Genetic Algorithms for **Multidimensional** S

- Let the decision be multidimensional so $S=(S_1,\ldots,S_n)$ is a vector. How do we represent this as a binary string?
- Assume S_i is an integer and $0 \leq S_i \leq 2^K - 1$, Then you can represent each S_i by a binary string of length k ?
- Assume the length is 4 for each of three variables, then your binary string is something like: $S = [0110 \mid 1111 \mid 0001 \mid \ldots]$ divider tells us where the substrings are
- And each of the variables is $\leq$ 15 iff k=4

S_1 S_2 S_3

handout 9-12-11

Warehouse Example on Board

- Assume you want to consider building a warehouse at each of Z locations (i=1, …,Z).
- The size of the warehouse can be between 10,000 sq ft up to 150,000sq ft so S_i = j means the warehouse size is j*10,000 sq ft. $\Rightarrow j \in [1,15] \cap \mathbb{Z}$
- However, I need to convert 1 to 15 into a binary string to solve this integer problem with Genetic Algorithms. $\Rightarrow k = 4$

$$\begin{array}{cccc} j = 1, & 5, & 2, & 7 \\ S = [0001 & 0101 & 0010 & 0111] \\ z_1 & z_2 & z_3 & z_4 \end{array}$$

- How long is the string for each S_i? $z \propto k$

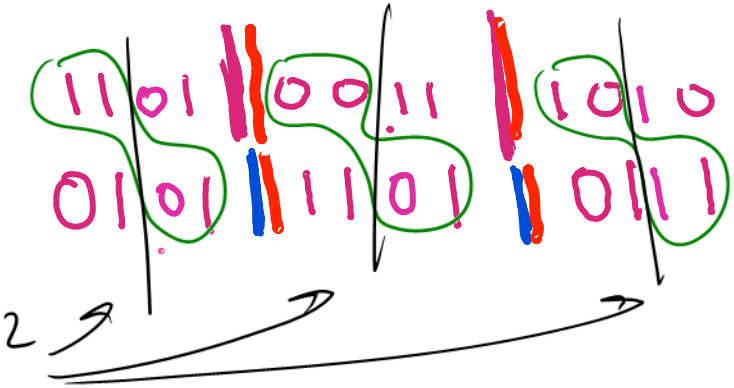
Warehouse Problem

- Assume I want to decide about building a warehouse in 3 locations. So if
- warehouse j(base 10) j(base2)
- A 5 0101
- B 3 0011
- C 12 1100

Then S=(0 1 0 1 | 0 0 1 1 | 1 1 0 0), which is a binary string of length 12.

So How do you Do Crossover with Multidimensional S ?

if $S = (S^1, S^2, S^3)$



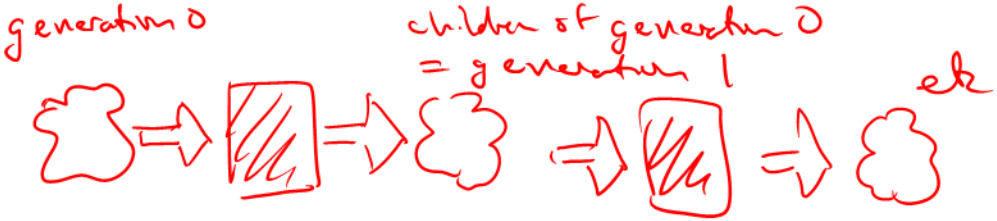
Parent 1

Parent 2

- Then you do all cross over operations for the part of the string related to one decision S^i (e.g. between the blue lines) So if cross over is at position 2 for each variable, a child could be 1101 | 0001 | 1011

what if $j \in [1,13] \cap \mathbb{Z}$?
we can generate out bounds children.
→ just ignore than children and continue
→ note 0 is also out of bounds

Elitism in Genetic Algorithms *(used both for single and multiple dimensional problems)*



- Suppose you find a solution S_i that is very good (has a low fitness).
- In the algorithm we have described the old population is discarded. Is that a good idea?
- We can modify this. How should we do it?

→ pick the elite of the current generation to accompany the children. This also means to control population size, we have to drop the worst children.

How large should E be?
E is number of "elite" parents brought to next generation

⇒ note that there will be a record of the best solution even if E = 0

- Advantages of large E?

  → preserve good solutions

- Disadvantages of large E?

  → limits diversity as there aren't many children

  → search of search space negatively impacted

- What do people usually use for E?

  → Idea: dynamically change E with time

  → Static configurations usually have E < 50%

Review of GA-Steps in the Algorithm

- 0. Initialization. Pick algorithm parameters and initial population
- Compute fitness F(CurSk) for all members of current **population** {CurSk| k=1,M}.
- **Crossover**:
  - Use fitness in random selection of parents. So probability of CurSk being selected as a parent is higher if F(CurSk) is high for both roulette and tournament selection.
  - Use crossover to create "children" (also called offspring) from each pair of parents.
- Use **mutation** to make some random change in the children.
- All children become the current **population** and go to step 1. (The previous population is now discarded.)
- can be modified for elitism.

