2022 IEEE International Conference on Robotics and Automation (ICRA)
May 23-27, 2022. Philadelphia, PA, USA

Stackelberg Strategic Guidance for Heterogeneous Robots Collaboration

Yuhan Zhao[1], Baichuan Huang[2], Jingjin Yu[2] and Quanyan Zhu[1]

***Abstract*— In this study, we explore the application of game theory, in particular Stackelberg games, to address the issue of effective coordination strategy generation for heterogeneous robots with one-way communication. To that end, focusing on the task of multi-object rearrangement, we develop a theoretical and algorithmic framework that provides strategic guidance for a pair of robot arms, a leader and a follower where the leader has a model of the follower's decision-making process, through the computation of a feedback Stackelberg equilibrium. With built-in tolerance of model uncertainty, the strategic guidance generated by our planning algorithm not only improves the overall efficiency in solving the rearrangement tasks, but is also robust to common pitfalls in collaboration, e.g., chattering.**

I. Introduction

With robotic technology research and development rapidly accelerating, one can expect an explosion in the number and type of robots to be deployed in the coming years. With this trend, there is an increasing need to have robots with different capabilities effectively collaborative to solve tasks, e.g., packing products at factories or in autonomous warehouses. For example, it can be that different batches of robots have different specifications and, as a result, have complementary capabilities, which can happen when a company purchases the batches years apart. In this case, having these robots work together can effectively extend the service life of older robots, thus delivering more value for the hardware investment. However, simply putting autonomous robots together is not sufficient; algorithms must be developed to ensure that collaboration drives more value than having the robots make individual decisions. In the same vein, with robots increasingly permeating our work and lives, it can be predicted that robots will be working and playing alongside humans. One would expect that the robot would observe and understand human behavior and assist accordingly with limited communication.

Motivated by the above-mentioned broadly applicable use cases, in this study, we explore the application of game theory, in particular Stackelberg games [1], for enabling heterogeneous autonomous robots to collaboratively solve manipulation tasks. Specifically, we develop a framework, Stackelberg Guided Collaborative Manipulation (SGCM), for coordinating two robot arms to jointly solve a multi-object rearrangement task. The two robots have different manipulation and computation capabilities, where one is a leader and the other is a follower. The leader is assumed to have knowledge of the follower's decision-making model, whereas the follower only makes decisions based on the leader's action. SGCM is shown to deliver more efficient solutions as compared with a greedy baseline and avoids potential pitfalls, e.g., chattering where the robots nullify each other's actions, with limited communication. In other words, SGCM provides a more resilient architecture.

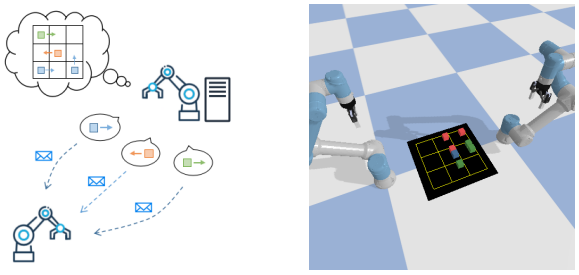


Fig. 1: Overview of Stackelberg Guided Collaborative Manipulation framework and multi-object rearrangement task settings. [left] Illustration of the setup, where one robot (upper right) has more computing power and physical capabilities would guide the other less capable robot (lower left). [right] The simulation environment in PyBullet [2].

This work's key contribution is a theoretical and algorithmic framework that applies Stackelberg games to robot collaboration. First, we propose a novel stochastic Stackelberg game framework (SGCM) to provide strategic guidance for heterogeneous robots, or agents in general, to collaboratively solve physical tasks with only leader-to-follower communication. Then, we develop a general algorithm, through dynamic programming and mixed integer programming, that computes the feedback Stackelberg equilibrium as the equilibrium policy for a leader-follower setting where the leader has a model of the follower's decision-making logic. Our algorithmic solution is instantiated and evaluated over a product-packing-like rearrangement task, which shows that the SGCM approach enables the involved robots to work together more efficiently and is robust to uncertainties.

II. Related Work

Stackelberg games are first proposed to study hierarchical competitions in the market where some companies possess dominant power [1]. In general, two players are involved in a Stackelberg game, a leader (she) and a follower (he). The leader first announces her strategy to maximize her utility by considering all possible reactions from the follower and sticks to that strategy. The follower best responds to the leader's strategy by maximizing his utility. Two strategies then constitute a *Stackelberg equilibrium*. The static Stackelberg game has been extended to dynamic contexts. One direct extension is Stochastic Stackelberg Game (SSG) [3],

[1]Y. Zhao and Q. Zhu are with the Department of Electrical and Computer Engineering, New York University. Email: {yhzhao, qz494}@nyu.edu.
[2]B. Huang and J. Yu are with the Department of Computer Science, Rutgers University. Email: {baichuan.huang, jingjin.yu}@rutgers.edu.
This work is supported by NSF awards IIS-1845888 and IIS-2132972; partially supported by grants ECCS-1847056 from NSF and grant W911NF-19-1-0041 from Army Research Office (ARO).

2022 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9681-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICRA46639.2022.9811678

[4]. SSG takes into consideration the feature of dynamic interactions. Therefore, the equilibrium is designed for the overall interaction process instead of a single stage. Many works have focused on SSG and its applications, for example, supply chain competitions [5], security and resource allocation [6], [7], and cooperative advertising [8], [9].

