A Topological Approach to Design Problems for Distributed Systems using Category Theory: A Focus on Robotic Swarms

1st Philipp Ahrendt
Quartz Laboratory
ISAE-Supméca
Saint-Ouen, France
https://orcid.org/0009-0004-6873-3718

2nd Mehdi Mcharek
ESTACA'Lab
ESTACA
Saint-Quentin-en-Yvelines, France
https://orcid.org/0000-0002-5813-7662

3rd Stéphane Dugowson
Quartz Laboratory
ISAE-Supméca
Saint-Ouen, France
https://orcid.org/0009-0006-8535-583X

4th Moncef Hammadi
Quartz Laboratory
ISAE-Supméca
Saint-Ouen, France
https://orcid.org/0000-0002-5050-7315

2025 IEEE International systems Conference (SysCon) | 979-8-3315-0818-0/25/$31.00 ©2025 IEEE | DOI: 10.1109/SYSCON64521.2025.11014826

***Abstract*—The mathematical formalization of design problems has received significant attention in recent years. A number of theoretical frameworks have been developed which express certain classes of design problems as mathematical optimization problems with constraints. Compositionality in complex systems has played a prominent role in structuring the reasoning behind these approaches, and category theory has been foundational in formalizing compositional structures. This work proposes an adaptation of the mathematical design problem to large distributed systems. A distributed system is modeled in terms of its local interactions, which are interpreted as a topological structure on the system and represented by a hypergraph. Category theory is used to match the structure of the design problem to the hypergraph of the system, and to formulate a general distributed design problem with global constraints. By applying concepts from algebraic topology, in particular homological algebra, the distributed design problem is brought into an explicit form which allows for the use of existing algorithms for optimization with consensus constraints. Compared to previous formalizations, this approach provides a finer structural view of systems with complex operational interdependencies, and has the potential to reduce the complexity of the design problem for large robotic swarms by exploiting the partial independences present in the system.**

***Index Terms*—Distributed Systems, Optimization Problems, Co-Design, Category Theory, Homological Algebra,**

I. Introduction

In recent years, significant progress has been made in the development of general theoretical frameworks for complex design problems in engineered systems. Mathematical formulations of co-design now exist which provide a way to systematically transform design problems on a given system architecture into explicit optimization or constraint satisfaction problems. However, the design of autonomous robotic swarms, characterized by their distributed functionality and operational interdependencies, still poses significant challenges which are not addressed by the existing theories. While existing models have the capacity to take into account the system architecture to some degree, the conditions for this are too rigid to be applicable to more complex swarm systems. More general formulations, which weaken the structural requirements in order to be applicable to swarms, are currently not expressive enough to explicitly account for the interdependencies within the system. As a result, these approaches are very hard to scale to large and complex collaborative systems, and quickly become computationally intractable [1].

This work proposes a structured formulation of design problems for generic collaborative swarms with fixed patterns of interaction. The key idea is that agents in a large swarm only have direct operational interdependencies with a small part of the swarm. This results in a decomposition of the swarm into smaller overlapping regions. By applying concepts from category theory and algebraic topology, this decomposition can be used to rephrase global design problems as multi-objective optimization problems with consensus constraints. This approach makes it possible to take into account more general dependency patterns than previous frameworks, and has the potential to significantly increase the scalability of mathematical design optimization techniques.

The paper is organized as follows: Section II reviews related work on formalizing design problems in engineered systems, with a focus on compositional approaches and their extensions to distributed systems. Section III introduces the necessary concepts from category theory, including categories, functors, and their use in system design. The proposed framework is detailed in Section IV, where the mathematical foundations for the design of large-scale robotic swarms are laid out, and optimization is formulated using homological algebra. Section VI provides a discussion of the framework's contributions, limitations, and potential applications. Finally, Section VII concludes the paper by summarizing the key contributions and outlining directions for future research.

II. Related Work

Previous works on the mathematical formalization of system design have advanced our understanding of compositionality in complex systems, but they exhibit limitations that motivate the approach proposed in this work.

Censi [2] introduced a theoretical formulation of design problems for engineered systems, defining them in terms of functionalities and requirements of system components. By assuming a monotonous relationship between functionality and requirements, the framework allowed for compositional modeling through a simple rule that generated an optimization problem from a given architecture. However, this assumption restricts its applicability to systems where dependencies are straightforward and monotonous, making it unsuitable for systems with complex or non-monotonic interactions.

Building on Censi's work, Carlone and Pinciroli [3] generalized the framework by removing the division between functionality and requirements and relaxing the assumption of monotonous relationships. Their formulation, based on performance metrics and system-wide constraints, extended applicability to systems with intricate interactions, such as collaborative robot swarms. However, this generalization came at the cost of compositionality. By treating the system as a whole, their approach devolves into a high-dimensional search problem, lacking mechanisms to explicitly link the problem structure to the swarm's inherent interdependencies.

Dickerson and Wilkinson [4] proposed a constraint-driven design framework that uses enriched category theory to capture system structures more explicitly. While their approach exploits architectural structure to reduce the complexity of the design problem, it is heavily reliant on a well-structured system architecture. For highly distributed systems, such as robotic swarms, their framework does not provide a direct means of modeling the dynamic interactions within the system.

A complementary direction has been explored through Multidisciplinary Design Optimization (MDO) [5], which focuses on optimizing the performance of systems involving multiple disciplines or subsystems. MDO methods effectively coordinate design decisions across subsystems, but their focus on subsystem coordination does not address the topological and compositional challenges posed by distributed systems, where interactions often resemble those of a hypergraph structure.

While these works highlight the importance of compositionality in structured system architectures, they reveal significant gaps in addressing systems with complex and dynamic interactions. The generalization of Censi's framework [3] sacrifices compositionality, while the approaches in [4] and [5] rely on strong assumptions about system structure. This work addresses these limitations by introducing a topological perspective, leveraging hypergraphs and category theory to refine the representation of distributed systems.

