The Complexity of Partition Tasks

**Fernando Esponda, Matías Vera-Cruz,
Jorge Tarrasó, and Marco Morales**
Departments of Computer Science and Economics
Instituto Tecnológico Autónomo de México
México, D.F.
fernando.esponda@itam.mx, matiasveracruz@gmail.com, jorge.tarraso@gmail.com

Abstract

In this paper we introduce the Partition Task problem class along with a complexity measure to evaluate its instances and a performance measure to quantify the ability of a system to solve them. We explore, via simulations, some potential applications of these concepts and present some results as examples that highlight their usefulness in policy design scenarios, where the optimal number of elements in a partition or the optimal size of the elements in a partition must be determined.

1 Introduction

Groups of agents are often faced with situations that require them to divide—specialize—into subgroups. For example, a colony of ants must divide its members into foragers, lookouts, and workers in different proportions at different times; Mexican citizens must vote, every six years, for one of three political parties; and the denizens of Santa Fe, N.M. must decide every Friday night whether or not to attend the El Farol restaurant. The outcome of these tasks can depend, among other things, on factors like the availability and accuracy of information, the agent's internal decision-making models, and the interactions with other agents.

Consider the El Farol [3] problem in which the experience at the restaurant is deemed pleasurable if its at most 60% full. A possible variant of this problem is to include the owner's requirement of having the restaurant at no less than 60% of its maximum capacity, in this way both customers and owners are gratified when total attendance is exactly 60%. Suppose we want to model this situation and study the factors that influence the outcome; for instance, the amount of memory agents have, their decision strategies, and the effects of embedding agents in a social network. What is the influence of each variable? How will a given design do with a more difficult or with an easier problem? What *is* a more difficult or easier problem?

In this paper we provide the first step necessary towards answering these kinds of questions by characterizing problems in a way that is independent of the strategies available for solving them. We precisely define the types of problems we are interested in, establish a measure for their complexity, and give a measure of how well a particular group of agents addresses a given problem over time. These tools allow us to study problems in terms of their complexity and strategies in terms of their ability to solve them.

In Sect. 2 we define the class of tasks we are interested in and in Sect. 3 give a measure for their complexity. Section 4 discusses how to quantify the error of trying to solve a given task and Sect. 5 presents some potential applications as well as the results of an agent-based simulation that illustrate the uses of our proposed measures and definitions. In particular, we investigate a version of the El Farol restaurant problem and point out some of the variables that affect its complexity. In addition, we show how embedding agents in a social network affects their task-solving capabilities and how our tools can help design policies that enable agents to face less complex tasks.

1.1 Related Work

Partition tasks (PTs) are related to combinatorial optimization [22, 31] problems which are characterized by the task of finding an unknown solution out of combinatorially many. In contrast, a PT is specified by the set of all its possible solutions (its criterion), and therefore solving a PT consists of reaching any of these (known) solutions. Hence, the criterion to a given PT could be the set of solutions to a combinatorial optimization problem induced by a certain objective function. PTs are also related to complexity. According to [19], common definitions of complexity are associated with: how difficult it is to describe an object; how difficult it is to construct an object; and how organized an object is. Our work relates the most to the second point.

Significant work regarding complexity has been carried out in different fields such as physics [24, 27, 2], biology [8, 25], computer science [12, 16, 23] and social sciences [9, 11, 7, 4]. In many of these cases, the framework presented in this paper can be useful since the PT problem class is very general. For example, in [3] customers have a positive experience if they attend El Farol restaurant and it is no more than 60% full. Alternatively, in the Minority Game [10], agents decide among two choices to attend and the players who end up among the minority win and in the Standing Ovation problem [20] agents decide whether to stand up and clap or remain seated. Similarly, in [6, 13, 1] agents must make decisions or adopt characteristics that can also be modeled as a group of agents dividing itself into categories. However, all these papers focus on the system dealing with the PT and its characteristics. In the present work we concentrate on the construction of PTs independent of the systems that are trying to solve them and present a measure that can be used as a point of reference for the study of such systems.

Another relation with the present work can be found in theoretical computer science which defines problem complexity in terms of the most efficient algorithm that solves the problem [12, 16, 23]. PTs are fundamentally computational tasks and as such can be assigned a complexity in this sense. The distinction between this measure and the ones introduced in the current work is that the latter are not concerned with the time and space required to arrive at a solution and instead focus on the characteristics of the problem itself; namely, on the relative size of the solution set. The complexity class #P (sharp P) is the class of function problems whose image is the number of accepting paths of an NP problem [32]. Problems in this class are concerned with finding how many solutions a given problem has; for instance, computing the number of Hamiltonian cycles of some cost in a given graph. Our definition of PTs requires enumerating the valid arrangements of agents into bins; thus, defining a PT is in fact a #P problem.