Stackelberg games are also popular in robotics research. Stanková et al. in [10] have proposed a Stackelberg game based approach for heterogeneous robotic swarm coverage problem, which outperforms the standard Lloyd's algorithm. Duan et al. have investigated the robot surveillance problem over graphs in [11]. The optimal interception strategy is studied by formulating the interaction between the intruder and the robot as a Stackelberg game. Hebbar and Langbort have discussed a collaborative human-robot search-and-rescue problem as a rescuer-rescuee Stackelberg game [12]. The optimal rescue plans are provided with the Stackelberg game framework. However, only a few works focus on the dynamic Stackelberg game in robotics. For example, Koh et al. in [13] have studied the bi-robot cooperative object transportation by formulating the problem as an SSG and applying Q-learning to solve the transportation strategy.

This work explores a new application of Stackelberg games to robotics, namely collaborative manipulation, a practical, high-utility task that finds a great many of real-world use cases. In robotics, much effort has been devoted to rendering multi-arm planning more efficient. In terms of general methods, Cohen et al. [14] adapts a heuristic search, e.g., A*, in an innovative manner for solving high dimensional planning problems, including dual-arm systems. Shome et al. [15] proposed the dRRT* algorithm that computes asymptotically optimal solutions that applies to multi-arm manipulation tasks. From an application perspective, the task of generating efficient plans for transporting objects using coordinated multi-arm rearrangement is tackled in [16]. Xian et al. [17] proposed new techniques for performing mode-switching in solving dual-arm closed-chain manipulation tasks. A two-arm rearrangement task, somewhat similar to the problem examined in this work, is systematically studied in [18]. To our knowledge, existing works on multi-object arrangement by multiple arms generally assume central planning, which limits their application; we do not in this study.

III. Problem Formulation

We consider two robotic agents A (the leader, she) and B (the follower, he) cooperatively rearranging different types of objects in a 2D workspace Ω. The workspace Ω is partitioned into sub-cells shown in Fig. 1. Due to the heterogeneity of different robotic agents, we assume that the two robotic agents differ in their manipulation and computation capabilities. Apart from having more feasible actions to manipulate the objects, the leader also possesses more computational resources to deal with complex scenario plannings. The follower, however, has fewer feasible actions to move the objects. His computational power is also limited: he can only sense and process the current situation instead of planning for the future, or only execute the pre-programmed policies. In addition to accomplishing the rearrangement task cooperatively, the leader can also plan ahead and guide the follower with her powerful computational resources, so that the follower achieves better utility. The full cooperation scheme ,including the guidance, can be formulated as a finite horizon Stochastic Stackelberg Game (SSG). The SGCM framework aims to provide the equilibrium policy of SSG, which is adopted for cooperation and strategic guidance in the rearrangement task.

We define $s \in \mathcal{S}$ as the environment state representing the positions of all objects in different cells of Ω. $\mathcal{S}$ is the set of all states. The action $a^i \in \mathcal{A}^i$ represents a specific action for robot $i = \{A, B\}$ from its action set. Each action corresponds to moving one object from one cell to another. In particular, $a^i = \varnothing$ represents no action. We denote T as the interaction horizon of SSG (also leader's prediction horizon) and use subscripts $t \geq 0$ to represent the stage. We assume both robots have a perfect observation of the current state. The game is played as follows.

- At stage $t < T-1$, both robots observe s_t. The leader first chooses her action $a_t^A \in \mathcal{A}^A$. With probability p_{fail}^A, the leader fails to execute a_t^A, resulting in an empty action. Then the follower reacts to a_t^A by taking the action $a_t^B \in \mathcal{A}^B$. With probability p_{fail}^B, the follower fails to execute a_t^B, leading to an empty action. Then both robots receive the utility $u^i(s_t, a_t^A, a_t^B)$ for $i = \{A, B\}$. The environment transits to a new state s_{t+1} with the the transition probability $p(s_{t+1}|s_t, a_t^A, a_t^B)$.
- ·····
- At stage $T-1$, both robots observes s_{T-1}. The leader and the follower take actions a_{T-1}^A and a_{T-1}^B sequentially. Failure probabilities for action execution are defined similarly. The environment transits to the new state s_T based on the transition probability $p(s_T|s_{T-1}, a_{T-1}^A, a_{T-1}^B)$. Apart from the utility $u_{T-1}^i(s_{T-1}, a_{T-1}^A, a_{T-1}^B)$, an additional terminal utility $u_T^i(s_T)$ is also incurred for robot $i = \{A, B\}$.

Remark. The interaction horizon T shows the leader's planning consideration in the cooperative rearrangement task. The leader can predict the next T stages, but the rearrangement task does not necessarily terminate after T stages. When $T = 1$, both robots only care about the current state, and SGCM reduces to a repeated static Stackelberg game. When $T > 1$, the leader will consider the impact from the future and compute strategies to maximize the overall utility over T stages.

