**Proceedings of the ASME 2023
International Design Engineering Technical Conferences and
Computers and Information in Engineering Conference
IDETC/CIE2023
August 20-23, 2023, Boston, Massachusetts**

DETC2023-115112

MULTI-AGENT BAYESIAN OPTIMIZATION FOR UNKNOWN DESIGN SPACE EXPLORATION

Siyu Chen[1], Alparslan Emrah Bayrak [2], Zhenghui Sha[1,*],

[1]Walker Department of Mechanical Engineering, The University of Texas at Austin, Austin, TX
[2]School of Systems and Enterprises, Stevens Institute of Technology, Hoboken, NJ

ABSTRACT

This paper proposes a multi-agent Bayesian optimization (MABO) framework as a reference model for rational design teams to study the effects of information exchange on a team's search performance in finding global optimum of complex objective functions with many local optima. The core idea of the framework has three main steps. First, the design space is divided into regions based on the number of agents involved in the search. In each region, only one agent works on the part of the objective function. Second, a global-local communication strategy is developed to allow agents in local searches to share their sampled design points with a global evaluator. The global evaluator computes the posterior mean and variance based on all sampled points from local agents and evaluates the acquisition function (e.g., the expected improvement) to recommend the next sampling decisions for local agents. Third, when making the decision about where to sample next, each local agent only has access to the expected improvement evaluated in its local region and chooses the design that yields the largest value locally. To evaluate how the information exchange between agents and between local and global impact the search results, our framework is compared with a multi-agent model that does not allow information sharing and global-local interaction. Furthermore, we evaluated the performance of the model based on benchmark functions with varying complexities and also investigated the impact of the number of agents on search performance. We observe that when information sharing is allowed and global-local interaction is enabled in all scenarios, there is a significant improvement in convergence speed as well as the success rate of convergence.

Keywords: Multi-agent System (MAS), Bayesian Optimization (BO), Design Team, Design Space Exploration

*Corresponding author: zsha@austin.utexas.edu

1. INTRODUCTION

Design space exploration is the process of finding the best design solution that meets all requirements and constraints by exploring and evaluating different design alternatives available for a given problem [1]. It is typically a sequential process where knowledge of the design space is acquired through a series of design assessments, rather than starting with a full understanding of the design space of evaluation metrics at the beginning. Furthermore, a design process, particularly for complex design problems, usually involves more than just one person making decisions. It is rather a team effort where decisions made by other members influence the decision of each member.

Coordinating the decisions of team members in such a process is an essential problem that could significantly influence the effectiveness of design teams [2]. The impact of mechanisms and the frequency of interactions between design team members on design outcomes have been studied in the literature [3, 4]. As an example, the work by McComb et al. finds that if design team members all work on the same configuration problem, the optimal number of interactions between team members should be zero [5]. However, in practice, it is not quite common to see teams where all members work on the same problem, but there is rather a division of labor where each member works on a different sub-problem within a system-level design problem. Such a case deserves a new investigation since the coordination of subproblems requires some information sharing between team members [6]. In this paper, we computationally study the impact of communication between team members (i.e., computational design agents) on design exploration performance when each agent works on a different portion of the design space. We use rational agents that act to maximize their own utility, meanwhile can share their respective design samples with a global evaluator for optimal decision-making, in order to study the impact of information sharing independent of confounding human factors.

Design space exploration can be formulated as a black-box optimization problem when the space or function form is unknown to designers. Bayesian optimization (BO) is knowledge-based reasoning to explore unknown design spaces informed by past experience, or data [7], which provides rational design recommendations with future sampling decisions. BO typically estimates the model using a Gaussian process, which takes into account the uncertainty during the search and adopts an acquisition function that determines where to sample next in the process. This approach effectively balances exploration and exploitation to achieve the optimum [8].

The idea of BO has been used for decades in the literature, including for well-known design optimization algorithms such as Efficient Global Optimization [9]. Other examples include the work in [10], which proposes a BO approach that can effectively handle variable-size design space problems, with results demonstrating superior performance compared to other optimization methods. BO has also been applied to practical design application contexts as well. The study in [11] argues that BO is an efficient and effective method of exploring the design space of hardware accelerators, which can significantly reduce the exploration time while still achieving high-quality designs. Another example in [12] introduces a data-driven approach to design space exploration and exploitation based on BO for additive manufacturing.

In spite of those successful applications, however, for problems with a large design space and complex non-linear functions, where many local optima exist, it is difficult for a single BO to efficiently determine the global optimal value within the design space. For complex design problems, the Multidisciplinary Design Optimization (MDO) literature promotes the use of decomposition-based approaches where a system-level problem is partitioned into smaller subproblems to be solved in a coordinated fashion [13]. For instance, Bayrak et al. show that partitioning a high-dimensional problem into smaller subproblems enables the finding of effective solutions with multiple agents compared to a case where the entire problem is addressed by a single agent [14].

This paper presents a multi-agent BO (MABO) framework to tackle the optimization of complex design problems, enabling a team of agents to learn from their local design space and collaborate by sharing global information with the other agents in the team. We use this framework as a reference model to computationally study the value of interactions and information sharing within design teams where multiple rational agents combine their information to solve local design problems and find the global optimal solution. To the best of our knowledge, the use of MABO as a model of rational design teams to study the impact of information sharing accounting for uncertainty in design space is new in the literature. Such a study can provide upper bounds on team performance with an appropriate information-sharing mechanism.

The MABO framework comprises three main steps: 1) *Design space partitioning*: The design space is divided into regions based on the number of agents involved in the search. Each region is searched by a single agent, which helps avoid redundant search efforts and reduce computational cost; 2) *Global-local communication strategy*: The MABO framework incorporates a global-local interaction that enables agents in local searches to share their sampled design points with a global evaluator. The global evaluator evaluates the acquisition function based on the sampled points from all agents to recommend the next sampling decisions for the agents. This communication strategy allows for effective information sharing and the utilization of global knowledge, which helps avoid getting trapped in local optima. 3) *Local decision-making*: Each agent only has access to the expected improvement evaluated in its local region and chooses the design that yields the largest value locally. This allows for an effective exploration of the design space while prioritizing local optima. The paper evaluates the performance of the proposed MABO framework in different information-sharing scenarios using benchmark functions of varying complexity and number of agents. The results demonstrate that the MABO framework significantly improves the convergence speed compared to a multi-agent model that does not allow information sharing and global-local interaction.