III. Basic Concepts from Category Theory

We recall here the most important concepts from category theory which will be used in our further development. We refer to the standard reference [6], as well as the more introductory work [7], for more detailed expositions of category theory.

A. Categories

A *category* $\mathscr{C}$ is a collection of objects and relations between objects, called *morphisms*. We represent a morphism f between two objects C, D as an arrow $C \xrightarrow{f} D$. Morphisms can be composed, such that any sequence $C \xrightarrow{f} D \xrightarrow{g} E$ induces a morphism $C \xrightarrow{f \circ g} E$. This composition is required to be associative, meaning that $h \circ (g \circ f) = (h \circ g) \circ f$ for any sequence of morphisms f, g, h. Lastly, in a category, each object C is required to have an *identity morphism* $C \xrightarrow{\mathrm{Id}} C$, which verifies $f \circ \mathrm{Id} = f$ for any outgoing morphism f from C, and $\mathrm{Id} \circ g$ for incoming morphism g into C.

Most mathematical structures fit into this basic framework. One basic example is the category **Set** with sets as objects, functions between sets as morphisms, and the usual composition of functions. It is easy to verify that function composition is associative, and that the identity functions of each set are indeed identity morphisms. Another important category, which will be used in our framework, is **Vect** with vector spaces as objects, and linear functions as morphisms. Composition is just the function composition inherited from **Set**, and thus to verify that this is a category one only needs to notice that identity functions are linear, and that the composition of two linear functions is again linear. Finally, a particular class of categories which will play an important role here are *partially ordered sets* (*posets* from now on). A poset is a set $\mathscr{A}$ equipped with a binary relation $\alpha \leq \beta$ on its elements, verifying the following three axioms:

1) *Reflexivity:* $\forall \alpha \in \mathscr{A}$, $\alpha \leq \alpha$.
2) *Transitivity:* $\forall \alpha, \beta, \gamma \in \mathscr{A}, \alpha \leq \beta \wedge \beta \leq \gamma \implies \alpha \leq \gamma$.
3) *Antisymmetry:* $\forall \alpha, \beta \in \mathscr{A}, \alpha \leq \beta \wedge \beta \leq \alpha \implies \alpha = \beta$.

A poset $\mathscr{A}$ can be seen as a category, where there is a single morphism $\alpha \to \beta$ whenever $\alpha \leq \beta$. axioms 1) and 2) ensure that this verifies the associativity and identity axioms of a category.

For any category $\mathscr{C}$, one can define the opposite category $\mathscr{C}^{\mathrm{op}}$ by reversing the directions on all its morphisms. For a poset $\mathscr{A}$, this amounts to saying that in $\mathscr{A}^{\mathrm{op}}$ there is a single morphism $\alpha \to \beta$ whenever $\alpha \geq \beta$, which is the same as reversing the order relation on $\mathscr{A}$.

We will often represent mathematical relations in a category through *commutative diagrams*. A diagram such as (1) is said to commute if any two directed paths between the same objects compose to the same morphism. In (1), there are two ways to construct a morphism from X to W, $c \circ a$ and $d \circ b$. Commutativity in this case means that $c \circ a = d \circ b$.

$$\begin{tikzcd} X \arrow[r, "a"] \arrow[d, "b"'] & Y \arrow[d, "d"] \\ Z \arrow[r, "c"'] & W \end{tikzcd} \tag{1}$$

From here on, all diagrams will be commutative unless stated otherwise.

B. Functors and natural transformations

If $\mathscr{C}$ and $\mathscr{D}$ are two categories, a *functor* $F : \mathscr{C} \to \mathscr{D}$ maps objects $C \in \mathscr{C}$ to objects $F(C) \in \mathscr{D}$, and morphisms $C \xrightarrow{f} D$ in $\mathscr{C}$ to morphisms $F(C) \xrightarrow{F(f)} F(D)$ in $\mathscr{D}$. These mappings should preserve identities and compositions, meaning that for any object $C \in \mathscr{C}$, $F(\mathrm{Id}_C) = \mathrm{Id}_{F(C)}$, and for any sequence $C \xrightarrow{f} D \xrightarrow{g} E$ in $\mathscr{C}$, $F(g \circ f) = F(g) \circ F(f)$. As a consequence of these two conditions, functors preserve commutative diagrams. Applying a functor $F : \mathscr{C} \to \mathscr{D}$ to the objects and morphisms of a diagram in $\mathscr{C}$ yields a new diagram in $\mathscr{D}$. If the original diagram in $\mathscr{C}$ was commutative, then so is the new one. Equation (2) illustrates this.

$$\begin{tikzcd} X \arrow[r, "a"] \arrow[d, "b"'] & Y \arrow[d, "d"] \\ Z \arrow[r, "c"'] & W \end{tikzcd} \quad \xrightarrow{F} \quad \begin{tikzcd} F(X) \arrow[r, "F(a)"] \arrow[d, "F(b)"'] & F(Y) \arrow[d, "F(d)"] \\ F(Z) \arrow[r, "F(c)"'] & F(W) \end{tikzcd} \tag{2}$$

An important and relatively simple class of functors are those of the form $F : \mathscr{A} \to \mathscr{C}$, where $\mathscr{A}$ is a poset and $\mathscr{C}$ is any other category. Usually we will take $\mathscr{C} = \textbf{Vect}$ or $\textbf{Set}$.