Remark. In the rearrangement task, since we assume p_{fail}^i for execution, the transition probability $p(s_{t+1}|s, a_t^A, a_t^B)$ is not necessarily binary given the action pair (a_t^A, a_t^B). There are four possibilities for the future state s_{t+1}, which corresponds to the action pair $(a_t^A, a_t^B), (a_t^A, \varnothing), (\varnothing, a_t^B)$, and $(\varnothing, \varnothing)$ in the deterministic scenario. We assume p_{fail}^i are independent for $i = \{A, B\}$, so that the transition probabilities can be computed via p_{fail}^i.

A policy is a state-dependent probability distribution over the action set for each robot. Given state s, we write

$\pi_t^A(a^A|s)$ and $\pi_t^B(a^B|s,\pi_t^A)$ to represent the probability[1] of choosing each action for robots A, B. For simplicity, we denote π_t^i as the vector form of the robot i's policy at time t, and write $\pi^i := \{\pi_t^i\}_{t=0}^{T-1}$ for $i = \{A, B\}$. The accumulated utility for robot $i = \{A, B\}$ is given by

$$J^i(\pi^A, \pi^B) = \mathbb{E}_{\pi^A,\pi^B}[\gamma^T u_T^i|s_0] + \sum_{t=0}^{T-1} \mathbb{E}_{\pi_t^A,\pi_t^B}\left[\gamma^t u_t^i|s_0\right],$$

where γ is the discount factor. By taking advantage of the stage-wise additive utility and the perfect state observation, we compute the *feedback Stackelberg equilibrium* (FSE) as the solution for cooperation and guidance in the rearrangement task. In the FSE $\{\pi_t^{A*}, \pi_t^{B*}\}_{t=0}^{T-1}$, at any stage of the game, both robots maximize their current and future accumulated utilities starting from the current stage.

A. Guidance in Rearrangement Tasks

The guidance aims to improve the follower's utility by taking advantage of the leader's powerful computation capabilities. It is embodied in two aspects: the leader plans the future for the follower and recommends the equilibrium policy to the follower. The follower can only maximize the current stage utility, but the game does not necessarily terminate in one step. Solely focusing on one-stage utility may not be optimal for the follower in the long run. However, the leader can consider the future impact and generate better strategies that are beneficial for the future. The leader can also recommend a better strategy to the follower, which cannot be computed with the follower's limited computation capability.

In SSG, one way to achieve successful guidance is to set the leader's utility the same as the follower's. In this way, as the leader maximizes her utility, she also helps maximize the follower's utility. So the equilibrium strategy is beneficial for both leader and follower. Another approach is to set the follower's utility as the potential function and construct the leader's utility-based the potential function, for example, affine transformation. In this way, the equilibrium strategy computed by the leader is also most beneficial for the follower.

IV. Stackelberg Guided Collaborative Manipulation Framework

In this section, we use dynamic programming to solve the FSE policies for both robots. In the equilibrium computation, we reformulate the leader's problem into a Mixed Integer Linear Programming (MILP) to reduce the problem complexity. Next, we propose the Stackelberg Guided Collaborative Manipulation (SGCM) framework to compute cooperation and guidance strategies for both robots with a rolling horizon approach and summarize the algorithm.

A. Dynamic Programming for Computing FSE

The FSE generates an optimal cooperation strategy that utilizes the leader's computational advantage. The leader can plan for T stages to envision the optimal action to reorganize the objects and the optimal guidance strategy to benefit the follower. The FSE can be solved by dynamic programming retrospectively.

We define $v_t^i(s)$ as the value function for the robot i at stage t, $i \in \{A, B\}$. The terminal value $v_T^i(s)$ is given by the terminal utility $u_T^i(s_T)$. At each stage of the game, two robots play a Stackelberg game with perfect state observation. At stage $t \le T-1$, after observing the state s_t, the t-th component of the equilibrium is computed by

$$\begin{aligned}
\max_{\pi_t^A} \quad & \sum_{a^A,a^B} \pi_t^{B^*}(a^B|s,\pi_t^A)\pi_t^A(a^A|s) \\
& \qquad \left[u_t^A(s,a^A,a^B) + \sum_{s'} p(s'|s,a^A,a^B)v_{t+1}^A(s')\right] \\
\text{s.t.} \quad & 0 \le \pi_t^A(a^A|s) \le 1, \quad \forall a^A \in \mathcal{A}^A, \\
& \sum_{a^A} \pi_t^A(a^A|s) = 1, \\
& \pi_t^{B*} \in \arg\max_{\pi_t^B} \ \mathcal{Q}_t^B(\pi_t^A).
\end{aligned} \tag{$\mathcal{Q}_t^A$}$$

where

$$\begin{aligned}
\max_{\pi_t^B} \quad & \sum_{a^A,a^B} \pi_t^B(a^B|s,\pi^A)\pi_t^A(a^A|s) \\
& \qquad \left[u_t^B(s,a^A,a^B) + \sum_{s'} p(s'|s,a^A,a^B)v_{t+1}^B(s')\right] \\
\text{s.t.} \quad & 0 \le \pi_t^B(a^B|s,\pi^A) \le 1, \quad \forall a^B \in \mathcal{A}^B, \\
& \sum_{a^B} \pi_t^B(a^B|s,\pi^A) = 1.
\end{aligned} \tag{$\mathcal{Q}_t^B(\pi_t^A)$}$$