This paper is structured as follows. Section 2 provides technical background on BO. Next, Section 3 delves into the technical details of the proposed MABO framework, outlining the problem formulation and the solution approach with multiple agents. Sections 4 and 5 present the experimental settings and results, comparing the proposed MABO method with a global-local interaction and a method without a global evaluator. Section 6 further discusses the research results and draws insights into the impact of information sharing on design team performance. Finally, Section 7 concludes the paper with a summary of findings and limitations that lead to future work.

2. BACKGROUND ON BAYESIAN OPTIMIZATION

We use BO to model the decision-making of each agent in a team to find the optimum of an unknown objective function in its local region defined by a design space partitioning. This section provides a brief introduction to the preliminaries of BO. A typical BO process comprises two major components, 1) a statistical inference method, typically Gaussian process (GP) regression, to model the unknown objective function value with uncertainty and 2) an acquisition function to decide where to sample in the design space [15]. In order to find the global optimum of a black-box objective function $f(\mathbf{x})$, where $\mathbf{x} \in A$; A is a d-dimensional design space domain $A \subseteq \mathbb{R}^d$, BO updates the posterior mean and variance of a Gaussian process given the prior data, and an acquisition function selects the next best guess for the optimum based on the updated posterior probability distribution. Note that this process assumes that agents always act rationally to maximize their own utility represented by the acquisition function and does not include confounding human factors in the analysis.

2.1 Gaussian process

Gaussian process is a commonly used statistical inference model, which defines a distribution over possible unknown functions [16]. BO realizes the reasoning about $f(\mathbf{x})$ by choosing an appropriate Gaussian process prior:

$$\mathbf{f}(\mathbf{x}_{1:k}) \sim \mathcal{GP}\left(\mu_0(\mathbf{x}_{1:k}), \boldsymbol{\Sigma}_0(\mathbf{x}_{1:k}, \mathbf{x}_{1:k})\right), \tag{1}$$

where the set of observations is $\mathcal{D} = (\mathbf{x}_{1:k}, \mathbf{f}(\mathbf{x}_{1:k}))$, $\mathbf{x}_{1:k} = [\mathbf{x}_1, \ldots, \mathbf{x}_k]$, $\mathbf{f}(\mathbf{x}_{1:k}) = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_k)]$,

$\mu_0(\mathbf{x}_{1:k}) = [\mu_0(\mathbf{x}_1), \ldots, \mu_0(\mathbf{x}_k)]$ is the mean vector by evaluating a mean function μ_0 at each $\mathbf{x}_1, ..., \mathbf{x}_k$, and $\boldsymbol{\Sigma}_0(\mathbf{x}_{1:k}, \mathbf{x}_{1:k}) = [\Sigma_0(\mathbf{x}_1, \mathbf{x}_1), \ldots, \Sigma_0(\mathbf{x}_1, \mathbf{x}_k); \ldots; \Sigma_0(\mathbf{x}_k, \mathbf{x}_1), \ldots, \Sigma_0(\mathbf{x}_k, \mathbf{x}_k)]$ is constructed by covariance $\Sigma_0(\cdot, \cdot)$ between each observation. Given the observation data $\mathcal{D}$, the posterior probability distribution is defined as [15]:

$$
\begin{aligned}
&f(\mathbf{x}) \mid \mathbf{f}(\mathbf{x}_{1:k}) \sim \mathcal{GP}\left(\mu(\mathbf{x}), \sigma^2(\mathbf{x})\right) \\
&\mu(\mathbf{x}) = \boldsymbol{\Sigma}_0(\mathbf{x}, \mathbf{x}_{1:k}) \boldsymbol{\Sigma}_0(\mathbf{x}_{1:k}, \mathbf{x}_{1:k})^{-1} (\mathbf{f}(\mathbf{x}_{1:k}) - \mu_0(\mathbf{x}_{1:k})) + \mu_0(\mathbf{x}) \\
&\sigma^2(\mathbf{x}) = \Sigma_0(\mathbf{x}, \mathbf{x}) - \boldsymbol{\Sigma}_0(\mathbf{x}, \mathbf{x}_{1:k}) \boldsymbol{\Sigma}_0(\mathbf{x}_{1:k}, \mathbf{x}_{1:k})^{-1} \boldsymbol{\Sigma}_0(\mathbf{x}_{1:k}, \mathbf{x}),
\end{aligned}
\tag{2}
$$

where $\mu(\mathbf{x})$ denotes the posterior mean and and $\sigma^2(\mathbf{x})$ denotes the posterior variance.

2.2 Acquisition function

An acquisition function is a heuristic used to determine the next point to sample in the search space. This function takes the probabilistic surrogate model, introduced in Section 2.1 that approximates the objective function as input. The next observation (sampling point) for the search is selected by optimizing the acquisition function [15] while balancing the search strategy between exploration and exploitation. Several acquisition functions are widely used in BO, such as Probability of Improvement (PI) [17], Expected Improvement (EI) [7], Lower Confidence Bound (LCB) [18] and Thompson Sampling (TS) [19]. In this study, we adopt EI for the acquisition function due to its high sensitivity to improvements and fast convergence speed to the optimum [9].

Assuming that the design problem is formulated as a minimization problem, the corresponding utility function can be defined as follows:

$$
u(\mathbf{x}) = max(0, f^*(\mathbf{x}) - f(\mathbf{x})), \tag{3}
$$

where $f^*(\cdot)$ is the minimum value of $f(\cdot)$ observed so far. Using this utility, the acquisition function with EI can be formulated as follows:

$$
\begin{aligned}
a_{\mathrm{EI}}(\mathbf{x}) =& \mathbb{E}[u(\mathbf{x}) \mid \mathbf{x}, \mathcal{D}] \\
=& \int_{-\infty}^{f^*} (f^* - f)\, \mathcal{N}(f^*; \mu(\mathbf{x}), \sigma^2(\mathbf{x})) \mathrm{d}f \\
=& (f^* - \mu(\mathbf{x}))\, \Phi\left(f^*; \mu(\mathbf{x}), \sigma^2(\mathbf{x})\right) \\
&+ \sigma^2(\mathbf{x}) \mathcal{N}\left(f^*; \mu(\mathbf{x}), \sigma^2(\mathbf{x})\right),
\end{aligned}
\tag{4}
$$

where μ and σ^2 are mean and variance functions of the posterior probability distribution for f given by Eq. (2).

