: Multi-agent Mission for Forest Fire Monitoring on AMASE.

Figure 1 describes the multi-UAVs mission for forest fire searching simulated on Aerospace Multi-agent Simulation Environment – AMASE (https://github.com/afrl-rq/OpenAMASE, 2019). The coloured triangular shapes represent the agents (UAVs) with their respective coloured dots destinations. The two irregular polygons represent the fire in the rectangular forest. Agents have an in-built Bayesian Belief Network updated using simple heuristics algorithms and information from the sensor data. For instance, whenever an agent detect a fire using its sensor, then it increase the probability of the true states of fire detection node while in the background the agent record all the mission data and possible uncertainties.

The agents' mission data are subjected to training purposes using the conjugate gradient descent algorithm or expectation maximization algorithms (Bottou, 2010; Dempster et al., 1977) to conduct the learning process. The learned network (i.e. output of the training process) could be used for making predictions on future variables. Figure 2 describes an example of simple BBN for tomorrow's rain forecasting based on today's temperature and rain.

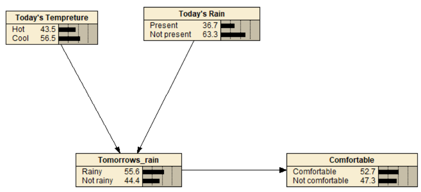

Figure 2: : Simple Bayesian Belief Network for Rain Forecast.

Figure 2 describe a simple BBN for rain forecast. The agents have similar BBN for probable heat source,wind speed, wind direction, etc. in forest fire monitoring. Experiment results from our multi-agent mission for forest fire searching proves that the learning algorithms work perfectly with uncertain data, though the uncertainty needs to be spread across BBN node's states (Figures 3 and 4). In the case of a highly dynamic environment, the learning process could use time-based learning algorithms like a time-base gradient descent algorithm of (Bottou, 2010). The learning process could occur concurrently with other agent's activities time or schedule after the mission (bad for a dynamic environment).

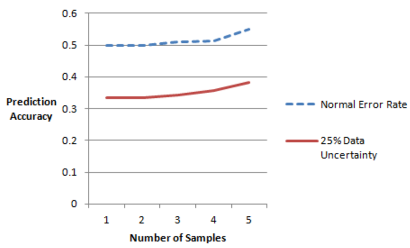

Figure 3: Prediction Perfection Comparison with 25% Uncertain Data.

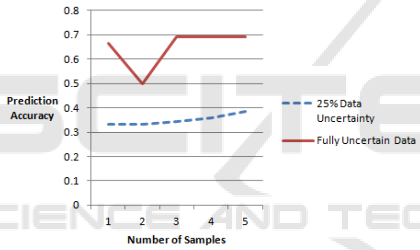

Figure 4: Prediction Perfection Comparison with full Uncertain Data.

Furthermore, Base on the experiment results on our simulation platform, like other learning approaches, agents need to generate a few training data for the learned network to be generated (which is a potential drawback). In the absence of training data, agents could use randomly assigned probabilities and learned with them before getting the real data or use equation (3). Agents often share learned cases during operations or after missions gathered by a central server and update their learned network. Figure 3 compares the prediction perfection of normal data (data without uncertainty) and uncertain data (data with 25% uncertainty) in wind direction node prediction from multi-agent forest fire monitoring. That is, we monitored the agents' prediction perfection in guessing future wind direction in a forest. We developed the model on the simulation platform Aerospace Multi-agent Simulation Environment- AMASE (https://github.com/afrl-rq/OpenAMASE, 2019) and gathered agents cases from 10, 100, 1000, 10000, and 100000 cases. The prediction perfection grows with the number of training samples. Surprisingly, the learning algorithms (both expectation-maximization and conjugate gradient) make better predictions using uncertainties due to a wider decision space. Figure 4 describes the complete uncertain in one of the states of two-state node, which made the prediction poor. Future works will look into training data utilization and spreading.

