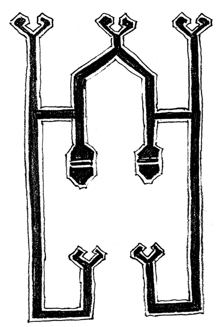
Notes on “Notes on the Synthesis of Form”

Richard P. Gabriel



Ron Goldman introduced me to the work of Christopher Alexander around 1990, maybe a little before. I started by reading "The Timeless Way of Building" and "A Pattern Language." I read his work in this order (dates indicate date of publication):

Order I Read Alexander

1. 1979: The Timeless Way of Building
2. 1977: A Pattern Language
3. 1975: The Oregon Experiment
4. 1987: A New Theory of Urban Design
5. 1985: The Production of Houses
6. 1973: The Grass Roots Housing Process
7. 1968: The Bead Game Conjecture
8. 1993: A Foreshadowing of 21st Century Art
9. 1965: A City is Not a Tree
10. 1966: BART: The Bay Area Takes a Million Dollar Ride
11. 1991: The Perfection of Imperfection
12. 1997: The Nature of Order, The Process of Creating Life (Draft; & the final version in 2007)
13. 1965: Notes on the Synthesis of Form [read in 2015]
14. 2012: Battle for the Life and Beauty of the Earth

Of the fourteen major works of his I read, "Notes" was the thirteenth, and I first read it in 2015.

Before reading "Notes," I considered Alexander a mystic & an artistic architect and builder who believed beauty was objective, how people lived in the built world was important, and iterative design and development was the only way to do things. The ideas that struck me were QWAN—the Quality Without a Name—habitability, piecemeal growth, and generativity—a form of emergence. Patterns seemed an OK idea for helping designers get to QWAN. Later came wholeness, life, centers, and the "Fifteen Properties."

The Fifteen Properties

Levels of Scale	Good Shape	Roughness
Strong Centers	Local Symmetries	Echoes
Boundaries	Deep Interlock and Ambiguity	The Void
Alternating Repetition	Contrast	Simplicity and Inner Calm
Positive Space	Gradients	Not-Separateness

A vivid quote I recall is this one:

> *In order for* [a] *building to be alive, its construction details must be unique and fitted to their individual circumstances as carefully as the larger parts. . . . The details of a building cannot be made alive when they are made from modular parts.*

Let me repeat the last sentence:

> *The details of a building cannot be made alive when they are made from modular parts.*

To give you a sense of Alexander's main ideas, I'll read parts of some quotes from Alexander's later work. About QWAN from "Timeless Way":

> *To reach the quality without a name we must then build a living pattern language as a gate. This quality in buildings and in towns cannot be made, but only generated, indirectly, by the ordinary actions of the people, just as a flower cannot be made, but only generated from the seed.*

About wholeness and "life":

> *What we call "life" is a general condition which exists, to some degree or other, in every part of space: brick, stone, grass, river, painting, building, daffodil, human being, forest, city. And further: The key to this idea is that every part of space—every connected region of space, small or large—has some degree of life, and that this degree of life is well-defined, objectively existing, and measurable.*

About God:

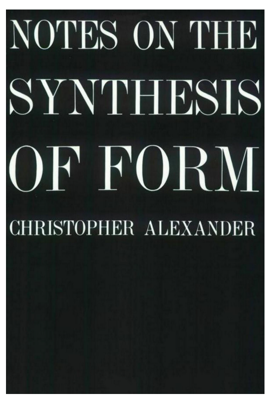
> *. . . we're trying to do something that no one else has ever tried to do in the 20th century . . . make God appear in the middle of a field.*

About sadness from "Nature of Order":

> *The I, that blazing one, is something which I reach only to the extent that I experience, and make manifest, my feeling. What feeling, exactly? What exactly am I aiming for in a building, in a column, in a room? How do I define it for myself, so that I can feel it clearly, so that it stands as a beacon to steer me in what I do every day?*
>
> *What I aim for is, most concretely, sadness. I try to make the volume of the building so that it carries in it all feeling. To reach this feeling, I try to make the building so that it carries my eternal sadness. It comes, as nearly as I can in a building, to the point of tears.*

Reading "Notes" I thought: Who wrote this???
How did the "Notes" Alexander become the "Blazing One" Alexander?

