Delft University of Technology

Delft Center for Systems and Control

Technical report 19-020

Bayesian-DPOP for continuous distributed constraint optimization problems*

J. Fransman, J. Sijs, H. Dol, E. Theunissen, and B. De Schutter

If you want to cite this report, please use the following reference instead:
J. Fransman, J. Sijs, H. Dol, E. Theunissen, and B. De Schutter, "Bayesian-DPOP for continuous distributed constraint optimization problems," *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS'19)*, Montreal, Canada, pp. 1961–1963, May 2019.

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: https://www.dcsc.tudelft.nl

* This report can also be downloaded via https://pub.bartdeschutter.org/abs/19_020.html

Bayesian-DPOP for Continuous Distributed Constraint Optimization Problems

Extended Abstract

JEROEN FRANSMAN, Delft University of Technology
JORIS SIJS, Netherlands Organisation for Applied Scientific Research (TNO)
HENRY DOL, Netherlands Organisation for Applied Scientific Research (TNO)
ERIK THEUNISSEN, Netherlands Defense Academy (NLDA)
BART DE SCHUTTER, Delft University of Technology

In this work, the novel algorithm Bayesian Dynamic Programming Optimization Procedure (B-DPOP) is presented to solve multi-agent problems within the Distributed Constraint Optimization Problem framework. The Bayesian optimization framework is used to prove convergence to the global optimum of the B-DPOP algorithm for Lipschitz-continuous objective functions. The proposed algorithm is assessed based on the benchmark problem known as dynamic sensor placement. Results show increased performance over related algorithms in terms of sample-efficiency.

Additional Key Words and Phrases: DCOP; DPOP; Bayesian Optimization; Distributed Optimization

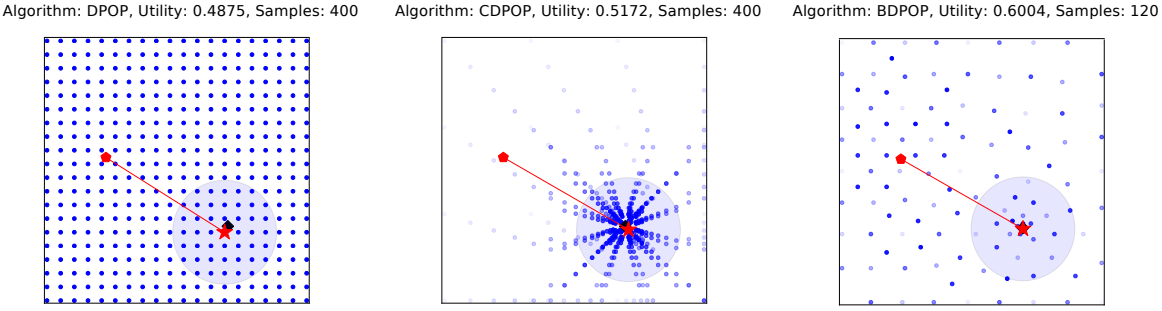

Fig. 1. Graphical demonstration of sampling within DPOP, C-DPOP, and B-DPOP for a 1-agent, 1-target (black diamond) dynamic sensor coordination problem. The sensor range is indicated by a blue disk, the agent starts at the red hexagon and moves towards the red star, and the samples are shown as blue dots of which transparency indicates the chronological order. DPOP has no adaptivity and applies uniform sampling. The sampling of C-DPOP is adjusted with respect to the optimum of the previous iteration. B-DPOP converges in a sample-efficient manner, requiring less samples and achieving a higher utility.

1 INTRODUCTION

Many real-world problems that involve multi-agent systems become intractable if solved through a centralized process, such as scheduling execution minimization [7], mobile sensor coordination [11], and hierarchical task networks mapping [8]. Such problems can be represented within the Distributed Constraint Optimization

Problem (DCOP) framework [6]. Numerous solvers for DCOP have been proposed; for a detailed overview and taxonomy, the reader is referred to [1], [4].

