ME537: Homework 2

Kevin Kemper

1 Simulated Annealing

k	Step
k_{max}	Maximum number of steps
s	State - a path
s_{best}	Best known path
s_{new}	New path to check
e	energy - length of path
e_{best}	Length of best path
e_{new}	Length of new path
T	Temperature - a path

$random()$	Returns random number between 0 and 1
$neighbor(s)$	Shuffles two adjacent cities in s
$energy(s)$	Returns the length of s

$s \leftarrow shuffl(cities)$
$s_{best} \leftarrow s_0$
$e \leftarrow \infty$
$e_{best} \leftarrow \infty$
$T \leftarrow 1000$
$k \leftarrow 0$
$k_{max} \leftarrow 500$
while $k < k_{max}$ **do**
  $s_{new} \leftarrow neighbor(s)$
  $e_{new} \leftarrow energy(s_{new})$
  if $e_{new} < e_{best}$ **then**
    $s_{best} \leftarrow s_{new}$
    $e_{best} \leftarrow e_{new}$
  end if
  if $\frac{e^{e-e_{new}}}{T} > random()$ **then**
    $s \leftarrow s_{new}$
    $e \leftarrow e_{new}$
  end if
  $T \leftarrow T\frac{kMax-k}{kMax}$
  $k \leftarrow k + 1$
end while

2 Evolutionary Algorithm

ϵ	Probability of choosing a random path
k	Step
k_{max}	Maximum number of steps
$pop[]$	Population of possible paths
s	State - a path
s_{new}	New path to check
e	energy - length of path
e_{new}	Length of new path

$random()$	Returns random number between 0 and 1
$random(pop)$	Returns a random path in the population
$neighbor(s)$	Shuffles two adjacent cities in s
$cost(s)$	Returns the length of s
$shuffle(cities)$	Shuffles the list of cities to genearate a random path
$worst(pop)$	Returns the worst in the population
$best(pop)$	Returns the best in the population

while $i < population\ size$ **do**
  $pop[i] \leftarrow shuffle(cities)$
end while
$\epsilon \leftarrow 0.1$
$k \leftarrow 0$
$k_{max} \leftarrow 500$
while $k < k_{max}$ **do**
  if $(1 - \epsilon) < random()$ **then**
    $s_{new} \leftarrow best(pop)$
  else
    $s_{new} \leftarrow random(pop)$
  end if
  $s_{new} \leftarrow neighbor(s_{new})$
  $e_{new} \leftarrow cost(s_{new})$
  if $eNew < cost(worst(pop))$ **then**
    $s[worst(pop)] \leftarrow s_{new}$
  end if
  $k \leftarrow k + 1$
end while

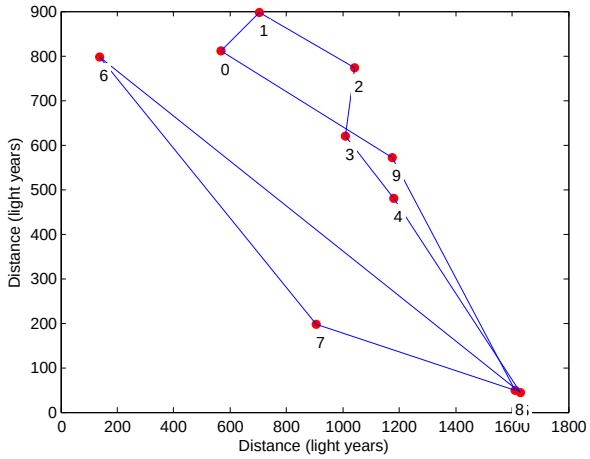


Figure 1: An example of a path chosen by the simulated annealing algorithm for 10 cities. The dots represent city locations with numbers indicating the order of the visits.