Pseudo Code for Genetic algorithm (page 120 of xeroxed reading from S&Y Text)

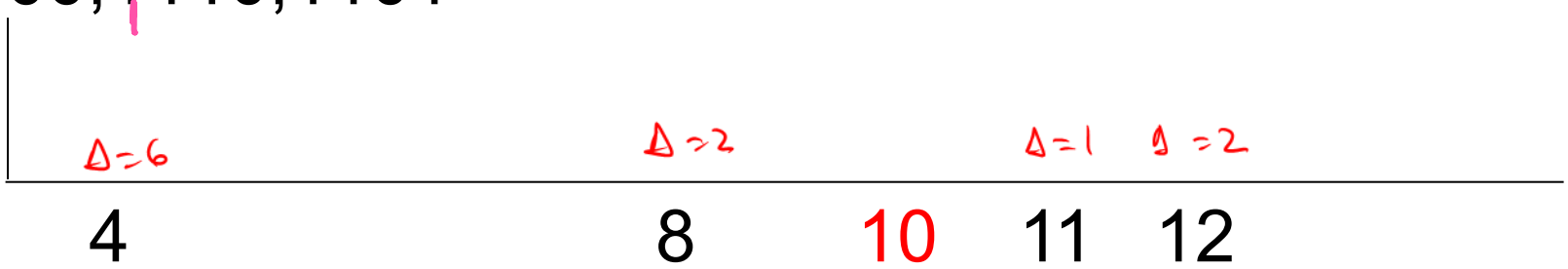
```
Procedure (Genetic_Algorithm)
    M= Population size.                          (*# Of possible solutions at any instance.*)
    N_g= Number of generations.                  (*# Of iterations.*)
    N_o= Number of offsprings.                   (*To be generated by crossover.*)
    P_μ= Mutation probability.                   (*Also called mutation rate M_r.*)
    P ← Ξ(M)                                     (*Construct initial population P.*)
                                                 (*Ξ is population constructor.*)
    For j = 1 to M                               (*Evaluate fitnesses of all individuals.*)
            Evaluate f(P[j])                     (*Evaluate fitness of P.*)
    EndFor
    For i = 1 to N_g
            For j = 1 to N_o
                (x, y) ← φ(P)                    (*Select two parents x and y from current population.*)
                offspring[j] ← χ(x, y)           (*Generate offsprings by crossover of parents x and y.*)
                Evaluate f(offspring[j])         (*Evaluate fitness of each offsprings.*)
            EndFor

            For j = 1 to N_o                     (*With probability P_μ apply mutation.*)
                mutated[j] ← μ(y)
                Evaluate f(mutated[j])
            EndFor
    P ← Select(P, offsprings)                    (*Select best M solutions from parents & offsprings.*)
    EndFor
    Return highest scoring configuration in P.
End
```

unnecessary

Figure 3.3: Structure of a simple genetic algorithm.

Problems with Neighborhoods of Binary Strings used to represent integers greater than 1

- Recall that a binary string can represent an integer in base 10.
- Assume CurS= 1100 = $1x2^3 + 1x2^2+0x2^1+0x2^0$=**10**
- Neighborhood from flipping one digit: 0100, 1000,1110,1101



What can you say about this range?

⇒ neighborhood may be very asymmetric
⇒ is this okay? → problem-specific

Both Binary Coded GA and Tabu Search Use Binary Strings

- For some problems the variables are really binary (0 or 1).
- However, in other cases the variables are real valued or integer or something else (e.g. perturbation) . In these cases the neighborhoods associated with the binary string representation does give an uneven perturbation. It might not cause a problem, but you should think about whether it might cause a problem.
- For simulated annealing there is no need to represent S as a binary string.

Questions

- Review: What are the parameters for binary coded GA?

  population size × generations = max eval of cost function

- So what happens to the mutations? What if it is a “bad” mutation?
- What happens to children of crossover operations that are not good?
- What are some of the differences between GA and SA?

  ⇒ you have a population of solutions

-

Selecting Parameters for GA and Real-Coded GA

- Setting parameters for GA is less straightforward than for SA.
- We will later discuss some methods that can be used for this.
- We have discussed here only the application of GA to integers or binary strings Later we will discuss other applications including “Real-Coded” GA for decision variables that are real numbers.

We'll Return Later to Genetic Algorithms

- We will return later to details of GA including setting parameter values and representation of decision variables that are not binary strings.
- Next I want to start on the third major heuristic algorithm Tabu Search so we have covered the three most widely used algorithms: SA, GA, and TS so you can start on your projects.

Project

- Teams of 2 to 4 students (unless PhD student with own project)
- Computer module provided for your “application”
- You will compare GA, SA, TS and your own improvements in these algorithms
- Later lectures will discuss how to do comparisons
- Written Report
- Short Oral Presentaton

14

Project Topics

- **Satisfiability Testing for Artificial Intelligence**: Prof. Bart Selman, Computer Science
- **Resource Allocation in Cellular Networks**: Prof. Steve Wicker, Electrical Engineering
- **Optimal Control of Finite Element Systems with Environmental Applications**: Prof. Christine Shoemaker, Civil and Environmental Engineering
- **Protein Folding**: Molecular and Cell Biology
- **ORIE Project** (maybe Job Shop Scheduling)

15

General Project Instructions

- By Thursday we will have posted the general instructions for the projects, which includes all the deadlines.

- Be sure to attend Prof. Wicker's lecture this Wednesday on heuristic optimization for optimizing cell phone networks.
- This is one possible topic for a projec.

Homeworks, Exams, Guest Lectures

- There will be homework (and possibly exam) questions on the two guest lectures (Sept. 14 and 21) so everyone should hear these lectures (not just the people doing those projects).
- The reason is that part of the process of learning heuristic methods is see by example how you convert a real problem into something that can be analyzed by optimization.