Other complexity definitions, such as Description Length, Algorithmic Information Content, Thermodynamic Depth, and Effective Complexity, amongst others, aim to characterize the nature of objects and the behavior of systems (see [26, 21] for a review). The main distinction with our work is that these definitions would characterize a task with respect to the structure of its valid configurations, whereas the definition provided here is concerned only with the *a priori* likelihood of arriving at one of them, regardless of the complexity assigned to the configuration itself by other measures.

2 Partition Tasks

Throughout this paper we consider an agent-set to be any collection of individual entities, such as a bag of rocks, a collection of ants, or a group of voters and tasks to be certain arrangements or configurations that agents must achieve. For example, agents can collectively set market prices, learn complicated functions, or elect a public official. In particular we are interested in tasks that require a system to configure an agent-set into a specific arrangement, such as a bag of rocks sorted into igneus, sedimentary and metamorphic; 40% of ants in an ant-hill foraging; or voters decisions splitting in favor of a political party. We refer to these sort of tasks as *Partition Tasks* (PT). Intuitively, PT are defined by the agent-set and a specification of how it should be partitioned. In what follows we give a precise definition for PT and explicitly construct their problem space.

2.1 Problem Space

Let $\mathcal{A}_k = \{1, \ldots, k\}$ be a set of agents with $|\mathcal{A}_k| = k$, and $\mathcal{B}_n = \{1, \ldots, n\}$ a set of bins with $|\mathcal{B}_n| = n$. For the purpose of this construction we are only concerned with the number of agents in $\mathcal{A}_k$, the number of bins in $\mathcal{B}_n$, and their identities, which we represent as consecutive integers. Given $\mathcal{A}_k$ and $\mathcal{B}_n$ a *PT* must specify how the agent-set can be partitioned in order to satisfy the task. We call the set of satisfying arrangements the *criterion*.

2.1.1 Criterion Space

A partition of $\mathcal{A}_k$ into $\mathcal{B}_n$ is an exhaustive and mutually exclusive assignment of the k agents into the n bins. The criterion space consists of all the possible partitions induced by $\mathcal{A}_k$ and is constructed in two steps. First, we present the set of possible partitions according to their cardinality (eqn. 1) and then assign agents to each bin (eqn. 2).

$$\Delta_k^n = \left\{ (\delta_1, \ldots, \delta_n) : \delta_j \in \mathbb{Z}_+, \sum_{j=1}^{n} \delta_j = k \right\}. \tag{1}$$

Consider a specific $\boldsymbol{\delta} \in \Delta_k^n$, each entry δ_j indicates the number of agents that are to be placed in bin j; notice that some δ_j's could be equal to zero, *e.g.*, $n = 2$ and all agents in the first component, zero in the second.

Now we specify the identity of agents in each of the n bins. First, let $2^{\mathcal{A}_k}$ denote the power set of $\mathcal{A}_k$. The set of all subsets of $\mathcal{A}_k$ of size $i \in \{1, \ldots, k\}$ is written as

$$P_i = \left\{ p : p \in 2^{\mathcal{A}_k}, |p| = i \right\}.$$

Let $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_n) \in \Delta_k^n$. The set of all possible assignments of δ_j agents into bin j, for all $j \in \{1, \ldots, n\}$, is given by

$$\mathcal{P}_{\boldsymbol{\delta}} = \left\{ (p_1, \ldots, p_n) : p_j \in P_{\delta_j}, \bigcap_{j=1}^{n} p_j = \emptyset \right\}. \tag{2}$$

The union over all $\boldsymbol{\delta} \in \Delta_k^n$ yields all possible assignments of k agents into n bins

$$\mathcal{P}_k^n = \bigcup_{\boldsymbol{\delta} \in \Delta_k^n} \mathcal{P}_{\boldsymbol{\delta}}. \tag{3}$$

The set of possible criteria—the criterion space—is

$$\mathbb{C} = \bigcup_{n,k \in \mathbb{N}} \mathcal{P}_k^n. \tag{4}$$

Definition 1. *Partition Task*
Given a set of agents $\mathcal{A}_k$, a set of bins $\mathcal{B}_n$, and a criterion $\mathcal{C}_{n,k} \subseteq \mathcal{P}_k^n$, the partition task specified by $\langle \mathcal{A}_k, \mathcal{B}_n, \mathcal{C}_{n,k} \rangle$ is to construct an assignment x of $\mathcal{A}_k$ into $\mathcal{B}_n$ such that $x \in \mathcal{C}_{n,k}$. A Partition Task $\langle \mathcal{A}_k, \mathcal{B}_n, \mathcal{C}_{n,k} \rangle$ is solved by $x \in \mathcal{P}_k^n$ if and only if $x \in \mathcal{C}_{n,k}$. [1]

For example, a marketing company looking to segment the customer pool such that 50% buy product X, 25% buy product Y, and the remaining get the competition's alternative can be described as a *PT* with an agent-set $\mathcal{A}_{1000}$, with 1000 customers in the customer base; a set of bins $\mathcal{B}_3$: one for product X, one for

[1] Although we define the criterion as the enumeration of all valid agent-to-bin arrangements, using a concise description of this set is also an acceptable way to formulate it as long as it can be mapped back to our criterion space. For example, instead of enumerating the $\binom{100}{50}$ ways in which 50 out of 100 agents can be arranged into two bins, one can simply say we require 50% of the agents in each bin.

product Y, and a third for the alternative; and a criterion $\mathcal{C}_{3,1000}$ in which all possible arrangements of 500 customers into bin X, 250 customers into bin Y, and the rest into the alternative are enumerated.