I first read "Notes" around 2015; the copy I had was a sweet first edition I found at Powell's Bookstore in Portland Oregon. Being a first edition is important: it did not contain the disclaimer that the 1971 paperback edition had. We'll see that at the end of this presentation.



In it Alexander presented a requirements-based, modularity discovery process based on a series of sometimes obvious, sometimes strange mathematical ideas.

Notes on the Synthesis of Form

- CA's PhD Dissertation
- misfits
- selfconscious / unselfconscious processes

Despite good material on misfits and the unselfconscious and selfconscious design approaches, the formal, mathematical method surprised me—and not in a happy way.

“Notes” and Computer Scientists

1. they love this book
2. because design as removing misfits sounds great
3. because CA demonstrates that some of it can be treated *mathematically*
4. except #3 is wrong, or at least, CA did a lousy job

These ideas prompted Peter Naur to declare the following at the super-famous 1968 NATO conference on Software Engineering:

> *The approach suggested by Christopher Alexander in his book: “Notes on the Synthesis of Form,” is to make a tree structure of the decisions, so that you start by considering together those decisions that hang most closely together, and develop components that are sub-systems of your final design. Then you move up one step and combine them into larger units, always based on insight, of some kind, as to which design decisions are related to one another and which ones are not strongly related. I would consider this a very promising approach.*

In “Notes” Christopher Alexander presents a detailed example problem: the redesign of a village in India of some 600 people to better suit present and future demands.

Indian Village

- an agricultural village of six hundred people
- to be reorganized to make it fit present and future conditions developing in rural India
- 141 “misfit” variables
- ∼1400 interactions among them

The essence of design presented in “Notes” is to minimize the number of misfits. In the Indian Village scenario the essential problem was to take a set of design requirements (141 of them) . . .

Here Are Some of Them

33. Fertile land to be used to best advantage.
34. Full collection of natural manure (animal and human).
35. Protection of crops from insects, weeds, disease.
36. Protection of crops from thieves, cattle, goats, monkeys, etc.
37. Provision of storage for distributing and marketing crops.
38. Provision of threshing floor and its protection from marauders.

. . . and a set of interactions among them (about 1400 of them), and then partition the requirements into groups that represent coherent design subtasks or components.

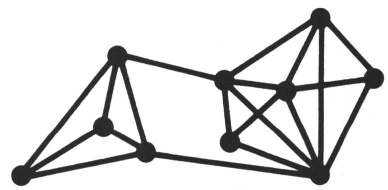
Here Are Some of Them

1 interacts with 8, 9, 12, 13, 14, 21, 28, 29, 48, 61, 67, 68, 70, 77, 86, 101, 106, 113, 124, 140, 141
2 interacts with 3, 4, 6, 26, 29, 32, 52, 71, 98, 102, 105, 123, 133
3 interacts with 2, 12, 13, 17, 26, 76, 78, 79, 88, 101, 103, 119
. . .
141 interacts with 1, 8, 9, 12, 13, 14, 15, 18, 21, 24, 48, 61, 62, 63, 65, 66, 68, 86, 89, 93, 95, 96, 100, 112, 115, 130, 138, 140

He approached the problem obliquely, starting with the (important) idea of “misfits” and what makes for a “good fit” between a design and its context. He writes:

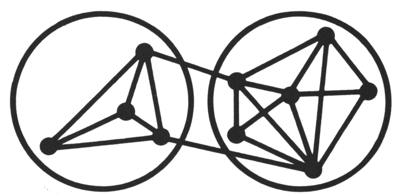
> *. . . the concept of good fit, though positive in meaning, seems very largely to feed on negative instances; it is the aspects of our lives which are obsolete, incongruous, or out of tune that catch our attention.*
>
> *We should find it almost impossible to characterize a house which fits its context. Yet it is the easiest thing in the world to name the specific kinds of misfit which prevent good fit. A kitchen which is hard to clean, no place to park my car, the child playing where it can be run down by someone else's car, rainwater coming in, . . .*

Alexander presents a visual example to explain the basic idea of decomposition—a set of misfit variables represented as a network with links between nodes that interact:

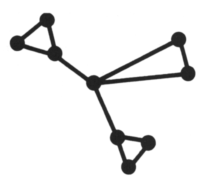


Then he writes:

> *Clearly these misfit variables, being interconnected, cannot adjust independently, one by one. On the other hand, since not all the variables are equally strongly connected (in other words there are not only dependences among the variables, but also independences), there will always be subsystems like those circled below, which can, in principle, operate fairly independently.*



Here is another picture from "Notes" showing the sorts of decompositions Alexander was thinking about:



Any computer scientist looking at these diagrams would immediately think of cohesion and coupling. But the work reported in "Notes" took place in the late 1950s, when these concepts were not known by these names. Instead, Alexander appeals to *homeostasis*, a concept described and explored by W. Ross Ashby in "Design for a Brain" first published in 1952.