For two functors $F, G : \mathscr{C} \to \mathscr{D}$ between the same categories, a *natural transformation* $\mu : F \Rightarrow G$ from one functor to another is a family of morphisms $F(C) \xrightarrow{\mu_C} G(C)$ in $\mathscr{D}$, for each object $C \in \mathscr{C}$, such that the diagram (3) commutes for all morphisms $C \xrightarrow{f} D$ in $\mathscr{C}$.

$$\begin{tikzcd} F(C) \arrow[r, "\mu_C"] \arrow[d, "F(f)"'] & G(C) \arrow[d, "G(f)"] \\ F(D) \arrow[r, "\mu_D"'] & G(D) \end{tikzcd} \tag{3}$$

C. Limits and colimits

Just as any poset can be seen as a category, one can construct an abstract category from any diagram. For example, the diagram (1) can be seen as an abstract diagram category with four objects, their identity morphisms, and the four morphisms shown in the diagram and their compositions. Let $\mathscr{I}$ is such an abstract diagram category, for example a generic poset. In any category $\mathscr{C}$, a diagram with the same "shape" as $\mathscr{I}$ can be interpreted as a functor $F : \mathscr{I} \to \mathscr{C}$. A functor into the category **Vect** would be a similar diagram where the objects are vector spaces and the morphisms are linear functions. A *cone* above a functor $F : \mathscr{I} \to \mathscr{C}$ is an object $C \in \mathscr{C}$, together with a family of morphisms $\mu_i : C \to F(i)$ for each object $i \in \mathscr{I}$, such that for all morphisms $f : i \to j$ in $\mathscr{I}$, the following diagram commutes

$$\begin{tikzcd} C \arrow[d, "\mu_i"'] \arrow[dr, "\mu_j"] & \\ F(i) \arrow[r, "F(f)"'] & F(j) \end{tikzcd} \tag{4}$$

The *categorical limit* of F is a cone $(\lim F, (\pi_i)_i)$ above F with the following property:

- For any cone $(C, (\mu_i)_i)$ above F, there is a *unique* morphism $\epsilon : C \to \lim F$ such that the following diagrams commute

$$\begin{tikzcd} C \arrow[r, dotted, "\epsilon"] \arrow[dr, "\mu_i"'] & \lim F \arrow[d, "\pi_i"] \\ & F(i) \end{tikzcd} \tag{5}$$

Dually, a *cocone* below F is an object C with morphisms $\lambda_i : F(i) \to C$ going in the opposite direction, such that the dual diagrams to (4) commute. A colimit is a cocone $(\operatorname{colim} F, (\iota_i)_i)$ such that for each cocone $(C, (\lambda_i)_i$ there is a unique morphism $\eta : \operatorname{colim} F \to C$ such that the dual diagrams to (5) all commute.

IV. Co-Design for Large Robot Swarms

A. Hypergraph representation of swarm systems

We consider a swarm of (possibly heterogenous) robots as a set $\mathbb{I}$, where each $i \in \mathbb{I}$ represents an individual robot. Each robot can be represented by a family of *design parameters* $(x_k)_{k \in K_i}$ which can be represented as vectors in $\mathbb{R}^{K_i}$. We denote $X = \prod_{i \in \mathbb{I}} \mathbb{R}^{K_i}$ the full configuration space of the swarm. For simplicity, we consider only design parameters which take real values, and suppose that these can be arbitrarily chosen. A complete formulation of global design problems for this context is given by Carlone and Pinciroli [3]. Although their work is based on a discrete set of possible design choices, the principle remains the same for continuous parameters. One associates to the swarm a system-level performance metric, which we take for simplicity to be a real-valued function $p : X \to \mathbb{R}$. Likewise, system-level equality constraints $(c_k,\ k \in C_=)$ and inequality constraints $(c_{k'},\ k' \in C_\leq)$ are imposed. At the full system level, this yields the design problem formulation (6).

$$\begin{gathered} \operatorname*{argmax}_{x \in X} p(x) \\ \begin{cases} c_k(x) = 0 \ \forall k \in C_= \\ c_{k'}(x) \leq 0 \ \forall k' \in C_\leq \end{cases} \end{gathered} \tag{6}$$

As is stated in [3], this type of problem generally grows exponentially with the size of $\mathbb{I}$, making it unsuitable for a direct application on large swarm structures. For this reason there is a need for additional structure in the problem formulation.

One characteristic of a large swarm is that the interactive behavior within it is decentralized and local. This can be due to limitations for instance in the communication distance, or a part of the problem formulation, for example when a global objective is divided into objectives for parts of the swarm. In any case, the local nature of interactions in the swarm can be understood as a property of *conditional independence*.

To illustrate this, consider a simple swarm $\{r_1, r_2, r_3\}$ of three robots working on a given task. We neglect for now any constraints on the system and focus on the performance metric p. If none of the three robots interact with each other, as in

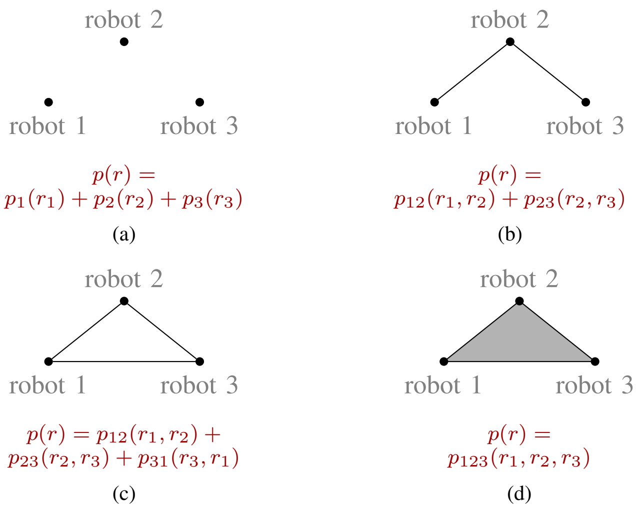

Fig. 1. Possible interaction patterns on a system leading to different structures on the performance metric. It can be observed that this allows to differentiate between "mutual interaction", as in (d), and 2 by 2 interaction (c).

figure 1a then we can measure the performance of the system simply by compounding the performance of each individual robot, and p is just a sum of three functions which can be optimized independently. Now, suppose instead instead r_1 and r_3 both interact with r_2 without any direct interaction between them, and that moreover the interactions of r_2 with the other two don't interfere with each other. This corresponds to figure 1b, and an example of this would be if r_2 emits a signal which is picked up by the other robots. In this case, we can write p as the sum of two terms which both depend on r_2. Figure 1 showcases four different interaction patterns with their corresponding decompositions of p.