Finally, the problem space $\mathbb{P}$ is given by

$$\mathbb{P} = \{\mathcal{A}_k \times \mathcal{B}_n \times \mathcal{P}_k^n : n, k \in \mathbb{Z}_+\}, \tag{5}$$

the set of all possible combinations of agent-sets, bins, and criteria. Therefore, any element of $\mathbb{P}$ specifies a PT.

It is important to differentiate the means to solve a task from the task itself. A PT is a static problem in the sense that the only thing that matters is whether a partition satisfies the criterion or not, independent of the process undergone in reaching it. In the following section we present a measure for the complexity of partition tasks.

3 Task Complexity

A partition task requires a system arranging an agent-set according to some criterion and its outcome depends on the details of the system's underlying mechanisms. However, it appears generally harder to get every voter to choose the same party than for only a majority to do so and easier to divide a pile of rocks in two than to separate them according to type. In this section we formalize this intuition and present a measure for the complexity of a PT independent of the details of the system trying to solve it. Our purpose is to provide a base line to study and evaluate the performance of different architectures and to allow the comparison of PTs. Formally, we seek to construct a total order[2] over our problem space $\mathbb{P}$.

We refer to a system as a mechanism that induces a probability distribution S over $\mathcal{P}_k^n$, for a given PT, conditional on the information available to that system $I(S)$, i.e., $p_S(c|I(S))$ is the probability that system S assigns to partition c given the available information $I(S)$. This information could include, for example, the initial conditions and the environmental feedback to the system.

We say that a system is un-intelligent if it does not have the capacity to take into account $I(S)$ in its probability assignments; that is, $p_S(c) = p_S(c|I(S))$ for all $c \in \mathcal{P}_k^n$. We also say that a system is unbiased[3] if it assigns equal probability to each $c \in \mathcal{P}_k^n$.

Our complexity measure reflects how likely it is for a system that is non-intelligent and unbiased to solve a given PT. This is formally stated in the following definition.

Definition 2. *Partition Task Complexity Measure*
The PT complexity measure is a function $M : \mathbb{P} \longrightarrow \mathbb{R}_+$ given by

$$M(\langle \mathcal{A}_k, \mathcal{B}_n, \mathcal{C}_{n,k} \rangle) = -\ln\left(\frac{|\mathcal{C}_{n,k}|}{|\mathcal{P}_k^n|}\right). \tag{6}$$

Recall that $|\mathcal{C}_{n,k}|$ is the number of arrangements satisfying the task—all the different ways in which the agent-set can be partitioned so as to accomplish the task—and $|\mathcal{P}_k^n|$ is the number of all the possible arrangements of the agent-set. Note that $M(\langle \mathcal{A}_k, \mathcal{B}_n, \mathcal{C}_{n,k} \rangle) \longrightarrow \infty$ as $|\mathcal{C}_{n,k}| \longrightarrow 0$, which is consistent with the idea of a non satisfiable task being the most complex (when $|\mathcal{P}_k^n| = 0$ our measure is set to zero). Notice the relation between M and the probability of selecting a valid arrangement uniformly at random ($|\mathcal{C}_{n,k}|/|\mathcal{P}_k^n|$).[4]

This quantity is an adequate *a priori* measure of complexity, since it relies only on the definition of the task and takes as reference the most impartial (unbiased) among all possible un-intelligent systems. By being unbiased it is the one with maximum entropy [5] and therefore best represents the available knowledge according to the Principle of Maximum Entropy [17].

[2] A total, antisymmetric, and transitive relation.

[3] Note that an intelligent system could be unbiased.

[4] We take the logarithm for practical reasons since probabilities alone could yield numbers too small to distinguish.

[5] The entropy of system S is defined as $-\sum_{c\in\mathcal{P}_k^n} p_S(c)\ln p_S(c)$. The maximum occurs when $p_S(c) = p_S(c')$ for all c and c'. In this particular case, the entropy of the system is the Boltzmann entropy with its constant set to one, where a satisfied PT corresponds to the macrostate and the elements of the criterion to the microstates of the system (see [5] for a discussion of Boltzmann entropy in agent systems).

A given PT might be readily solved by some systems and impossible for others. In order for our measure to be particular to the PT itself rather than to the system that will attempt to solve it, we have taken a neutral system as reference. In particular, this system has no path dependence in the sense that it assigns equal probability to all partitions independent of any initial conditions. Therefore, this measure of complexity does not consider path dependency, which is consistent with it being a measure of complexity of PTs alone. The complexity of a PT for a specific system could also be of interest, however, such a complexity measure should be path dependent. Although considerations of this sort are beyond the scope of this paper, they could be addressed by changing the reference system to the one of interest.