With the acquisition function a_{EI} shown in Eq. (4), the next sampling point $\mathbf{x}$ is selected as the one that maximizes EI. The two terms in Eq. (4) can be viewed as a trade-off between exploiting the information from evaluating the points with low mean values and exploring the points with high uncertainty.

3. MULTI-AGENT BAYESIAN OPTIMIZATION

3.1 Problem setup

In this section, we show the problem formulation to find the minimum of a black-box function in a d-dimensional design space $A \subseteq \mathbb{R}^d$ with N agents in a team. The goal of agent i, where $i \in \{1, 2, ..., N\}$ is to find the location of global minimum $\mathbf{x}^*$

$$
\mathbf{x}^* = argmin_{\mathbf{x} \in A} f(\mathbf{x}), \tag{5}
$$

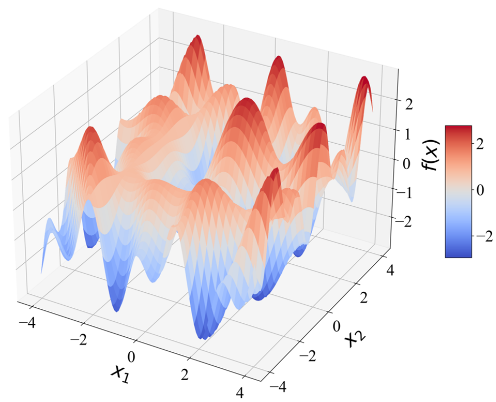
where $f(\cdot)$ is a black-box objective function and $\mathbf{x} = (x_1, x_2, ..., x_d) \in A$.

We assume that the design task is divided into N regions and that each agent is assigned to one unique region in the design space $A_i \subseteq A$ to divide the labor among the agents in a team. We assume that the division of labor is done at the beginning before the design search starts. In this formulation, none of the agents is allowed to search beyond their assigned region but only communicate by sharing the local points they sampled in the past. The search stops either after a predefined number of steps or when an agent arrives sufficiently close to the local minimum (e.g. when the smallest convergence rate among N agents is lower than a predefined threshold). Figure 1a shows an example to illustrate this idea. In this example, the objective function is an unknown function of an MAS consisting of three agents. The design space has been partitioned into three local regions depicted in Figure 1b. In each region, only one agent is assigned to search the space locally to find the global minimum (marked as a red star), whereas none of them knows where the global minimum is. We do not allow multiple agents in one region because that does not fundamentally change the problem to be solved and only influences the convergence speed [5] because that simply increases the number of sampling points in each search.

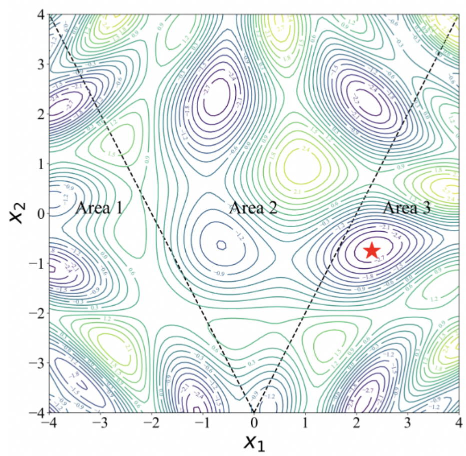
3.2 Solution process with MABO

To computationally study the solution to the problem presented in the preceding section using a team of agents, we develop a multi-agent Bayesian Optimization (MABO) process that allows all agents to work collaboratively to find the optimal solution in an unknown design space. The algorithm for our proposed MABO process is described in Algorithm 1. In this MAS, an important feature is that it has a global evaluator and the agents are allowed to contribute their locally sampled points to the global; meanwhile, the information processed at global will be shared back to each individual for their decision-making on future moves. We use this global-local interaction mechanism to study the value of interactions among team members (i.e., agents) in solving a complex problem (with multiple local optima) when there is a division of labor among team members. We perform this analysis in comparison to a case where the agents do not use this global evaluator, i.e., they do not communicate with each other.

Specifically in this MAS, when agents are allowed to communicate, a Gaussian process, as introduced in Section 2.1, uses Bayesian inference to estimate the probability distribution of potential values for $f(\mathbf{x})$ at a candidate point $\mathbf{x}$. The posterior distribution is iteratively updated by the global evaluator collecting observation data contributed by all agents. The acquisition function is located at the global level that evaluates the value of the EI at a new point $\mathbf{x}$ based on the current posterior distribution over f. After global evaluation, each agent can only access the EI information defined in its local region A_i and will choose the next sampling point that produces the local maximum of EI.



(a) Objective function



(b) Contour plot

FIGURE 1: AN EXAMPLE OF THE OBJECTIVE FUNCTION IN 2D DESIGN SPACE.

With the objective function shown in Figure 1a, one snapshot of an iteration in MABO using Algorithm 1 is illustrated in Figure 2. Each agent picks one new point in each iteration in its local region divided by the dashed line. The global evaluator collects all sampled points (the black points) from each agent to evaluate the posterior means and variance shown in the subfigures of the first and second columns. Based on this evaluation, the acquisition function value is updated, as shown in the subfigures in the third column. To move forward, the agents select the location of the maximum value of the acquisition function in their respective local region (the red points) as the next sampling point. In Figure 3, the sampled points of each agent are marked with different colors, where the numbers on the points indicate the search in each iteration step.

This approach (when agents share design samples with each other through a global evaluator) can effectively find the global optimum with high convergence speed (albeit without any theoretical guarantees) by involving multiple agents working on their local regions in each iteration. The global evaluator can acquire multiple observations within a single iteration, thereby enabling a comprehensive evaluation of the objective function across all local regions at each iteration.

Algorithm 1 MULTI-AGENT BAYESIAN OPTIMIZATION

Set local region A_i for each agent in an unknown design space A
Place a Gaussian process prior $\mathcal{D}_i = (\mathbf{x}_{1:k}, \mathbf{f}(\mathbf{x}_{1:k}))$ in A_i for each agent
Observe f at the initial step in design space A
for $Step = 1$ to $MAX - Step$ **do**
  Update the posterior probability distribution with all available data on each local region A_i as Eq. (2)
  Update the acquisition function Eq. (4)
  for agent 1 to N **do**
    Let $\mathbf{x}_i$ be a optimizer of acquisition function Eq. (4) in the local region A_i
    Observe $f(\mathbf{x}_i)$.
  end for
  Collect all data $(\mathbf{x}_i, f(\mathbf{x}_i))$ from agents
end for
Return a solution: the point $\mathbf{x}$ evaluated with the global optimum $f^*(\mathbf{x})$.