This formalism is equivalent to a *factor graph*. Factor graphs are commonly used to represent the dependence structure of random variables, and provide a convenient framework for a wide range of algorithms to analyze a function which decomposes according to it [8], [9]. Factor graphs have moreover been applied to decentralized constraint optimization problems, which are relevant for network optimization problems in robotic swarms [10]. Another equivalent formalism used in network theory are *hypergraphs*, which are typically used in a more generic fashion as means to represent a system structure [11]. A hypergraph can be defined as a pair $(\mathbb{I}, \mathscr{A})$, where $\mathbb{I}$ is a set of vertices and $\mathscr{A} \subseteq \mathscr{P}(\mathbb{I})$ is the set of hyperedges α connecting the vertices $i \in \alpha$. For example, if $\mathbb{I} = \{1, 2, 3, 4\}$, a possible hypergraph would be $\mathscr{A} = \{\{1, 2\}, \{2, 3, 4\}\}$.

B. Consistency as categorical limit

In light of the preceding discussion, we will view a swarm as a hypergraph $(\mathbb{I}, \mathscr{A})$, and we will show how a design problem can be formulated explicitly in terms of this structure. For this, a formulation in category theoretic terms, as it is outlined in [12], is the most convenient.

We interpret $\mathscr{A}$ as a poset ordered by inclusion of subsets $\alpha \subseteq \beta$, and define a functor $F : \mathscr{A}^{\mathrm{op}} \to \mathbf{Vect}$, which to each hyperedge $\alpha \in \mathscr{A}$ associates the vector space $F_\alpha = \prod_{i \in \alpha} \mathbb{R}^{K_i}$, and to each inclusion $\alpha \subseteq \beta$ the canonical projection $\pi_\alpha^\beta : F_\beta \to F_\alpha$. We call this the *design functor* of the system. This associates to each group of interacting agents the corresponding design space. Now the performance metric p is supposed to be decomposable over $\mathscr{A}$. The performance metric is thus determined by a family of local functions $p_\alpha : F_\alpha \to \mathbb{R}$ for each interaction group α.

The fact that the hyperedges in $\mathscr{A}$ may overlap makes it impossible to optimize the local performance metrics p_α independently, since doing this might lead to incompatible local solutions. The incompatibility of local design parameters can be detected within the functor F through the projection maps. The easiest way to see this is by using the categorical limit of F. The limit $\lim F$ is given by

$$\lim F = \left\{ (x_\alpha)_{\alpha \in \mathscr{A}} \in \prod_{\alpha \in \mathscr{A}} F_\alpha,\ \forall \alpha \subseteq \beta, x_\alpha = \pi_\alpha^\beta(x_\beta) \right\} \tag{7}$$

Using this, the following result can be obtained.

Proposition 1. *Let $(\mathbb{I}, \mathscr{A})$ be a swarm with total design space X and design functor F. Then the following conditions are equivalent:*

1) *For all $\alpha, \beta \in \mathscr{A}$ we also have $\alpha \cap \beta \in \mathscr{A}$.*
2) *$X = \lim F$.*

As a consequence, it is possible to rephrase the optimization of $p(x)$ as an optimization of independent local objectives $p_\alpha(x_\alpha)$, subject to a global "compatibility" constraint, which is $(x_\alpha)_\alpha \in \lim F$. Although this only recovers the actual optimization problem if $\mathscr{A}$ is closed by intersections, the formulation can be used for a decomposition on any hypergraph, by recursively adding any missing intersections and setting the corresponding local performance metrics equal to zero. The effect of this is that there are new variables which are not an explicit part of the optimization problem, but which appear implicitly when formulating the consistency constraint. By the equivalence in proposition 1, adding these to the constraint is necessary for the optimization problem to be well-defined on X. This condition ensures equality between shared variables because the shared variables of two sets will be projected in the intersection of these two sets. The intersection then automatically enforces the equality of these projections. From here on, all hypergraphs will be supposed to be closed by intersections unless stated otherwise, with missing intersections added if necessary.

C. Global constraints as categorical colimit

Similarly to the optimization of global performance, constraint satisfaction on the system level can also be formulated in terms of the associated hypergraph $(\mathbb{I}, \mathscr{A})$. However, for this to work in the present framework all constraints have to be linear. Possible generalizations which include other classes of constraints, such as polynomial constraints, are left for future work.

Suppose all the equality constraint functions $(c_k, k \in C_=)$ and inequality constraint functions $(c_{k'}, k' \in C_\leq)$ are real-valued affine functions on the full design space X. Then each constraint function has a canonical expression as a sum of linear functions on the subspaces F_α given by the design functor F. On the toy system with three robots represented by figure 1b, a linear constraint function $c(x_1, x_2, x_3)$ can be written as

$$c(x_1, x_2, x_3) = c(x_1, x_2, 0) + c(0, x_2, x_3) - c(0, x_2, 0) + K \tag{8}$$

where K is a constant which depends on the constant term in c and on the decomposition. In (8), one would always have $K = 0$. Such a decomposition into affine functions is almost never unique. Any parameters that appear together in more than one hyperedge can be linearly transformed in different functions of a decompositions, in such a way that the sum remains unmodified. Equation (9) gives all the possible affine decompositions of $c(x_1, x_2, x_3)$ which respect the hypergraph and leave the constant coefficients invariant.

$$\begin{aligned} c(x_1, x_2, x_3) = c(x_1, \lambda x_2, 0) + c(0, \mu x_2, x_3) \\ - c(0, (\lambda + \mu - 1)x_2, 0) + K \end{aligned} \tag{9}$$

where λ, μ is any pair of real numbers. All of these decompositions are equally valid and can be used to express design constraints on the parts of system, in such a way that the corresponding system level constraint is verified. However, choosing a particular decomposition would unnecessarily restrict the possible design choices and lead to suboptimal solutions to the design problem. The functor F can be used to express the fact that the local design parameters verify *any one* of the collections of local constraints which reconstitute the system constraint, without having to specify one. By only considering transformations which do not alter any of the constant terms, the analysis can be restricted to linear constraints.