Homeostasis is the tendency toward a relatively stable equilibrium between interdependent elements, and was one of the bases for thinking formally about living systems in the 1950s.

CA's Homeostatic Lights

- *n* lights with connections between some
- start off with all lights on
- on each cycle, for each light:
  - 50% probability a turned on light will turn off
  - 50% probability a turned off light connected to a turned on light will turn on
- if all lights turn off, they never turn on again
- W. Ross Ashby

Alexander asks readers to imagine a set of lights that behave as follows:

- if a light is on, every second there is a 50% probability it will turn off
- if a light is off, every second there is a 50% probability it will turn on if it is connected to a light that is on

If all the lights happen to be off, there is no way for any of them to turn on again. Alexander explains that each light can be thought of as a misfit variable: the off state corresponds to fit; the on state corresponds to misfit. He continues as follows:

> *The fact that a light which is on has a 50–50 chance of going off every second, corresponds to the fact that whenever a misfit occurs efforts are made to correct it. The fact that lights which are off can be turned on again by connected lights, corresponds to the fact that even well-fitting aspects of a form can be unhinged by changes initiated to correct some other misfit because of connections between variables.*
>
> *The state of equilibrium, when all the lights are off, corresponds to perfect fit or adaptation. It is the equilibrium in which all the misfit variables take the value 0. Sooner or later the system of lights will always reach this equilibrium. The only question that remains is, how long will it take for this to happen? It is not hard to see that apart from chance this depends only on the pattern of interconnections between the lights.*

From this initial idea, Alexander goes on to try to come up with a suitable goodness measure for potential decompositions. Let's be more explicit: even though Alexander states the underlying algorithmic problem like this:

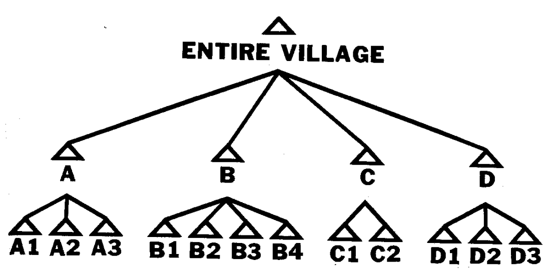
> . . . *how should* [a] *system be decomposed into a set of subsystems, so that the information transfer between the subsystems is a minimum?*

. . . the information transferred is in terms of the probabilities of a system of lights turning on and off. I don't know about other folks, but I found this a little batty.

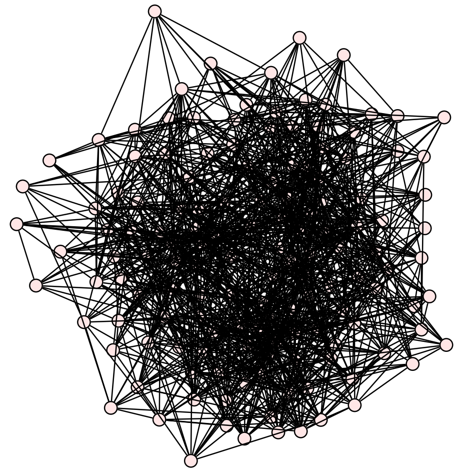
(Did I mention that Alexander wrote a program to solve the Indian Village design problem? Yeah, that's pretty important.)

I was interested in Alexander's program. It was written around 1962 in assembly language and ran on an IBM 7090 mainframe—a tiny computer by today's reckoning.

I tried several now-classical algorithms and several of my own devising. None came close to reproducing the decomposition from "Notes."

