Production structures as Complex Adaptive Systems

L. Monostori¹,²(1), B.Cs. Csáji¹

¹Computer and Automation Research Institute, Hungarian Academy of Sciences, Budapest, Hungary
²Department of Manufacturing Science and Technology, Budapest University of Technology and Economics, Budapest, Hungary

Complexity at every level of manufacturing / 1
The importance of complexity

“The need to be able to measure the complexity of a problem, algorithm or structure, and to obtain bounds and quantitative relations for complexity arises in more and more sciences: besides computer science, the traditional branches of mathematics, statistical physics, biology, medicine, social sciences and engineering are also confronted more and more frequently with this problem.

In the approach taken by computer science complexity is measured by the quantity of computational resources (time, storage, program, communication) used up by a particular task.” (Lovász and Gács, 1999)

We will consider computational (or algorithmic) complexity, Kolmogorov (or algorithmic information) complexity, information complexity (information entropy), topological (graph) complexity, and, finally, Complex Adaptive Systems (CASs).
Computational Complexity - Time

Time complexity refers to a function describing how much time it will take an algorithm to solve a problem, based on the size of its input. We say that a problem has time complexity at most $f(n)$, if it can be solved by a Turing machine with time demand at most $f(n)$. Note that there may not be a unique best Turing machine, hence the asymptotic phrase “at most” in the definition.

Example: the traveling salesman problem (the problem to find a Hamiltonian circle with minimum cost in a weighted graph) can be solved in time $O(n^22^n)$, where $n$ is the size of the network to visit.

Classes of time complexity: e.g., P and NP type problems. (It is not known whenever $P$ differs from $NP$, however, most mathematicians believes that it is so.)

Computational Complexity - Space

The storage requirement of solving a problem (space complexity) is defined similarly as time complexity.

Conversely, Savitch’s theorem asserts that $P$-Space = $NP$-Space, thus, nondeterminism does not really make a difference regarding space complexity, namely polynomial storage should be enough.
Computational Complexity - Communication

Communication (and its costs) becomes important if distributed systems are considered. With many algorithmic and data processing problems, the main difficulty is the transport of information between different processors. The key concept regarding communication is the protocol. Communication problems can be completely coded by communication matrices and the concept of protocol can be defined by these matrices. Communication complexity of a matrix is defined as the smallest possible time requirement of all protocols solving it.

Kolmogorov Complexity

In 1960s Solomonoff, Kolmogorov and Chatin (independently) introduced a complexity concept which is often called algorithmic information complexity. Given a particular universal Turing-machine, the Kolmogorov-complexity of a (bit)string (description) is the length of the shortest program that generates the description and halts. In other words Kolmogorov defined the complexity of a structure as the length of its shortest description (namely, on a TM).

The Kolmogorov complexity of the Mandelbrot set fractal is almost zero.
Information Complexity

The concept of entropy in information theory describes how much information (or disorder) there is in a signal or event. An intuitive understanding of information entropy relates to the amount of uncertainty about an event associated with a given probability distribution.

Shannon defines entropy (Shannon, 1948) in terms of a discrete random variable $X$, with possible states (or outcomes) $x_1, \ldots, x_n$ as

$$H(X) = \sum_{i=1}^{n} p(x_i) \log \left( \frac{1}{p(x_i)} \right) = -\sum_{i=1}^{n} p(x_i) \log p(x_i),$$

where $p(x_i)$ denotes the probability of the $i$-th outcome of $X$, the basis of log is 2 and with the convention that $0 \cdot \log(0) = 0$.

The definition above satisfies Shannon’s desiderata that entropy should be a (complexity) measure of the average information content associated with a random outcome.
Topological (graph) Complexity - Entropy

Using the idea of entropy, the measure of vertex degree distribution is defined as

\[ I_v(G) = \sum_{i=1}^{n} a_i \log_2 a_i \]

where \( a_i = \sum_{j=1}^{n} a_{ij} \).

This measure has the property that it increases with the connectivity and with other complexity factors, such as, the number of branches, cycles, cliques, etc.

Topological (graph) Complexity - Subgraphs

The topological complexity of a graph can also be expressed by the total number of subgraphs.

The number of all subgraphs containing two edges has an important role in chemistry and it has a special name, called Platt’s index.

The number of subgraphs containing three edges are called Gordon-Scantlebury index.

Later, total subgraph count was introduced to measure the complexity of a graph, where subgraphs of all sizes (even the graph itself) was counted. In case of large networks, often subgraphs containing less then \( x \) (e.g. \( x = 3 \)) edges are counted in practice, in order to avoid combinatorial explosion.

Using subgraphs, one can also define the overall connectivity index of a graph as a sum of all adjacencies of all \( k \)-th order subgraphs.
Topological (graph) Complexity - Walks

An alternative way to define the topological complexity of a graph is to count all paths from any vertex to any other vertex, called total walk count.

Note that the length of a walk is limited by the number of vertices, \( n \), and there could be only finite number of possible walks.

A natural extension of this measure is when (edge or vertex) weights are also taken into account.

Complex adaptive systems (CAS) / 1


Study the structures and dynamics of systems and the question, how the adaptability of systems creates complexity.

CAS: a multi agent system in which “a major part of the environment of any given adaptive agent consists of other adaptive agents, so that a portion of any agent’s efforts at adaptation is spent adapting to other adaptive agents”.

Complex adaptive systems (CAS) / 2

The most remarkable phenomenon exhibited by CAS is the emergence of highly structured collective behaviour over time by the interaction of simple subsystems, usually, without any centralised control.

Both the CAS and its environment simultaneously co-evolve in order to maintain themselves in a state of quasi-equilibrium, i.e., on the edge of chaos (Waldrop, 1992)

In designing CAS, non-linear phenomena, incomplete data and knowledge, a combinatorial explosion of states, dynamic changes in environment and the frame problem are some notable examples of difficulties to be faced.