For simplicity, we write $\sum_{a^A \in \mathcal{A}^A}$ as $\sum_{a^A}$. The same applies to $\sum_{a^B}$. The summation over s' contains four possibilities as discussed in Sec. III. We assume the optimal solution set of $(\mathcal{Q}_t^B(\pi_t^A))$ is a singleton for all $t = 0, \ldots, T-1$. Then $\pi_t^{B*}(a^B|s,\pi_t^A)$ is unique given the leader's policy π_t^A. After solving for π_t^{A*} and π_t^{B*}, we update the value function $v_t^i(s)$, $i = \{A, B\}$, by setting them as the optimal objective values of $(\mathcal{Q}_t^A)$ and $(\mathcal{Q}_t^B(\pi_t^A))$. The FSE can be obtained by solving $(\mathcal{Q}_t^A)$ and $(\mathcal{Q}_t^B(\pi_t^A))$ backward from stage $T-1$ to stage 0.

B. MILP Reformulation

To find the t-th component of the FSE, we need to solve a bilevel optimization problem $(\mathcal{Q}_t^A)$, which is in general hard. However, we notice that the problem $(\mathcal{Q}_t^B(\pi_t^A))$ is an LP and is linear in π_t^B. By utilizing this structure, we can use KKT conditions to equivalently represent $(\mathcal{Q}_t^B(\pi_t^A))$. Note that the constraint $\pi_t^B(a^B|s,\pi^A) \le 1$ is in fact redundant because it is guaranteed by $\pi_t^B(a^B|s,\pi^A) \ge 0$ and $\sum_{a^B} \pi_t^B(a^B|s,\pi^A) = 1$. Therefore we omit this inequality and simplify $(\mathcal{Q}_t^B(\pi_t^A))$ as

$$\begin{aligned}
\max_{x} \quad & (\pi_t^A)^{\mathsf{T}} U_t^B \pi_t^B \\
\text{s.t.} \quad & \pi_t^B \ge 0, \quad \sum_{a^B} \pi_t^B = 1,
\end{aligned} \tag{$\widetilde{\mathcal{Q}}_t^B(\pi_t^A)$}$$

where the utility matrix $U_t^B \in \mathbb{R}^{|\mathcal{A}^A| \times |\mathcal{A}^B|}$ and $U_{t,ij}^B = u_t^B(s,a^A,a^B) + \sum_{s'} p(s'|s,a^A,a^B)v_{t+1}^B(s')$.

[1]The follower's policy is paratermized by the leader's policy in the Stackelberg equilibrium. So we include π_t^A into π_t^B.

Let $\lambda \in \mathbb{R}$ and $\mu \in \mathbb{R}^{|\mathcal{A}^B|}$ be the dual variables associated with the equality and the inequality constraints in $(\widetilde{\mathcal{Q}}_t^B(\pi_t^A))$, respectively. We obtain the KKT conditions

$$\begin{aligned} &\lambda \mathbf{1}_{|\mathcal{A}^B|} - (U_t^B)^\mathsf{T}\pi_t^A \geq 0, && \pi_t^B \geq 0, \\ &(\pi_t^B)^\mathsf{T}(\lambda \mathbf{1}_{|\mathcal{A}^B|} - (U_t^B)^\mathsf{T}\pi_t^A) = 0, && \sum_{a^B} \pi_t^B = 1, \end{aligned} \tag{1}$$

where $\mathbf{1}_{\{\cdot\}}$ is the all-ones vector with proper dimensions.

For a Stackelberg game, the pure strategy for the follower always exists, which means that π_t^B can only have one non-zero element. If π_t^{B*} of $(\widetilde{\mathcal{Q}}_t^B(\pi_t^A))$ is not a singleton, we can show that any pure strategy in the support of π_t^{B*} is also optimal [19]. Therefore, we only focus on the pure strategy for the follower, which can be represented by binary variables. Furthermore, we can use the binary variable to linearize the complementarity condition in (1) and obtain

$$0 \leq \lambda \mathbf{1} - (U_t^B)^\mathsf{T}\pi_t^A \leq M(1-\pi_t^B), \quad \pi_t^B = \{0,1\}^{|\mathcal{A}^B|},$$

where M is a large number. We substitute the inner optimization problem in $(\mathcal{Q}_t^A)$ with KKT conditions and obtain

$$\begin{aligned} \max_{\pi_t^A, \pi_t^B, \lambda_t} \quad & (\pi_t^A)^\mathsf{T} U_t^A \pi_t^B \\ \text{s.t.} \quad & 0 \leq \pi_t^A \leq 1, \qquad \sum_{a^A} \pi_t^A = 1, \\ & \pi_t^B \in \{0,1\}^{|\mathcal{A}^B|}, \quad \sum_{a^B} \pi_t^B = 1, \\ & 0 \leq \lambda_t \mathbf{1}_{|\mathcal{A}^B|} - (U_t^B)^\mathsf{T}\pi_t^A \leq M(1-\pi_t^B), \end{aligned} \tag{$\widetilde{\mathcal{Q}}_t^A$}$$

where the utility matrix $U_t^A \in \mathbb{R}^{|\mathcal{A}^A| \times |\mathcal{A}^B|}$ and $U_{t,ij}^A = u_t^A(s, a^A, a^B) + \sum_{s'} p(s'|s, a^A, a^B) v_{t+1}^A(s')$.