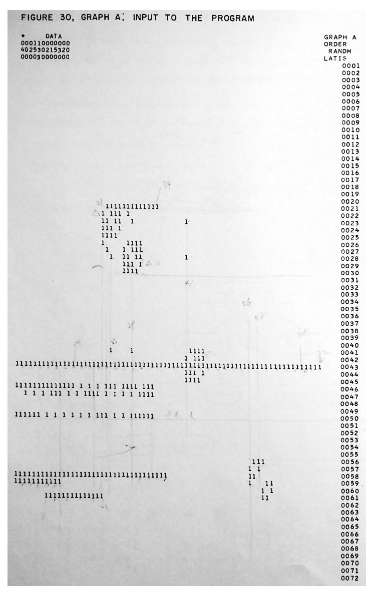


But, the Indian Village problem is not as simple as the examples shown in "Notes." Here is the Indian Village diagram:



Would Alexander have persisted in trying to decompose this if he had seen such a visualization?

I ran into some problems. First were errors in his data. Alexander states that interactions should be symmetric, but I found 50 requirements that were unidirectional. Did Alexander notice them? Yes. While analyzing some of Alexander's other early papers, I discovered that such errors were not uncommon. At first I was shocked to see so many errors.



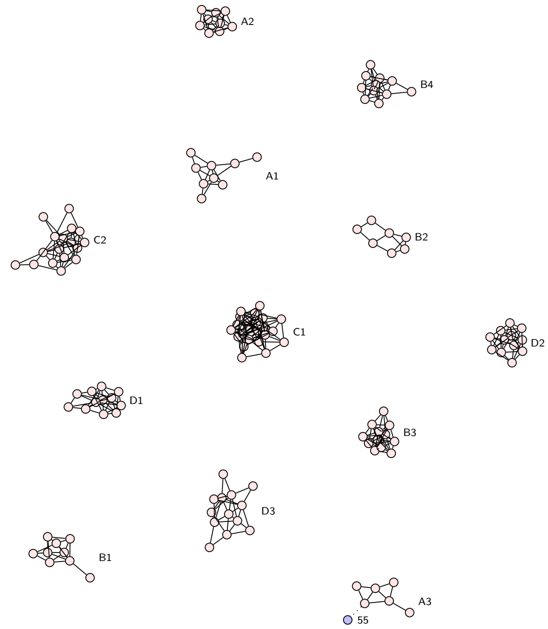
But I was less shocked when I saw that the input of interactions to his programs took the form of putting 1s on punch cards, so that misplacing a 1 is not outrageous. My shock was partly restored when I read that Alexander used a subroutine called "SYMET" to eliminate non-symmetric interactions:

> *With the program SYMET in operation . . . it is possible to generate the data at the card punch without handchecking it, with the assurance that it will be machine checked and that only the "most certain" pairs . . . will be treated as linked.*

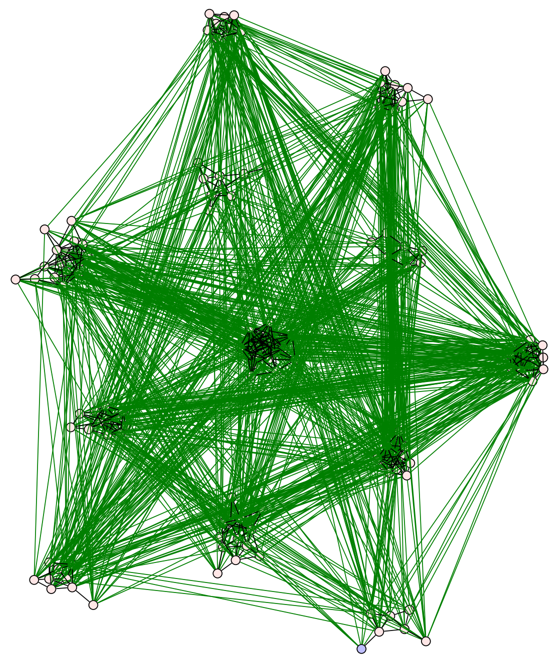
But . . . why not have the computer help find punchos?

I eventually figured out that Alexander's goodness measure was mostly trying to minimize coupling, with an "adjustment" to discourage clusters of widely varying sizes.