The agents segment their learning activity into two, inter and intra agent learning. In connection with other agents (i.e., global learning), agents learn to monitor their variables and learn new training data from other agents (i.e., inter-agent learning). The potential issues arise in managing the fusion of learning information from different agents. The learned network could be used to make predictions and reduces the use of communication and stochastic variables in solving DCOP. Figure 5 describe the architecture of the model.

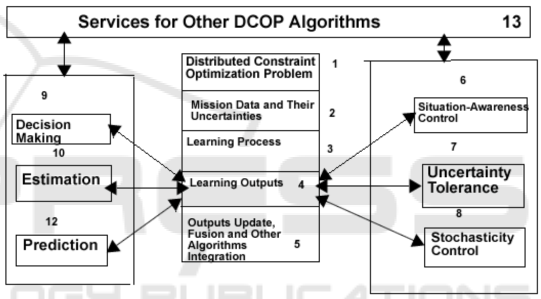

Figure 5: The Model Architectural Description in Multi-agent System

From figure 5, the agents identify their objective functions and constraint in step labelled (1) as Distributed Constraint Optimization Problems (DCOPs). Mission data and their uncertainties are labelled at (2) and send for learning purposes to layer (3)which contain all the learning algorithms in the in-built BBN form. The output of the learning process is a well trained network that will help agents in making predictions (11), estimations (10), decision making (9), agents situation-awareness control (6), uncertainty tolerance system (7), stochastic variable control (8) as in MGM, DSA, DPOP algorithms etc. Layer (5) is responsible for agents learned network update and knowledge fusion. Modules labelled 6,7,8,9,10, and 11 are the services for integration with other existing DCOP algorithms such as DPOP (Petcu and Faltings, 2005; Fransman et al., 2019), MGM(Maheswaran et al., 2004), (Maheswaran et al., 2004) etc as layer labelled (12).

4.1 Fitting the Model with DCOP Algorithms

In DCOP algorithms such as Maximum Gain Message (Maheswaran et al., 2004), Distributed Stochastic Algorithms (Zhang et al., 2005), etc. Agents start with random allocation of variables and communicate to neighbouring agents to optimize their variables. Instead of such blind random variables allocation, our model proposes the use of learned network (learned BBN) and Bayesian inference rule (3) in such variables allocations. The agents learn individually as well as learn collectively with other agents. Figure 6 describes the agents self-learning process.

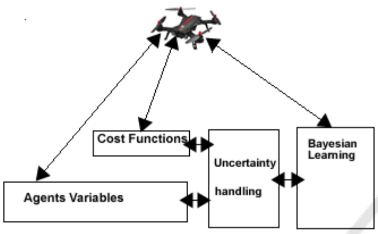

Figure 6: Agent Bayesian Local Learning Process.

From figure 6, the agents maintain its DCOP variables and cost functions, then assign the uncertainty in their values as describe in table 1. The mission data will be subjected to Bayesian learning process as describe in figure 5. The agents learn their variables and cost functions optimization transition by supplying the learning cases to the Bayesian learning algorithms. At each cycle, the agents determine the level of uncertainty to each variable and costs functions. The supplied data will be sent for optimal network training. The agents keep updating their knowledge on time bases to avoid wasting computation power. Due to the rate of changing the environment, the learning algorithms provide a priority-based training approach to cope with the changing environment (i.e., agents treat recent cases with higher priority). Another approach of learning is the central learning process, whereby agents learn from other agents and update their own knowledge and knowledge about those agents. In a centralized system, the server is responsible for the learning process and agents' knowledge updates. In a decentralized system, when agents come within the communication range, agents learn by combining all their training data. The training case file is small in size to which could be replicated on all agents' memory (e.g., UAVs). The number of iterations could be limited in order to avoid large executions. Figure 7 describes the central learning process, agents B, C, D, and E are within the communication range and learn from each other (through communication). As such, they can learn together and share experiences. Agents A and F are not within the communication range as such using their own mission data for learning purposes (local learning).