We define the complexity relation $\precsim_M$ over $\mathbb{P}$ by

Definition 3. *Complexity Relation*
Let $PT_a, PT_b \in \mathbb{P}$,

$$PT_a \precsim_M PT_b \iff M(PT_a) \leq M(PT_b)$$

which reads "PT_b is at least as complex as PT_a".

To see that M is a total order over $\mathbb{P}$ it suffices to note that it is defined for every PT ($|\mathcal{P}_k^n| \geq 0$ and $|\mathcal{C}_{n,k}| \geq 0$) and that it is a real-valued function ($M : \mathbb{P} \to [0, \infty)$).

For example, the task of having exactly 60% occupancy in a restaurant that has 100 potential customers $PT_{rest60} = \langle \mathcal{A}_{100}, \mathcal{B}_2, \mathcal{C}_{2,100} \rangle$ has a criterion $\mathcal{C}_{2,100}$ that enumerates all the possible arrangements of exactly 60 customers at the restaurant with the remaining 40 elsewhere. The complexity of PT_{rest60} is $M(PT_{rest60}) = -\ln\left(\frac{\binom{100}{60}}{2^{100}}\right)$. An occupancy requirement of exactly 80% PT_{rest80} yields $M(PT_{rest80}) = -\ln\left(\frac{\binom{100}{80}}{2^{100}}\right)$, and allowing a range between 55% and 65% of its capacity ($PT_{rest[55,65]}$) has a complexity of $M(PT_{rest[55,65]}) = -\ln\left(\frac{\sum_{i=55}^{65}\binom{100}{i}}{2^{100}}\right)$. Therefore, $PT_{rest[55,65]} \precsim_M PT_{rest60} \precsim_M PT_{rest80}$ since $M(PT_{rest[55,65]}) \leq M(PT_{rest60}) \leq M(PT_{rest80})$.

Knowing the complexity of a PT is much like knowing the expected outcome of a coin flip, it helps to identify special behaviors. Furthermore, having a complexity measure is useful for characterizing the performance and plasticity of different architectures and in guiding their design. In section 5 we provide a series of examples that illustrate the value of having PTs thus classified, but first we consider how well a given system addresses some task.

4 Task Completion

In this section we propose a generalized way to calculate the discrepancy between a PT and any given partition—a distance metric between a partition and a PT. First we specify a distance metric between elements of $\mathcal{P}_k^n$ (partitions) and then between a partition and a set of partitions (between a partition and a criterion).

Let $x, y \in \mathcal{P}_k^n$ and let $d : \mathcal{P}_k^n \times \mathcal{P}_k^n \to \mathbb{R}$, the distance between x and y is given by

$$d(x, y) = \sum_{j=1}^{n} \mid x_j \ominus y_j \mid,$$

where $x_j \ominus y_j$ is the symmetric difference between sets x_j and y_j—the set of elements that are in x_j or y_j but not both[6]. Since $d(x, y)$ is a metric, $\mathcal{P}_k^n$ is a metric space.

The distance between a partition $x \in \mathcal{P}_k^n$ and a set of partitions $Y \subseteq \mathcal{P}_k^n$ is the minimum of all distances from each point in Y to x, *i.e.*

$$d(x, Y) = \min_{y \in Y} d(x, y).$$

[6] $x_j \ominus y_j = (x_j - y_j) \cup (y_j - x_j) = (x_j \cup y_j) - (x_j \cap y_j)$

Finally, the discrepancy between a PT $p = \langle \mathcal{A}_k, \mathcal{B}_n, \mathcal{C}_{n,k} \rangle$ and a partition $x \in \mathcal{P}_k^n$ is given by

$$D(x, p) = d(x, \mathcal{C}_{n,k}). \tag{7}$$

The ability to quantify the distance between agent-set configurations (partitions) and PTs will be useful for quantifying the error of a system's attempt at addressing it. In the following section we define one such error using this notion of distance and present a series of examples that illustrate the usefulness of this definition along with those of PTs and their complexity.

5 Applications

In this section we provide some examples as evidence of the usefulness of our definitions and measures. For this purpose we created an agent-based simulation to assess their value empirically. We first summarily describe the details of the simulation and then present some interesting results.

5.1 An Agent-based model

Agent-based models are a useful resource for studying phenomena resulting from the interactions of agents and have been widely used to illustrate concepts regarding agent-sets and their mechanisms [28, 3, 20, 30]. In this paper we present a model in which agents take discrete decisions regarding which bin to occupy. These decisions are influenced by the (social) network in which the agents are embedded. We perform simulations that are divided into events within which agents receive input from their neighbors and the environment, update their internal state, and choose a bin. The following list explains the main components of the simulation.