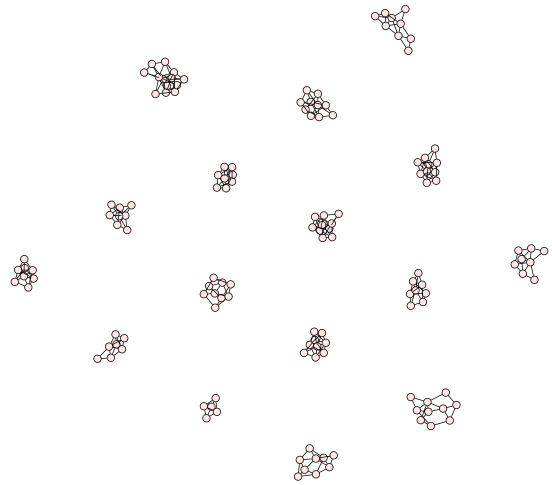
Here is Alexander's decomposition:



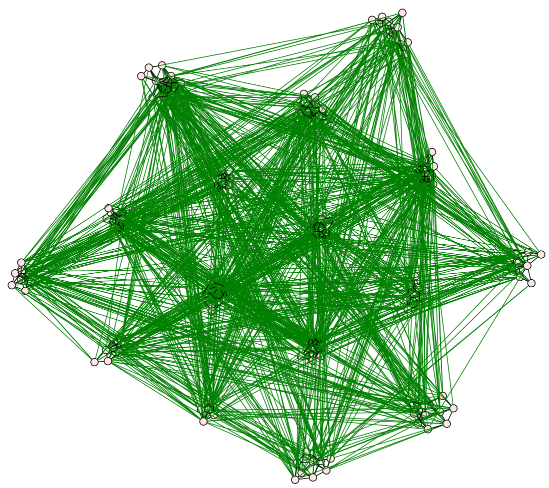
Looks good, right? But notice cluster A3; node 55 (“Cattle access to water”) is not connected to any other requirements in its cluster. Moreover, if node 55 is moved to cluster D1, Alexander’s measure improves. Now let’s look at how well this decomposition is at minimizing coupling:



I wrote my own goodness measure which tries to balance cohesion and coupling; it uses a different way to balance cluster size, and here’s how it did:

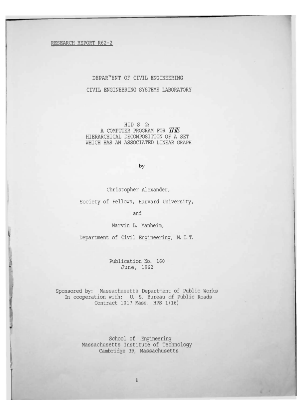


and coupling:



It measured out better than Alexander's solution using his goodness measure, but it's still lousy. But lousy because the whole idea is crazy of expecting an algorithmic decomposition of something as intertwined as the Indian Village problem to be simple.

You probably think that my essay and this talk are about the mathematics, algorithms, and programming of a network theory problem considered in the early 1960s. But it's really about the transformation of Alexander from a mathematics-minded architect creating a formal design method to a visionary philosopher and phenomenologist of beauty and design "drunk in God."



RESEARCH REPORT R62-2

DEPAR™ENT OF CIVIL ENGINEERING

CIVIL ENGINEBRING SYSTEMS LABORATORY

HID S 2:
A COMPUTER PROGRAM FOR THE
HIERARCHICAL DECOMPOSITION OF A SET
WHICH HAS AN ASSOCIATED LINEAR GRAPH

by

Christopher Alexander,

Society of Fellows, Harvard University,

and

Marvin L. Manheim,

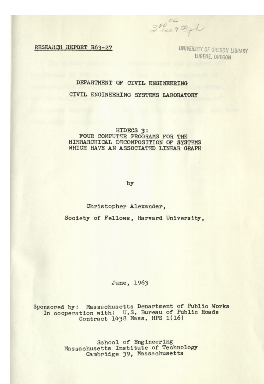
Department of Civil Engineering, M. I. T.

Publication No. 160
June, 1962

Sponsored by: Massachusetts Department of Public Works
In cooperation with: U. S. Bureau of Public Roads
Contract 1017 Mass. HPS 1(16)

School of .Engineering
Massachusetts Institute of Technology
Cambridge 39, Massachusetts

i



3^{00}_{no} ^{cc} 9 50 ph

RESEARCH REPORT R63-27

DEPARTMENT OF CIVIL ENGINEERING

CIVIL ENGINEERING SYSTEMS LABORATORY

HIDECS 3:
FOUR COMPUTER PROGRAMS FOR THE
HIERARCHICAL DECOMPOSITION OF SYSTEMS
WHICH HAVE AN ASSOCIATED LINEAR GRAPH

by

Christopher Alexander,

Society of Fellows, Harvard University,

June, 1963

Sponsored by: Massachusetts Department of Public Works
In cooperation with: U.S. Bureau of Public Roads
Contract 1438 Mass. HPS 1(16)

School of Engineering
Massachusetts Institute of Technology
Cambridge 39, Massachusetts

The last endnote in "Notes" mentions two reports on the programs Alexander wrote: "HIDECS 2: A Computer Program for the Hierarchical Decomposition of a Set with an Associated Graph"; and "HIDECS 3: Four Computer Programs for the Hierarchical Decomposition of Systems Which Have an Associated Linear Graph." They were never formally published.