If $(\mathbb{I}, \mathscr{A})$ is the hypergraph representing the system, X its design space and F is its design functor, we associate to each vector space F_α its dual space F_α^* of real-valued linear functions $c_\alpha : F_\alpha \to \mathbb{R}$. Likewise we note X^* the dual space of the design space, which is the space containing the system level constraint functions. For each $\alpha \subseteq \beta$, the corresponding projection π_α^β induces an injection $\iota_\alpha^\beta : F_\alpha^* \to F_\beta^*$, defined as $\iota_\alpha^\beta(c_\alpha)(x_\beta) = c_\alpha(\pi_\alpha^\beta(x_\beta))$. This gives a new functor $F^* : \mathscr{A} \to \mathbf{Vect}$, which we will call the *constraint functor*[1]. Using F^* we can formulate the problem of determining equivalent constraint decompositions as a dual problem to the compatibility problem expressed by the design functor. The equivalence of local constraints can be expressed with the categorical colimit of F^*, which is given as the following quotient space $(\prod_{\alpha \in \mathscr{A}} F_\alpha^*)/\sim$, where $\sim$ is the linear equivalence relation on $\prod_{\alpha \in \mathscr{A}} F_\alpha^*$ generated by

$$\forall \alpha \subseteq \beta,\ \forall c_\alpha \in F_\alpha^*,\ \langle c_\alpha \rangle \sim \langle \iota_\alpha^\beta(c_\alpha) \rangle \tag{10}$$

Where $\langle c_\alpha \rangle$ is the vector with the coordinates of c_α on F_α^* and zeros elsewhere. Equation (10) yields the dual result to proposition 1.

Proposition 2. *Let $(\mathbb{I}, \mathscr{A})$ be a swarm with total design space X and constraint functor F^*. Then the following conditions are equivalent:*

1) *For all $\alpha, \beta \in \mathscr{A}$ we also have $\alpha \cap \beta \in \mathscr{A}$.*
2) $X^* = \operatorname{colim} F^*$.

As a consequence, we can identify system level constraint functions with equivalence classes $[c]$ of families of local constraints c_α in $\operatorname{colim} F^*$. With this, we can state the full design problem on $(\mathbb{I}, \mathscr{A})$ in terms of the functors F and F^*.

$$\begin{gathered} \underset{x \in \hat{X}}{\operatorname{argmax}} \sum_{\alpha \in \mathscr{A}} p_\alpha(x_\alpha) \\ \begin{cases} x \in \lim F \\ [c_k](x) = A_k \ \forall k \in C_= \\ [c_{k'}](x) \leq A_{k'} \ \forall k' \in C_\leq \end{cases} \\ [c_k], [c_{k'}] \in \operatorname{colim} F^* \end{gathered} \tag{11}$$

D. Formulation in terms of homological algebra

In order to make the very general formulation in (11) tractable as a concrete optimization problem, we will express it in terms of homological algebra. The resulting constructions are algebraically explicit, and can be exploited in optimization algorithms.

Homological algebra originates in the classification of topological spaces, and has traditionally been used to define algebraic invariants preserved by continuous deformation. It has since evolved into a rich subject strongly connected to category theory, and with applications in topology, group theory, and algebraic geometry. We refer to [13] for a detailed account of homological algebra and its applications in mathematics. In recent years there has been a significant increase in applications of homology outside of pure mathematics, most prominently in data analysis [14]. Recent applications of homological algebra to distributed optimization problems, similar to the one presented in this work, have led to new optimization algorithms which account for the presence of interdependence constraints [15], [16], [12].

In order to define homological structures, we use the theoretical framework described in [12].

Let $C^0 = \prod_{\alpha \in \mathscr{A}} F_\alpha$ be the full product of local design spaces. We define another product space $C^1 = \prod_{\alpha \subset \beta} F_\alpha$, where the product runs over all strictly ordered pairs $\alpha \subset \beta$ in $\mathscr{A}$. By relating the spaces C^0 and C^1, it will be possible to characterize the elements of C^0 which belong to $\lim F$. We can define a linear map $\delta_0 : C^0 \to C^1$, given as follows

[1]This can be seen as the composition of functors $\mathscr{A} \xrightarrow{F^{\mathrm{op}}} \mathbf{Vect}^{\mathrm{op}} \xrightarrow{-^*} \mathbf{Vect}$.

$$\delta_0 x_{\alpha\beta} = \pi^\beta_\alpha(x_\beta) - x_\alpha \tag{12}$$

this is the first of a sequence of linear maps $0 \to C^0 \xrightarrow{\delta_0} C^1 \xrightarrow{\delta_1} C^2 \xrightarrow{\delta_2} ...$, which constitute a *cochain complex*. The maps δ_n are the *codifferentials*, and have the crucial property

$$\delta_{n-1} \circ \delta_n = 0 \tag{13}$$

Property (13) is equivalent to the inclusion relation $\operatorname{im}\delta_{n-1} \subseteq \ker\delta_n \subseteq C_n$, where $\operatorname{im}\delta_{n-1}$ is the image space $\{\delta_{n-1}x,\ x \in C^{n-1}\}$ and $\ker\delta_n$ is the kernel $\{x \in C^n,\ \delta_n x = 0\}$. By this inclusion, we can define a sequence of spaces $H^n(C) = \ker\delta / \operatorname{im}\delta_{n-1}$, which are the *cohomology groups* associated to the cochain complex. Concretely, the cohomology group $H^n(C)$ is the supplementary vector space to $\operatorname{im}\delta_{n-1}$ inside $\ker\delta_n$, i.e the subspace of $\ker\delta_n$ which verifies $\operatorname{im}\delta_{n-1} \oplus H^0(C) = \ker\delta_n$.