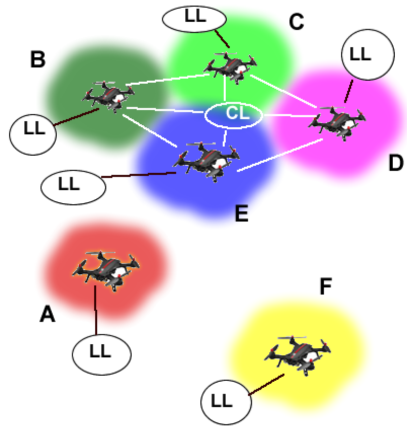

Figure 7: Centralized and Local Multi-agent Process

Therefore, Local learning allows the agents to learn from its generated sensor data and give it an opportunity to optimize its sensor data, sensor use scheduling, perform local data check off, etc. On the other hand, the agent acquire knowledge of unvisited areas from other co-agents through the centralized learning (figure 7).

5 CONCLUSIONS AND FUTURE WORK

We proposed the use of Bayesian inferential reasoning and learning to tackle dynamism and uncertainty in Distributed Constraint Optimization (DCOP) algorithms. The agents learn in two steps, local learning (self-learning) and central learning (where agents learn from other agents). In each learning strategy, the agents run the conjugate gradient descent algorithm or expectation maximization algorithms, in dealing with the uncertainty problem. Experiment results using multi-agents missions for forest fire prove that the learning process works best by having uncertainties spread across agents states, which perfect better than real cases. To our knowledge, this is the first time to tackle uncertainties in DCOP using Bayesian inferential reasoning and learning, modelling forest fire monitoring as DCOP, and introduction of situation-awareness to DCOP.

Agents use the learned network (BBN) to make estimations or optimization predictions instead of random variables selections, as in Maximum Gain Message (MGM), Distributed Stochastic Algorithms(DSA), etc.(Fioretto et al., 2018; Maheswaran et al., 2004; Petcu and Faltings, 2005) As such, it will reduce the complexity of the DCOP algorithms my removing, communications, and computation cost of DCOP algorithms. It would also improve scalability and made them usable in dangerous and non-communication environment. The propose model differ with Bayesian Distributed Pseudo Tree Optimization of Fransman et al (Fransman et al., 2019) by introducing learning opportunities, Situation Awareness, and uncertainties handling.

Future work focus attention on training data utilization and agents situation awareness. That is, we are going to look at the minimum amount of data needed for the production of accurate predictions tools. We are intended in improving the agents ability to consider current environmental situation and future activities as well.

The propose architecture will later on undergo comparative analysis and evaluation with other DCOP algorithms operating in higly dynamic or uncertain environment. We will also look at agents Bayesian learning in a highly changing environment together with architectural fusion. with other learning algorithms

ACKNOWLEDGEMENTS

The authors wish to express their gratitude and appreciation for any comments that help in making this paper a great one. The authors wish to also express their appreciation to Petroleum Technology Trust Fund (PTDF) of Nigeria for the sponsorship of this research.

REFERENCES

Bevacqua, G., Cacace, J., Finzi, A., and Lippiello, V. (2015). Mixed-initiative planning and execution for multiple drones in search and rescue missions. In *Proceedings of the Twenty-Fifth International Conference on International Conference on Automated Planning and Scheduling*, ICAPS'15, pages 315–323. AAAI Press. event-place: Jerusalem, Israel.

Billiau, G., Chang, C. F., Ghose, A., and Miller, A. A. (2012). Using distributed agents for patient scheduling. Lecture Notes in Computer Science, pages 551–560, Berlin, Heidelberg. Springer.

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. page 10.

Brys, T., Pham, T. T., and Taylor, M. E. (2014). Distributed learning and multi-objectivity in traffic light control. *Connection Science*, 26:65.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. [Online; accessed 2019-11-14].

Fioretto, F., Pontelli, E., and Yeoh, W. (2018). Distributed constraint optimization problems and applications: A survey. *J. Artif. Int. Res.*, 61(1):623–698.

