Modeling Crop Rotation with Discrete Mathematics

Kellen Myers*, Victoria Ferguson, Yekaterina Voskoboynik

0.1 Objective

The purpose of this module is to provide a framework by which crop rotations may be scheduled using discrete mathematics models. This module is applicable to three levels of coursework to meet the following objectives:

1. **Advanced discrete mathematics course**: This option provides an example of discrete mathematics applied in the context of sustainability. A formal description mathematics that can be used to construct crop-rotation models is given, including block designs and finite state machines. Contextual information as to crop rotation techniques are provided, without emphasis. Students will construct crop rotation schedule(s) using models based on the mathematics discussed and a provided list of crops and their characteristics.

2. **Basic discrete or applied mathematics course**: this option provides an example of utilizing discrete mathematics in the context of sustainability. Definitions and examples of relevant structures and concepts, including block designs, are provided. References and extensions of this material are available for those students interested in learning more about the mathematical proofs or the horticultural/agricultural research. Students will complete a schedule using a given model and a given list of crops with their characteristics.

3. **Applied horticulture/agriculture course**: this option provides basic background on a model appropriate for scheduling the rotation of crops. More background on the crops and the research behind these rotations are provided. Students will complete a rotation schedule using a given model and an extended list of crops from which to build the appropriate model for their intended market(s).

In order to use this module, one should read the introduction and background material carefully to contextualize the lesson plan. References should be consulted as necessary. Once this material is well understood, the lesson plan can be adapted to one's classroom setting and implemented. Sections on assignments and follow ups may also be applied according to one's classroom parameters. The background and lesson sections are divided into four parts, according to the subject matter. It is suggested that this material is presented to students in this the order it appears here.

The lesson plan section is meant to be a succinct plan that draws upon the information in the introduction and background sections, which are not presented or represented within the lesson plan in order to keep the lesson plan less cluttered. Those sections contain the bulk of the "material" in this module, whereas the lesson plan is meant to outline the presentation of this material. It also utilizes some of the appendices, which provide more compartmental data, including large tables and illustrations. The background second does contain recommendations and details regarding the intended or recommended methods of presentation, assessment, structure, etc. of the lesson's material.

0.2 Introduction

The production of crops is essential to human life. Producing these crops, however, comes with it a cost to the environment. The commonly used agricultural techniques in place in the United States rely on machinery and chemicals in order to keep up with the demand. However, over time, these methods strip the soil of nutrients which in turn require added inputs to produce the crops in the first place. In order to continue to produce crops at current levels of need and, by extension, to increase production, methods to counter these deficiencies must be utilized.

One of these methods, used in both organic and "conventional" agriculture, is that of crop rotation. Crop rotation may be as simple as the current corn/bean rotation in the conventional system, or more elegantly implemented with organic techniques such as intercropping, "green manure," and "catch cropping." Certain crops have characteristics which lend themselves to rotation. These crops are then rotated with crops which provide income or fodder for animals. However, the creation of a rotational schedule over a plot of subdivided land is a challenge which faces the agriculturist in planning the most efficient use of land so that demand is met, soils are sustained, and economic impact is reduced when leaving land off of a "production schedule." The use of discrete mathematics allows for creation of schedules meeting a variety of needs.

There is no single aspect of discrete mathematics that explicitly addresses crop rotation (in the way that, say, graph theory provides models for networks in biology and computation).. However, many concepts from discrete mathematics may be used in constructing models for crop rotation, as crop rotation is a complex and multifaceted process.

In constructing a model for crop rotation, various plots of land are assigned crops which are rotated on some regular basis. If we consider some set of crops from which these assignments are made, and each plot of land is assigned some subset of these crops, then essentially this is a block design. A block design is, essentially, a base set S and a set of blocks **B**, each of which is a subset of S. Certain aspects of the model are not encompassed by block designs (e.g. the order of the rotations). However, various aspects of the theory (e.g. regularity of the block design), are applicable to the crop rotation model.

Because there are only finitely many ways in which crops could be planted at any given time, and because they will be organized and planted in some deterministic way (see note), the notion of a finite state machine is also applicable to this model. This better encapsulates the transitions in the model, from one state to the next, because there should be some particular relation between one crop in a plot and the next crop that is planted. However, this structure is very abstract and open-ended, and may have too little structure to provide on its own any precise guidelines for crop rotation.

Note: Although students may introduce randomization into their model, this lesson will not venture so far as to consider random or partially random methods. This module may also invoke from students many other concepts or ideas, some of which may extend or supplement the methods and concepts provide in the lesson. For example, various methods in optimization (optimizing yield or profit).

For such reasons, crop rotation provides a good exercise in both advanced and basic discrete and/or applied mathematics courses. Many different concepts from discrete mathematics may be applied to the problem, as well as ad hoc methods generated by students, and these methods may be combined and modified to best suit the problem. There is a substantial opening for students to use creative and innovative methods to solve this problem in a critical and analytical way, rather than just apply a theorem or use a standard model and be done – this module will provide students with "hands on" experience with modeling, in an open-ended and creative context, and will eschew the "plug and chug" methods that some students may expect or prefer.

1. Background

1.1 Agriculture

Planning a rotational schedule requires knowledge of multiple factors: knowing how many crops will be planted, into how many segments the land area may be divided (keeping in mind that smaller agriculture is permissible), and the number of crops which must return on investment. Each land segment is considered a field; each field has its own rotation schedule.

The rotation itself is determined by biophysical needs as well as weather, market conditions, and logistical factors: thus, a rotation planned on paper may need to be changed in practice if any of these non-biophysical needs change [NEO]. Rotations which are multiples of 4 and 5 are typically used in the northeastern United States, especially when growing Solanaceous crops, which must be rotated out for three years between growth due to the propensity for soil borne diseases.

Plants are divided into three primary groups: cash, soil, and forage crops. They are defined as follows:

1. Cash crops are those that will be used for income, or processed for the community or private good. These are such thing as sweet corn, popcorn, field corn (for animal use), tomatoes, peppers, potatoes, cabbages, squash, cucumbers, beans, some flowers, herbs.

2. Soil crops are those that give back to the soil, or trap nutrients so that they are not leached out of the soil. These are also known as cover crops, which are further divided into catch crops ('catching' nitrogen) and green manure (which deposit nitrogen through their own processes). Most of these are not income producers.

Forage crops are those which are used for grazing animals. Many of these are also cover crops. Forage crops are not usually income producers, but they may be when dried and sold to other livestock farmers. One of the primary concerns addressed by crop rotation is nitrogen depletion in soil.
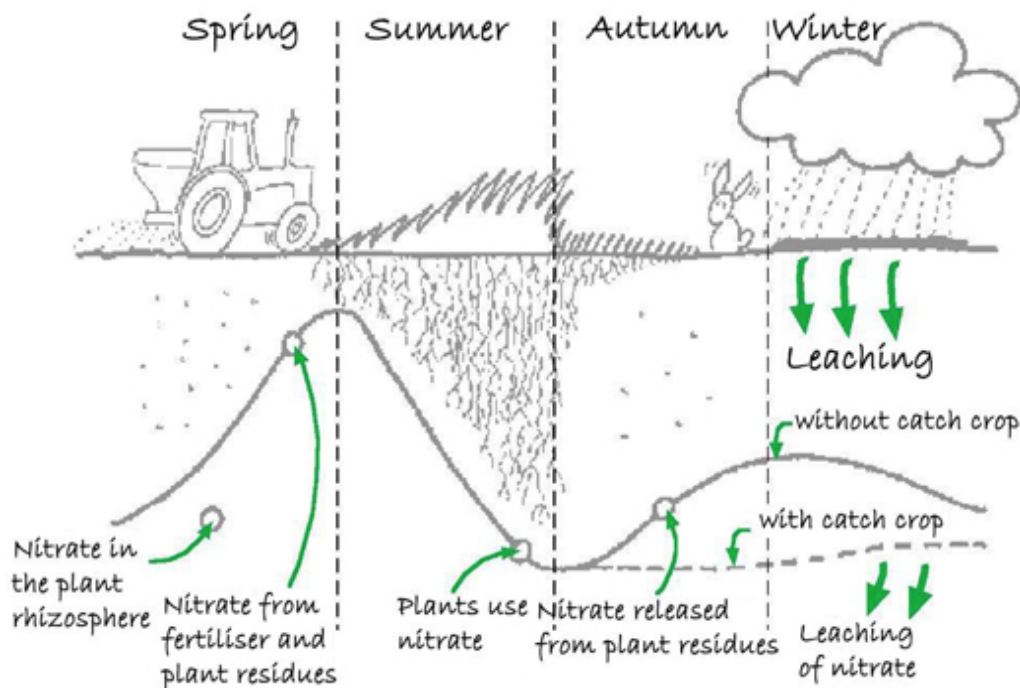
Typical Nitrogen Cycle over a year [ØST]

In most parts of the United States, crops are grown on a four-season basis. Winter, even in most Northern states, can be used as a cover crop preparation for crops that are sown in early Spring. These seasonal aspects of crop rotation are particularly important in sustainable agriculture, and the use of cover crops provides means of Cover crops may serve specific purposes. These include the following:

1. increasing carbon to nitrogen ratio by increasing organic matter;

2. increasing nitrogen levels;

3. easily "killed" - require no herbicides for end-of-run kill;

4. compaction reduction due to deep roots;

5. grazing crop/quick forage;

6. soil erosion prevention;

7. recapture nutrients;

8. weed suppression/natural herbicides;

9. beneficial insect attraction;

10. heat/drought tolerance;

11. wet soil tolerance;

12. cold tolerance;

13. nurse crops.

List taken from [HOO].

Considering even just a subset of these factors, we can begin to see that a mathematical model might be needed to capture the complex requirements and structure of crop rotations, and to construct and to analyze such crop rotations.

1.2 Modeling

Once the number of crops and their properties and relationships are determined, the model for scheduling the crops may be built. In mathematical modeling, the first step is to formulate a real world problem, scenario, or structure in a mathematically meaningful way. Some properties are ignored or simplified, generally in ways that are intended to clarify, simplify, or support – not invalidate – the mathematical analysis and conclusions of modeling.

Example: Systems of roads may be considered as an abstract network or "graph," which is a mathematical representation of roadways that illustrates some of their properties.

Once the model has been constructed in a way that represents mathematically the important aspects of the real world subject, the model is subject to mathematical scrutiny. One can analyze the model for known properties. This type of analysis helps serve to validate the model. The model may also be used to predict or suggest events, facts, or outcomes in the real world that can be observed or predicted in the model more easily.

Example: model of a traffic network should correctly give us information we already know like the number of intersections in the network. We may also use the model to try to determine whether one can get from a certain point to another point without crossing a particular road, or to predict whether

traffic flow will be obstructed by making a particular street one-way instead of two-way.

