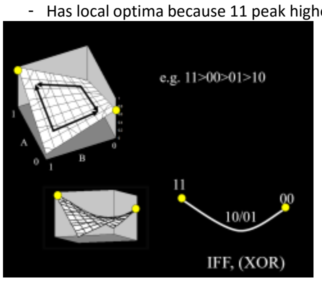
Modularity Continued

17 November 2015 10:04

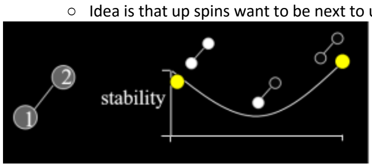
Two types
- Functional

Example

Boolean Fitness function with a fitness saddle.
- If and only if or XOR functions (depending on how you look at it)
- Has local optima because 11 peak higher than 00 peak.



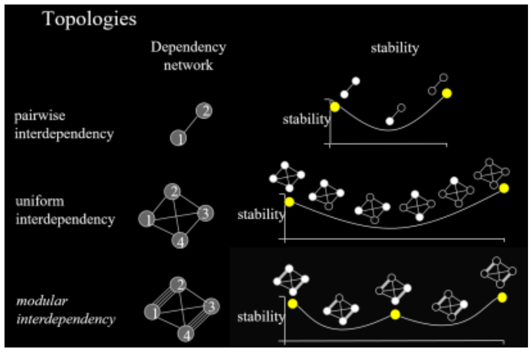
Consider this system in a local lattice
- See "Ising model" (often referred to as "spin glasses")
  - The model used for spin in ferromagnetism
  - Idea is that up spins want to be next to up spins and vice versa.



- In 1-D there is no local optima
  - This is because the boundary can change without making it worse/better
    000011111000 --> 001111100000
- In 2-D there is local optima
  - Because you have to change entire rows / columns of the lattice to have a neutral mutation.

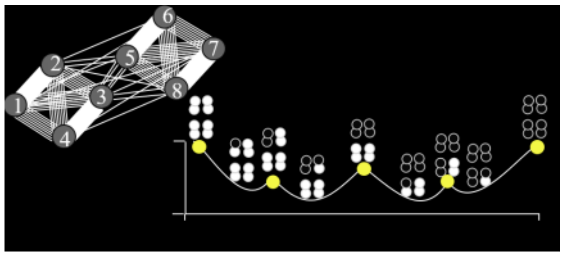
Irregular limited connectivity
- NK landscape

Fully connected
- If everything is connected then the entire system converges very quickly to global optima so not very interesting.



Modular interdependency:
Fully connected but:
What if we make some connections better than others
  - This can produce local optima.

Hierarchical if and only if function: HIFF
  - We can increase the number of interdependencies and have hierarchical modules
    - These increase the number of local optima.
    - Large changes in state are required to increase fitness
  - This function is used to prove what GA with crossover can do
    - Because each module can be solved separately (without solving the interdependencies)
    - Then different combinations of modules can be permuted through the crossover mechanics
    - **Problem:** This is only possible when the alleles for each module are next to each other.
  - **HIFF function is similar to royal road** with the difference that both all 0s and all 1s solves a module. This is the same idea as the Herb Simon "*nearly decomposable systems*".
    - Royal road is a completely decomposable system like the Safe cracking example
    - HIFF function is like that but if the safe made clicks in two places so is now no longer completely decomposable.



There are different algorithms that discover modularity/automatic problem decomposition.
- Idea that these search algorithms can solve modular problem in polynomial time vs exponential time.

Summary:
- Modular systems can have significant dependencies between modules
- Simon provides a hierarchical notion of modularity (in complex dynamical systems), and combinatorial stories of modularity effecting evolvability of complex systems: but the examples don't quite bridge former to the latter
- An example dynamical system that is structurally modular and showed that its dynamics are strongly interdependent. But it can be decomposed..
- Suggested a means to quantify the value of a proposed decomposition given a using equivalence classes of contexts

Conclusion
- Don't assume that modularity ≈ independent sub-parts
- 'decomposability' - only need to consider limited number of configurations of a subsystem
  - Decomposable but not separable, is where the interesting cases are.
- Evolvability of complex systems
  - Separable and nearly separable systems are easy to optimise/evolve
  - Decomposable systems are easily evolvable with appropriate mechanisms (see 'compositional evolution' lecture) but hard for naive methods