Fioretto, F., Yeoh, W., and Pontelli, E. (2015). Multi-variable agents decomposition for dcops to exploit multi-level parallelism (extended abstract). page 2.

Fioretto, F., Yeoh, W., and Pontelli, E. (2017). A multiagent system approach to scheduling devices in smart homes. AAMAS '17, page 981–989, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems. event-place: São Paulo, Brazil.

Fransman, J., Sijs, J., Dol, H., Theunissen, E., and De Schutter, B. (2019). Bayesian-dpop for continuous distributed constraint optimization problems. AAMAS '19, page 1961–1963, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems. event-place: Montreal QC, Canada.

Hale, J. Q. and Zhou, E. (2015). A model-based approach to multi-objective optimization. WSC '15, page 3599–3609, Piscataway, NJ, USA. IEEE Press. event-place: Huntington Beach, California.

Hoang, K. D., Fioretto, F., Hou, P., Yokoo, M., Yeoh, W., and Zivan, R. (2016). Proactive dynamic distributed constraint optimization. page 9.

Hoang, K. D., Hou, P., Fioretto, F., Yeoh, W., Zivan, R., and Yokoo, M. (2017). Infinite-horizon proactive dynamic dcops. AAMAS '17, page 212–220, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems. event-place: São Paulo, Brazil.

Khan, M. M. (2018). *Speeding up GDL-based distributed constraint optimization algorithms in cooperative multi-agent systems*. PhD thesis. [Online; accessed 2019-11-21].

Léauté, T., Faltings, B., and De, c. P. F. (2011). *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence Distributed Constraint Optimization under Stochastic Uncertainty*.

Le, T., Fioretto, F., Yeoh, W., Son, T. C., and Pontelli, E. (2016). Er-dcops: A framework for distributed constraint optimization with uncertainty in constraint utilities. page 9.

Maheswaran, R. T., Pearce, J. P., and Tambe, M. (2004). Distributed algorithms for dcop: A graphical-game-based approach. page 8.

Mandt, S. and Hoffman, M. D. (2017). Stochastic gradient descent as approximate bayesian inference. page 35.

Petcu, A. and Faltings, B. (2005). A scalable method for multiagent constraint optimization. IJCAI'05, page 266–271, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. event-place: Edinburgh, Scotland.

Pujol-Gonzalez, M. (2011). Multi-agent coordination: Dcops and beyond. IJCAI'11, page 2838–2839. AAAI Press. event-place: Barcelona, Catalonia, Spain.

Ramchurn, S. D., Farinelli, A., Macarthur, K. S., and Jennings, N. R. (2010). Decentralized coordination in robocup rescue. *The Computer Journal*, 53(9):1447–1461.

Romanycia, M. (2019). Netica-j reference manual. page 119.

Stanton, N. A., Stewart, R., Harris, D., Houghton, R. J., Baber, C., McMaster, R., Salmon, P., Hoyle, G., Walker, G., Young, M. S., Linsell, M., Dymott, R., and Green, D. (2006). Distributed situation awareness in dynamic systems: theoretical development and application of an ergonomics methodology. *Ergonomics*, 49(12-13):1288–1311.

Stranders, R., Delle Fave, F. M., Rogers, A., and Jennings, N. (2011). U-gdl: A decentralised algorithm for dcops with uncertainty. [Online; accessed 2019-11-25].

Wang, J. and Xu, Z. (2014). Bayesian inferential reasoning model for crime investigation. page 11.

Williamson, J. (2001). Bayesian networks for logical reasoning. page 19.

Xiang, Y. (2002). Probabilistic reasoning in multiagent systems - a graphical models approach.

Yeoh, W., Varakantham, P., Sun, X., and Koenig, S. (2011). Incremental dcop search algorithms for solving dynamic dcops (extended abstract). page 2.

Zhang, W., Wang, G., Xing, Z., and Wittenburg, L. (2005). Distributed stochastic search and distributed breakout: properties, comparison and applications to constraint optimization problems in sensor networks. *Artificial Intelligence*, 161(1-2):55–87.