We note that $(\mathcal{Q}_t^A)$ is a mixed integer quadratic programming (MIQP). To facilitate the computation, we follow [19] to further cast $(\widetilde{\mathcal{Q}}_t^A)$ to an MILP by changing of variables $z_t(a^A, a^B) = \pi_t^A(a^A|s)\pi_t^B(a^B|s, \pi_t^A)$. Then $z_t \in \mathbb{R}^{|\mathcal{A}^A| \times |\mathcal{A}^B|}$ and $(\widetilde{\mathcal{Q}}_t^A)$ becomes

$$\begin{aligned} \max_{z_t, \pi_t^B, \lambda_t} \quad & U_t^A \odot z_t \\ \text{s.t. } & \pi_t^B \in \{0,1\}^{|\mathcal{A}^B|}, \quad \sum_{a^B} \pi_t^B = 1, \\ & 0 \leq z_t \leq 1, \quad \sum_{a^A, a^B} z_t = 1, \quad z_t \mathbf{1}_{|\mathcal{A}^B|} \leq \mathbf{1}_{|\mathcal{A}^A|}, \\ & \pi_t^B \leq z_t^\mathsf{T} \mathbf{1}_{|\mathcal{A}^A|} \leq \mathbf{1}_{|\mathcal{A}^B|}, \\ & 0 \leq \lambda_t \mathbf{1}_{|\mathcal{A}^B|} - (U_t^B)^\mathsf{T}\left(z_t \mathbf{1}_{|\mathcal{A}^B|}\right) \leq M(1-\pi_t^B), \end{aligned} \tag{$\bar{\mathcal{Q}}_t^A$}$$

where $\odot$ represents element-wise multiplication. Since z_t is a matrix variable, we use $\sum_{a^A, a^B} z_t$ to denote the sum of all the element in z_t.

Remark. The finite horizon game allows us to define $T+1$ subsets of $\mathcal{S}$ given the initial state s_0: $\{\mathcal{S}_0, \mathcal{S}_1, \ldots, \mathcal{S}_T\} := \{\mathcal{S}_t\}_{t=0}^T$ and $\mathcal{S}_0 = \{s_0\}$. Each $\mathcal{S}_t$ contains all possible states occurred at time t. The value function $v_t^i(s)$ $(i = \{A, B\})$ is only defined for the state $s \in \mathcal{S}^t$ instead of all the states in $\mathcal{S}$. In this way, we do not need to compute $v_t^i(s)$ for state $s \notin \mathcal{S}_t$, $i = \{A, B\}$, which saves computation time. It is clear that $\mathcal{S}_t \subset \mathcal{S}$ for $t = 0, \ldots, T$, but we do not necessarily have $\mathcal{S}_m \cap \mathcal{S}_n = \varnothing$ for $m \neq n$. This means that the interactions between two robots may lead to some old states.

C. Rolling Horizon Computation for Rearrangement Task

We mention that the cooperative rearrangement task does not necessarily terminate after T stages. Therefore, we propose the SGCM framework which adopts the rolling horizon approach to compute FSE and execute the first-stage FSE policy. Similar to Model Predictive Control (MPC), the rolling horizon approach improves the robustness and resiliency of the cooperation. The follower may not precisely execute the leader's recommended strategy if he is susceptible to uncertainties. The leader can readjust her action and guidance policy accordingly to minimize the impact of uncertainties. The task eventually terminates when all objects are reorganized to the goal position, which corresponds to s_{goal}. We summarize the SGCM framework for cooperative rearrangement with strategic guidance in Alg. 1.

Algorithm 1: SGCM framework

Initialize: s_0, s_{goal}, T ;
for *iter* = $1, 2, \ldots$ **do**
  `// Forward prediction`
  $t \leftarrow 0$, $\mathcal{S}_0 = \{s_0\}$;
  while $t < T$ **do**
    for $s \in \mathcal{S}_t$ **do**
      for *action pair* $(a^A, a^B) \in \mathcal{A}^A \times \mathcal{A}^B$ **do**
        predict s_{new} with $s, (a^A, a^B)$;
        add s_{new} to $\mathcal{S}_{t+1}$;
        compute and store $u^i(s, a^A, a^B)$, $i = \{A, B\}$;
    $t \leftarrow t + 1$;
  `// Dynamic programming`
  $v_T^i(s) \leftarrow u_T^i(s)\ \forall s \in \mathcal{S}_T$, $i = \{A, B\}$;
  $t \leftarrow T - 1$;
  while $t \geq 0$ **do**
    for $s \in \mathcal{S}_t$ **do**
      formulate utility matrices U_t^A, U_t^B ;
      $(\pi^{A*}, \pi^{B*}) \leftarrow$ solve MLIP $(\bar{\mathcal{Q}}_t^A)$;
      $v_t^i(s) \leftarrow (\pi^{A*})^\mathsf{T} U_t^i \pi^{B*}$, $i = \{A, B\}$;
    $t \leftarrow t - 1$;
  `// Game execution and observation`
  leader executes π_0^{A*} and recommend π_0^{B*} ;
  observe new state s_{obs};
  if $s_{\text{obs}} = s_{\text{goal}}$ **then**
    break;
  $s_0 \leftarrow s_{\text{obs}}$;

Remark. In Alg. 1, for problems with small size, we can always enumerate all possible states in the forward prediction and perform value iterations to compute the FSE policy as shown in the algorithm. We mention that other simulation-based approaches can also be easily incorporated in Alg. 1 if the problem size is large. For example, we can adopt Monte Carlo Tree Search (MCTS) to explore part of $\mathcal{S}_t$ at different stage t if the size of $\mathcal{S}_t$ is huge. Then we only perform value iterations on the simulated states and compute approximate FSE policy. The simulation-based approaches are not guaranteed to provide the global optimal equilibrium, although they may be faster for online computation. However, for a specific task such as the rearrangement task, we can always pre-process states. Then we only need to perform a state search in the forward prediction for online computation.