4. EXPERIMENT SETUP

To investigate the impact of information exchange and interactions between local agents (through the global evaluator), we created another MABO process that does not have a global evaluator and each agent solves a BO in its own region as a baseline model. Therefore, we compare the following two MABO processes in terms of their convergence speeds.

- *Method 1: the MABO process without a global evaluator*;
- *Method 2: the proposed MABO proposed with a global evaluator enabled.*

We test these two processes on two different benchmark functions with varying complexities, as shown in Table 1 and displayed in Figure 4: 1) the Cosines function and 2) the Eggholder function. These two functions are widely recognized within the global optimization literature, as they have been frequently utilized as benchmarks in the context of BO. This is evidenced by their previous use in research studies such as [20, 21]. Compared to the former function, the latter is more complex as it contains many more local minima and maxima in the search space.

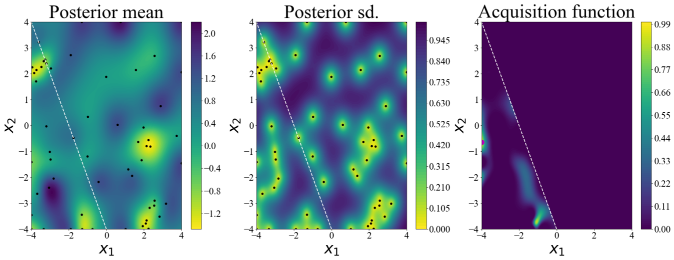
To study the scalability of the findings and evaluate the impact of the number of agents on search performance, we performed computational studies with three and five agents. As a result, we create three different experimental scenarios: 1) MABO of the Cosines function with an MAS of three agents, 2) MABO of the Eggholder function with an MAS of three agents, and 3) MABO of the Eggholder function with an MAS of five agents. To keep a fair comparison, the number of maximum iterations for sampling is set to 50 for all experiments, and the initial number of samples for the Gaussian process prior $\mathcal{D}_i = (\mathbf{x}_{1:k}, \mathbf{f}(\mathbf{x}_{1:k}))$ in A_i is set to $k = 5$ in each method and scenario in Algorithm 1.

5. RESULTS

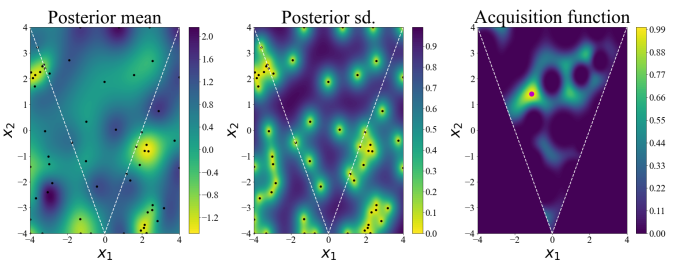
Scenario 1: MABO of the Cosines function with an MAS of three agents. Each agent is assigned to its local region defined

TABLE 1: TWO BLACK-BOX FUNCTIONS

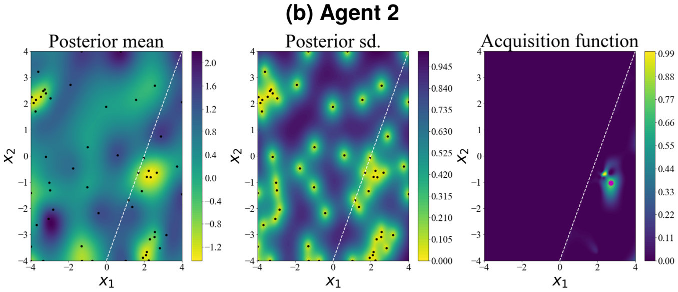
Name	Formula	Global minimum	Global domain A
Cosines	$f(\mathbf{x}) = 1 - (x_1^2 + x_2^2 - 0.3cos(3\pi x_1) - 0.3cos(3\pi x_2))$	$f(0.314, 0.303) = -1.596$	$\{\mathbf{x} \mid -1 \leq \mathbf{x} \leq 1\}$
Eggholder	$f(\mathbf{x}) = -(x_2 + 47)\sin\left(\sqrt{\left\|x_2 + \frac{x_1}{2} + 47\right\|}\right)$ $-x_1 \sin\left(\sqrt{\|x_1 - (x_2 + 47)\|}\right)$	$f(512, 404.232) = -959.641$	$\{\mathbf{x} \mid -520 \leq \mathbf{x} \leq 520\}$



(a) Agent 1



(b) Agent 2



(c) Agent 3

FIGURE 2: THE GP MODEL AND ACQUISITION FUNCTION WITH A GLOBAL EVALUATOR ARE ENABLED FOR AGENTS 1, 2, AND 3.

in Table 2. The sampling process and the search trajectory (indicated by the number index) of each agent in its own region are displayed in Figure 5. The best $f(\mathbf{x})$ (i.e., f^* in Eq. (4)) observed so far in each step shown in Figure 6 describes the convergence speed for an MAS. According to Figure 6a, Agent 2 in Method 1 (i.e., the MABO without a global evaluator) was able to successfully find the global minimum $f(0.314, 0.303) = -1.596$ in 31 steps. Agent 1 and Agent 3 reached their local minima in 20 and 22 steps, respectively. With Method 2 (i.e., the MABO with a global evaluator), Agent 2 reached the global minimum in only six steps, as shown in Figure 6b. Agent 1 and Agent 3 found their local minima in 9 and 18 steps, respectively.

Scenario 2: MABO of the Eggholder function with an MAS of three agents. The local region for each agent is defined in Table 3 and the global minimum of this Eggholder function is located in area 3, $f(512, 404.232) = -959.641$. Due to the increased complexity of the Eggholder function, finding the global minimum using MABO without global-local information exchange is a challenging task. As shown in Figure 8a, Agent 3 cannot reach the global minimum even after 50 steps. Actually, at Step 49, the best performance achieved is from Agent 2. In Method 2 with a global evaluator, MAS accomplished the search in 32 steps, as illustrated in Figure 8b.