A DCOP is a problem of optimizing a global utility function in a distributed manner through the exchange of messages between agents. Typically, problems that include variables with discrete domains are considered. For problems with continuous domains, a typical solution has been to discretize the continuous domains via a uniform grid, which introduces drawbacks such as exponential growth in the problem complexity and cannot guarantee to find the optimal solution.

In this paper, the novel solver Bayesian-DPOP (B-DPOP) extends the Dynamic Programming Optimization Procedure (DPOP) algorithm [6], to remove the need for discretizing the continuous domains while retaining convergence guarantees and efficient use of function sampling.

2 BAYESIAN DPOP

DCOP defines a global utility function $G = \sum_{f_i \in F} f_i$, that is the sum of all local utility functions f_i in function set F, where i is the function index. The B-DPOP algorithm optimizes a DCOP by executing four phases; (1) Pseudo tree construction; (2) Assignment of local utility functions; (3) Sample propagation, based on Bayesian optimization; (4) Value propagation. In the third phase, the local utility functions are modeled by Gaussian processes defined by a kernel $\kappa_i(\cdot)$ as $f_i(\cdot) \approx \mathrm{GP}(0, \kappa_i(\cdot))$.

Kernels can be combined through addition or multiplication into a composite kernel [2]. This allows the modeling of the local functions by separate kernels (specific to the function properties) and the resulting composition of these kernels will resemble the global utility function. Therefore, the search for the global optimum within B-DPOP can be reduced to the search for optima of the Gaussian processes.

The convergence to the optimum ($\boldsymbol{y}^* = f_i(\boldsymbol{x}^*)$) can be guaranteed through *dense sampling* of the utility function [9]. Samples are defined as $\mathcal{O}_{1:s} = \{\mathcal{O}_1, \ldots, \mathcal{O}_s\}$, where $\mathcal{O}_j = (\boldsymbol{x}_j, \boldsymbol{y}_j)$ is the input/output pair of a local utility function $\boldsymbol{y}_j = f_i(\boldsymbol{x}_j)$.

2.1 Acquisition function

From these samples, the mean function ($\mu(\boldsymbol{x}|\mathcal{O}_{1:s}) = \boldsymbol{k}^T \boldsymbol{K}^{-1} \boldsymbol{y}_{1:s}$) and the variance function ($\sigma^2(\boldsymbol{x}|\mathcal{O}_{1:s}) = \kappa(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}^T \boldsymbol{K}^{-1} \boldsymbol{k}$) can be defined, where $\boldsymbol{K}$ is the Gramian matrix, i.e. $K_{n,m} = \kappa(\boldsymbol{x}_n, \boldsymbol{x}_m)$ for all $n, m \in \{1 : s\}$, and $\boldsymbol{k} = [\kappa(\boldsymbol{x}_1, \boldsymbol{x}), \ldots, \kappa(\boldsymbol{x}_s, \boldsymbol{x})]^T$ denotes the cross-correlation vector between the samples and input $\boldsymbol{x}$. The acquisition function q^{EI} can be defined based on the mean and variance functions, parameter ξ, and the maximum sampled output $\boldsymbol{y}^+_{1:s} = \max_{\mathcal{O}_| \in \mathcal{O}_{1:s}} \boldsymbol{y}_j$ as

$$q^{\mathrm{EI}}(\boldsymbol{x}, \xi) = \begin{cases} c(\boldsymbol{x}) \Phi\left(\frac{c(\boldsymbol{x})}{\sigma(\boldsymbol{x})}\right) + \sigma(\boldsymbol{x}) \phi\left(\frac{c(\boldsymbol{x})}{\sigma(\boldsymbol{x})}\right) & \text{if } \sigma(\boldsymbol{x}) > 0 \\ 0 & \text{if } \sigma(\boldsymbol{x}) = 0 \end{cases} \tag{1}$$

$$c(\boldsymbol{x}) = \mu(\boldsymbol{x}) - \left(\boldsymbol{y}^+_{1:s} + \xi\right)$$