This illustration from [ROB] shows the three steps described above (numbered in the order presented here). Note that this presentation emphasizes the cyclic nature of modeling – requiring refinement and adjustment over time. Some of the assignments and follow-ups presented with this module focus on analysis of the model and its real-world implications in order to refine or improve the next model. Even though we cannot literally test the model (this would require many years, costly resources, and much labor), we can analyze it mathematically, simulate it thoroughly, and otherwise obtain the feedback that is an important part of this process of modeling.
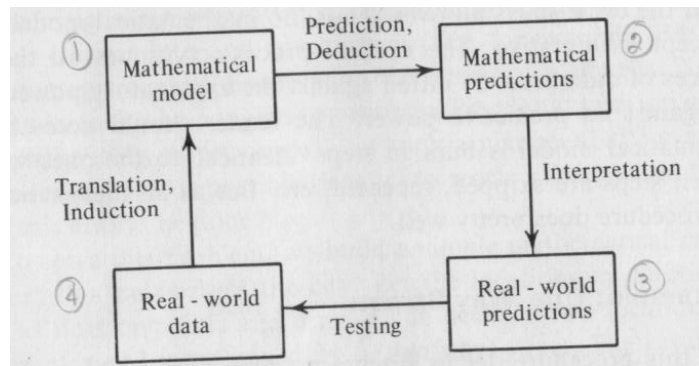
A more complex diagram from [CLE] is included in Appendix 3, and this diagram illustrates the feedback in the modeling process that solely in the realm of mathematics, simulation, and/or computation. [ROB2] presents in its introduction a take on modeling that is particularly attuned to simulation models. Simulations are important for (at least) three reasons:

1. They save money by avoiding real-world implementations of tests/computations/measurements.

2. They allow us to obtain mathematical/abstract data from systems for which it may not be feasible to simply compute all possible outcomes or to otherwise analyze mathematically (i.e. computationally intractable problems).

3. They save time by allowing many tests/computations/measurements in short spans of time.

When students are asked to analyze their models for crop rotation, they should not only be presented with simulation as one method of analysis, but they should question precisely how simulation works. Having constructed models that account for nitrogen in soil, issues with various crops being adjacent, and the yield of various crops providing food and/or income, students should consider how a simulation will account for these factors.

Note: It is amusing to consider that our description of modeling is, itself, a model of a particular academic/scientific matter that occurs in the real world. It is represented mathematically in various diagrams and descriptions but is, itself, subject to feedback that makes it more effective and meaningful. For this reason, students should be encouraged to think critically about the process of modeling itself.

For more on this four-step model of modeling, [ROB] and [BEN] provide great exposition and examples, and even exposition about particular examples or classes of examples. The introduction of [BEN] gives an interesting small example wherein effective deployment of sales employees is considered. This, in particular, is an example more like our question, where we want to know how to do

something – not just how something deterministic happens in nature (i.e. a natural phenomenon not created or affected by our actions). [CLE] gives an account of the historical development of mathematical modeling, and is a book that has a particular focus on teaching about modeling, in its own right, not just how to do it. [ROB2] provides, in its second chapter, important exposition on the nature of modeling in ecology. They assert that modeling in ecology is unique (or at least, different from some other kinds of modeling) for a variety of reasons:

1. Ecological systems are interconnected and cannot be subdivided – a single cell can be observed, but human cells when isolated no longer behave as they normally do as a part of the body.

2. Ecological systems are incredibly rich in structure on many levels – although we can consider crop rotation for a single field, the practice of crop rotation may also have impacts on larger scales (the local or global ecosystems) or on a smaller scale (insect and bacteria populations). The subsequent effects on other scales then feeds back into the scale we are considering.

3. Ecological systems contain objects, quantities, and ideas that are not necessarily strictly or precisely defined – the notion of "species" is subject to much dispute. Studies of predator-prey interaction can be modeled based on individual animal actions, or by larger-scale considerations that consider only the larger trends in population.

4. Ecological systems are subject to wild fluctuations for which one cannot account, and worse, limitation on time and resources severely limits the reproducibility of ecological experiments or the testing of the conclusions of various ecological models. For this reason, biological models are often tested for sensitivity of initial/external conditions.

Once the desired characteristics of the process to be modeled have been determined, it is important to consider what mathematical structure(s) and theory(ies) might best describe it. In this case, students can consider the two such structures, block designs and finite state machines, as well as other ideas they may introduce themselves into the model – previously known mathematical concepts, concepts they look up on their own, and/or ad hoc methods they devise. These methods allow students to construct crop rotations and analyze their properties for strengths and weaknesses according to the parameters of the model and the underlying natural process.

For the purposes of this lesson, we devise the following model for a crop rotation:

A mathematical block design consists of three structures with the following properties:

1. a set of crops, wherein each crop has three attributes:

    (a) the type of the crop (e.g. consumer crop)

    (b) the soil properties of the crop (e.g. nitrogen fixer)

    (c) a set of crops that are contraindicated as neighbors for the crop

2. a planar region, subdivided into smaller subregions

3. a function that assigns to each subregion an ordered list of crops

This presentation of structure should be familiar to advanced mathematics students, and can even be rewritten as (S,R,f), where S is a set of triples (t,s,v), R is a set of (presumably but not necessarily contiguous) regions in the plane, and f is a function on the members of R that outputs finite sequences of members of S. From this structure, other mathematical structures may be formed (e.g a graph with vertices R that indicates adjacency in the plane with adjacency in the graph). The "rules" of a good crop

rotation can be presented in similarly abstract ways, but should first be presented as a part of the model and then be subjected to abstraction.

Note: In order to best teach this lesson, we recommend that the model be taken as mostly fixed, at least at first, so that block designs and FSMs are actually applicable. Discussion about how to construct the model is beneficial, but we recommend that students be asked to accept the given model as the first starting-point in the next step as given, rather than have students attempt to apply the following mathematical methods to models to which they are inapplicable.

1.3 Block Designs

A block design is an abstract mathematical structure. The most basic notion of a block design is simply a set S, sometimes called the varieties, and a set **B** of which each element B is a subset of S. Each B is called a block, and **B** itself is sometimes referred to as the design or block design. This is fair, in that S could be inferred from **B**, at least in the sense that any elements of S that do not occur in any member of **B** are mostly irrelevant.

Two parameters inherent to all block designs are v = |S| and b = |**B**|. The v stands for variety, instead of using s or some other letter. There are other properties that may also be imposed on block designs, some of which result in further useful parameters:

1. If every block has the same size, then **B** is said to be uniform.* We let k = |B| for all B in **B**.

2. If **B** is uniform and if every element of S occurs in the same number of blocks, then we let r be the number of blocks in which each element of S occurs. Here **B** is said to be regular.

3. If **B** is regular, and if r = k, then **B** is called complete.

*Note: The term "uniform" here is not necessarily standard to block designs. Generally, designs are considered to be either regular or not regular. The term "uniform" here is borrowed from the terminology of hypergraphs (although the two structures are defined similarly or even identically, the theory surrounding each one is quite different).

Note that the three definitions are nested: To be complete, **B** must be regular; To be regular, **B** must be complete. Complete block designs are not very interesting in their own right because the design is literally **B** = { S, S, S, …., S , r copies of S. The notion of being complete is so restrictive that if **B** is complete, **B** is completely determined by S and r. However, shortly examples will be given where complete designs are more interesting if we consider other properties of **B** that we could explore.

Block designs can be used to mathematically describe our model because each could be given a block, where the block is the set of crops to be grown in this block.

Note: Although this is not a part of our model, it is important to note that one technique in agriculture is [need elaboration about that thing where multiple crops go in the same place – which is also quite amenable and natural to be studied by block designs].

Latin squares are an example of complete block designs that give interesting structure – each block is endowed with an order (something quite relevant to our model, since this is a part of our subject that is not incorporated into basic block designs). A complete block design is a Latin square if each block is given an order B(i) = B(i,1), B(i,2) , … B(i,k), and no two blocks ever agree at a given "time" that is, if for any i, j, and t, if B(i,t) = B(j,t), then i = j.

Latin squares designs are "squares" in the sense that they can be arranged with blocks listed in order as rows, and the number of rows and columns is equal. Each row contains each element of S exactly once,

and so too does each column – that is what the $B(i,t) = B(j,t)$ condition means. In this way, Latin squares also reflect a geometric property, similar (indeed, a much stronger condition) to some of the conditions we might impose about crop adjacency.

There are many thorough references, including [STR] and [WAL], that elaborate on the theory of block designs and related structures. They are all generally very technical and theoretical, and extend far beyond what is relevant to this module – usually the introduction and first chapter is about as much as could be relevant to the module and its presentation to students. However, chapter 14 of [STR] treats block designs in which the blocks are ordered and adjacency in this ordering is restricted. This is similar to the restrictions of Latin squares, but is closer to the kinds of restrictions we would put on crops. Unfortunately, it is mathematically beyond the bounds of the module.

Chapter 10 of [HAL] treats block designs in a text that explores many aspects of combinatorics and discrete math. His approach to designs is fairly advanced, but it may be recommended as extended or further reading for advanced mathematics students who are interested in block designs and their context in the larger body of mathematics.

Different approaches do block designs yield different models for crop rotation. A generic approach does not provide very much insight. Our model is, essentially, a bunch of columns listing crops in order of their rotation. Appendix 2 provides an example of what this looks like. The layout of the plots in the field of land is irrelevant until adjacency restrictions are considered, and those should only be considered for advanced math students.

For this reason, a block design gives some idea of how to distribute the crops to each plot. Putting them in a particular order is the only extra step, which is why block design + ordering is an interesting approach. Once the order and crop properties are introduced, students can analyze the benefits of various methods and ideas from block designs (or from elsewhere).

Students should also be asked to discuss the benefits of having regular or non-regular block designs as crop rotations – this is a major point, and Appendix 2 illustrates the potential problem of "just making it up" – a regular block design avoids the periodic "too much of _____" seasons, and allows for more uniform and regular yields and output. However, regular block designs are also more structured and may restrict options and give less flexibility in varying the yields and other parameters.

1.4 Finite State Machines

Finite state machines, and related abstract structures, provide a different framework for imagining our model of crop rotations. There are several similar, but slightly different, definitions of various machines. In this context, a "machine" is an abstract model of computation or machination of some sort. For example, a VCR or washing machine could be modeled by a relatively simple FSM.

The model of computation is one of states – some set of states of the machine. Certain input (or in some cases, no input at all) give the machine instructions about how to transition from one state to another. A blinking light is one of the most simple objects that could be modeled as a FSM. It has no input, and it simply transitions between on and off forever.

Such a device could have input, however. The input could be an on/off switch. In the case that the light is on, it will always turn off again, but if the switch is also in the off position, the light will stay off. Note that the input from the switch determines which transition to take from each state to the next.

The most basic structure is a Finite State Automaton (FSA), which is a machine that is fixed throughout its entire computation. The FSA may have input, but this input is a finite amount of input that is known

when the machine begins computing. In that sense, the input determines a fixed machine that then carries out the computation with no further input during the computation.

A Finite State <u>Machine</u> (FSM) is an FSA but with the additional parameter that input at any point during the computation might determine different transitions. For this reason, an FSM has much more complexity than an FSA, although neither is as complex as the standard model of computation, the Turing Machine (which is, itself, similar to actual computers).
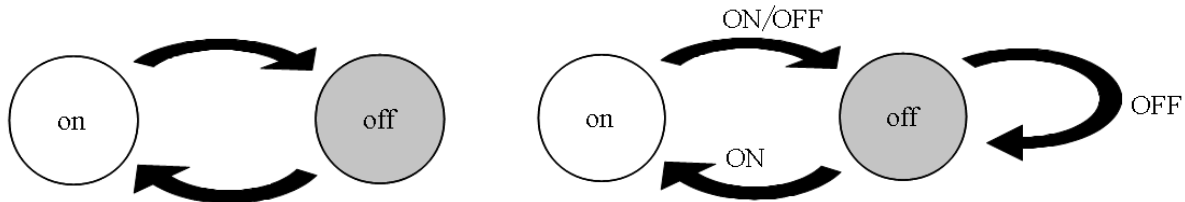
A blinking light, as an FSA and an FSM (with ON/OFF switch)

A <u>Deterministic</u> Finite State Automaton/Machine (DFSA or DFSM) is the same as above, except that certain states are considered to be "final" states, at which point the computation is considered to be finished (and, if "output" is desired, the output is the corresponding final state).