V. Experiments and Evaluations

In this section, we evaluate our SGCM framework and demonstrate the strategic guidance with a multi-object rearrangement task [20], where two heterogeneous robotic arms (also called robots) cooperatively rearrange the objects to the goal position. The basic settings are shown in Fig. 1. Two robots are distinguished by their manipulation and computation capabilities. The leader can move the object along horizontal, vertical, and diagonal directions; she is also capable of making complex planning and predictions to accomplish the task by sensing the environment. The follower, however, can only manipulate the objects along the horizontal and vertical directions. His limited computation capability only allows him to consider the current situation rather than the future. We represent objects of different types with different colors for visualization purposes. The goal is to rearrange the red, green, and blue objects to bottom-left, bottom-middle, and bottom-right cells, respectively. Every action of the two robots is assigned a cost, and the cost of manipulating the object from a specific cell may double, depending on how many objects are in that cell. The state is defined in Sec.III. At the beginning of each round of interaction, a reward is assigned to the current state, which is proportional to the distance of the current state to the goal state. Each robot's utility is the reward minus the cost.

The experiment is carried out by simulation (PyBullet [2]), and the perception to the environment is vision-based. We assume that the ground truth segmentation of objects is accessible in the simulation, which is reasonable due to the practicality in real-world object detection methods such as Mask R-CNN [21]. The overall system works as follows. The leader perceives the environment state (positions of all objects) to compute high-level object manipulation command: what to grasp and where to place. After receiving the command, a low-level controller executes the command. The low-level controller adopts a Grasp Network [22], [23] to propose the grasp position of the selected object. Then the PyBullet's internal inverse kinematics module is used for manipulation motion planning. The position for placement is obtained by pixel test, where all pixels in the destination cells are iteratively tested until a pixel is found such that the object can be placed at this pixel as the center and has no collisions with other objects.

A. SGCM Framework and Greedy Approach

We set the interaction horizon $T = 2$ and assume the failure probabilities $p^A_{\text{fail}} = p^B_{\text{fail}} = 0.1$. Then two robots follow the SGCM framework to cooperatively work on the rearrangement task. For comparison, we also implemented a greedy rearrangement strategy. In the greedy approach, there is in fact no cooperation between two robots. Each robot observes the current environment and moves sequentially, only maximizing his/her current-stage utility. The reason for using the one-stage utility is because of no cooperation. Each robot has no need to consider the impact of the other robot. We select 10 different cases for testing. Each case corresponds to a different initial object setting. In these cases, the number of objects in the same type may differ, but the goal positions do not change. The initial configurations of 10 cases are shown in Fig. 2.

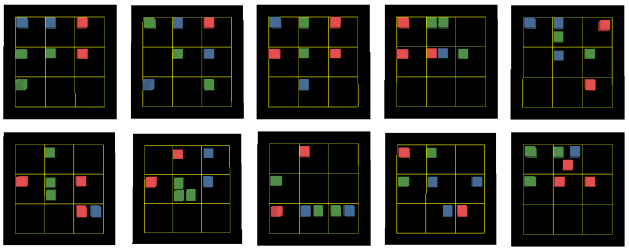


Fig. 2: The initial configurations of 10 cases in the experiment. The 10 cases are numbered from left to right.

We summarize the experiment results of using the SGCM framework and greedy approaches in Tab. I.

	Greedy			SGCM		
case #	status	rounds	utility	status	rounds	utility
1	No	> 7	283.5	Yes	6	309
2	No	> 7	288.5	Yes	6	313.5
3	No	> 7	300.5	Yes	6	315
4	Yes	5	278	Yes	5	283
5	No	> 8	306.5	Yes	7	344.5
6	Yes	6	289.5	Yes	5	306.5
7	Yes	5	297.5	Yes	5	298.5
8	Yes	3	219.5	Yes	3	219.5
9	Yes	7	249	Yes	5	275
10	Yes	7	277.5	Yes	6	301

TABLE I: Comparison between SGCM framework and greedy approach for 10 cases. Status refers to the completion of the task. The utility is the sum of single-stage utility over interaction rounds.

Note that robots do not care about cooperation in the greedy approach. To measure the performance of the greedy approach and to compare with the SGCM framework, we define the stage-wise utility in the greedy approach as the current state reward minus the total cost of two robots after each round of manipulation. The utility in Tab. I is the sum of all stage-wise utility and the values are comparable. Also, note that the SGCM framework and the greedy approach may have different interaction rounds. For comparison, the utility in Tab. I is computed based on the rounds from the SGCM framework because the SGCM framework always yields a smaller one. It means that we only sum up the single-stage utility in the greedy approach before the specified rounds, regardless of the completion status.