2.2 Sampling and converge

Samples are iteratively selected by optimization of the acquisition function, as $\boldsymbol{x}_{s+1} = \arg\max_{\boldsymbol{x}} q(\boldsymbol{x}|\mathcal{O}_{1:s})$. In this work, the expected improvement acquisition function $q^{\mathrm{EI}}(\cdot)$ [5] is combined with the Matérn kernel in order to produce dense samples [10] within a search region $\mathcal{S}$. The search region, $\mathcal{S}_{1:s} = \{\boldsymbol{x} \mid q^{\mathrm{EI}}_{1:s}(\boldsymbol{x}, 0|\mathcal{O}_{1:s}) > 0\}$,

will tend to an empty set when the number of samples tends to infinity. Therefore, if the search region includes the optimal input, $\boldsymbol{x}^* \in \mathcal{S}_{1:s}$, the solution will convergence to the global optimum. In order to find the conditions for which $\boldsymbol{x}^* \in \mathcal{S}_{1:s}$ holds, the set $\mathcal{U}_{1:s} = \{\boldsymbol{x} \mid \bar{f}_i(\boldsymbol{x}|\mathcal{O}_{1:s}) > \boldsymbol{y}^+_{1:s}\}$ is defined. The function $\bar{f}_i(\boldsymbol{x}|\mathcal{O}_{1:s})$ is the upper bound function based on the Lipschitz constant L_{f_i} of the function and $\mathcal{O}_{1:s}$ as, $\bar{f}_i(\boldsymbol{x}|\mathcal{O}_{1:s}) = \min_{\mathcal{O}_| \in \mathcal{O}_{1:s}} L_{f_i}\|\boldsymbol{x} - \boldsymbol{x}_j\| + \boldsymbol{y}_j$. By definition, $\mathcal{U}_{1:s}$ does not include the samples $\mathcal{O}_{1:s}$ because the upper bound function is equal to the sample output for all sample inputs, i.e. $\bar{f}_{i_j}(\boldsymbol{x}_j|\mathcal{O}_j) = \boldsymbol{y}_j$. Therefore, the set $\mathcal{U}_{1:s}$ includes the optimal input $\boldsymbol{x}^* \in \mathcal{U}_{1:s}$ or the optimal input is included in the sample set, $\boldsymbol{x}^* \in \mathcal{O}_{1:s}$. This attribute can be exploited by proper selection of the kernel parameters such that $\mathcal{U}_{1:s} \subseteq \mathcal{S}_{1:s}$. Consequently, if $\mathcal{S}_{1:s} \to \emptyset$, then $\mathcal{U}_{1:s} \to \emptyset$, guaranteeing the optimal input will be sampled.

2.3 Definition of kernel parameters

Based on the Lipschitz constants of the local utility functions, the parameters of κ_i can be determined such that $\mathcal{U}_{1:s} \subseteq \mathcal{S}_{1:s}$ In this work, the Matérn kernel with parameter $\nu = 3/2$ and the expected improvement acquisition function q^{EI} with $\xi = 0$ is chosen to exemplify the selection of the kernel parameters λ and l. The kernel length scale l will be used to bound the derivative of the mean function μ and the kernel scale λ will be used to scale the standard deviation σ by considering two initial samples $\mathcal{O}_1$ and $\mathcal{O}_2$. The derivative of the mean function can be bounded from above to the Lipschitz constant L_{f_i} by setting $l \geq \frac{\sqrt{3}e^{-1}y_1}{L_{f_i}}$.

Due to the bounded derivative of $\mu(\cdot)$ and the fact that standard deviation σ is only dependent on the distance between samples, the required value of the standard deviation $\sigma(\cdot)$ can be defined as $\sigma_{1:s}(x) \geq \bar{f}_i(\boldsymbol{x}|\mathcal{O}_{1:s}) - \mu_{1:s}(\boldsymbol{x}) \geq L_{f_i}\|\boldsymbol{x}_2 - \boldsymbol{x}_1\|$.