For advanced (or some basic) mathematics students, it is useful to give the standard mathematical definition of each structure. A FSA consists of four things, the tuple (a,S,s,f), where:

- A is the input alphabet, the set of all possible inputs;

- S is the set of all possible states;

- s1 is the initial state, a member of S;

- f is the transition function, a function that maps SxA to S.

Here, the transition function is dependent on the input a, but there is only one initial input. So the function would be computed as s2 = f(s1,a), then s3 = f(s2,a), s4 = f(s3,a), etc.

A FSM is defined only slightly differently (A,S,s,f):

- A is the input alphabet;

- S is the set of all possible states;

- s1 is the initial state;

- f is the transition function, a function that maps SxA to S;

Everything here is the same except that there is a different input at each step, so instead of what is above, we get s2 = f(s1,a1), then s3 = f(s2,a2), s4 = f(s3,a3), etc.

And in the case of a deterministic FSA or FSM, the set F is also added to the structure:

- F is the set of final states, a subset of S, for which f is not necessarily defined.

One important caveat for DFSA or DFSM is that, in addition to allowing terminal states, we must

require that no cycles exist in the machine – in other words, that a final state *must* be reached, regardless of the input. If the machine is drawn as a directed graph, this requirement is that such a graph is acyclic.

These machines were originally invented to model various real life machines that perform some computation, algorithm, or mechanized function. Below are other sorts of "real world" examples that are hopefully familiar, but illustrate the difference between each type of machine.

- FSA: A set of traffic light on a timer. (Traffic lights with sensors are FSM.)

- FSM: A digital thermostat, which measures temperature and turns climate control devices on and off depending on that temperature.

- DFSA: A treasure hunt game where the player is presented with a clue, which leads to the next clue, and so on.

- DFSM: A "choose your adventure" book, where the player is presented with options about choices to make, leading him/her from page to page in the book until a conclusion is reached.

Appendix 4 contains examples of each kind of machine, representing crop rotations, drawn out in a diagram. This representation is often useful to illustrate FSA/FSM/etc. but is impractical when the set of states or number of inputs is too large.

References for FSMs are similar to those for designs, in that usually texts venture immediately into mathematics that are far beyond that which is needed for this module. [GIL] is one such example – the reference would be good for advanced mathematics student and/or students looking for extended mathematical study in this area. [CAR] provides a gentler introduction suitable for advanced (or perhaps even basic) math course, and may be more readable as a reference as well.

Finite state automata/machines provide a reasonable way to construct crop rotations because the parameters of the model include conditions that have to do with the transition from one crop to another (e.g. the nitrogen fixers and nitrogen consumers). An FSM would also allow for input dynamically, that is, the algorithm accepts input at each step and may have different transitions depending on the input. For us, such input might include soil conditions, weather conditions, desired yields, or others. For details on how this would work, see Appendix 4, which includes a small example that has a FSM that has the season as input.

2. Learning Objectives

The emphasis and parameters of each learning objective will vary depending on the focus of the course in which the module is presented. Assessment and evaluation will similarly depend upon this context. Generally, upon completion of the module, students will be able to:

1. Understand the basic concepts of crop rotation, and its importance for sustainable agriculture;

2. Understand the fundamental (four step) process of mathematical modeling;

3. Understand the benefits of modeling as a process with feedback;

4. Understand the role of simulation in the process of modeling, differentiated from real data;

5. Apply block designs, finite state machines, and ad hoc methods to create a crop rotation;

6. Compare and contrast block design and finite state machines in crop rotation planning;

7. Create a crop rotation using block design, finite state machines, and/or ad hoc methods, based on the restrictions and parameters prescribed in the module;

8. Present their rotation schedule to peers and be able to explain and justify its construction;

9. Analyze a given crop rotation, provided by the instructor/module or constructed by peers, based on real-world parameters prescribed in the module.

3. Lesson Outline

The outlines provided in this section are intended to help an instructor create a lesson plan to teach the concepts described in the background section and to effect the outcomes described above. Some examples, exercises, and other items are recommended for a certain category of class. Others are recommended at the instructor's discretion to be either in-class or take-home exercises, in groups or individually, as discussion or presented as lecture. An instructor should take care to construct the lesson appropriately. The major exercise – the construction of a crop rotation – is itself subject to modification depending on the type of class being taught.

Examples, exercises, and exposition are clearly labeled, and as applicable are labeled with the type of course (**AG** for an agriculture or horticulture class, **BM** for basic math, **AM** for advanced math). This module is intended to span at least one class period, but could occupy two or three, depending on the parameters of the course. Included with this module is a separate document that is the "student packet" which includes content from the module that may be used as handouts for students, either as a part of lecture/exposition or as part of the assignments/exercises.

<u>3.1 Agriculture</u>

1. **Exposition**: Introduce the concept of crop rotation

    A. Fields divided into plots, crops are planted in each, on some schedule

    B. Rotation influenced by:

        i.   desired yields (for consumption or sale)

        ii. weather conditions

        iii. soil conditions and related matters

        iv. interactions between crops in the same field at different times

        v.   interactions between crops in adjacent fields at the same time

2. **Exposition**: Three types of crops:

    A. Cash crops – used for income or consumption

    B. Cover crops – used to enrich or improve soil health or related matters

    C. Forage crops – used for animals to graze, often also cover crops

3. **[AG] Example**: Common practices:

    A. In the Northeast US, crops are rotated often in periods of 4 or 5

    B. Crops can be planted in all four seasons – winter mostly cover crops

4. **[AG] Exposition**: Cover crops serve many purposes

    A. increasing carbon/nitrogen ratio (see illustration in background section)

    B. prevent soil erosion and degradation

    C. maintain ecological structure (bacteria, insects, etc.)

3.2 Mathematical Modeling

1. **Example:** How long does it take an object to fall to the ground?

   A. determine the appropriate variable / parameter (à la Galileo) – height of object

   B. model this relationship in the most simple way – a function of one variable

   C. collect/simulate/chart data for various objects/heights (see appendix 5)

0.5
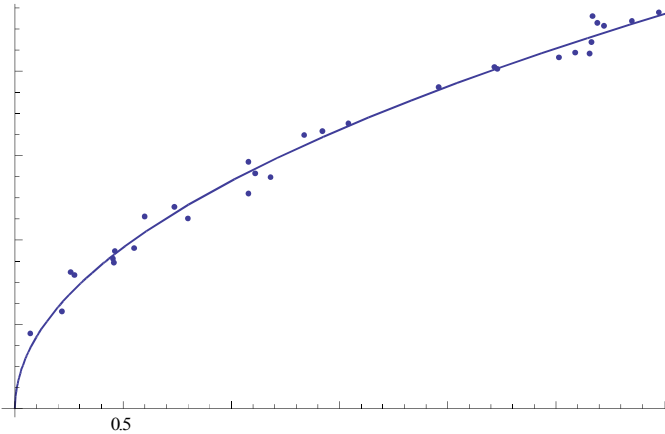
   D. use mathematics to find $t \approx c \sqrt{h}$, where c is a constant that depends on the units involved

      ○ In particular, $c \approx 0.54$ for SI units (meters and seconds)

   E. **[BM,AM]** Ask students to make a more careful analysis of this model

      i. Hopefully they suggest aspects of the physical subject of the model that are not included in the model (e.g. friction from the air – drop a piece of paper vs. a pencil).

      ii. But, compare the conclusion derived from the data to the prevailing theory that frictionless falling is dictated by $h(t) = h(0) - \frac{1}{2} g t^2$.

Note that solving $h(t) = 0$ gives the parabola indicated

2. **Exposition**: Describe the cyclic process of modeling – simplification and representation

   A. Explain the four steps in detail, especially emphasizing the translations between each of the four steps – from real world to mathematics, from data to revisions of the model, etc.
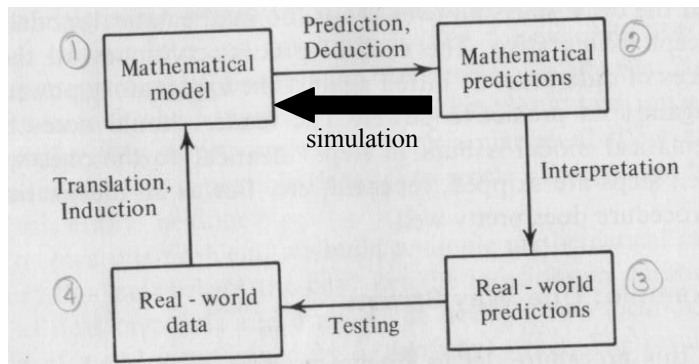
   B. Ask students to consider real world models – they could discuss, brainstorm, etc.

      ○ *Suggestions: political election prediction, weather predictions, building bridges, etc.*

C. Introduce simulation as a way to bypass the "real world" side of the model

- *See additional arrow added to diagram, labeled "simulation"*

3. Discuss with students how biological and especially ecological models are difficult or unique compared to models in physics, chemistry, etc.

A. Ecological systems cannot be subdivided – observations on one scale are not independent of the other scales (cells vs. organs, single animals vs. groups, one species vs. ecosystem)

B. Models simplify the structure of the real world, and in ecological systems, disregarding small scale (e.g. bacteria) or large scale (e.g. global warming) aspects may not be valid.

C. Ecological systems contain structure not well defined – notions like "species" are in dispute, and objects and objectives of study are often not inherently recognizable.

D. Ecological systems are sensitive to various conditions that cannot be known or measured. Small fluctuations at one time, in one way, can lead to drastic variation in outcome.

   i. **[BM, AM]** What is a mathematical system that may or may not be sensitive to initial conditions? Most likely answer: differential equation(s).

   ii. **[AG]** What kinds of small scale items might destabilize a crop rotation? One possible answer: bacteria.

4. For this lesson, the model is a mathematical structure defined as follows:

A. a set of crops, each with three attributes:

   i. the type of crop (e.g. cash crop)

   ii. the soil properties of the crop (e.g. nitrogen fixer)

   iii. the set of forbidden neighbors of a crop

B. a region in the plane, divided into smaller subregions

C. a function assigning to each subregion an ordered list of crops (i.e. the schedule)

5. **[AM]** This can be presented as a tuple (S,R,f) where R is a set of subregions, S is a set of crops which are themselves ordered triples, and f is the scheduling function. Like graph G = (V,E).

6. To this structure, we also impose rules or parameters for evaluation (forbidden neighbor needs to have a meaning, crop yields, soil health, etc. can be parametrized or evaluated).

