Applications of order based GAs (1)

- Sorting (easy)
- N-queens (not very difficult)
- Routing (tough)
- Scheduling (tough)
- Graph colouring (tough)
- etc.

Applications of order based GAs (2)

Precedence constrained job shop scheduling problem

- J is a set of jobs.
- O_j ($j \in J$) is a set of operations ($O = \cup O_j$)
- M is a set of machines
- *Able* $\subseteq O \times M$ defines which machines can perform which operations, and
- *Pre* $\subseteq O \times O$ defines which operation should precede which
- *Dur* : $\subseteq O \times M \rightarrow \mathbb{R}$ defines the duration of $o \in O$ on $m \in M$

The goal is now to find a schedule such that:

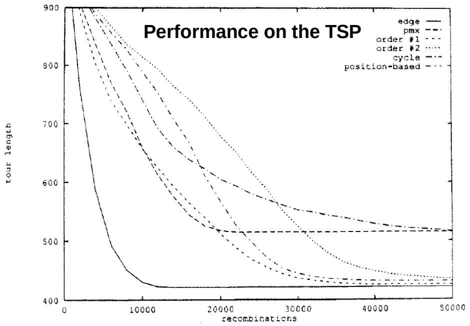
- All jobs are scheduled
- All conditions defined by *Able* and *Pre* are satisfied
- The total duration of the schedule is minimal

Applications of order based GAs (2)

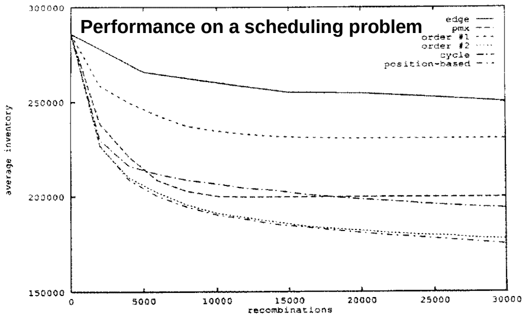
Precedence constrained job shop scheduling GA

- individuals are permutations of operations
- permutations are decoded to schedules by a decoding procedure
  - take the first (next) operation from the individual
  - look up its machine
  - assign the earliest possible starting time on this machine, subject to
    - machine occupation
    - precedence relations holding for this operation in the schedule created so far
- fitness of a permutation is the duration of the corresponding schedule (to be minimized)
- use any ob-mutation and any ob-crossover
- use roulette wheel selection on inverse fitness
- use random initialization

Performance of order based crossovers (1)



Performance of order based crossovers (2)



Performance of order based crossovers (3)

Conclusions:

- Different operators can perform differently on the same problem.
- The same operator can perform differently on different problems.

Corollary (bad news):

There is no generally good advise on the best operator.

Selection (1)

Fitness proportional selection (FPS):

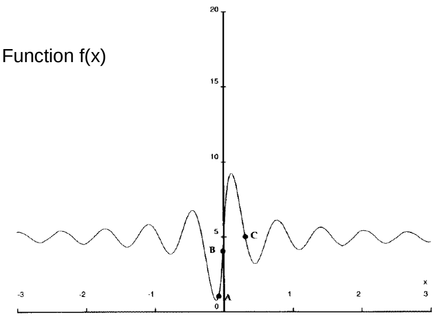
Expected number of times f_i is selected for mating is: $\frac{f_i}{\bar{f}}$

Disadvantages:

1 Outstanding individuals take over the entire population very quickly $\Rightarrow$ danger for premature convergence.
2 Low selection pressure when fitness values are near each other.
3 Behaves differently on transposed versions of the same function.

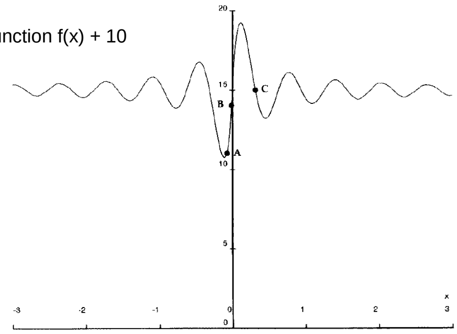
Selection (2)

Function f(x)



Selection (3)

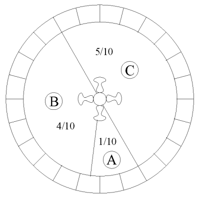
Function f(x) + 10



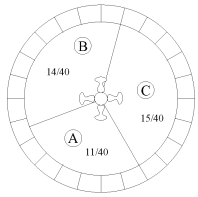
Selection (4)

Selection probabilities of the same population on

Function f(x)



Function f(x) + 10



Selection (5)

A cure for FPS: fitness scaling

Procedure:

1 Start with the raw fitness function *f*.
2 Standardise to ensure:
   – Lower fitness is better fitness.
   – Optimal fitness equals 0.
3 Adjust to ensure:
   – Fitness values range from 0 to 1.
4 Normalise to ensure:
   – The sum of the fitness values equals 1.

Selection (6)

A cure for FPS: fitness scaling (continued)

Details of procedure:

- Let P_t be the population at time *t*.
- Standardisation yields

$$f_t^s(x) = \begin{cases} f(x) - \min_t(f) & \text{if } f \text{ is to be minimized} \\ \max_t(f) - f(x) & \text{if } f \text{ is to be maximized} \end{cases}$$

- Adjusting yields f_s^a

$$f_t^a(x) = \frac{f_t^s(x)}{\max_t(f_t^s) - \min_t(f_t^s)} = \frac{f_t^s(x)}{\max_t(f_t^s)}$$ By standardisation

Selection (7)

- Normalisation yields :

$$f_t^n(x) = \frac{f_t^a(x)}{\sum_{x \in P_t} f_t^a(x)}$$

Note: max_t and min_t are taken over P_t

Selection (8)

Ranking selection

- Rank individuals according to their fitness
- Use the ranks, rather than the fitness values, to determine the probability of selection
- Mapping from ranks to selection probabilities is arbitrary, for instance linear

Selection (8)

Ranking selection example

3 individuals A, B, C and linear mapping for maximization problem:

- Fitness: f (A) = 1, f (B) = 4, f (C) = 5.
- Ranking: r(A) = 1, r(B) = 2, r(C) = 3.
- Linear function:

$$h(x) = \min + (\max - \min) \times \frac{(r(x)-1)}{n-1}$$

h(A) = 1, h(B) = 3, h(C) = 5

- selection probabilities proportional to h values: h(x)/9
  $p_{rank}(A) \approx 11\%$, $p_{rank}(B) \approx 33\%$, $p_{rank}(C) \approx 56\%$.
- selection probabilities with roulette wheel proportional to f values: f(x)/10
  $p_{rw}(A) = 10\%$, $p_{rw}(B) = 40\%$, $p_{rw}(C) = 50\%$.

Selection (9)

Tournament selection:

1 Pick k individuals randomly, without replacement
2 Select the best of these k comparing their fitness values

k is called the size of the tournament

selection is repeated as many times as necessary

Selection (9)

Tournament selection example