3 SIMULATION RESULTS

The performance of B-DPOP is now evaluated for randomly generated dynamic sensor coordination problems (adapted from [12]) where mobile sensors with limited sensing range coordinate their positions (from initial locations) to sense targets, within a two-dimensional plane. The initial location I of the agents and location T of the targets are chosen at random.

The problem is modeled within the DCOP framework and has been solved by B-DPOP and compared to the DPOP [6] and C-DPOP [3] algorithms. As DPOP cannot handle continuous domains, the domains are discretized via a uniform grid. The C-DPOP algorithm is used as comparison as it is designed to solve DCOP for continuous domains. The resulting sampling strategies of the algorithms can be seen in Figure 1, where the distinction between the static DPOP and dynamic C-DPOP and B-DPOP sampling is clearly visible.

The achieved global utility is compared based on the number of samples of the local utility functions of the algorithms to provide a common metric for the computational requirement. Simulation results are shown in Figure 2, where the amount of samples is kept equal for all experiments. The sample-efficiency of B-DPOP results in a higher overall achieved utility.

REFERENCES

[1] Jesus Cerquides, Alessandro Farinelli, Pedro Meseguer, and Sarvapali D Ramchurn. 2013. A tutorial on optimization for multi-agent systems. *Comput. J.* 57, 6 (2013), 799–824.

[2] David K Duvenaud, Hannes Nickisch, and Carl E Rasmussen. 2011. Additive gaussian processes. In *(NIPS) advances in Neural Information Processing Systems*. 226–234.

[3] Jeroen Fransman, Joris Sijs, H.S. Dol, E. Theunissen, and Bart De Schutter. 2018. Distributed constraint optimization for continuous mobile sensor coordination. In *(ECC) European Control Conference*.

[4] Allan R. Leite, Fabrício Enembreck, and Jean-Paul A. Barthès. 2014. Distributed Constraint Optimization Problems: Review and perspectives. *Expert Systems with Applications* 41, 11 (2014), 5139–5157.

[5] J Mockus. 1982. The Bayesian approach to global optimization. *System Modeling and Optimization* 38 (1982), 473–481.

[6] Adrian Petcu and Boi Faltings. 2005. DPOP: A scalable method for multiagent constraint optimization. In *(IJCAI) International Joint Conference on Artificial Intelligence*. 266–271.

[7] Denise M.V. Sato, André P. Borges, Peter Márton, and Edson E. Scalabrin. 2015. I-DCOP: Train classification based on an iterative process using distributed constraint optimization. *Procedia Computer Science* 51 (2015), 2297–2306.

[8] Evan Sultanik, Pragnesh Jay Modi, and William C Regli. 2007. On modeling multiagent task scheduling as a distributed constraint optimization problem. In *(IJCAI) International Joint Conference on Artificial Intelligence*. 247–253.

[9] Aimo Törn and Antanas Žilinskas. 1989. *Global optimization*. Vol. 350. Springer.

[10] Emmanuel Vazquez and Julien Bect. 2010. Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. *Journal of Statistical Planning and Inference* 140, 11 (2010), 3088–3095.

[11] Roie Zivan, Tomer Parash, and Yarden Naveh. 2015. Applying max-sum to asymmetric distributed constraint optimization. In *(IJCAI) International Joint Conference on Artificial Intelligence*. 432–439.

[12] Roie Zivan, Harel Yedidsion, Steven Okamoto, Robin Glinton, and Katia Sycara. 2015. Distributed constraint optimization for teams of mobile sensing agents. *Autonomous Agents and Multi-Agent Systems* 29, 3 (2015), 495–536.

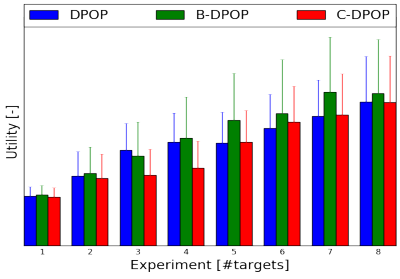

Fig. 2. Achieved utility comparison for a 3-agent problem based on 20 randomly generated problems with constant amount of samples.