After a lot of searching, I finally found them in 2022. I also found a translation of the HIDECS 2 program into Python.

HIDECS 2 reports on the program Alexander and his colleague Marvin Manheim wrote, which Alexander claimed had created his Indian Village decomposition. That program proceeds top down, breaking the set into two random partitions, then hill climbing to find the best using the goodness measure. This basic step is repeated several times.

With the Python program in hand, I was able to code it up myself in a style that made it easier for me to understand. And my 21st century computer was able to do at least 500 times more hill climbing than the 1962 IBM 7090, despite my use of real data structures instead of bit arrays.

The HIDECS 3 report, written in 1962 about a year after HIDECS 2 tells a story of slowly dawning insights. That story starts like this:

> *HIDECS 2 has three important weaknesses:*
>
> *1. The fact that the decomposition is made in a series of binary steps leads to certain 'mistakes,' since the holistic relatedness of system and subsystems is not properly taken into account.*
>
> *2. The fact that the decomposition criterion INFO* [the goodness measure] *is based on very stringent assumptions about the nature of the system $G(M, L)$. Namely, that the elements of M are binary variables, that the two variable correlations are very small, and that the many variable correlations vanish altogether. These assumptions make it hard to find systems in the real world which the formalism of HIDECS 2 can adequately represent.*
>
> *3. The fact that the subsets of elements which make the most natural subsystems of a system are not always disjoint, but often overlap.*

I will bluntly paraphrase these points:

Slowly Dawning Insights

1. Binary partitioning proceeding top down is a mistake.
2. Treating the network as probabilistic binary variables is a mistake. That is, homeostasis was the wrong starting point.
3. Disjoint decomposition into tree structure is a mistake—overlap is necessary.

In the HIDECS 3 report, Alexander presents four approaches and programs that try to fix the problems with HIDECS 2.

The first step is to look at bottom-up approaches, paying attention to the gestalt or holistic nature of the system. Alexander writes:

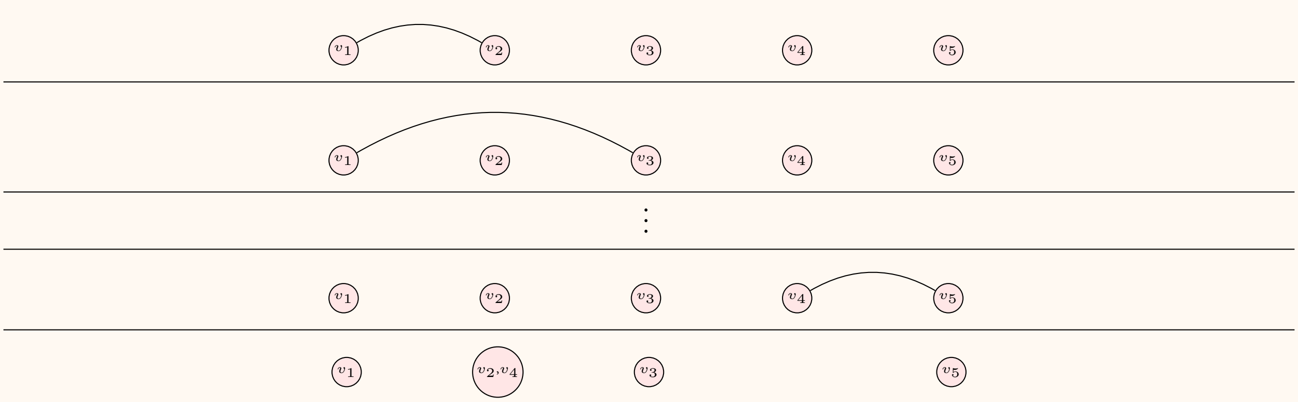
> *However, the defect of any algorithm which partitions M into two subsets at a time, is that it does not pay sufficient attention to the gestalt, or overall pattern of the subsystem, and therefore introduces a bias which by any reasonable intuition is a 'mistake.'*
>
> *...the decomposition into subsystems need not be defined stepwise, but can be defined all at once, and the holistic nature of the system thereby better preserved.*

The first program uses as a goodness measure a generalization of the original HIDECS 2 measure to multiple sets. It proceeds bottom-up, starting with each vertex in its own partition;

Initial

Then for each step, pairs of partitions are combined, the result measured for total goodness, and the best selected. The process continues until there is no improvement.