3.3 Block Designs

1. **Exposition**: Define block design

A. Base set S, set of blocks **B**, each member B of **B** is a subset of S

B. give small example, e.g. S={1,2,3,4, **B**={{1,2,{3,{1,3,{1,2,4

C. Define parameter v = |S|, b = |**B**|

- **[AG/BM] note:** |X| is the number of elements of the set S – illustrate w/example

D. Uniform: k = |B| when B in **B** all have the same size

    E. Regular: Define r = # B in **B** that contains a given s in S (if this is constant for all s)

    F. Complete: k = v (and this implies b = r)

2. **[AG/BM] Exercise**: Demonstrate or explain why if k = v, then b = r.

3. **[AM] Exercise**: Prove the formula kb = vr.

4. **Exposition:** Define Latin square

    A. a special complete block design where blocks are ordered and arranged in a grid

    B. **Example**: Latin square of order 4

$$
\begin{array}{cccc}
1 & 2 & 3 & 4 \\
2 & 1 & 4 & 3 \\
3 & 4 & 1 & 2 \\
4 & 3 & 2 & 1
\end{array}
$$

    C. **[AG/BM] Exercise**: Give examples of Latin squares in real life. Common answer: Sudoku.

    D. **[BM/AM] Exercise**: Prove there are 2 Latin squares with 123 as the top row. How many 3x3 Latin squares are there total? What about 4x4?

    E. **[AM] Exercise**: Define isomorphism of block designs. Apply this to Latin squares – give a reasonable interpretation of what an isomorphism does to rows and columns of a Latin square. Prove that every isomorphism class of Latin squares has a representative whose first row and first column are both in order (1234....n). Demonstrate, however, that there are non-isomorphic Latin squares (by example).

5. As a design, a Latin square is very restrictive – uniform, regular, complete, ordered.

6. **Examples:** Appendix 6 contains examples of various types of designs

7. **Exposition**: Discuss the application of designs to our model

    A. Use appendix 2 to illustrate the differences between regular and non-regular

    B. Emphasize here that there is benefit – it is not so good to just randomly generate a rotation

    C. Brainstorm or list shortcomings of block designs as techniques for designing rotations

       ◦ Represents each field's crops well, but no ordering for crops, no repetition allowed

    D. Consider whether any ad hoc methods might help.

3.4 Finite State Machines

1. **Exposition**: A different idea – Finite State Machines

    A. Recap what our model's structure is – an ordered list of crops for each plot in a field

    B. There is no "wrong" approach to constructing a rotation in this model – FSM is good too

    C. Contrasting and combining these two approaches is beneficial, we can use them together

2. **Exposition**: Define FSA, FSM, etc.

    A. Formal structures that mimic and model real life computational and mechanical instruments

    B. Informally, they are defined as machines that have states and move between states

        i. allows our model to repeat states (states means what is planted at a given time)

        ii. accounts for ordering and some other aspects block designs may not capture

        iii. harder to impose structure on FSA/FSM, harder to see global properties (e.g. yield)

        iv. more flexibility in a methodology means there are more variables to worry about

    C. Formally, a Finite State Automaton can be described by:

        i. states, including the initial state

        ii. rules for transitioning between states

        iii. (optional) input, given at the beginning, to determine how rules are followed

    D. A Finite State Machine allows input at any time, allowing dynamic transition control

    E. A Deterministic FSA or FSM has unavoidable "final" states that end the computation

3. **[AM]** Introduce a more formal model (the tuple of each definition)

4. **Exercise**: Design a DFSA representing a soda machine of some sort

5. **Examples**: Appendix 4 gives some small examples of FSM for crop rotation

6. **[BM/AM] Exercise**: Prove that any finite state automaton will either wind up in a final state (or a state that is repeated forever), or in a cycle that is repeated forever. *This proof could be solicited verbally from a small class – the key (and only) idea to the proof is that if neither of these occurs, then the machine can never repeat itself (once it repeats, this is the desired cycle). If the machine never repeats, it does not have finitely many states. This proof, for **[AM]** students could also be extended – it can be proved for FSM when the input is periodic.*

7. **Exposition**: Refer to appendix 2 again and continue discussing pros & cons of methods

    A. Now compare & contrast various FSM and block designs.

    B. Ask students to examine Appendix 7, which contains a partial construction of a rotation design and some of the underlying methods. Students should be challenged, using data from Appendix 1, to fill in missing parts of this design as best as they can.

    C. They should also then discuss and analyze the strengths and weaknesses of this rotation. This should lead them into the major assignment for this module – constructing their own rotation schedules.

4. Assignment, Exercises, Assessment, and Follow-Up

4.1 Assignment & Assessment

The table in Appendix 1 describes a set of crops by family and their characteristics for inclusion in a rotation using our model. The subsequent appendices 1A, 1B, and 1C reduce these tables in different ways, tailored to suit different levels of mathematical and agricultural backgrounds.

Students should be assigned, in some way or another, the task of creating a crop rotation. This task can be done in class or as a take-home assignment, but it is lengthy and is probably better as a take-home assignment unless one can set aside at least an hour of class time. Students can devise rotations in groups, large or small, or individually. Groups of 3-4 are recommended.

Students should be instructed to use their knowledge – depending on the course, this includes mathematical or agricultural knowledge, as well as the model provided in this module and the methods suggested (designs and FSM) to construct a crop rotation. Students should submit not just their rotation, but a summary of the methods they used, including any block designs ans FSM the used and some explanation of how/why they were constructed as such. Unless necessary for some reason, students are not required to use both methods – but of course, students should not be permitted to simply fill in a rotation arbitrarily.

A universal grading rubric for this assignment might be impossible to construct, but students should generally be graded based on the following criteria (modified as necessary). Relevant learning objectives are given in parentheses. The entire assignment is, literally, item (7).

1. Modeling – does the solution show an understanding of modeling? (1,2)

2. Agriculture – does the design reflect the structure of the model and the agricultural parameters of that model? (1,5,9)

3. Methods – does the solution utilize mathematics to create a rotation? (5)

4. Methods – does the design implement these methods properly and reasonably? (5,6,9)

5. Mathematics – are the mathematics of the design consistent and correct? (5,6)

Each group can decide whether the agricultural endeavor is a commercial, community, or private when selecting crops and building a rotation. Their assignment should include some explanation of their methods, which should reflect their goals (high yield of commercial crops, balanced yields, etc.)

For **[AG]** classes, an additional component is a focus on seasons as parameters for the model. These students can be expected to submit designs that rotate crops effectively based not only on basic parameters like not depleting nitrogen, but putting crops in the correct season. It is recommended that, despite being present in exposition, these students simplify the model by ignoring parameters of forbidden adjacency. The mathematics become more complicated / difficult otherwise.

For **[BM/AM]** classes, seasonal restrictions can be ignored. In **[AM]** classes instead, the adjacency restrictions should be considered.

Crop season noted are harvest times – rotations should be labeled accordingly (i.e. the "winter" row of some rotation indicates the crop planted with the intention of being harvested in winter, not the crop planted in winter to be harvested in spring). This distinction is important to **[AG]** classes, and if mistaken, would lead to an "off by one" error in the rotation (or for rotations designed using transitions that depend on the season, more serious errors). Math classes can simplify by ignoring seasons.

Adjacency restrictions are given in the column marked "Interactions." This has serious effects on the assignment and how students could go about solving it. First, the plots of land are no longer independent and analysis requires consideration of what happens not just globally (how much total of each crop do we get each year?) but also of specific local analysis of a particular plot (what are the neighbors of this plot?). It also requires the geometry of the field to be considered. Placing a crop with forbidden neighbors in a corner minimizes the number of neighbors. Grouping forbidden crops away from one another may help avoid adjacency issues. Students familiar with graph theory might consider the plots as nodes in a graph, where one attempts to avoid edges between such forbidden pairs.

While it is easy to avoid, with limited numbers of crops with interactions, models with parameters closer to real-world conditions will have more limitations and more complex interactions. Students should not be encouraged to necessarily just "try to make it work arbitrarily" – this advice becoming somewhat of a theme. Students should find a systematic way of avoiding these (e.g. finding a crop rotation using FSM and then "shifting" or "staggering" it in ways that avoid forbidden adjacency).

But, due to the increased complexity, it is recommended that students' mathematical background is not advanced, that adjacency be discussed in the presentation of the model and of the crops, but that no adjacency be required in the assignment. If included, it is important for students to consider whether a model that violates the adjacency restrictions is allowed, but penalized, or if these rules are absolute.

Note: Although it is not suggested that this idea be introduced to students, it is possible that even the shape of the plots could be changed. Each plot could be subdivided into right triangles, thus reducing the number of adjacent plots for each individual to three. Students who use this method should not be discouraged or forbidden, and advanced students may find that plane tilings / tessellations are also an interesting mathematical tool in designing the geometry of a field in which we have crop rotations.

4.2 Exercises

In addition to the large project/assignment described in the previous section, it may be useful in some courses to ask students to complete (in class or at home, in groups or individually) some of the exercises suggested in the lesson outline or background sections of this module. Here may be better opportunities for individual assessment, as some of these are more easily completed by single students. These may be compiled in some way and assigned, or may simply be suggested or assigned verbally, during the course of a lecture or discussion.

4.3 Follow up

Students can present their rotations, either as just lists – on poster-board, chalkboard, overhead, etc. – or with a detailed explanation presented orally. Students who do present their methods should be directed to make clear and precise diagrams where needed, an important part of presenting mathematical models and other mathematical ideas.

Students (or just the instructor) can offer analysis of rotations. Students could either analyze and critique their own rotations, or other groups. They can either offer ad hoc feedback and analysis, or they could be provided with a rubric or set of guidelines to guide this process.

It may be important to discuss precisely what a good rotation is – what parameters, measurements, properties, etc. define a good rotation schedule. All of the parameters should be discussed, and various ways of measuring the effectiveness of a rotation could be discussed and used in their analysis.

The feedback in this part of the modeling addresses learning objective 3 (and potentially 4). The student presentations address learning objective 8. The analysis itself addresses objective 9.

If computer programming is in the course, students can use computers to simulate the rotations and tabulate their output. They can also track the soil properties, nitrogen levels, etc. with additional parameters (either tracking real world variables, i.e. the actual nitrogen level one could measure, or by tabulating the number of nitrogen depleting crops that have been planted in a given time interval).

During this process, it is important to repeat and reinforce the discussion of the feedback portion of the process of modeling – students can suggest or even implement improvements to their rotations. Real world or simulated output, or even just further mathematical analysis, can provide insight into how to revise and improve their methods.

To wrap up the final discussion or lecture, focus on the overall process of modeling – how a real world situation gave us a model, how math was applied, how the agricultural parameters influenced our application of these mathematical methods, and then how the rotations were constructed and analyzed, eventually leading to revision of the rotations or even of the model itself.

5. References & Further Reading

5.1 Agriculture & Horticulture

[BUL] Bullock, D.G., *Crop Rotation*, Critical Reviews in Plant Sciences, 1992.

[DAB] Dabney et al., *Advances in Crop Management*, 2008.

[HAA] Haas, G., H. Brand, M. Puente de la Vega, *Nitrogen from Hairy Vetch (Vicia villosa Roth) as Winter Green Manure for White Cabbage in Organic Horticulture*, Biological Agriculture and Horticulture, 2007.

[HOO] Hoorman, J.J., et al., *Sustainable Crop Rotations with Cover Crops*, Fact Sheet: Agriculture and Natural Resources, The Ohio State University Extension, 2009.