Scenario 3: MABO of the Eggholder function with an MAS of five agents. In this scenario, the MABO methods were experimented with the same function, i.e., the Eggholder function, but the MAS in each method was expanded to five agents. The local region of each agent is defined in Table 4. Figure 9 demonstrates the sampling points and the corresponding trajectories for both methods. It is observed that many sampling points are clustered in

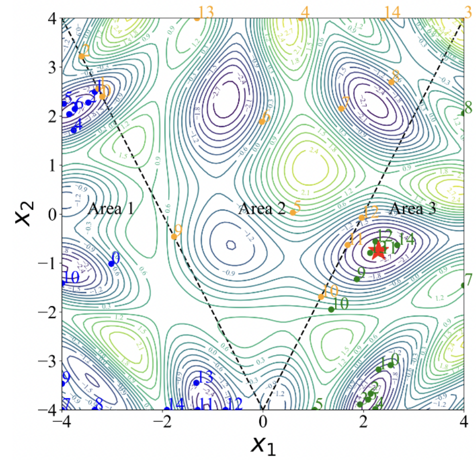


FIGURE 3: SAMPLING PROCESS FOR THREE AGENTS. THE RED STAR IS THE TRUE OPTIMUM. THE NUMBERS LABELED ON THE POINTS ARE THE SEARCH IN EACH ITERATION STEP.

TABLE 2: COSINES FUNCTION. LOCAL DESIGN SPACE DOMAIN FOR THREE AGENTS.

	Local region A_i
Agent 1	$\{\mathbf{x} \mid -2x_1 + x_2 + 2 \leq 0\}$
Agent 2	$\{\mathbf{x} \mid -2x_1 - x_2 - 2 \leq 0, 2x_1 - x_2 - 2 \leq 0\}$
Agent 3	$\{\mathbf{x} \mid -2x_1 + x_2 + 2 \leq 0\}$

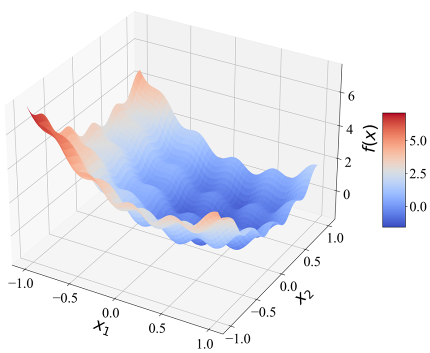
TABLE 3: EGGHOLDER FUNCTION. LOCAL DESIGN SPACE DOMAIN FOR THREE AGENTS.

	Local region A_i
Agent 1	$\{\mathbf{x} \mid -2x_1 + x_2 + 520 \leq 0\}$
Agent 2	$\{\mathbf{x} \mid -2x_1 - x_2 - 520 \leq 0, 2x_1 - x_2 - 520 \leq 0\}$
Agent 3	$\{\mathbf{x} \mid -2x_1 + x_2 + 520 \leq 0\}$

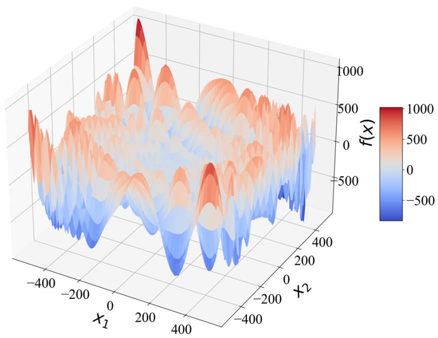
local minima. This indicates that agents in both methods tend to continue to exploit a location region if the previous sampling point in that region yields the best $f(\mathbf{x})$ observed so far. The agents start exploring other unknown spaces if no further improvement can be made since the last-found best $f(\mathbf{x})$ in their current local search regions. As shown in Figure 10a, Agent 5 in Method 1 reaches the global minimum $f(512, 404.232) = -959.641$ in 43 steps, while Agent 5 in Method 2 finds the minimum in just seven steps, as illustrated in Figure 10b.

6. DISCUSSION

The findings regarding the convergence speed of the methods in different scenarios are summarized in Table 5. The comparison between Scenario 1 and Scenario 2 indicates that, as the complexity of the function increases, more iterations are needed during the search for convergence. However, using the global



(a) Cosines function

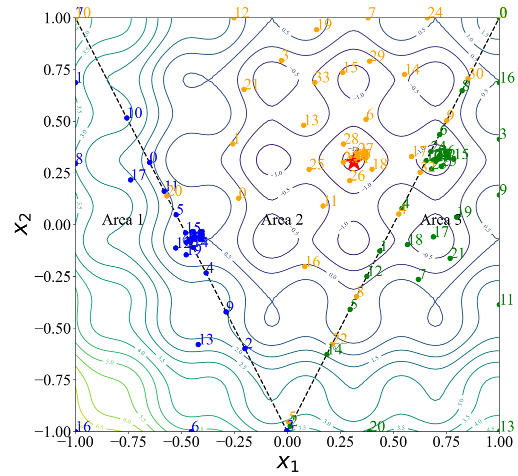


(b) Eggholder function

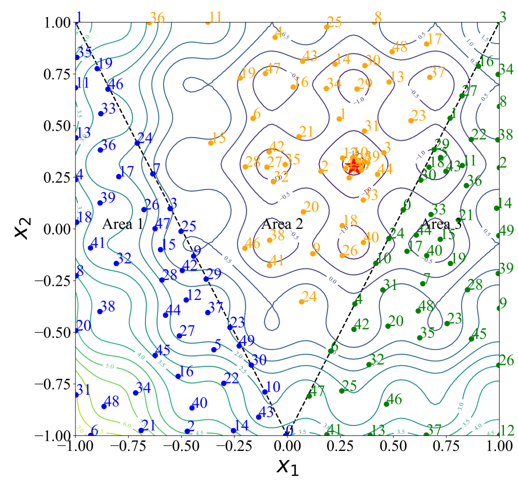
FIGURE 4: TWO BENCHMARK FUNCTIONS. THE EGGHOLDER FUNCTION IS MUCH MORE COMPLEX THAN THE COSINE FUNCTION, WITH MUCH MORE LOCAL OPTIMA THAN THE LATTER.

TABLE 4: EGGHOLDER FUNCTION. LOCAL DESIGN SPACE DOMAIN FOR FIVE AGENTS.