Then, like this



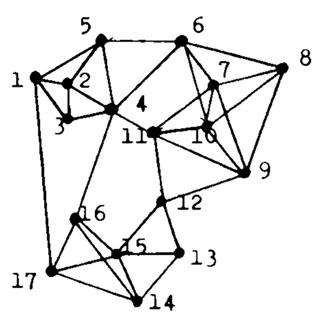
The coupling based measure worked so poorly that Alexander immediately gave up.

Alexander's next change is to give up on the idea that vertices are binary stochastic variables; instead, vertices are just plain-old nodes. Links are just links.

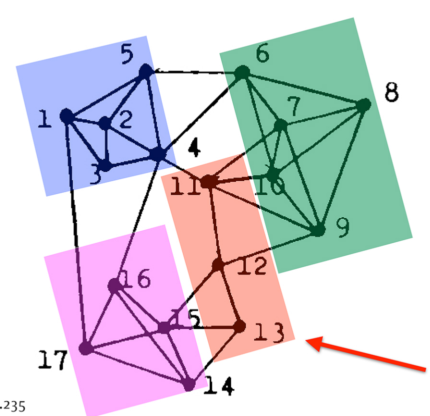
This next program uses a new goodness measure that looks at both cohesion and coupling. It proceeds by assigning each vertex to its own partition, then tries moving vertices one at a time from one partition to another, measuring the goodness of each such move, and choosing the best for the next step.

How well did this do?

For all the programs in the HIDECS 3 report, Alexander uses this graph as a test:



His program partitions it like this:

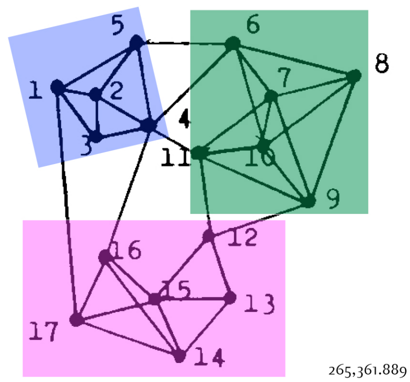


36,862.235

. . . which includes the interesting partition $\{11, 12, 13\}$. This odd partition comes from what I consider a novice programmer's mistake when coding searches based on measures.

Alexander says "the best of all the decompositions so obtained is thus the best decomposition to be obtained by moving a single element," but he overlooks the possibility of ties, and thus either his program must make a choice and live with it moving forward, or it must do a tree-search—not a simple hill-climbing search.

My version of this program performs a tree search, I got this decomposition:



Using Alexander's measure, his decomposition measures out at about $37,000$; mine at about $265,000$. The upshot: same measure, same basic program, but one picks one from a set of "best" measures, and the other searches all possible choices.

For the last two programs, Alexander gave up on trees—that is, disjoint partitions—and admitted semi-lattices. More than that, he adopted a more standard clique-detection algorithm developed by Frank Harary and Ian Ross.

> *Finally, in SIMPX and EQCLA, the subsystems are defined in such a way that they overlap instead of being disjoint. In fact, in these two cases the decomposition, instead of being a tree, is a lattice.*

At this point I was done trying to replicate and understand his programs.

"Notes" was published in 1964. In 1965 Alexander published an essay called "A City is Not a Tree" in which he wrote the following:

> *For the human mind, the tree is the easiest vehicle for complex thoughts. But the city is not, cannot and must not be a tree. The city is a receptacle for life. If the receptacle severs the overlap of the strands of life within it, because it is a tree, it will be like a bowl full of razor blades on edge, ready to cut up whatever is entrusted to it. In such a receptacle life will be cut to pieces. If we make cities which are trees, they will cut our life within to pieces.*
>
> ⋮
>
> *The structural simplicity of trees is like the compulsive desire for neatness and order that insists that the candlesticks on a mantelpiece be perfectly straight and perfectly symmetrical about the centre. The semilattice, by comparison, is the structure of a complex fabric; it is the structure of living things—of great paintings and symphonies.*
>
> ⋮
>
> *It must be emphasized, lest the orderly mind shrink in horror from anything that is not clearly articulated and categorised in tree form, that the ideas of overlap, ambiguity, multiplicity of aspect, and the semilattice, are not less orderly than the rigid tree, but more so. They represent a thicker, tougher, more subtle and more complex view of structure.*