For the design problem at hand it suffices consider the cohomology group $H^0(C)$, which is simply $\ker\delta_0$. The definition of δ_0 in (12) gives an algebraic interpretation of the consistency discussed in IV-B. More precisely, we have the following

$$\lim F = H^0(C) \tag{14}$$

A dual construction can be defined on the functor F^*. Similarly to the preceding case, we define $C_0 = \prod_{\alpha\in\mathscr{A}} F^*_\alpha$ and $C_1 = \prod_{\alpha\subset\beta} F^*_\alpha$. Thus, C_0 and C_1 are the dual vector spaces to C^0 and C^1. To the map $\delta_0 : C^0 \to C^1$ there is a dual map $\partial_0 : C_1 \to C_0$. An explicit formula for $\partial_0\Delta$, where $\Delta = (\Delta_{\alpha\beta})_{\alpha\subset\beta} \in C_1$, can be given as follows

$$\partial_0\Delta_\alpha = \sum_{\gamma\subset\alpha} \iota^\alpha_\gamma(\Delta_{\gamma\alpha}) - \sum_{\alpha\subset\beta} \Delta_{\alpha\beta} \tag{15}$$

As in the previous case, ∂_0 is the beginning of a sequence $0 \leftarrow C_0 \xleftarrow{\partial_0} C_1 \xleftarrow{\partial_1} C_2 \xleftarrow{\partial_2} ...$, this time in the opposite direction, and with the property $\partial_{n-1} \circ \partial_n = 0$. This is a *chain complex*, and ∂_n are the *differentials*. This yields a sequence of *homology groups* $H_n(C) = \ker\partial_{n-1} / \operatorname{im}\partial_n$. Once again, only the group $H_0(C)$ will be used here.

Equation (15) is closely related to the equivalence relation in (10). If we take $\alpha \subset \beta$, and $c_\alpha \in F^*_\alpha$, then we can define $\Delta \in C_1$ as $\Delta_{\alpha\beta} = c_\alpha$, with $\Delta_{\epsilon\eta} = 0$ for all other sequences $\epsilon \subset \eta$. Now, we have $\langle c_\alpha \rangle \sim \langle \iota^\beta_\alpha(c_\alpha) \rangle$. Using (15), we obtain

$$\begin{aligned} \left(\langle \iota^\beta_\alpha(c_\alpha) \rangle - \langle c_\alpha \rangle\right)_\alpha &= c_\alpha = \partial_0\Delta_\alpha \\ \left(\langle \iota^\beta_\alpha(c_\alpha) \rangle - \langle c_\alpha \rangle\right)_\beta &= \iota^\beta_\alpha(c_\alpha) = \partial_0\Delta_\beta \end{aligned} \tag{16}$$

Everything else being zero, we have $\langle \iota^\beta_\alpha(c_\alpha) \rangle - \langle c_\alpha \rangle = \partial_0\Delta$. Using the fact that the equivalence relation is generated by these pairs, we can construct a Δ for any equivalent pair $c \sim c'$ verifying this equation. Going back to the definition of $H_0(C)$, the following can be shown to hold

$$\forall c, c' \in C_0,\ c \sim c' \iff [c] = [c'] \in H_0(C) \tag{17}$$

Lastly, to complete the design problem, local constraints valid for groups of interacting agents can be formulated in C_0, as collections of equality and inequality constraints given by functions c_{i_α} for the groups $\alpha \in \mathscr{A}$. These local constraints, even when patched together, do not represent a homology class like proper global constraints, since they are "fixed" locally and cannot be arbitrarily distributed.

Putting everything together, the following purely algebraic formulation of the design problem for distributed systems is obtained.

$$\begin{gathered} \underset{x\in C^0}{\operatorname{argmax}} \sum_{\alpha\in\mathscr{A}} p_\alpha(x_\alpha) \\ \begin{cases} x \in H^0(C) \\ c_{i_\alpha}(x_\alpha) = 0\ \forall\alpha \in \mathscr{A},\ i_\alpha \in C^\alpha_= \\ c_{i'_\alpha}(x_\alpha) \leq 0\ \forall\alpha \in \mathscr{A},\ i'_\alpha \in C^\alpha_\leq \\ [c_k](x) = A_k\ \forall k \in C_= \\ [c_{k'}](x) \leq A'_k\ \forall k' \in C_\leq \end{cases} \\ [c_k], [c_{k'}] \in H_0(C) \end{gathered} \tag{18}$$

V. Case study: UAV swarms in Search-and-Rescue missions

In this case study, we apply our mathematical framework to a search-and-rescue (SAR) mission, where a swarm of unmanned aerial vehicles (UAVs) is deployed over a disaster-affected area. The mission aims to locate survivors efficiently by maximizing area coverage, minimizing response time, and ensuring accurate localization.

The operation involves a decentralized UAV swarm with autonomous decision-making capabilities and collaborative behavior. Focusing for now on objectives and constraints which can be defined for the swarm as a whole, or for a group of interacting UAVs, a design problem can be formulated as follows

- **Maximize coverage:** The swarm as a whole has to cover as large an area as possible.
- **Respect communication constraints:** Limits in the amount of communication of the swarm, both within the swarm and with an outside control center, are imposed.