[ØST] Østergaard HS, *Catch Crops*, Baltic Deal, www.balticdeal.eu, 2010 .

[REE] Reeves D.W., *Cover Crops and Rotations*, CRC Press, Inc, 1994

[SUL] Sullivan P., *Overview of Cover Crops and Green Manures: Fundamentals of Sustainable Agriculture*, ATTRA, www.attra.ncat.org, 2003

5.2 Block Design

[BET] Beth, Thomas, et al., *Design Theory*, Cambridge University Press. 2nd ed., 1999.

[CAM] Cameron, P. J. & J. H. van Lint, Designs, Graphs, Codes and their Links, Cambridge University Press, 1991.

[HAL] Hall Jr., Marshall, *Combinatorial Theory*, 2nd ed., Wiley-Interscience, 1986.

[HUG] Hughes, D.R. & E.C. Piper, *Design theory*, Cambridge University Press, 1985.

[STI] Stinson, Douglas R., *Combinatorial Designs: Constructions and Analysis*, Springer, 2003.

[STR] Street, Anne P. & Deborah J. Street, *Combinatorics of Experimental Design*, Oxford University, 1987.

[WAL] Wallace, W.D., *Combinatorial Designs*, Marcel Dekker, 1988.

5.3 Finite State Machines

[CAR] Carroll, J. & D. Long, *Theory of Finite Automata with an Introduction to Formal Languages*, Prentice Hall, 1989.

[KOH] Kohavi, Z., *Switching and Finite Automata Theory*, McGraw-Hill, 1978.

[GIL] Gill, A., *Introduction to the Theory of Finite-state Machines*, McGraw-Hill, 1962.

[GIN] Ginsburg, S., *An Introduction to Mathematical Machine Theory*, Addison-Wesley, 1962.

5.4 Mathematical Modeling

[ARI] Aris, Rutherford, *Mathematical Modelling Techniques*, Dover, 1978.

[BEN] Bender, E.A., *An Introduction to Mathematical Modeling*, Dover, 1978.

[CLE] Clements, R.R., *Mathematical Modeling: A Case Study Approach*, Cambridge University, 1989.

[LIN]   Lin, C.C. & L. A. Segel, *Mathematics Applied to Deterministic Problems in the Natural Sciences*, SIAM, 1988.

[GER] Gershenfeld, N., *The Nature of Mathematical Modeling*, Cambridge University, 1998.

[ROB] Roberts, Fred. *Discrete Mathematical Models with Applications to Social, Biological, and Environmental Problems*, Prentice-Hall, 1976

[ROB2] Robertson, David et al., *Eco-Logic, Logic-Based Approaches to Ecological Modelling*, Massachusetts Institute of Technology, 1991.

Appendix 1 – Table of Crops

Crop Family	Common name crops	Type (Season*)	Properties	Interactions**	Notes
Grains **N depletion** [x]	corn (maize) cereal rye annual ryegrass sorghum-Sudan grass oats	cash, forage forage, cover forage, cover forage, cover cash, forage, cover	1,4-8,11-13 1,4-8,11 1,4-8,10 1,3-8,11,13		
Brassicacea	cabbage mustard kale turnip	cash (F) cash, cover cash (Sp, F, W) cash, cover	8, deters worms deters worms deters worms	dill, beets, lavend	
Fabaceae	clover vetch alfalfa cowpea mungbean	cover cover forage, cover cover cash, cover	2,7,9-11 2,7,10 2 2,3,6,7,10 2,6,10	garlic garlic garlic garlic garlic	
Solanaceae **must not be in same field within 3 years**	tomatoes potatoes peppers eggplants	cash cash cash cash		dill, beets pumpkins	
Cucurbitaceae	cucumbers melons squash pumpkins	cash (Su) cash (Su) cash (Su, F, W) cash (Su, F)	deters aphids	potato	
Herbs [y]	dill lavender calendula	cash (F) cash (Su,F) cash/cover (Su,F)	8,9	tomato, cabbage cabbage	medicinal
Aliaceae	garlic onion leek scallion	cash (Sp) cash (Sp) cash (Sp) cash (Sp)		all beans	
Betaceae	beets	cash (Sp,W)		tomato, cabbage	cool weather

Key:

Sp	Spring		6	prevents soil erosion	
Su	Summer		7	recaptures nutrients	
F	Fall		8	weed suppressant	
W	Winter		9	attracts beneficial insects	
1	increases carbon : nitrogen ratio		10	heat/drought tolerance	
2	increases nitrogen		11	wet soil tolerance	
3	easy non-herbicidal removal		12	cold tolerance	
4	compaction reduction (deep roots)		13	nurse crops	
5	grazing crop				

* Crop season noted are harvest times – rotations should be labeled accordingly. No label means all seasons.

** These are the "forbidden adjacency" parameters. Crops cannot be adjacent to other crops listed in this column.

x Notes in this column apply to all members of the family.

y Herbs are not all from the same family, but are listed together here as a matter of simplicity.

Appendix 1A – Table of Crops for use by agriculture/horticulture [AG] course

Crop Family	Common name crops	Type (Season*)	Properties	Notes
Grains **N depletion** [x]	corn (maize) cereal rye annual ryegrass sorghum-Sudan grass oats	cash, forage forage, cover forage, cover forage, cover cash, forage, cover	1,4-8,11-13 1,4-8,11 1,4-8,10 1,3-8,11,13	
Brassicacea	cabbage mustard kale turnip	cash (F) cash, cover cash (Sp, F, W) cash, cover	8, deters worms deters worms deters worms	
Fabaceae	clover vetch alfalfa cowpea mungbean	cover cover forage, cover cover cash, cover	2,7,9-11 2,7,10 2 2,3,6,7,10 2,6,10	
Solanaceae **must not be in same field within 3 years**	tomatoes potatoes peppers eggplants	cash cash cash cash		
Cucurbitaceae	cucumbers melons squash pumpkins	cash (Su) cash (Su) cash (Su, F, W) cash (Su, F)	deters aphids	
Herbs [y]	dill lavender calendula	cash (F) cash (Su,F) cash/cover (Su,F)	8,9	medicinal
Aliaceae	garlic onion leek scallion	cash (Sp) cash (Sp) cash (Sp) cash (Sp)		
Betaceae	beets	cash (Sp,W)		cool weather

Key:

Sp	Spring		6	prevents soil erosion
Su	Summer		7	recaptures nutrients
F	Fall		8	weed suppressant
W	Winter		9	attracts beneficial insects
1	increases carbon : nitrogen ratio		10	heat/drought tolerance
2	increases nitrogen		11	wet soil tolerance
3	easy non-herbicidal removal		12	cold tolerance
4	compaction reduction (deep roots)		13	nurse crops
5	grazing crop			

* Crop season noted are harvest times – rotations should be labeled accordingly. No label means all seasons.

x Notes in this column apply to all members of the family.

y Herbs are not all from the same family, but are listed together here as a matter of simplicity.

Appendix 1B – Table of Crops for use by Basic Math [BM] course

Crop Family	Common name crops	Type
Grains **N depletion** [x]	corn (maize) cereal rye oats	cash, forage forage, cover cash, forage, cover
Brassicacea	cabbage mustard kale turnip	cash cash, cover cash cash, cover
Fabaceae	clover vetch mungbean	cover cover cash, cover
Solanaceae **must not be in** **same field** **within 3 years**	tomatoes potatoes peppers eggplants	cash cash cash cash
Cucurbitaceae	cucumbers melons squash pumpkins	cash cash cash cash
Herbs [y]	dill lavender	cash cash
Aliaceae	garlic onion	cash cash
Betaceae	beets	cash

x Notes in this column apply to all members of the family.

y Herbs are not all from the same family, but are listed together here as a matter of simplicity.

Appendix 1C – Table of Crops for use by Advanced Math [AM] course

Crop Family	Common name crops	Type	Properties	Interactions*
Grains	corn (maize) cereal rye annual ryegrass sorghum-Sudan grass oats	cash, forage forage, cover forage, cover forage, cover cash, forage, cover	3 1,3 1,3 1,3 1,3	
Brassicacea	cabbage mustard kale turnip	cash cash, cover cash cash, cover		dill, beets, lavender
Fabaceae	clover vetch alfalfa cowpea mungbean	cover cover forage, cover cover cash, cover	2 2 2 2 2	garlic garlic garlic garlic garlic
Solanaceae **must not be in** **same field** **within 8 turns** ˣ	tomatoes potatoes peppers eggplants	cash cash cash cash		dill, beets pumpkins
Cucurbitaceae	cucumbers melons squash pumpkins	cash cash cash cash	deters aphids	potato
Herbs ʸ	dill lavender calendula	cash cash cash/cover		tomato, cabbage cabbage
Aliaceae	garlic onion leek scallion	cash cash cash cash		all beans
Betaceae	beets	cash		tomato, cabbage

Key:
1 increases carbon : nitrogen ratio
2 increases nitrogen
3 nitrogen depletion – unless labeled (2), all crops deplete nitrogen to some amount! These are just the worst offenders and nothing but those labeled (2) should follow.

* These are the "forbidden adjacency" parameters. Crops cannot be adjacent to other crops listed in this column.

x Notes in this column apply to all members of the family.

y Herbs are not all from the same family, but are listed together here as a matter of simplicity.

Appendix 2 – Example Rotation with Bad Outcomes

| Plot 1	Plot 2	Plot 3	Plot 4	Plot 5
corn	vetch	beans	corn	eggplant
vetch	beans	rye	rye	vetch
beans	corn	vetch	potatoes	
rye		corn	vetch	
potatoes		potatoes		
eggplant				

This example should be considered carefully by students. Suggest, if necessary, that they look at the vetch. Even by brute-force working out the yields for each year, they should find that in the eighth year, all but one plot is growing vetch – which means very little to sell or eat. Similar coincidences may be observable for other crops in this rotation.

This example may allow students to discuss why a more regular block design would work, or why using finite state machines that have the same (or at least similar) periods would be useful. In the discussion, it may be helpful to discern what is meant, then, by using FSMs with similar periods (or block designs that are partially or totally regular). In this way, they would have periods that have many prime factors in common (e.g. 3, 6, and 12). This would provide a more regular cycle in which one could avoid having too many of the same crop most/all of the time.

Another way to allow for different cycle lengths without too much of one crop at any given time is to require the opposite, that all cycle lengths be relatively prime. Here the problem is not avoided, but it is minimized in some sense. This could be asked of students as a question – if none of the cycles are the same length*, but vetch must be planted at one point in each cycle, how do you minimize the frequency with which the yield is overwhelmingly or entirely vetch? This causes the most uniform sort of distribution of all possible outcomes. It is (hopefully) evident that part of the problem with the example given is that Plots 1, 4, and 5 tend to "sync up" very often.

*Note: Here it may be important to note that it is not clear what "same length" means – indeed, periodicity is really a matter of the period<u>s</u> of a periodic function – which are all multiples of the fundamental period. For this reason, plots 4 and 5 are clearly of the same period, which is 4. To achieve the largest possible period for the entire field, relatively prime cycle lengths would be used. Students may have differing interpretations of the question itself, which may be an interesting point for group discussion.