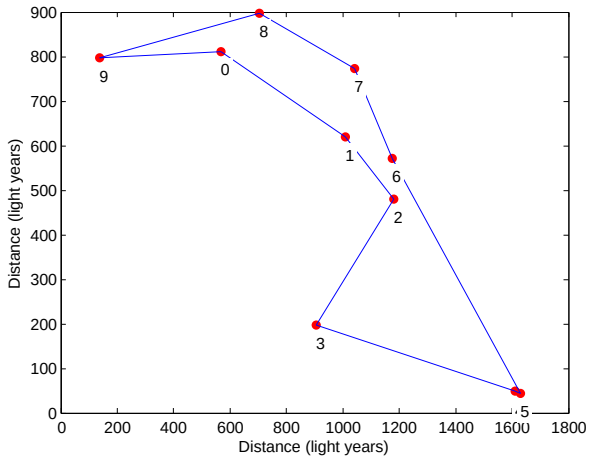
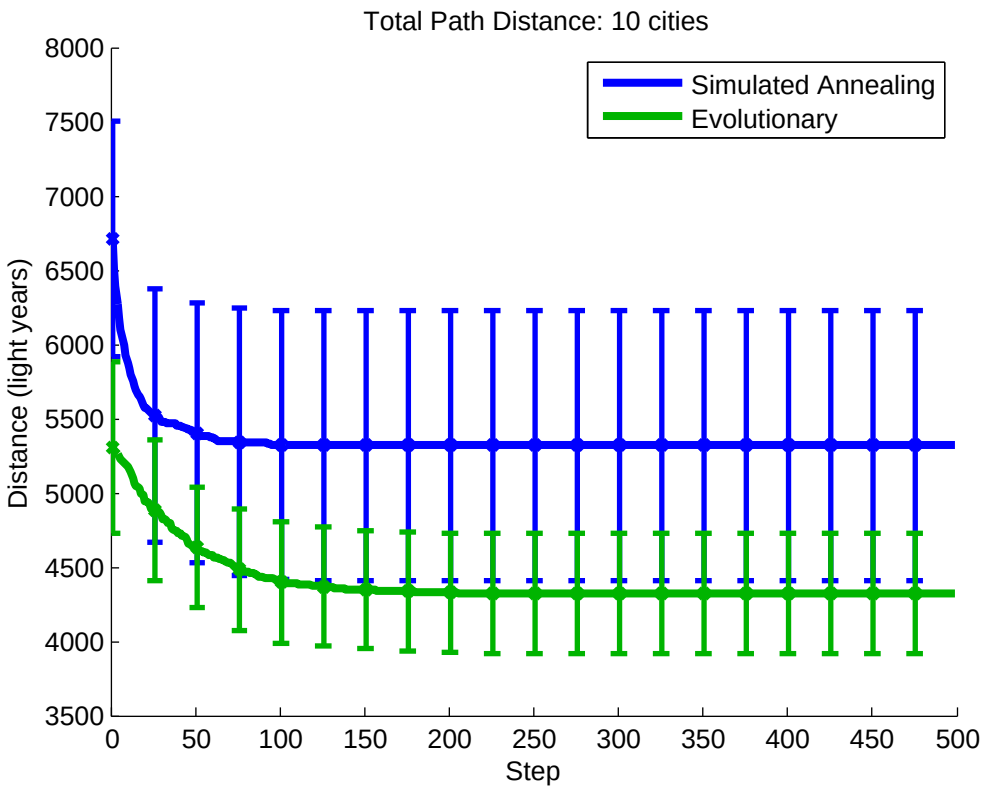
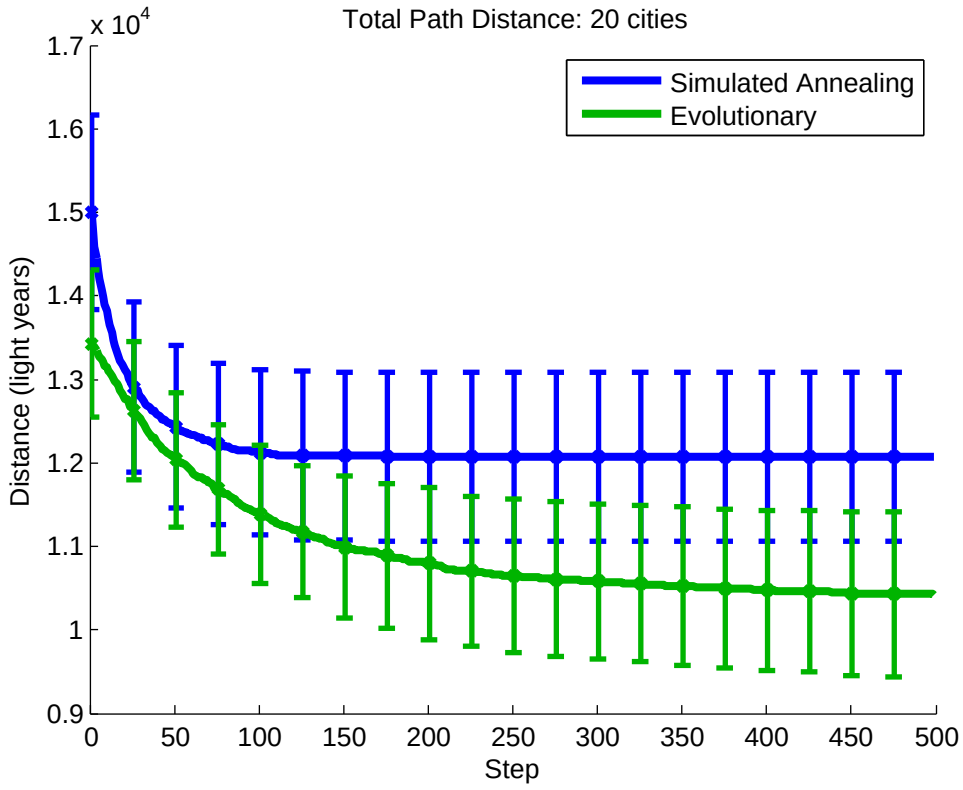


Figure 2: An example of a path chosen by the evolutionary algorithm for 10 cities. The dots represent city locations with numbers indicating the order of the visits.