Within the swarm $\mathbb{I}$, individual UAVs will only interact locally, which allows for the definition of a hypergraph $(\mathbb{I}, \mathscr{A})$ on the swarm. The objective of maximal area coverage can be decomposed in terms of regions of interaction. If $\mathbb{A}(x_1, ...x_n)$ is the area coverage of the swarm as a function of the configurations of the individual drones, the partial operational independence gives a decomposition of $\mathbb{A}$ as a sum

$$\mathbb{A} = \sum_{\alpha\in\mathscr{A}} \kappa_\alpha \mathbb{A}_\alpha \tag{19}$$

where $\mathbb{A}_\alpha$ is the area covered by the UAVs in α, and κ_α are correction constants to avoid overcounting of areas covered by several groups. In general the area function cannot be decomposed into a sum of functions for the individual UAVs, since covered areas may overlap. However, under the assumption that non-interacting UAVs do not have overlapping coverage, the decomposition (19) is always possible. Equation (19) is defined for $x \in C^0$, which also includes UAVs in overlapping regions counted several times with inconsistent configurations. Imposing the consistency constraint $x \in H^0(C)$ ensures that UAVs on intersections are assigned only one configuration.

The communication limits are a combination of global and local constraints. Firstly, each interaction group will have a communication limit $c_\alpha(x_\alpha) \leq 0$ of information sharing between UAVs in $\mathscr{A}$. Secondly, outward communication of the whole swarm, for example with a control center, is limited. If we consider a limit imposed by the data processing capacity of the external entity, this gives a global constraint $c \leq A$ which can be distributed in any way among the regions of the swarm. Thus, we can view c as a homology class in $[c] \in H_0(C)$, supposing that the constraint is linear. Consider a toy problem involving three UAVs with a communication pattern as in figure 1b. The graph representation in the figure can be taken directly as a definition for the associated hypergraph $(\mathbb{I}, \mathscr{A})$, leaving out the vertices at the edges, since they will not appear separately in this problem.

- $\mathbb{I} = \{1, 2, 3\}$.
- $\mathscr{A} = \{\{2\}, \{1, 2\}, \{2, 3\}\}$.

In this case the total area coverage is $\mathbb{A} = \mathbb{A}_{12} + \mathbb{A}_{23} - \mathbb{A}_2$. Communication constraints can be defined on the edges as c'_{12} and c'_{23}, and an outward communication constraint c can be defined on the whole system. Since c can be arbitrarily distributed among the subgroups of the system, it is equivalent to a homology class $[c] \in H_0$, whereas the local constraints c'_{12} and c'_{23} are not. This gives the following optimization problem

$$\underset{x \in C^0}{\operatorname{argmax}}\, \mathbb{A}_{12}\left(x_1, x_2^{(1)}\right) + \mathbb{A}_{23}\left(x_2^{(3)}, x_3\right) - \mathbb{A}_2\left(x_2^{(2)}\right)$$
$$\begin{cases} x \in H^0(C) \\ c'_{12}\left(x_1, x_2^{(1)}\right) \leq 0 \\ c'_{23}\left(x_2^{(3)}, x_3\right) \leq 0 \\ [c](x) \leq A \end{cases} \tag{20}$$

The consistency constraint $x \in H^0(C)$ can be made even more explicit. The space C^1 is particularly simple since there are only two strict inclusion pairs $\{2\} \subset \{1, 2\}$ and $\{2\} \subset \{2, 3\}$. We have $C^1 = F_2^2$, where F_2 is the space of the variable x_2, and the codifferential δ_0 is given by

$$\delta_0\left(x_1, x_2^{(1)}, x_2^{(2)}, x_2^{(3)}, x_3\right) = \left(x_2^{(1)} - x_2^{(2)}, x_2^{(3)} - x_2^{(2)}\right) \tag{21}$$

We can take the transposed map $\delta_0^* : C^1 \to C^0$ and set $L = \delta_0^* \delta$, which is a square matrix acting on C^0, and verifies

$$x \in H^0(C) \iff Lx = 0 \tag{22}$$

Plugging this back into the optimization problem 20, we obtain a fully explicit constrained optimization problem:

$$\begin{aligned} \max_{x \in C^0} \quad & \mathbb{A}_{12}(x_1, x_2^{(1)}) + \mathbb{A}_{23}(x_2^{(3)}, x_3) - \mathbb{A}_2(x_2^{(2)}) \\ & \text{subject to:} \\ & Lx = 0, \\ & c'_{12}(x_1, x_2^{(1)}) \leq 0, \\ & c'_{23}(x_2^{(3)}, x_3) \leq 0, \\ & [c](x) \leq A, \end{aligned} \tag{23}$$

where:

- $L = \delta_0^* \delta_0$ is the Laplacian matrix derived from the codifferential δ_0 based on the hypergraph structure.
- $\mathbb{A}_{12}$, $\mathbb{A}_{23}$, and $\mathbb{A}_2$ represent local area coverage functions for subsets of UAVs, with $\mathbb{A}_2$ subtracted to avoid double counting.
- $c'_{12}(x_1, x_2^{(1)})$ and $c'_{23}(x_2^{(3)}, x_3)$ are local communication constraints.
- $[c](x)$ represents the global communication constraint as a homology class in $H_0(C)$.

The problem can now be solved using variational optimization techniques or distributed optimization algorithms that respect the problem's structure. Techniques such as Lagrange multipliers, gradient-based methods, or decentralized optimization algorithms (e.g., ADMM [17]) can be employed.

VI. Discussion

The proposed framework provides a novel approach to the design and optimization of collaborative robotic swarms. By utilizing category theory and algebraic topology, the framework enables the decomposition of complex swarm behaviors into well-defined local optimization problems, ensuring compatibility and consistency across the swarm. This structured approach is particularly useful for addressing scalability challenges associated with large swarm systems.