Environment The environment evaluates agents decisions and keeps track of the distribution of agents in bins. This distribution is represented by a vector of numerical values indicating how many agents are in each bin—e.g., the number of agents inside El Farol thus far. Both the criterion and the distribution of agents in bins are kept from the agents. An event starts with all bins empty and ends after all agents have chosen a bin, that is, once a partition is achieved. At the end of an event the environment provides each agent with a binary signal depending on the bin they chose, agents with the same choice receive the same signal, given by

$$signal(k) = \begin{cases} 1, & \text{if the number of agents in bin } k \text{ is short of the number established by the task} \\ -1, & \text{otherwise} \end{cases}$$

Agent Agents are the principal actors of our simulation and have the following characteristics:

- At the beginning of the simulation each agent is assigned, uniformly at random, a default bin, which we refer to as its predisposition.
- Each agent has a fixed set of neighbors and a weight assigned to each of them. The neighbor relationship is symmetrical while the weights assigned between neighbors may differ. Agents keep an additional weight for their predisposition. Weights are initialized at a random value between -1 an 1.
- Each agent makes a single discrete decision per event—a bin number. An agent chooses a bin by querying its neighbors and weighing their decisions (along with its predisposition). The decision of agent i is computed as:

$$D(i) = \underset{k}{\operatorname{argmax}} \quad B_i(k),$$

–Repeat
  –Set the task vector to zero
  –Randomly select an agent (*Agent*)
  –Repeat
    –Query *Agent*'s neighbors and compute its decision
    –Report *Agent*'s decision to the environment and update the partition vector
    –Select another unqueried agent (*Agent*) according to the breath-first order
  –Until all agents have had their turn
  –Update each agent's weights according the environment's signal. Increment the number of events by 1
–Until the target number of events has been reached

Figure 1: Simulation steps

where $B_i(k)$ is the aggregate weight of agent i's neighbors whose decision is bin k: $B_i(k) = \sum w_{i,j}$ such that $D(j) = k$. Ties are settled by randomly choosing between the competing options. See Sect. 5.2.3 for an example of the consequences of resolving ties differently.

- At the end of each event agents update their weights using the signal provided by the environment:

$$w_{i,j} = \begin{cases} w_{i,j}, & \text{if } D(i) \neq D(j) \\ sgn(w_{i,j}) \cdot \min\{|w_{i,j}| + \eta \cdot signal(D(i)), 1\}, & \text{if } D(i) = D(j), \end{cases}$$

where $w_{i,j}$ denotes the weight assigned by agent i to agent j and η is a learning parameter (set to 0.01 throughout).

- Each agent reports its most recent decision when queried, *i.e.*, its decision for the current event if it has already made a choice or its decision for the previous event (or its predisposition during the first event) if it has not. Agents make their decisions sequentially following a breath-first order starting from a randomly chosen agent. If the network is not connected the procedure is repeated until all agents have made a choice.

Network For any given run, an agent has a fixed set of neighbors to which it is connected. In this paper we experiment with the Erdös Rényi (ER) model of network topology [14]. ER networks are characterized by a single parameter that indicates the probability of including any given link in the graph.

The agent-set, the agent's characteristics and the network make up the system that will address the given PTs. The simulations go through the steps listed in Fig. 1.

Each simulation event (iteration) concludes with a specific partition of the agent-set. At the end of the simulation there are I data points—one for each event—from which we compute the average error (at each event) in addressing task $\langle \mathcal{A}_k, \mathcal{B}_n, \mathcal{C}_{n,k} \rangle$, given by

$$Error(\langle \mathcal{A}_k, \mathcal{B}_n, \mathcal{C}_{n,k} \rangle) = \frac{1}{I} \sum_{t=1}^{I} d(x_t, \mathcal{C}_{n,k}) \tag{8}$$

where $x_t \in \mathcal{P}_k^n$ is the specific partition corresponding to the t^{th} event (iteration).

The examples presented in Sect. 5.2 report the results of running our simulation on different tasks. Each data point represents the average of eqn. 8 over 30 simulations—each of the 30 simulations varies only in the random values of the parameters used to run it.

5.2 Experiments

In this section we provide four simulated scenarios using the agent-based model presented above. The intention here is not to assess the adequateness of the model—agent's decision models, network topology, etc., but rather to highlight the potential value and usefulness of our definitions and measures. Notice that changing the model may change the data but not the approach to interpret it.

5.2.1 Measures as a tool to evaluate a model's performance

Consider the restaurant occupancy problem once again. In Sect. 3 we discussed how different requirements on the occupancy of the restaurant yield different complexities according to our measure; in particular, that the further away a problem is from requiring exactly 50% occupancy, the harder it gets. Figure 2 shows the results of our simulation for population sizes of 100. The plots show how the error in satisfying the given partition task is consistent with the task's complexity measure, irrespective of the network connectivity parameters. Having the ability to measure problem complexity enables us to evaluate, for instance, the effect of different network connectivities in achieving a particular task.