Perhaps the thrust of these quotes isn't obvious: Alexander is repudiating the results in "Notes." "Notes" and an earlier, related paper became central to a movement in the 1960s called "Design Methods," whose proponents admired Alexander's "rational approach to combining problem requirements into solution structures." In an interview in 1971, Alexander said:

> *I've disassociated myself from the field. . . . There is so little in what is called "design methods" that has anything useful to say about how to design buildings that I never even read the literature anymore. . . . I would say forget it, forget the whole thing. . . .*

In a 1965 essay he wrote:

> *Myself, as some of you know, originally a mathematician, I spent several years, in the early sixties, trying to define a view of design, allied with science, in which values were also let in by the back door. I too played with operations research, linear programming, all the fascinating toys, which mathematics and science have to offer us, and tried to see how these things can give us a new view of design, what to design, and how to design.*
>
> ⋮
>
> *Finally, however, I recognized that this view is essentially not productive, and that for mathematical and scientific reasons, if you like, it was essential to find a theory in which value and fact are one, in which we recognize that there is a central value, approachable through feeling, and approachable by loss of self, which is deeply connected to facts, and forms a single indivisible world picture, within which productive results can be obtained.*

In his Preface to the 1971 paperback edition of "Notes," Alexander seems to apologize for thinking of design as a rational method that leads from requirements to a formal design.

> *At the time I wrote this book, I was very much concerned with the formal definition of "independence," and the idea of using a mathematical method to discover systems of forces and diagrams which are independent.* [diagrams are patterns] *But once the book was written, I discovered that it is quite unnecessary to use such a complicated and formal way of getting at the independent diagrams.*
>
> ⋮
>
> *If you understand the need to create independent diagrams, which resolve, or solve, systems of interacting human forces, you will find that you can create, and develop, these diagrams piecemeal, one at a time, in the most natural way, out of your experience of buildings and design, simply by thinking about the forces which occur there and the conflicts between these forces.*
>
> ⋮
>
> *I have written about this realization and its consequences, in other, more recent works. But I feel it is important to say it also here, to make you alive to it before you read the book, since so many readers have focused on the method which leads to the creation of the diagrams, not on the diagrams themselves, and have even made a cult of following this method.*
>
> ⋮
>
> *Indeed, since the book was published, a whole academic field has grown up around the idea of "design methods"—and I have been hailed as one of the leading exponents of these so-called design methods. I am very sorry that this has happened, and want to state, publicly, that I reject the whole idea of design methods as a subject of study, since I think it is absurd to separate the study of designing from the practice of design.*

What can we learn from these investigations? Christopher Alexander's journey was of slowly dawning insights not a grand "Aha!" First are the small insights about the problems of decomposition, cohesion, and coupling viewed during the early days of computing.

Although Alexander did not have the concepts of cohesion and coupling as they are now known, he navigated the waters between them. He was not shy about using techniques and algorithms invented by others: some randomized algorithms already existed and were generally known in the late 1950s; clique detection algorithms were known and Alexander acknowledges using one. Alexander and Manheim were not inept programmers—the HIDECS programs were written in assembly language and exhibited a sophisticated use of "bumming."

Next is that Alexander was using the software he was creating as a colleague; his understanding of the problem improved as the flaws in his programs revealed themselves; sometimes he tried to improve the programs, and other times he reformulated the problem.

Finally—and most importantly—Alexander's struggles taught him to look away from formalism to understand design—to look elsewhere. Note the progression of thought from these very early investigations to those near the end of his career. This is his big insight: design requires human feeling.

The reality of the computer and the poverty of programming languages were stern teachers, teaching Alexander that cold abstraction requires a warm human hand and experienced (tear-filled) eyes, that machines can be partners for exploration, and that a city is not a tree.

Expanded Essay: `https://dreamsongs.com/Files/NotesOnNotesExpanded.pdf`
Journal Version: `https://programming-journal.org/2023/7/8`