	Local region A_i
Agent 1	$\{\mathbf{x} \mid 3x_1 + x_2 + 1040 \leq 0\}$
Agent 2	$\{\mathbf{x} \mid -3x_1 - x_2 - 1040 \leq 0\}$
Agent 3	$\{\mathbf{x} \mid 6x_1 + x_2 - 520 \leq 0, -6x_1 + x_2 - 520 \leq 0\}$
Agent 4	$\{\mathbf{x} \mid 3x_1 - x_2 - 1040 \leq 0, -6x_1 - x_2 + 520 \leq 0\}$
Agent 5	$\{\mathbf{x} \mid -3x_1 + x_2 + 1040 \leq 0\}$



(a) Method 1



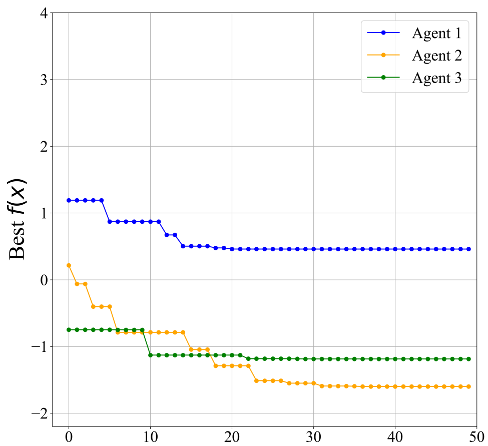
(b) Method 2

FIGURE 5: DESIGN SPACE EXPLORATION IN SCENARIO 1

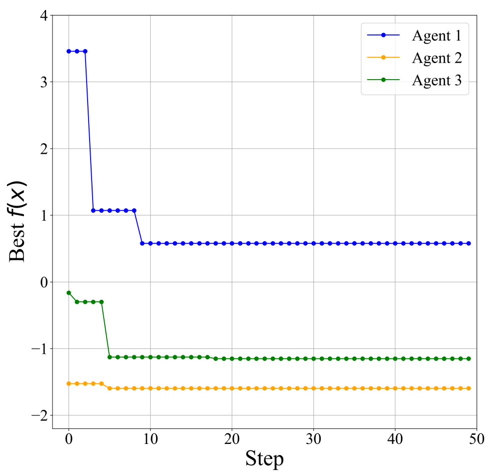
evaluator (and interactions among agents) is found to be more robust against the increased complexity because Method 2 successfully finds the global minimum within 50 steps, yet Method 1 failed in Scenario 2. Comparing the results of Scenario 2 and Scenario 3 indicates that both MABO methods become more efficient when more agents are available in a team. In particular, the convergence speed of our proposed MABO method (i.e., Method 2) with five agents is about 4.5 times faster than that with three agents when dealing with the same Eggholder function. In all scenarios, the results indicate that the MABO with global-local information exchange outperforms the one without such information exchange.

TABLE 5: CONVERGENCE SPEED FOR EACH SCENARIO

Obj. Func.	MAS with three agents		MAS with five agents	
	Method 1	Method 2	Method 1	Method 2
Cosines	31	6	N/A	N/A
Eggholder	>200	32	43	7