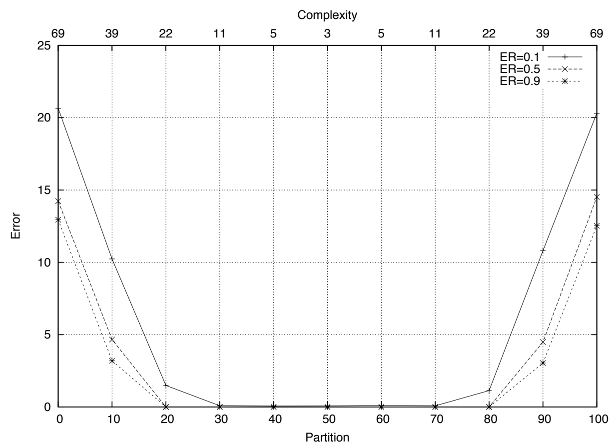

Figure 2: Average error (see eqn. 8) for different Erdös Rényi networks across several task complexities. The bottom x-axis indicates the size of the partition and top x-axis the complexity of the task according to eqn. 6.

5.2.2 Measures as a tool of policy design: Choosing the number of partitions

Imagine an industrial city with many car factories that need to choose a warehouse in which to store their vehicles. Cars are temporarily kept in warehouses before being shipped to their selling agencies. The factories produce a total of 100 cars per day and each car is stored for one day in a warehouse.

A profit maximizing agent wants to invest in the warehouse business and faces three options that have the same cost: to rent a warehouse with a capacity for 87 cars (PT_{WH87}), to rent two warehouses with capacities for 50 and 37 cars ($PT_{WH50,37}$), or to rent three warehouses with capacities for 50, 25 and 12 cars each ($PT_{WH50,25,12}$). Suppose that companies pay per car stored and that having to turn away a costumer, because of insufficient room, represents some cost—they may not come back to that particular warehouse. Given limited knowledge about how companies decide, which option is the wisest investment?

The complexity measures (see eqn. 6) for the three scenarios are presented in the following table:

$M(PT_{WH87})$	$-\ln\left(\frac{\binom{100}{87}}{2^{100}}\right)$	32.8
$M(PT_{WH50,37})$	$-\ln\left(\frac{\binom{100}{50}\binom{50}{37}}{3^{100}}\right)$	16.5
$M(PT_{WH50,25,12})$	$-\ln\left(\frac{\binom{100}{50}\binom{50}{25}\binom{25}{12}}{4^{100}}\right)$	23.9

The value of M for $PT_{WH50,37}$ is the lowest and therefore $PT_{WH50,37} \precsim_M PT_{WH87}$ and $PT_{WH50,37} \precsim_M PT_{WH50,25,12}$. It is wisest to invest in the two warehouse option. Figure 3 shows the results of a simulation for different connectivities of the agent network. The task with the lowest average error is indeed $PT_{WH50,37}$ followed by $PT_{WH50,25,12}$ and PT_{WH87}. Notice, however, that there is an exception when agents are not connected at all (an ER connectivity of zero). We address this in the following example.

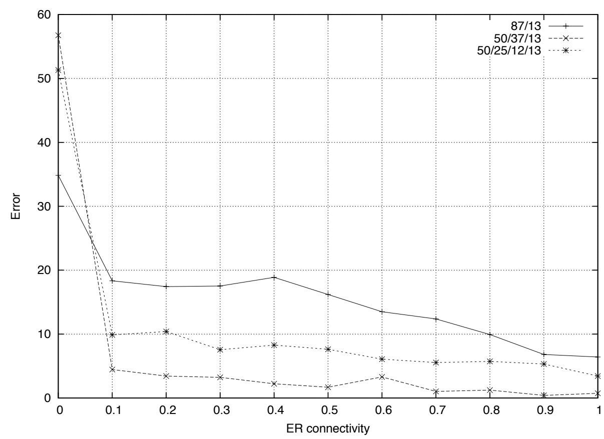

Figure 3: Average error (see eqn. 8) for three distinct tasks across different network connectivities.

5.2.3 Measures as a tool to notice unexpected behaviors

The complexity measure introduced in Sect. 3 is meant to provide a reference point against which a particular agent-set's performance can be evaluated and which can help identify interesting or unusual behavior—much like knowledge of the expected behavior of a fair coin helps to identify a biased one.

In the previous example we presented three partition problems that had a particular ordering with regards to their complexity. The experiments reported in Fig. 3 show that the simulation results are consistent with this ordering except in the case where agents have no connections with each other where they do not consult any other agents to make a choice. This inconsistency is an indication of noteworthy behavior, not necessarily wrong but worth explaining.

A close inspection of the model's implementation revealed that agents by themselves do not have complete knowledge of all their options—the existence of all available warehouses (as ties where not initially settled randomly)—and that this information percolates throughout the network once agents are connected. This fact explains the departure from expected behavior (once knowledge of all options is provided to each isolated agent the curves cease to cross) and would have been difficult to notice without the baseline provided by the complexity measure.

5.2.4 Measures as a tool of policy design: Choosing the size of the partition

The local government of a small town of 1,000 people is planning on building a public swimming pool. Suppose that the cost of constructing the pool is the same irrespective of the capacity and that the government must incur in a cost that depends on the number of workers it hires to service the pool, which does depend on its size—workers are hired on a per year basis.