Appendix 3 – Various Representations of the Modeling Process
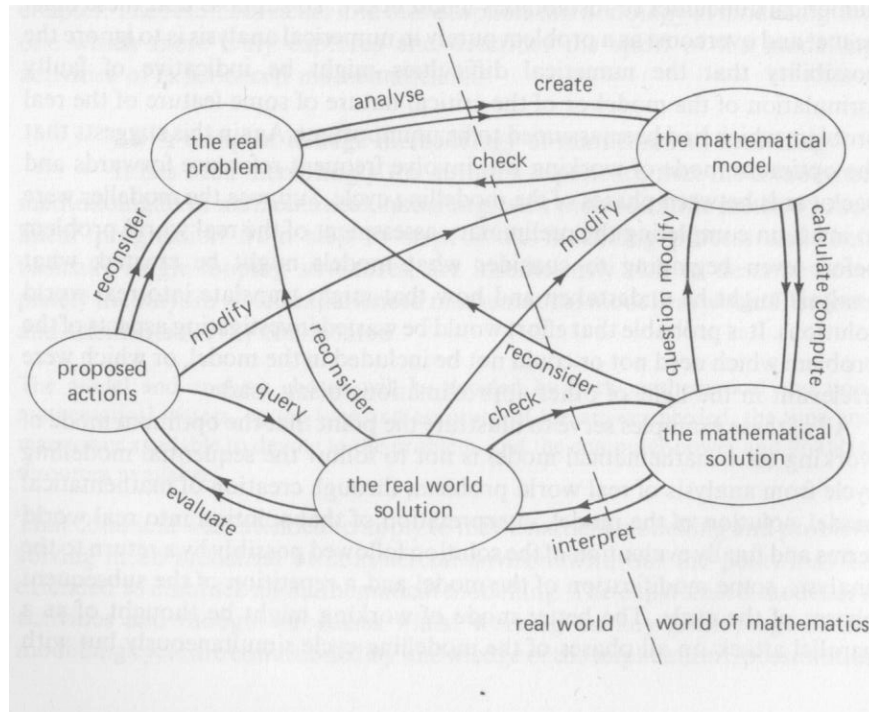
An illustration that presents a more robust network of modeling. The double-arrows represent the normal "cycle" of modeling described and illustrated in the text of this module, while other arrows represent various forms of feedback – in particular, the "question modify" feedback that occurs between the "mathematical solution" (which may also include the results of computations and simulations). Taken from [CLE] p 22.
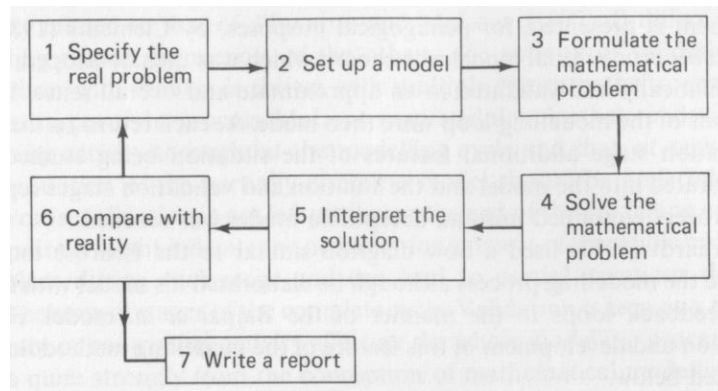
An illustration presenting a similar version of the 4-step procedure described in the text of the module (here some of the arrows have become boxes). This version was proposed in 1978 by Penrose, and was the first presentation of such a simplified notion of modeling. Taken from [CLE] p 15.
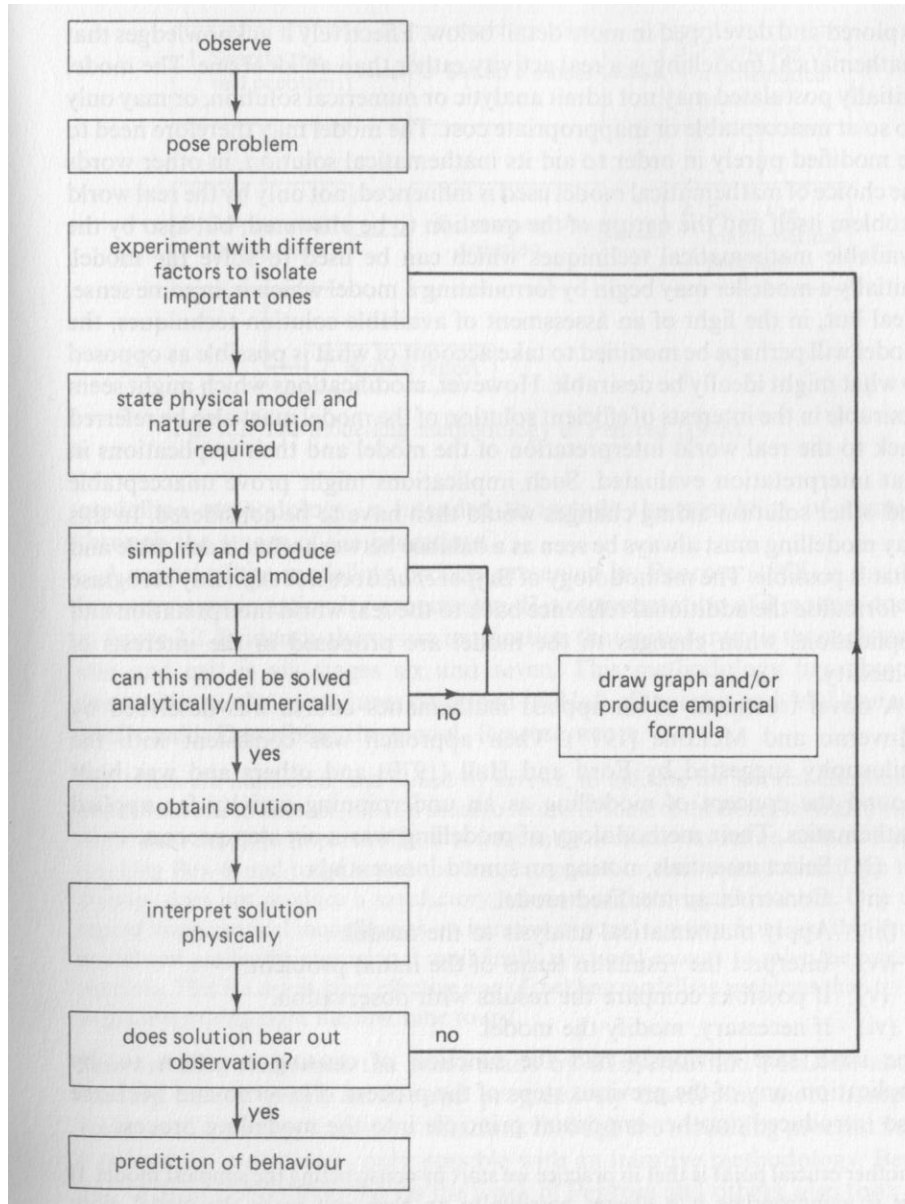
A more complex diagram illustrating the process of mathematical modeling, introduced by Bajpai et al. in 1975. This diagram was designed in relation to modeling in engineering/physics. Although introduced in 1970 and 1972 by Hall and Ford, this diagram clearly illustrates a series of steps used to attempt to produce from observation a prediction, and within these steps is included a method for refinement of the model. Taken from [CLE] p 13.

Appendix 4 – Examples of very simple FSA and FSM representing crop rotations

Example 1 – a finite state automaton that is very rudimentary. This may be subject to analysis, as it may not be a very good example. The states are, themselves, entire layouts of the field, with all nine plots filled in in a somewhat arbitrary way. This method would be not much more advanced that just filling in a list for each plot arbitrarily. The corresponding list is included below the FSA diagram.

ryegrass	vetch	corn
cowpea	cabbage	clover
turnip	onion	beet

→

turnip	beet	vetch
cowpea	clover	corn
ryegrass	cabbage	onion

cabbage	turnip	dill
peppers	potatoes	beets
vetch	beets	alfalfa

←

beet	corn	cowpea
vetch	onion	vetch
corn	ryegrass	clover
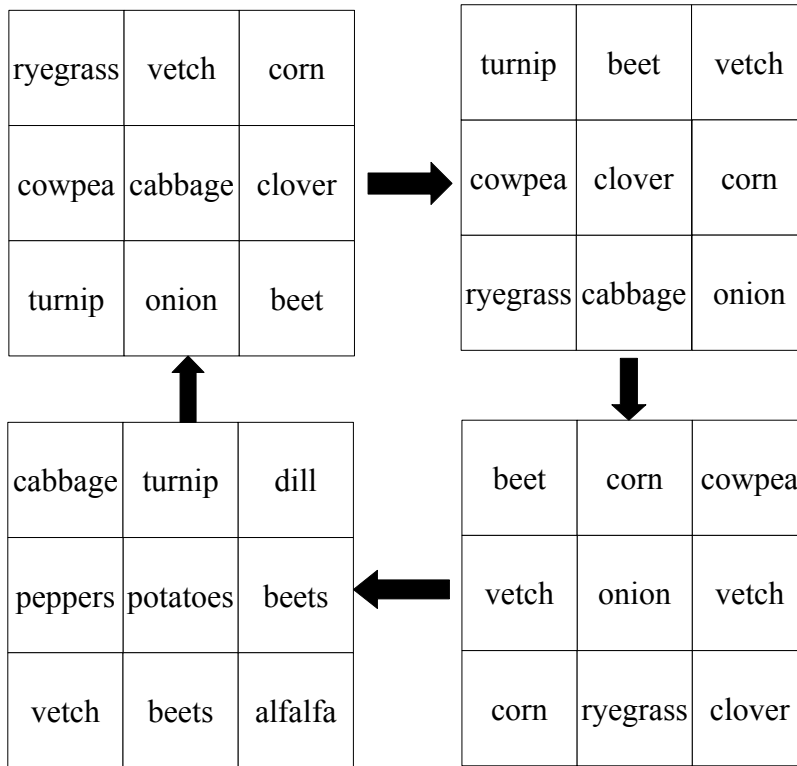
Plot (1,1)	**Plot (1,2)**	**Plot (1,3)**	**Plot (2,1)**	**Plot (2,2)**	**Plot (2,3)**	**Plot (3,1)**	**Plot (3,2)**	**Plot (3,3)**
ryegrass	vetch	corn	cowpea	cabbage	clover	turnip	onion	beet
turnip	beet	vetch	cowpea	clover	corn	ryegrass	cabbage	onion
beet	corn	cowpea	vetch	onion	vetch	corn	ryegrass	clover
cabbage	turnip	dill	peppers	potatoes	beets	vetch	beets	alfalfa

Students could be asked to analyze this rotation schedule. In this case, they should point out various strengths and weaknesses, like that cowpeas and vetch occur in sequence in (1,3), which may be unnecessary for keeping nitrogen levels up. Students can find other examples of good and bad qualities of this model according to various parameters.