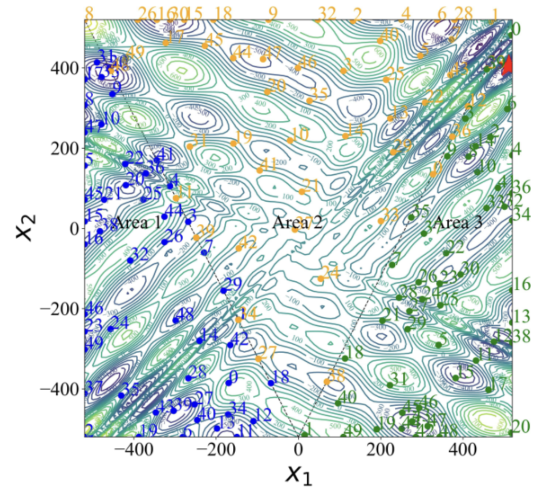


(a) Method 1

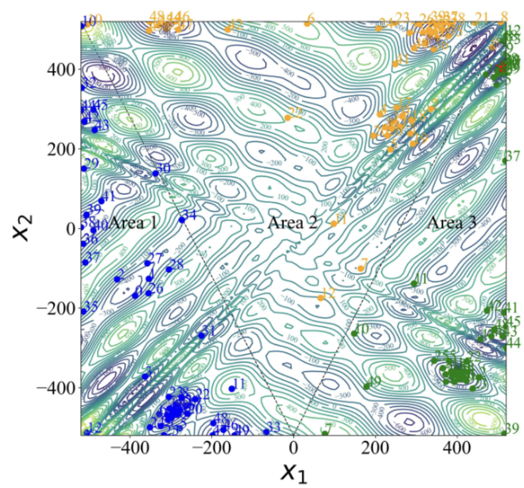


(b) Method 2

FIGURE 6: CONVERGENCE SPEED IN SCENARIO 1



(a) Method 1



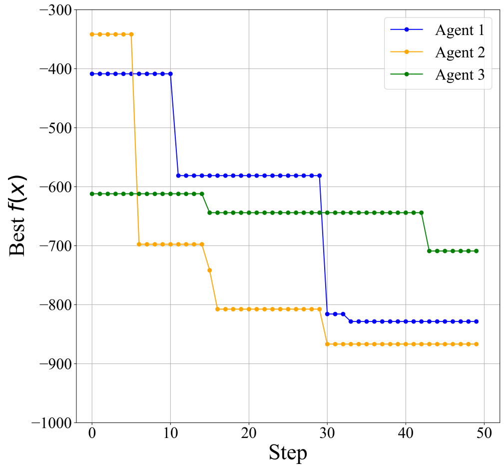
(b) Method 2

FIGURE 7: DESIGN SPACE EXPLORATION IN SCENARIO 2

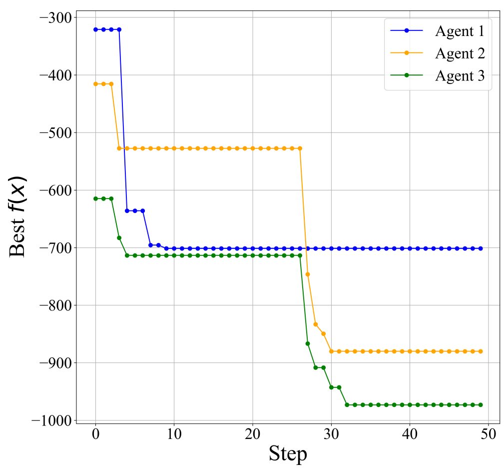
Although these results are intuitive, they lead to some useful take-aways for design teams in practice. First, they provide a different conclusion about interactions between team members from the computational findings in the literature. Recall that the study by McComb et al. suggests that team members should not interact with each other when they all work on the same problem without a division of labor [5] whereas the existence of a division of labor in a team benefits from interactions for effective solutions based on our findings. Note that this result does not contradict this literature since the study conditions in terms of task allocations are different. Combining our results with those from [5], we can see when team communication is beneficial and when it is not. In our study, even though the agents are responsible for their own regions, information from other regions in the design space leads these agents to find solutions in their own regions faster than in the case where there is no interaction among agents. The convergence results also support the argument that teams are as good as the most vital link in the team [22], i.e., the performance of a team is determined by the best agent in the team. Note that these results do not include any human factors that might add some adverse effects due to communication problems, trust and cognitive biases.

7. CONCLUSION AND FUTURE WORK

The study is motivated to answer the following question: What is the impact of information exchange among agents in a multi-agent system (MAS) on their performance in searching a complex unknown design space collaboratively for the global optimum? To this end, the paper presents a multi-agent Bayesian optimization (MABO) framework that addresses the challenges of finding the global optimum of complex objective functions with many local optima. The framework involves dividing the design space into local regions and allowing agents in local searches to share their sampled design points with a global evaluator. The global evaluator computes the posterior mean and variance and evaluates the acquisition function to recommend the next sampling decisions for local agents. To answer the motivation question, the proposed method was compared with a MABO method that does not allow information exchange. The results show that

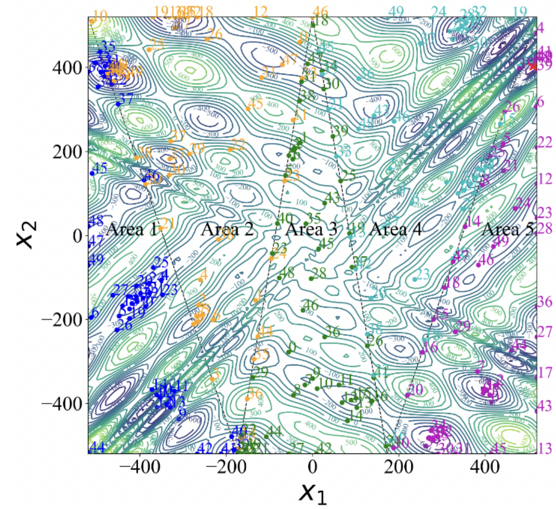


(a) Method 1

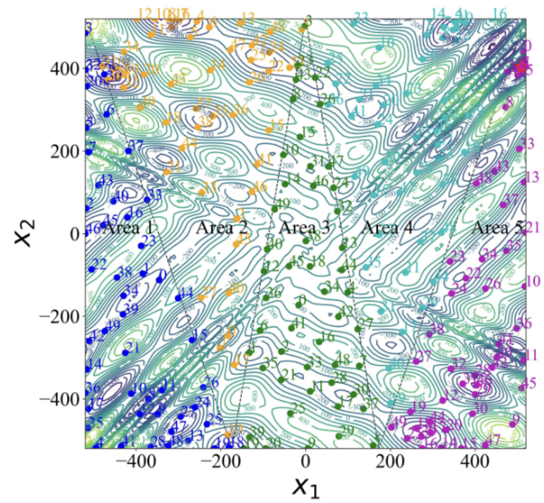


(b) Method 2

FIGURE 8: CONVERGENCE SPEED IN SCENARIO 2



(a) Method 1



(b) Method 2

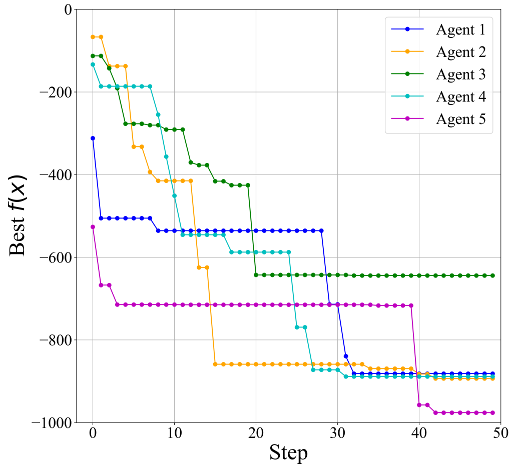
FIGURE 9: DESIGN SPACE EXPLORATION IN SCENARIO 3

allowing information exchange and global-local interaction significantly improves convergence speed (more than five times on average), regardless of function complexity and MAS team size. Therefore, we conclude that the proposed MABO framework is effective in exploring complex design spaces and finding the optimal solution.

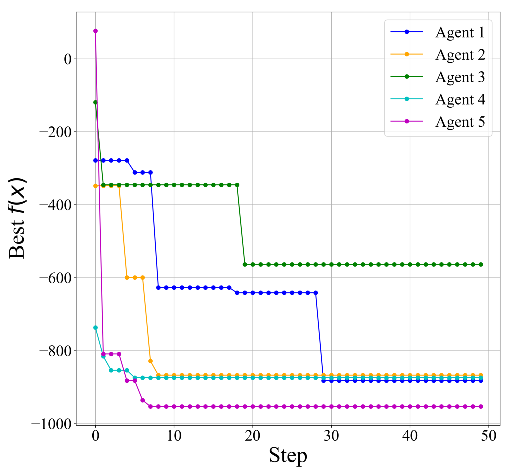
Although promising, there are a number of limitations in our current study, which lead to several future directions. First, we did not consider the cost in the current framework. However, in reality, the cost could be associated with search and information sharing. In our future study, in addition to evaluating search performance quantified by convergence speed, cost shall be the other dimension in performance evaluation. Second, an unknown space must involve unknown constants. In our current study, agents are assumed to have access anywhere in the designated region. In future work, this assumption can be relaxed by introducing "infeasible area" in the local search regions, so agents are not allowed to sample candidate points in the infeasible areas. Moreover, we can investigate how the location of the infeasible area would influence the agents' search performance. Third, in the current study, we conducted the experiment with the MABO framework using one particular acquisition function and a special kernel setting of the Gaussian process. In future studies, experimental results will be collected with more comprehensive hyperparameter settings, so a complete picture of the agents' search performance can be obtained. Fourth, this paper only presents the MABO solution to 2-dimensional design space exploration. In higher-dimensional cases, more complex division strategies could be required to partition the space. Lastly, since information exchange and global-local interaction are enabled in our framework, this opens many research questions for the community. For example, when the amount of information and frequency can be controlled, how could the variations influence the agents' search performance? In our future work, we are motivated to answer these questions based on the work presented.