The townspeople's preferences are such that:

- They enjoy the pool if it is at most 50% full
- They dislike the government spending their taxes on unnecessary expenses; that is, seeing that the government hired more workers than the ones needed to service the pool on a particular day
- They dislike receiving a bad service at the pool that is caused by a lack of workers in relation to the attendees on that day

The government wishes to select a pool capacity such that the number of expected attendees is maximized subject to not wasting resources—hiring more workers than the ones needed. Given the pool capacity (Cap), this problem can be formulated as a partition problem $PT_{pool} = \langle \mathcal{A}_{1000}, \mathcal{B}_2, \mathcal{C}_{2,1000} \rangle$ where $\mathcal{A}_{1000}$ represents the town's inhabitants, $\mathcal{B}_2$ represents the two options of attending the pool and not attending the pool, and $\mathcal{C}_{2,1000}$ all the possible agent arrangements in which $0.5 * Cap$ of the people attend the pool and the remaining $1000 - 0.5 * Cap$ go elsewhere. Each value of Cap defines a distinct partition problem. The government's job is thus to find the right partition problem for its constituency.

Assuming that our agent-based model captures correctly how agents behave and that the social network in which they are embedded can be modeled by an ER network whose parameter we ignore, we can run a simulation to determine the right size for the pool.

Figure 4 presents the dispersion of the average error for different attendance objectives (partition sizes of the agent-set) and for several parameters of the ER network—notice that setting a pool capacity implicitly fixes the attendance objective due to the townspeople's preferences.

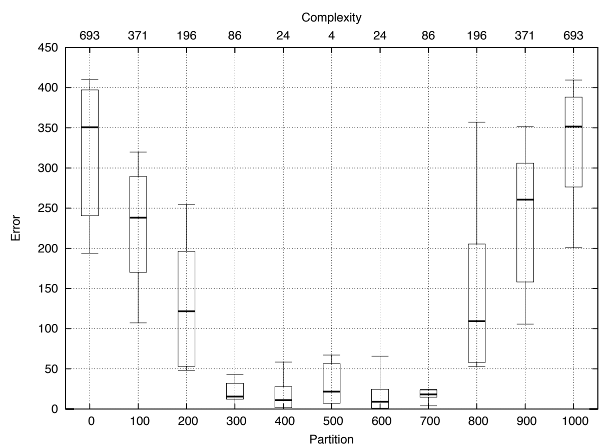

Figure 4: Dispersion of the average error (see eqn. 8) across network connectivities. The bottom x-axis indicates the size of the partition and top x-axis the complexity of the task according to eqn. 6.

Turnouts between 300 and 700 have an error that is close to zero across all connectivity parameters, whilst the rest have errors that reach magnitudes as high as 400, i.e., the expected outcome can vary in up to 400 attendantes. Selecting a pool capacity such that less than 300 or more than 700 people visit the pool will result in drastic fluctuations in attendance and, therefore, in many days with too many workers and many days with too few. On the other hand, having the objective number of townspeople be between 300 and 700 will minimize unnecessary expenditures and uneasiness due to bad service. Given that government's objective is to maximize the number of people it pleases, it should build a pool that can hold 1,400 of its constituents, since this translates into 700 happy townspeople per day with little overspending.

6 Conclusions

The present work was motivated by a particular kind of task that a group of agents often encounteres, one in which the group ends up partitioned into a set of discrete subgroups; for instance, agents chosing where to park, where to have dinner, or a geologist choosing how to sort his findings. In this paper we defined a class of problems, named *Partition Tasks*, which encompass these kinds of actitivities, and a measure that allows for the comparison of problems in terms of their *a priori* complexity. Moreover, having problems thus defined made it possible to quantify the distance between a problem and a specific partition and to establish the error of an agent-set's attempt at solving a PT.

The construction of the problem space $\mathbb{P}$ here presented not only provided the framework under which we were able to properly define a PT along with a measure for its complexity, but also sets the context under which a series of interesting questions can be formally addressed. For example, given a definition of equivalency between PTs, can a PT always be transformed in such a way that the resulting PT is equivalent to the first? If so, can a PT always be transformed in such a way that the resulting PT is not only equivalent but also at most as complex as the original? It is possible that these questions will not have an affirmative response when considering $\mathbb{P}$ as a whole. Nonetheless, one could use the equivalence relation $\sim_c$ to partition $\mathbb{P}$ (into equivalence classes or *problem classes*) and pose the same questions considering only certain cells of the partition.

The aim of the measures and definitions here introduced is to provide researchers with useful tools for analyzing the properties and mechanisms of agents engaged in a Partition Task, and as a first step into the study of the tasks themselves.

In the second half of the paper we presented a series of examples that illustrate how having a precise definition, a complexity measure, and a performance error, are useful in studying how agents behave and in designing PTs. For example, we showed how these measures and definitions, together with a computational model, help to analyze the effect of a social network's topology on the resolution of a task and how this knowledge can be used, in turn, to design the proper task for a group of agents so as to fullfill some given (exogenous) objective.