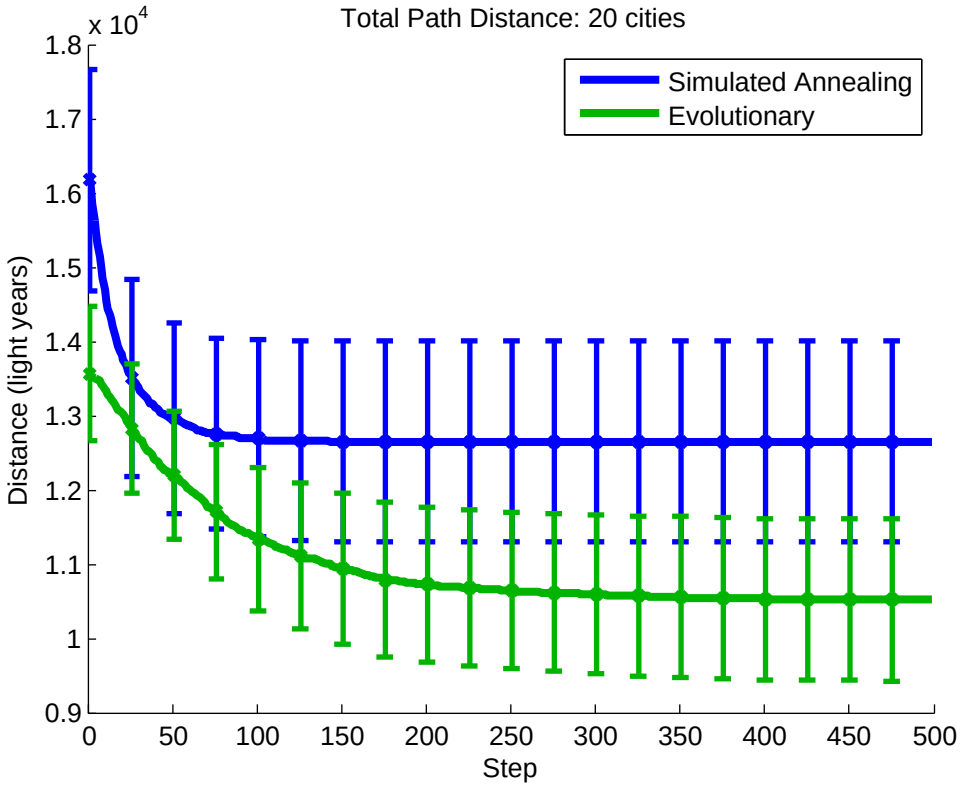


(a)

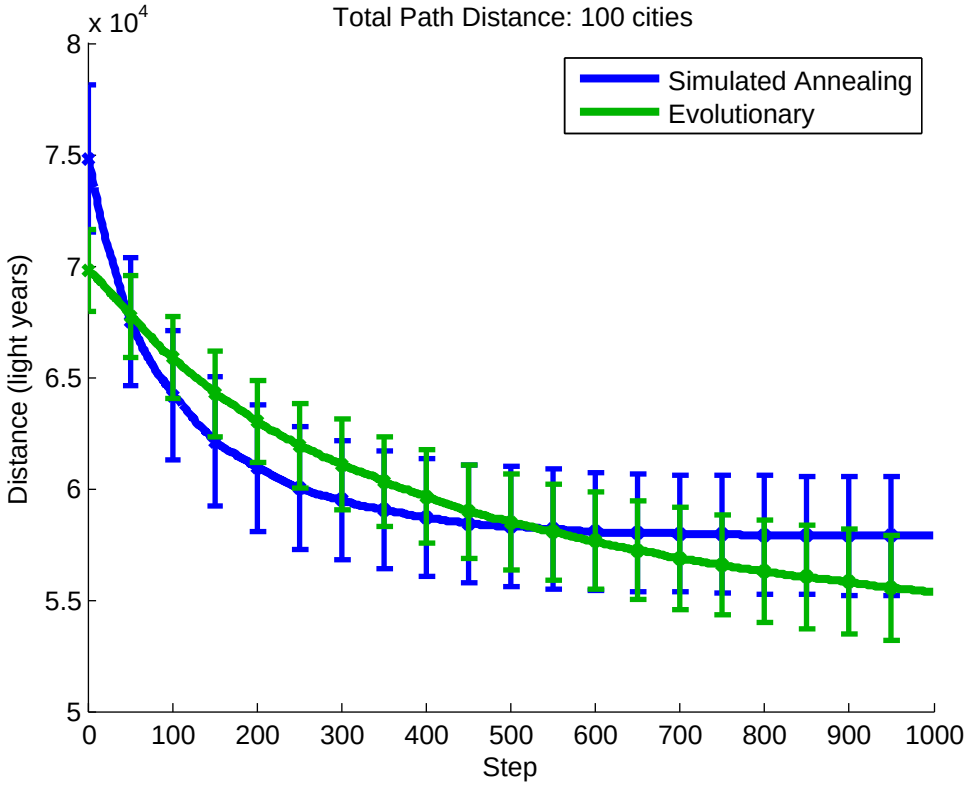


(b)

Figure 3: Performance of simulated annealing vs. an evolutionary algorithm.



(a)



(b)

Figure 4: Performance of simulated annealing vs. an evolutionary algorithm.