The central question is realising an artifactual system that achieves its purpose in unpredictable conditions. It is difficult to approach problems like this by using only existing principles, such as analysis and determinism.

Classification of Emergent Synthesis problems (Ueda, K.)

Class I: Problem with complete description: if all the information concerning the environment and specification are given, then the problem is completely described. However, it is often difficult to find an optimal solution.

Class II: Problem with incomplete environment description: the specification is complete, but the information on the environment is incomplete. Since the problem is not wholly described in this case, it is difficult to cope with the dynamic properties of the unknown environment.

Class III: Problem with incomplete specification: not only the environment description but also the specification is incomplete. Problem solving, therefore, has to start with an ambiguous purpose, and the human interaction becomes significant.

A CAS approach to production scheduling

To design a fast, adaptive and approximate method that can effectively perform closed-loop control on resource allocation problems in uncertain, dynamic and large-scale environments.
Reinforcement learning

The environment evolves by probabilistically occupying a finite set of discrete states. For each state there is a finite set of possible actions that may be taken. Every time the learning system takes an action, a certain reward is incurred. States are observed, actions are taken, and rewards are incurred at discrete time steps.

---

Core idea - Iterative probing

Resource allocation is formulated as a stochastic shortest path problem. The system makes stochastic iterative probes in the search space (the space of all possible schedules). From the performance measures of the achieved schedules, the system modifies the distribution of the random variable that gives the schedules. The aim is to learn a distribution in which good schedules have high probability. The system learns a distribution like this by neurodynamic programming.
The building of schedule can be represented as a tree (graph) in a natural way. The nodes represent (partial) schedules. The edges represent consigning an operation to a machine that can process that operation. The root of the tree is the empty schedule. The leaves represent complete schedules.

In each iteration the system builds a schedule by moving from the root to a leaf. At each node the system has to make a decision which way to go. The system learns the decision probabilities with neurodynamic programming (reinforcement learning + artificial neural networks).
Three levels of learning

Pseudo-code

1. Simulate a state-action trajectory from the starting state using the model and the policy generated by the Boltzmann formula.
2. After a terminal state is reached, back propagate the achieved final performance and update the Q estimates of the visited states according to Q-learning.
3. Fit a smooth approximating function to all of the available Q estimates by n-SVR type Kernel Regression (Gaussian kernels).
4. Decrease the temperature, except there were changes and disturbances in the system.
5. Increase iteration counter and, unless some terminating conditions are met, go back to (1).

Benchmark flexible job-shop problems

Hurink’s benchmark dataset: 6-30 jobs (30-225 tasks) and 5-15 machines

Hard problems where standard dispatching rules or heuristics perform poorly on them

<table>
<thead>
<tr>
<th>dataset</th>
<th>parallel</th>
<th>1000 iter.</th>
<th>5000 iter.</th>
<th>10000 iter.</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdata</td>
<td>1</td>
<td>8.54 %</td>
<td>5.69 %</td>
<td>3.57 %</td>
</tr>
<tr>
<td>edata</td>
<td>1.2</td>
<td>12.37 %</td>
<td>8.03 %</td>
<td>5.26 %</td>
</tr>
<tr>
<td>rdata</td>
<td>2</td>
<td>16.14 %</td>
<td>11.41 %</td>
<td>7.14 %</td>
</tr>
<tr>
<td>vdata</td>
<td>5</td>
<td>10.18 %</td>
<td>7.73 %</td>
<td>3.49 %</td>
</tr>
<tr>
<td>average</td>
<td>2.3</td>
<td>11.81 %</td>
<td>8.21 %</td>
<td>4.86 %</td>
</tr>
</tbody>
</table>
Machine breakdown (t=100)

New job entered the system (t=100)
New machine is available (t=100)

20 machines
25 orders
4-7 operations per order

A job is cancelled (t=100)

20 machines
25 orders
4-7 operations per order
Coll-Plexity
Collaborative Complexity, Collaborations as Complex Systems

Coordinator:
RWTH Aachen - Lab. For Machine Tools and Prod. Engineering (WZL)

Partners:
Global Research and Financing (Israel)
Universität St. Gallen, ITEM-HSG
MTA SZTAKI
Schiesser AG
Virtuelle Fabrik AG

Collaborative Enterprises as CASs

Complex Adaptive Systems constitute a natural framework to model collaborative enterprises and to investigate complexity drivers in it. An enterprise can be associated with an agent that interacts with other agents in an uncertain and changing environment. Multi-agent based supply chain management models are already available. After a model is defined, computer-based simulation can be used to evaluate and test the system. Our (scientific) results should be verified or confirmed and in a lot of cases simulation constitutes the only available way (of our days) to investigate and analyse extremely complex systems, such as collaborative enterprises.
Models and potential complexity measures

(1) Environment model: sequence of random variables (time series $X_1, X_2, \ldots, X_t, \ldots, X_n$) Potential complexity measure: Information complexity (entropy)

(2) Collaboration model: complex adaptive systems Potential complexity measure: computational complexity (measure of applied resources)

(3) Enterprise network model: network and graph theory Potential complexity measure: topological (graph) complexity (adjacency, components, walks)

Conclusions

Production systems are really complex systems exposed to changes and disturbances

The CAS-approach seems to be an appropriate one


Agent-based systems for manufacturing, by Monostori, Váncza and Kumara, CIRP Key-note paper, 2006:

The further evolution of multi-agent systems and manufacturing will probably proceed hand in hand: the former can receive real challenges from the latter, which, in turn, will have more and more benefits in applying agent technologies, presumably together with well-established or emerging approaches of other disciplines.