The next example rotation is illustrated in three stages. The first includes crop selection – crops are selected based on some desired output or yield (if you want tomatoes, pick tomatoes). Then, seasons in which to grow the crops are selected (this FSM will use seasons as its input – the input alphabet is the set of seasons, and presumably they are input correctly to correspond to real life). Each season in which the crop is planted will be an incoming arrow, and the subsequent season an outgoing arrow.
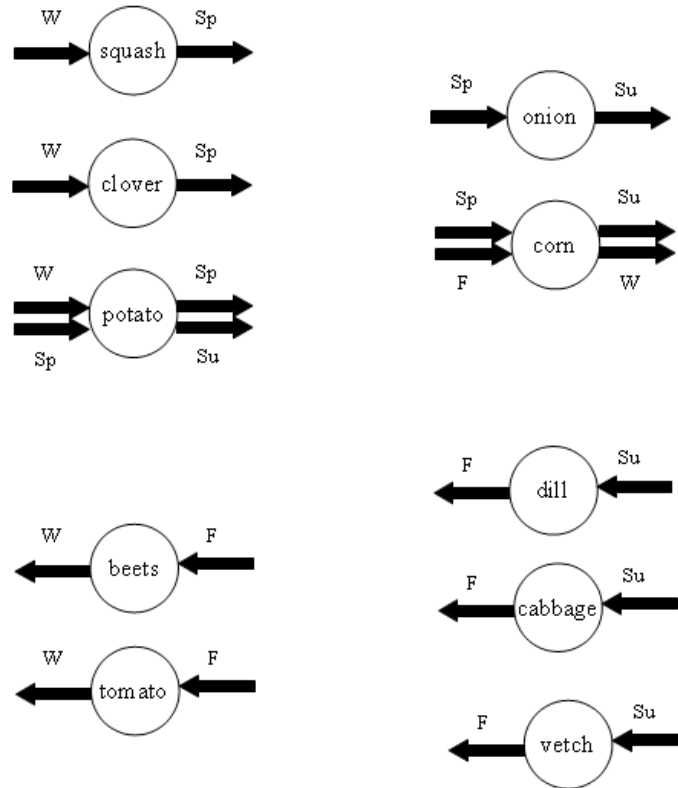
Note that these are mostly grouped by season, to allow for the easiest possible connections (in terms of drawing/layout). In a more complex drawing, each crop will have more inputs and outputs, which will lead to more complex rotations.

In the next stage, these are connected in ways that are consistent – seasons must match along each arrow, and altogether the rotation should obey the parameters being used. Here, most crops have only one or two outgoing edges, which means that at a specific season, if we are on a crop (say cabbage) and the input is "spring" we have no option. However, because of our design, we know the only input would be "fall" because the previous input must have been "summer" (that is the only incoming edge for cabbage). This is totally fine – we don't need to add extra meaningless edges that will never get used, so long as we know our input will never use them.
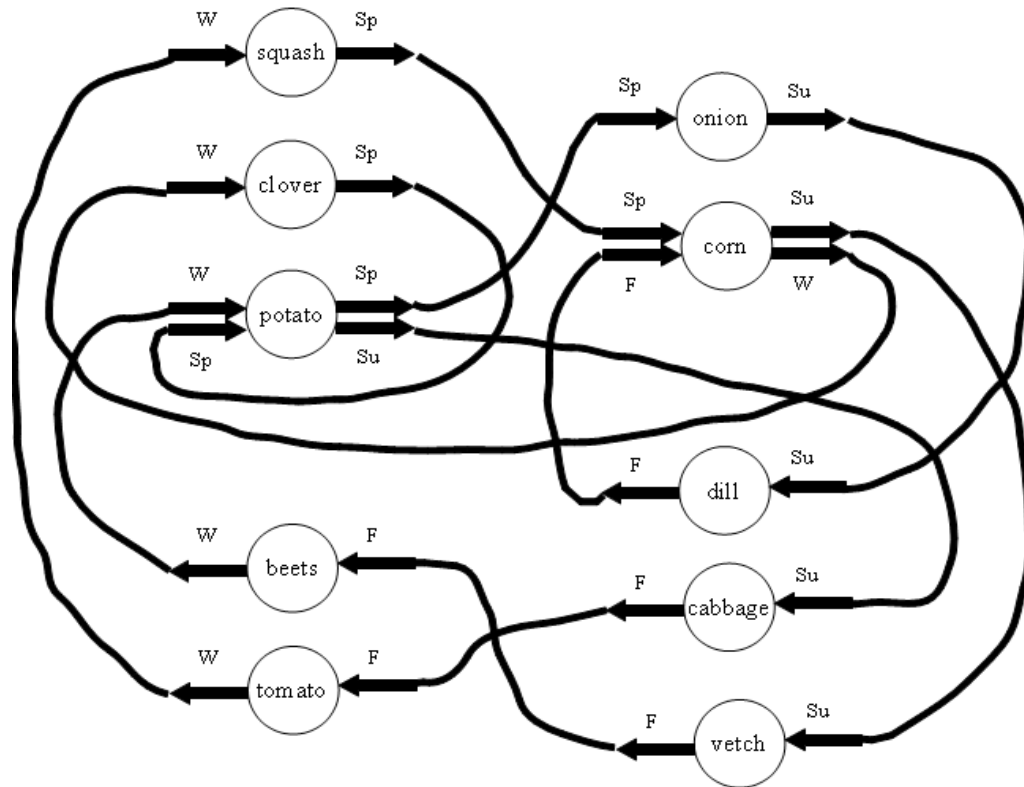
The result of this is a single large cycle of length 12 which is:

squash, corn, vetch, beets, potato, onion, dill, corn, clover, potato, cabbage, tomato

Now, this might not be ideal in some situations or according to our particular agricultural parameters. In particular, potato is repeated twice in too short an interval. There are a few ways to fix this. One is to make another set of rotations:
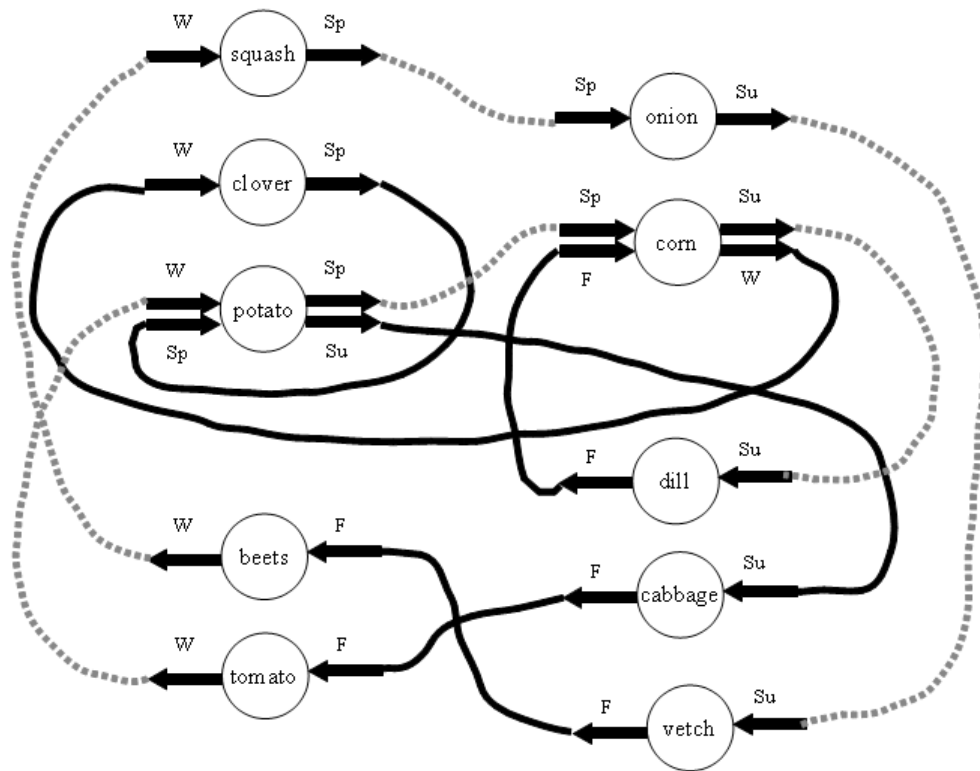
Note that the arrows that have changed are shaded differently. All other connections are the same as in the previous FSM, but by switching a few connections, the FSM changes drastically.

In coming up with this rotation, we notice that there is a cycle of length 4, which will give us something that repeats every year, a very short cycle. The following diagram highlights the two disjoint cycles in the second FSM we have constructed:

Now the two types of arrows show the two disjoint cycles in the FSM (assuming Sp-Su-F-W input).

In fact, in the larger cycle (length 8) we have also created another shorter cycle *in the graph* – dill and corn are both connected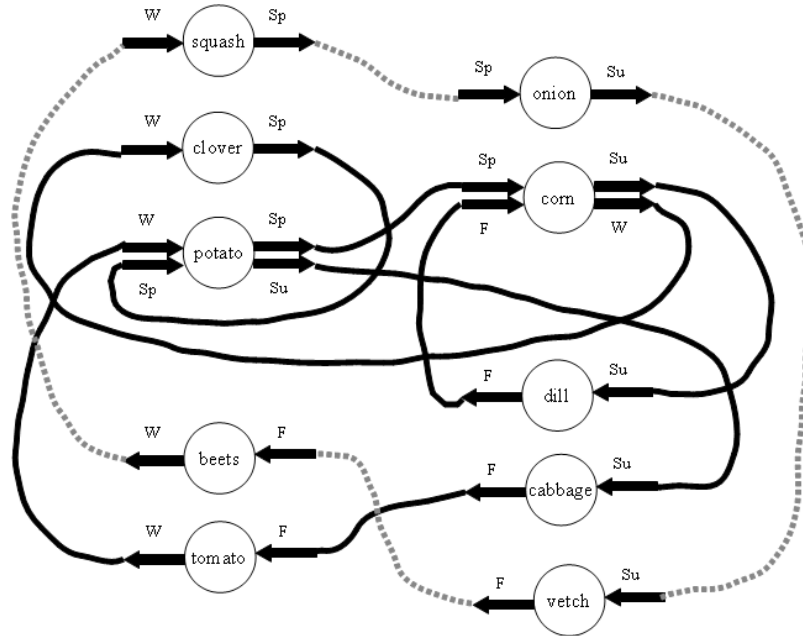 to the other. Although the seasons do not dictate that they simply repeat one after the other – instead, this produces (in the graph) a short cycle (length 2) and in the rotation, gives us two corn harvests in a short span of time (for a single plot). So, generally speaking, one might want to avoid short cycles or at least watch for them. The three cycles we have generated are as follows:

Season	Cycle 1	Cycle 2	Cycle 3
Sp	squash	squash	potato
Su	corn	onion	corn
F	vetch	vetch	dill
W	beets	beets	corn
Sp	potato		clover
Su	onion		potato
F	dill		cabbage
W	corn		tomato
Sp	clover		
Su	potato		
F	cabbage		
W	tomato		

We still need to worry about the potatoes – they repeat too often. We can "tweak" the rotation using ad hoc methods. We can simply replace one second occurrence of potato with onion (which goes in the same season). And as for the corn-dill-corn part of the cycle, we can replace dill with vetch in order to bring nitrogen levels up. This sort of ad hoc tweaking is sometimes an important part of this kind of construction, since real world parameters are often too complex to be fully satisfied in a simple mathematical construction. Our final set of possible rotations (changes in bold) is:

Season	Cycle 1	Cycle 2	Cycle 3
Sp	squash	squash	potato
Su	corn	onion	corn
F	vetch	vetch	**vetch**
W	beets	beets	corn
Sp	potato		clover
Su	onion		**onion**
F	dill		cabbage
W	corn		tomato
Sp	clover		
Su	**onion**		
F	cabbage		
W	tomato		

So, these two FSM generates three crop different cycles of crops to use. Multiple such FSM can make many more lists, which can be used one per plot, or one for several plots. In our case we will use these three for all our plots, but in a way that respects reasonable conditions – we will stagger this list *and* make sure to assign adjacency in a way that avoids having neighboring crops that are forbidden – in this case, dill-tomato, dill-cabbage, beets-tomato, and beets-cabbage, which if we'd like to be succinct, can be represented by a graph of forbidden pairings:
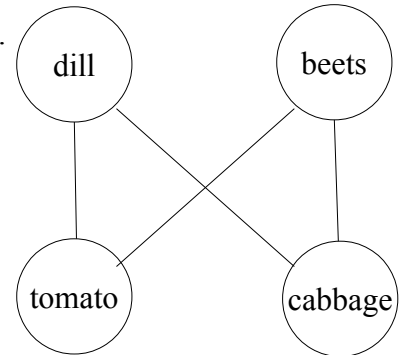
The three cycles we get from our FSM do not have particular starting points. We can choose arbitrarily, and this allows us to choose several starting points and assign different starting points in the rotation to each plot. One way of doing so gives us this (for a 3x3 plot):

	Plot (1,1)	**Plot (1,2)**	**Plot (1,3)**	**Plot (2,1)**	**Plot (2,2)**	**Plot (2,3)**	**Plot (3,1)**	**Plot (3,2)**	**Plot (3,3)**
Sp	squash	clover	potato	clover	potato	clover	squash	squash	squash
Su	onion	onion	corn	onion	corn	onion	onion	corn	onion
F	vetch	cabbage	vetch	cabbage	vetch	cabbage	vetch	vetch	vetch
W	beets	tomato	corn	tomato	corn	tomato	beets	beets	beets
Sp	squash	potato	clover	squash	clover	squash	squash	potato	squash
S	onion	corn	onion	corn	onion	corn	onion	onion	onion
F	vetch	vetch	cabbage	vetch	cabbage	vetch	vetch	dill	vetch
W	beets	corn	tomato	beets	tomato	beets	beets	corn	beets
Sp	squash	clover	potato	potato	potato	potato	squash	clover	squash
S	onion	onion	corn	onion	corn	onion	onion	onion	onion
F	vetch	cabbage	vetch	dill	vetch	dill	vetch	cabbage	vetch
W	beets	tomato	corn	corn	corn	corn	beets	tomato	beets
Sp	squash	potato	clover	clover	clover	clover	squash	squash	squash
Su	onion	corn	onion	onion	onion	onion	onion	corn	onion
F	vetch	vetch	cabbage	cabbage	cabbage	cabbage	vetch	vetch	vetch
W	beets	corn	tomato	tomato	tomato	tomato	beets	beets	beets
Sp	squash	clover	potato	squash	potato	squash	squash	potato	squash
Su	onion	onion	corn	corn	corn	corn	onion	onion	onion
F	vetch	cabbage	vetch	vetch	vetch	vetch	vetch	dill	vetch
W	beets	tomato	corn	beets	corn	beets	beets	corn	beets
Sp	squash	potato	clover	potato	clover	potato	squash	clover	squash
S	onion	corn	onion	onion	onion	onion	onion	onion	onion
F	vetch	vetch	cabbage	dill	cabbage	dill	vetch	cabbage	vetch
W	beets	corn	tomato	corn	tomato	corn	beets	tomato	beets

Note that the length of the table is LCM(4,12,8) = 24. Bad adjacency in bold (see below).

Geometrically, each cycle can be assigned to plots in the field according to this layout:

Cycle 2 (+0)	Cycle 3 (+4)	Cycle 3 (+0)
Cycle 1 (+4)	Cycle 3 (+0)	Cycle 1 (+8)
Cycle 2 (+0)	Cycle 1 (+0)	Cycle 2 (+0)

The parentheses indicate the offset in the cycle.

By laying out the cycles in this way (regardless of the way the cycles are staggered), we avoid most of the adjacency issues automatically – the only pairs of adjacent cycles are 2-1 and 2-3. Cycle 2 only contains one potential problem, beets, which cannot be adjacent to cabbage or tomato. Beets only occur in winter, and cabbage does not, so the only bad pair we need to check is beet-tomato adjacency.

However, this adjacency is unavoidable. This can serve as a number of things for instruction:

1. A place where students can analyze the rotation and say "there is a negative feature";

2. Something students could improve upon or fix in their own rotations;

3. Something students can analyze more deeply (esp. **[AM]** students). There can be significant questions like "assuming three of each cycle must occur in the layout, in some staggered way, how can we minimize the number of tomato-beet adjacency?"

This substantial "error" in the rotation is deliberate – students should not, after all, be using this rotation as too much of a basis for their own. When they construct their rotations, they should be starting from scratch even if they wind up mimicking the methods of this example (or the one in appendix 7 too). This might be a good test, in cases where students are suspected of trying to tweak this rotation without actually doing much work of their own, to see if they've fixed this problem (which requires a substantial change in the rotation.

The reason this error is so substantial can be easily isolated – beets occur every winter for cycle 2. It is unavoidable, and thus cycle 2 cannot be next to cycle 1 or 3. In this construction, some attempt is made to minimize the adjacency by putting cycle 2 in corners and placing it next to the longer cycle (cycle 1) as much as possible, so that this issue occurs less frequently.

Students should consider this example as one potential way of constructing rotations. It can be analyzed for strengths and weaknesses like the one mentioned (it is not necessarily the best or worst possible construction, and may have pros and cons).

Appendix 5 – data used for chart in example (simulated drop time)

height	drop time
0.277454	0.316615
0.454152	0.356153
1.10958	0.557436
1.95683	0.762136
2.2312	0.804299
0.600281	0.456651
0.738612	0.477814
2.58771	0.843939
1.17919	0.548234
2.85075	0.918419
1.07743	0.51068
2.66808	0.930758
0.799976	0.450971
2.21392	0.809947
2.6904	0.9151
0.257097	0.323524
0.459196	0.347605
2.65682	0.841634
2.72058	0.908641
0.0691029	0.178125
1.5414	0.676535
0.218348	0.231829
0.463968	0.373768
0.550272	0.381364
2.66399	0.869332
1.33635	0.648566
2.97359	0.939772
1.07925	0.585594
2.5144	0.833181
1.42243	0.657192

Simulated using the correct ideal formula, and adding a small random number(between -0.1 and 0.1).

Appendix 6 – Examples of block designs

This first example is a Latin square. It is complete, and thus is regular and uniform. Because it is a Latin square, the order of the rows and columns is meaningful.

$$
\begin{array}{cccccc}
1 & 2 & 3 & 4 & 5 & 6 \\
2 & 3 & 4 & 5 & 6 & 1 \\
3 & 4 & 5 & 6 & 1 & 2 \\
4 & 5 & 6 & 1 & 2 & 3 \\
5 & 6 & 1 & 2 & 3 & 4 \\
6 & 1 & 2 & 3 & 4 & 5 \\
\end{array}
$$

The next example is a complete design that is not a Latin square (it is not even square). Its parameters are $v=5$, $k=5$, $r=7$, $b=7$. The order of the blocks, and of elements within blocks, is irrelevant.

$$
\begin{array}{ccccc}
1 & 2 & 3 & 4 & 5 \\
1 & 2 & 3 & 4 & 5 \\
1 & 2 & 3 & 4 & 5 \\
1 & 2 & 3 & 4 & 5 \\
1 & 2 & 3 & 4 & 5 \\
1 & 2 & 3 & 4 & 5 \\
1 & 2 & 3 & 4 & 5 \\
\end{array}
$$

This example is a regular (thus uniform) block design that is more interesting than the previous example of a complete block design. Its parameters are $v=6$, $k=4$, $r=2$, $b=3$. Notice $v\,r = b\,k$, which must be true of a block design that is regular.

$$
\begin{array}{cccc}
1 & 2 & 4 & 5 \\
1 & 3 & 5 & 6 \\
2 & 3 & 4 & 6 \\
\end{array}
$$

The final two examples have $v=4$ and $b=3$. In one case (left), the design is uniform with $k=3$. On the right, the design would be regular (with $r=2$) except that it is not uniform (one element is "missing"). Note that if $v=4$, $b=3$, and $k=3$ in a uniform design, there is no way to make it regular because if so, the above-mentioned formula gives $r = b\,k\,/\,v = 3*3\,/\,4$, which is 9/4 (not an integer).

$$
\begin{array}{ccc}
1 & 2 & 3 \\
2 & 3 & 4 \\
1 & 2 & 4 \\
\end{array}
\qquad\qquad
\begin{array}{ccc}
1 & 2 & 3 \\
2 & 3 & 4 \\
1 & & 4 \\
\end{array}
$$

Appendix 7 – example crop rotation (with some missing data) for use as exercise/discussion

Appendix 4 gives an example of using a FSM to build a crop rotation. For that reason, this exercise will instead make use of a block design – it will be simpler, in some sense, than the example given in appendix 4. This appendix, like other material, is repeated in a student-ready format in the supplemental document meant for distribution to students. Students may be asked to do the following:

1. Find positive and negative features of this rotation;

2. Fill in the gaps, marked [_____?], with the best possible choice; and

3. Suggest improvements to the rotation that would correct negative features.

The rotation is based on the following block design:

4	5	6	7	8	9
1	2	3	7	8	9
1	2	3	4	5	6
2	3	5	6	8	9
1	3	4	6	7	9
1	2	4	5	7	8
2	3	4	5	7	9
1	3	5	6	7	8
1	2	_?	6	8	9
2	3	4	_?	7	8
1	3	4	5	8	9
1	2	5	6	7	9

This design is [_____?], and so it has parameters

$\qquad$ v = [_____?] $\qquad$ b = [_____?] $\qquad$ k = [_____?] $\qquad$ r = [_____?]

Because each rotation should have at least one nitrogen fixer, our first priority might be to use vetch and [_____?] to act as nitrogen producers in each block, and thus we assign:

5 = vetch

7 = clover

Notice that it is perfectly acceptable to assign one crop to more than one number (so, if clover were not preferable, both of these could be vetch). Other assignments can be made, taking care not to give any block too many nitrogen-depleting crops and to plant a sufficient number of cash crops (assuming some goal of having high yields of cash crops).

Students can fill this in on their own choices for the rest of this table:

4	vetch	6	clover	8	9
1	2	3	clover	8	9
1	2	3	4	vetch	6
2	3	vetch	6	8	9
1	3	4	6	clover	9
1	2	4	vetch	clover	8
2	3	4	vetch	clover	9
1	3	vetch	6	clover	8
1	2	_?	6	8	9
2	3	4	_?	clover	8
1	3	4	vetch	8	9
1	2	vetch	6	clover	9

Once they have done so, they should find a way of rotating these crops in order to compensate for nitrogen depletion, forbidden adjacencies, etc. Here is where block-designs fall short and ad hoc methods (or methods of FSM) can be used. This assignment could be expanded (expanding in the spirit of item 2 at the beginning of this appendix) to test particular parts of the content in this module, agricultural or mathematical.