ACKNOWLEDGMENTS

The authors greatly acknowledge the financial support provided by the Walker Department of Mechanical Engineering at



(a) Method 1



(b) Method 2

FIGURE 10: CONVERGENCE SPEED IN SCENARIO 3

the University of Texas at Austin.

REFERENCES

[1] Nardi, Luigi, Koeplinger, David and Olukotun, Kunle. "Practical design space exploration." *2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*: pp. 347–358. 2019. IEEE.

[2] Gyory, Joshua T, Soria Zurita, Nicolás F, Cagan, Jonathan and Mccomb, Christopher. "Independence or Interaction? Understanding the Benefits and Limitations of Nominally-Inspired and Interacting Sub-Structured Teams in a Virtual and Interdisciplinary Engineering Design Task." *Journal of Mechanical Design* : pp. 1–42.

[3] Gyory, Joshua T, Kotovsky, Kenneth and Cagan, Jonathan. "The influence of process management: uncovering the impact of real-time managerial interventions via a topic modeling approach." *Journal of Mechanical Design* Vol. 143 No. 11 (2021).

[4] McComb, Christopher, Cagan, Jonathan and Kotovsky, Kenneth. "Lifting the Veil: Drawing insights about design teams from a cognitively-inspired computational model." *Design Studies* Vol. 40 (2015): pp. 119–142.

[5] McComb, Christopher, Cagan, Jonathan and Kotovsky, Kenneth. "Optimizing design teams based on problem properties: computational team simulations and an applied empirical test." *Journal of Mechanical Design* Vol. 139 No. 4 (2017): p. 041101.

[6] Allison, James T., Kokkolaras, Michael and Papalambros, Panos Y. "Optimal Partitioning and Coordination Decisions in Decomposition-Based Design Optimization." *Journal of Mechanical Design* Vol. 131 No. 8 (2009). 081008.

[7] Snoek, Jasper, Larochelle, Hugo and Adams, Ryan P. "Practical bayesian optimization of machine learning algorithms." *Advances in neural information processing systems* Vol. 25 (2012).

[8] Greenhill, Stewart, Rana, Santu, Gupta, Sunil, Vellanki, Pratibha and Venkatesh, Svetha. "Bayesian optimization for adaptive experimental design: A review." *IEEE access* Vol. 8 (2020): pp. 13937–13948.

[9] Jones, Donald R, Schonlau, Matthias and Welch, William J. "Efficient global optimization of expensive black-box functions." *Journal of Global optimization* Vol. 13 No. 4 (1998): p. 455.

[10] Pelamatti, Julien, Brevault, Loïc, Balesdent, Mathieu, Talbi, El-Ghazali and Guerin, Yannick. "Bayesian optimization of variable-size design space problems." *Optimization and Engineering* Vol. 22 (2021): pp. 387–447.

[11] Reagen, Brandon, Hernández-Lobato, José Miguel, Adolf, Robert, Gelbart, Michael, Whatmough, Paul, Wei, Gu-Yeon and Brooks, David. "A case for efficient accelerator design space exploration via bayesian optimization." *2017 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*: pp. 1–6. 2017. IEEE.

[12] Xiong, Yi, Duong, Pham Luu Trung, Wang, Dong, Park, Sang-In, Ge, Qi, Raghavan, Nagarajan and Rosen, David W. "Data-driven design space exploration and exploitation for design for additive manufacturing." *Journal of Mechanical Design* Vol. 141 No. 10 (2019).

[13] Tosserams, S, Hofkamp, AT, Etman, LFP and Rooda, JE. "A specification language for problem partitioning in decomposition-based design optimization." *Structural and Multidisciplinary Optimization* Vol. 42 (2010): pp. 707–723.

[14] Bayrak, Alparslan Emrah, McComb, Christopher, Cagan, Jonathan and Kotovsky, Kenneth. "A strategic decision-making architecture toward hybrid teams for dynamic competitive problems." *Decision Support Systems* Vol. 144 (2021): p. 113490.

[15] Frazier, Peter I. "A tutorial on Bayesian optimization." *arXiv preprint arXiv:1807.02811* (2018).

[16] Rasmussen, Carl Edward. "Gaussian processes in machine learning." *Summer school on machine learning*: pp. 63–71. 2003. Springer.

[17] Kushner, H. J. "A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of

Noise." *Journal of Basic Engineering* Vol. 86 No. 1 (1964): pp. 97–106.

[18] Srinivas, Niranjan, Krause, Andreas, Kakade, Sham M and Seeger, Matthias. "Gaussian process optimization in the bandit setting: No regret and experimental design." *arXiv preprint arXiv:0912.3995* (2009).

[19] Russo, Daniel J, Van Roy, Benjamin, Kazerouni, Abbas, Osband, Ian, Wen, Zheng et al. "A tutorial on thompson sampling." *Foundations and Trends® in Machine Learning* Vol. 11 No. 1 (2018): pp. 1–96.

[20] González, Javier, Dai, Zhenwen, Hennig, Philipp and Lawrence, Neil. "Batch Bayesian optimization via local penalization." *Artificial intelligence and statistics*: pp. 648–657. 2016. PMLR.

[21] De Palma, Alessandro, Mendler-Dünner, Celestine, Parnell, Thomas, Anghel, Andreea and Pozidis, Haralampos. "Sampling acquisition functions for batch Bayesian optimization." *arXiv preprint arXiv:1903.09434* (2019).

[22] Brownell, Ethan, Cagan, Jonathan and Kotovsky, Kenneth. "Only as strong as the strongest link: The relative contribution of individual team member proficiency in configuration design." *Journal of Mechanical Design* Vol. 143 No. 8 (2021).