An example of the evolution of the interactions in the rearrangement task is visualized in Fig. 3.

$t = 0$ $t = 1$ $t = 2$ $t = 3$ $t = 4$ $t = 5$ $t = 6$

(a) Rearrangement status at each stage with SGCM framework.

$t = 0$ $t = 1$ $t = 2$ $t = 3$ $t = 4$ $t = 5$ $t = 6$

(b) Rearrangement status at each stage with greedy approach.

Fig. 3: Interaction evolution for Case 1 using SGCM framework and greedy approach. SGCM framework completes the task after 6 steps, while the greedy approach fails to rearranges objects and gets stuck in the last state.

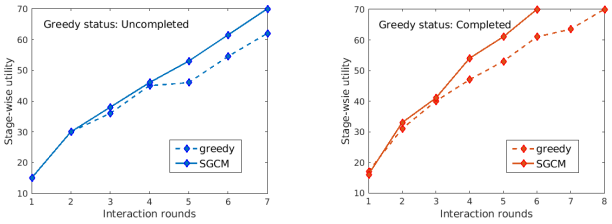
From Tab. I, we can observe several advantages of our

SGCM framework over the greedy approach in the cooperative rearrangement task:

- Two robots using the greedy approach can get stuck in some states due to myopic strategies. In these states, two robots repeat single actions, and the objects will never be reorganized to the target position. The SGCM framework can avoid such situations by taking advantage of the leader's computation capabilities. It also shows the significance of the planning and strategic guidance to the integrity of the cooperative rearrangement task. See Case 1, 2, 3, 5.
- When two robots are able to finish the rearrangement task with the greedy approach, the SGCM framework can either reduces the number of interactions and saves more actions (Case 6,9,10), or achieves higher utility when the number of interactions are the same (Case 4,7), showing the outperformance of the SGCM framework.

Remark. For some simple cases where the objects are easy to rearrange, for example, Case 8, the SGCM framework has the same performance as the greedy approach. However, this does not harm the effectiveness of the SGCM framework. In practice, we do not always have simple cases to rearrange. Then the advantage of the SGCM framework starts to appear, as demonstrated in other cases.

In order to have a clear view of how the SGCM framework outperforms the greedy approach, we plot the stage-wise utility along with the interactions in Fig. 4. We observe that at the beginning of the interaction, two approaches have the same stage-wise utility. But as the interaction evolves, the SGCM framework starts to yield a higher stage-wise utility than the greedy approach, showing the power of the strategic guidance in the rearrangement task.

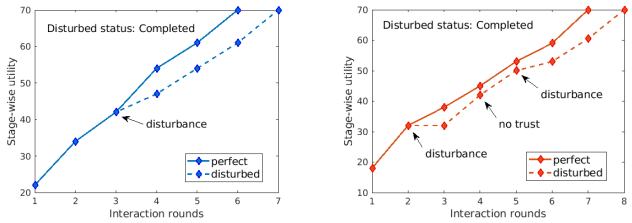


(a) For test case 2. **(b)** For test case 9.

Fig. 4: Stage-wise utility for two different cases, which demonstrates that the SGCM framework outperforms the greedy approach.

B. SGCM with Disturbance and Zero Trust

We demonstrate that our SGCM framework is robust and resilient to uncertainties and disturbances. In practice, the follower may not precisely execute the leader's recommended strategy at every stage due to the following reasons. First, the external disturbance may lead to hardware failure; second, the follower may not trust the leader's recommendation; third, the robot may be infected or hijacked by malware due to cybersecurity issues. In this situation, the follower may seek the strategy by himself, or randomly select an action, or even becomes adversarial to the leader. Therefore, resiliency is indispensable for cooperation. We illustrate the resiliency of our SGCM framework by injecting disturbances during the interaction, i.e., the follower does not follow the recommended strategy at certain steps. Our framework allows the leader to sense the abnormality and to adjust her strategy as well as the new recommended strategy in time, so that the impact of the disturbance is minimized.



(a) For test case 4. **(b)** For test case 3.

Fig. 5: Two plots shows how the disturbance and zero trust affects the stage-wise utility for two different cases. In both cases two robots are able to complete the rearrangement, which demonstrates the resiliency and robustness of the Stackelberg game approach.

In Fig. 5, the follower randomly selects a feasible action instead of the leader's recommended strategy to manipulate the objects when a "disturbance" occurs. The "no trust" in Fig. 5b means that the follower does not trust the leader's recommendation and selects the greedy strategy for manipulation. It is not surprising to see the performance degeneration after the disturbance or the zero trust. However, the deviation between the perfect and disturbed cases is controlled by SGCM and does not explode. Although suffering the disturbance and trust issues, we see that the SGCM framework can still ensure two robots accomplish the rearrangement task successfully. It shows that our SGCM framework is resilient and robust to random failure and zero trust during the two-robot cooperation in the rearrangement task.