Acknowledgments

The authors gratefully acknowledge the support provided by Asociación Mexicana de Cultura, A. C. and Conacyt grant 119162. We would also like to thank Martín Puchet, Drew Fudenberg, and two anonymous reviewers whose comments greatly helped improve the presentation and content of this paper.

References

[1] Daron Acemoglu, Munther A. Dahleh, Ilan Lobel, and Asuman Ozdaglar. Bayesian learning in social networks. NBER Working Papers 14040, National Bureau of Economic Research, Inc, May 2008.

[2] Reka Albert and Albert-Laszlo Barabasi. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47, 2002.

[3] W. Brian Arthur. Inductive reasoning and bounded rationality: The El Farol problem. *Amer. Econ. Rev.*, 84:406–411, May 1994.

[4] V. Bala and S. Goyal. Learning from neighbours. *Review of Economic Studies*, 65:595–621, 1998.

[5] G. Baldassarre. Self-organization as phase transition in decentralized groups of robots: A study based on boltzmann entropy. In Mikhail Prokopenko, editor, *Advances in Applied Self-organizing Systems (Advanced Information and Knowledge Processing)*, pages 127–146. Springer-Verlag, Berlin, 2008.

[6] A. Barabasi, H. Jeong, Z. Neda, E. Ravasz, A. Schubert, and T. Vicsek. Evolution of the social network of scientific collaborations, 2001.

[7] A. Barrat, M. Barthelemy, R. Pastor-Satorras, and A. Vespignani. The architecture of complex weighted networks. *PNAS*, 101(11):3747–3752, March 2004.

[8] M. Barthélemy, A. Barrat, R. Pastor-Satorras, and A. Vespignani. Dynamical patterns of epidemic outbreaks in complex heterogeneous networks. *J. Th. Bio.*, 235:275, 2005.

[9] Scott A. Boorman. A combinatorial optimization model for transmission of job information through contact networks. *Bell Journal of Economics*, 6(1):216–249, Spring 1975.

[10] Damien Challet and Yi-Cheng Zhang. Emergence of cooperation and organization in an evolutionary game. *Physica A*, 246(407), 1997.

[11] Michael Suk-Young Chwe. Communication and coordination in social networks. *Review of Economic Studies*, 67(1):1–16, January 2000.

[12] Stephen A. Cook. The complexity of theorem proving procedures. In *3rd Annual ACM Symposium on Theory of Computing*, pages 151–158, New York, 1971.

[13] Glenn Ellison and Drew Fudenberg. Rules of thumb for social learning. IDEI Working Papers 17, Institut d'?conomie Industrielle (IDEI), Toulouse, 1992.

[14] P. Erdős and A. Rényi. On random graphs. I. *Publ. Math. Debrecen*, 6:290–297, 1959.

[15] Michael R. Garey and David S. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness*. W.H. Freeman & Company, San Francisco, 1978.

[16] E. T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106(4):620–630, 1957.

[17] Seth Lloyd. Measures of complexity: A non-exhaustive list. *IEEE Control Systems Magazine*, august 2001.

[18] John H. Miller and Scott E. Page. The standing ovation problem. *Complexity*, 9:8–16, April 2004.

[19] Melanie Mitchell. *Complexity: A Guided Tour*. Oxford University Press, 2009.

[20] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial optimization: algorithms and complexity*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1982.

[21] Christos M. Papadimitriou. *Computational complexity*. Addison-Wesley, Reading, Massachusetts, 1994.

[22] R. Pastor-Satorras and J. Wagensberg. The maximum entropy principle and the nature of fractals. *Physica A: Statistical and Theoretical Physics*, 251(3-4):291–302, March 1998.

[23] Romualdo Pastor-Satorras and Alessandro Vespignani. Epidemic dynamics and endemic states in complex networks. *Phys. Rev. E*, 63:066117, 2001.

[24] Mikhail Prokopenko, Fabio Boschetti, and Alex J. Ryan. An information-theoretic primer on complexity, self-organization, and emergence. *Complexity*, 15(1):11–28, 2009.

[25] Erzsebet Ravasz and Albert-Laszlo Barabasi. Hierarchical organization in complex networks, September 2002.

[26] Thomas C. Schelling. *Micromotives and Macrobehavior*. W. W. Norton, September 1978.

[27] Leigh Tesfatsion and Kenneth L. Judd. *Handbook of Computational Economics, Volume 2: Agent-Based Computational Economics (Handbook of Computational Economics)*. North-Holland Publishing Co., Amsterdam, The Netherlands, The Netherlands, 2006.

[28] Levent Ülkü. Optimal combinatorial mechanism design. Technical Report 09-03, Centro de Investigación Económica, Instituto Tecnológico Autónomo de México, march 2009. Discussion Paper Series.

[29] Leslie G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, pages 189–201, 1979.