A key advantage of the framework is its ability to incorporate more general dependency patterns than traditional compositional methods. The use of homology and cohomology to model interactions and constraints allows for the explicit representation of both local and global design objectives. This capability enhances the framework's applicability to a wide range of collaborative systems, including UAV swarms in dynamic environments.

Despite its strengths, the framework does have certain limitations. The reliance on linear constraints simplifies the mathematical formulation but may restrict its applicability to more complex, nonlinear real-world scenarios. Additionally, the computational cost of homology and cohomology computations for large systems remains a challenge. These aspects highlight the need for further research to develop efficient algorithms and extend the framework to handle nonlinearities and dynamic swarm interactions. One possible direction is a possible generalization to polynomial constraints. Homological algebra has strong links to algebraic geometry, where the cohomology of certain functors plays an important role.

However, it is not yet clear what such a generalization would look like.

The framework aligns well with principles of multidisciplinary design optimization (MDO), facilitating the coordination of local and global design objectives. This alignment underscores its potential to serve as a foundation for hybrid approaches that integrate optimization techniques with machine learning or reinforcement learning for real-time decision-making.

In addition, there is significant potential for further refinement within the framework. The interaction model used in this work is relatively simple, and in the present problem formulation the conditions are favorable for well-posed design problems. In fact, our framework leads to a homology and cohomology which is trivial in higher degrees, and this fact is directly related to the well-posedness of the problem. This is no longer the case in more refined interaction models, and a more detailed analysis of topological properties will be required to analyze this more general setting. Homological algebra is uniquely equipped for this, and more progress can be made in the understanding of interactions in a distributed system.

VII. Conclusion

This work introduces a structured mathematical framework for the design and optimization of collaborative robotic swarms. By leveraging category theory and algebraic topology, the framework addresses the inherent complexities of swarm systems, enabling a systematic decomposition of global design problems into local optimization tasks. This approach ensures that the local solutions are globally consistent, thus overcoming the challenges posed by large-scale and distributed interactions.

The integration of homological and cohomological concepts into the framework provides a robust mathematical foundation for modeling both local and global constraints. This duality enables the representation of complex interdependencies within the swarm, offering a pathway to scalable and efficient optimization strategies. The framework's compatibility with multidisciplinary design optimization principles further enhances its applicability to diverse engineering problems.

While the framework offers significant advancements, its limitations also open avenues for future exploration. Extending the framework to handle nonlinear constraints, dynamic environments, and stochastic elements, as well as a more refined topological model of interdependency, will be essential for real-world applications. Additionally, developing efficient computational methods for large-scale systems remains a critical area of research.

The potential applications of this framework span a wide range of domains, including UAV swarms for search-and-rescue missions, autonomous sensor networks, and multi-robot coordination in industrial settings. By addressing these challenges, the framework lays the groundwork for future innovations in the design and operation of complex, autonomous systems.

References

[1] A. Prorok, M. Malencia, L. Carlone, G. S. Sukhatme, B. M. Sadler, and V. Kumar, "Beyond robustness: A taxonomy of approaches towards resilient multi-robot systems," 2021. [Online]. Available: https://arxiv.org/abs/2109.12343

[2] A. Censi, "A mathematical theory of co-design," 2016. [Online]. Available: https://arxiv.org/abs/1512.08055

[3] L. Carlone and C. Pinciroli, "Robot co-design: Beyond the monotone case," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3024–3030.

[4] C. E. Dickerson and M. K. Wilkinson, "Architecture, analysis, and design of systems using extensions of category theory," *IEEE Open Journal of Systems Engineering*, vol. 2, pp. 105–118, 2024.

[5] J. R. Martins and A. B. Lambe, "Multidisciplinary design optimization: a survey of architectures," *AIAA journal*, vol. 51, no. 9, pp. 2049–2075, 2013.

[6] S. Mac Lane, *Categories for the Working Mathematician*, 2nd ed., ser. Graduate Texts in Mathematics. Springer, sep 1998. [Online]. Available: http://www.worldcat.org/isbn/0387984038

[7] B. Fong and D. I. Spivak, *An Invitation to Applied Category Theory: Seven Sketches in Compositionality*. Cambridge University Press, Jul. 2019. [Online]. Available: http://dx.doi.org/10.1017/9781108668804

[8] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[9] O. Peltre, "Local max-entropy and free energy principles, belief diffusions and their singularities," 2023. [Online]. Available: https://arxiv.org/abs/2310.02946

[10] F. Fioretto, E. Pontelli, and W. Yeoh, "Distributed constraint optimization problems and applications: A survey," *Journal of Artificial Intelligence Research*, vol. 61, p. 623–698, Mar. 2018. [Online]. Available: http://dx.doi.org/10.1613/jair.5565

[11] L. Torres, A. S. Blevins, D. S. Bassett, and T. Eliassi-Rad, "The why, how, and when of representations for complex systems," 2020. [Online]. Available: https://arxiv.org/abs/2006.02870

[12] G. Sergeant-Perthuis, "Regionalized optimization," 2022. [Online]. Available: https://arxiv.org/abs/2201.11876

[13] C. A. Weibel, *An Introduction to Homological Algebra*, ser. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 1994.

[14] E. Munch, "A user's guide to topological data analysis," *Journal of Learning Analytics*, vol. 4, no. 2, Jul. 2017. [Online]. Available: http://dx.doi.org/10.18608/jla.2017.42.6

[15] J. Hansen and R. Ghrist, "Distributed optimization with sheaf homological constraints," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2019, pp. 565–571.

[16] O. Peltre, *A Homological Approach to Belief Propagation and Bethe Approximations*. Springer International Publishing, 2019, p. 218–227. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-26980-7_23

[17] Y. Wang, W. Yin, and J. Zeng, "Global convergence of admm in nonconvex nonsmooth optimization," *Journal of Scientific Computing*, vol. 78, pp. 29–63, 2019.