VI. Conclusion

In this paper, we have proposed a Stackelberg Guided Collaborative Manipulation (SGCM) framework for heterogeneous robots collaboration. Focusing on the multi-object rearrangement task, the SGCM framework enables the leader robot to strategically guide the follower robot with her more powerful manipulation and companion capabilities to achieve better performance. The feedback Stackelberg equilibrium is adopted as the guidance strategy in the SGCM framework, which can be computed effectively by the developed algorithm. The SGCM framework only requires one-way communication to work on the rearrangement task cooperatively. In addition, our SGCM framework is also robust and resilient to uncertainties, disturbances, and trust issues during cooperation. Besides the theoretical guarantees, the effectiveness of the SGCM framework is thoroughly evaluated and validated over many different test cases of rearrangement tasks, where our approach displayed several advantages over greedy approaches. For the future, we intend to extend our framework to a multi-follower setting, where the benefit of the Stackelberg approach is expected to become more prominent. More learning aspects such as learning the follower's behavior pattern will also be considered.

REFERENCES

[1] H. Von Stackelberg, *Market structure and equilibrium*. Springer Science & Business Media, 2010.

[2] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2021.

[3] T. Başar and G. J. Olsder, *Dynamic noncooperative game theory*. SIAM, 1998.

[4] A. Bensoussan, S. Chen, and S. P. Sethi, "The maximum principle for global solutions of stochastic stackelberg differential games," *SIAM Journal on Control and Optimization*, vol. 53, no. 4, pp. 1956–1981, 2015.

[5] V. DeMiguel and H. Xu, "A stochastic multiple-leader stackelberg model: analysis, computation, and application," *Operations Research*, vol. 57, no. 5, pp. 1220–1235, 2009.

[6] S. E. Albarran and J. B. Clempner, "A stackelberg security markov game based on partial information for strategic decision making against unexpected attacks," *Engineering Applications of Artificial Intelligence*, vol. 81, pp. 408–419, 2019.

[7] M. Tambe, *Security and game theory: algorithms, deployed systems, lessons learned*. Cambridge university press, 2011.

[8] A. Bensoussan, S. Chen, A. Chutani, S. P. Sethi, C. C. Siu, and S. C. Phillip Yam, "Feedback stackelberg–nash equilibria in mixed leadership games with an application to cooperative advertising," *SIAM Journal on Control and Optimization*, vol. 57, no. 5, pp. 3413–3444, 2019.

[9] X. He, A. Prasad, and S. P. Sethi, "Cooperative advertising and pricing in a dynamic stochastic supply chain: Feedback stackelberg strategies," in *PICMET'08-2008 Portland International Conference on Management of Engineering & Technology*. IEEE, 2008, pp. 1634–1649.

[10] K. Stanková, B. Ranjbar-Sahraei, G. Weiss, and K. Tuyls, "Staco: Stackelberg-based coverage approach in robotic swarms," *Proceedings of ADAPTIVE 2013*, pp. 71–76, 2013.

[11] X. Duan, D. Paccagnan, and F. Bullo, "Stochastic strategies for robotic surveillance as stackelberg games," *IEEE Transactions on Control of Network Systems*, 2021.

[12] V. Hebbar and C. Langbort, "A stackelberg signaling game for human-uav collaboration in a search-and-rescue context," *IFAC-PapersOnLine*, vol. 53, no. 5, pp. 297–302, 2020.

[13] J. J. Koh, G. Ding, C. Heckman, L. Chen, and A. Roncone, "Cooperative control of mobile robots with stackelberg learning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 7985–7992.

[14] B. Cohen, S. Chitta, and M. Likhachev, "Single-and dual-arm motion planning with heuristic search," *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 305–320, 2014.

[15] R. Shome, K. Solovey, A. Dobson, D. Halperin, and K. E. Bekris, "drrt*: Scalable and informed asymptotically-optimal multi-robot motion planning," *Autonomous Robots*, vol. 44, no. 3, pp. 443–467, 2020.

[16] R. Shome and K. E. Bekris, "Synchronized multi-arm rearrangement guided by mode graphs with capacity constraints," in *Algorithmic Foundations of Robotics XIV*, S. M. LaValle, M. Lin, T. Ojala, D. Shell, and J. Yu, Eds. Springer International Publishing, 2021, pp. 243–260.

[17] Z. Xian, P. Lertkultanon, and Q.-C. Pham, "Closed-chain manipulation of large objects by multi-arm robotic systems," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 1832–1839, 2017.

[18] R. Shome, K. Solovey, J. Yu, K. Bekris, and D. Halperin, "Fast, high-quality two-arm rearrangement in synchronous, monotone tabletop setups," *IEEE Transactions on Automation Science and Engineering*, 2021.

[19] P. Paruchuri, J. P. Pearce, J. Marecki, M. Tambe, F. Ordonez, and S. Kraus, "Playing games for security: An efficient exact algorithm for solving bayesian stackelberg games," in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*, 2008, pp. 895–902.

[20] S. D. Han, N. M. Stiffler, A. Krontiris, K. E. Bekris, and J. Yu, "Complexity results and fast methods for optimal tabletop rearrangement with overhand grasps," *The International Journal of Robotics Research*, vol. 37, no. 13-14, pp. 1775–1795, 2018.

[21] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[22] B. Huang, S. D. Han, A. Boularias, and J. Yu, "DIPN: Deep interaction prediction network with application to clutter removal," in *IEEE International Conference on Robotics and Automation*, 2021.

[23] B. Huang, S. D. Han, J. Yu, and A. Boularias, "Visual foresight trees for object retrieval from clutter with nonprehensile rearrangement," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 231–238, 2022.