CS 5722, Fall 2014 **Homework 8**
Group: Michael Jalkio, Kevin Li, Daniel Sperling
NetIDs: mrj77, kyl27, dhs252

1

a

See graph.

b

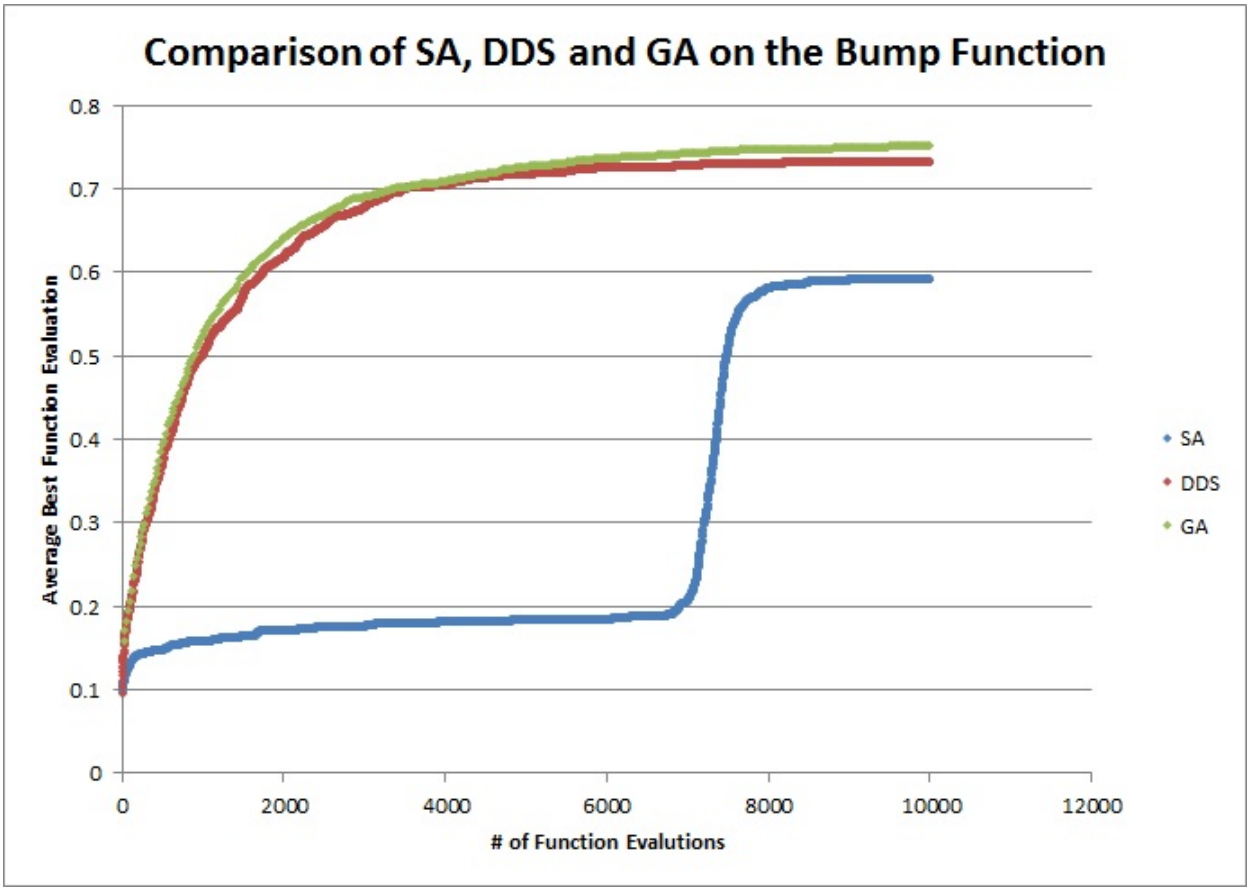
i

See graph.

ii

Since the dimensions are chosen randomly for each iteration, it would be better if we added some constraints on how they're chosen. We could avoid choosing dimensions that were chosen in the recent past, like tabu search. This will reduce "backtracking" and encourage searching of unseen solutions etc.

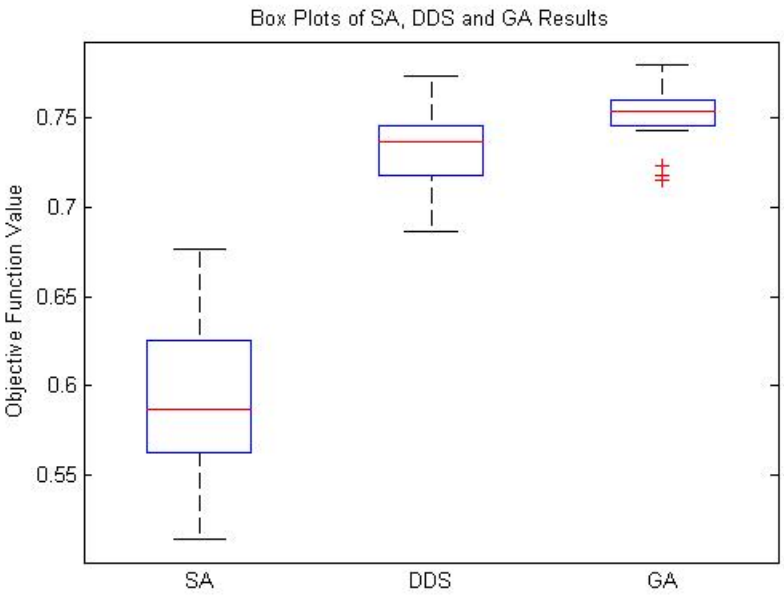
c



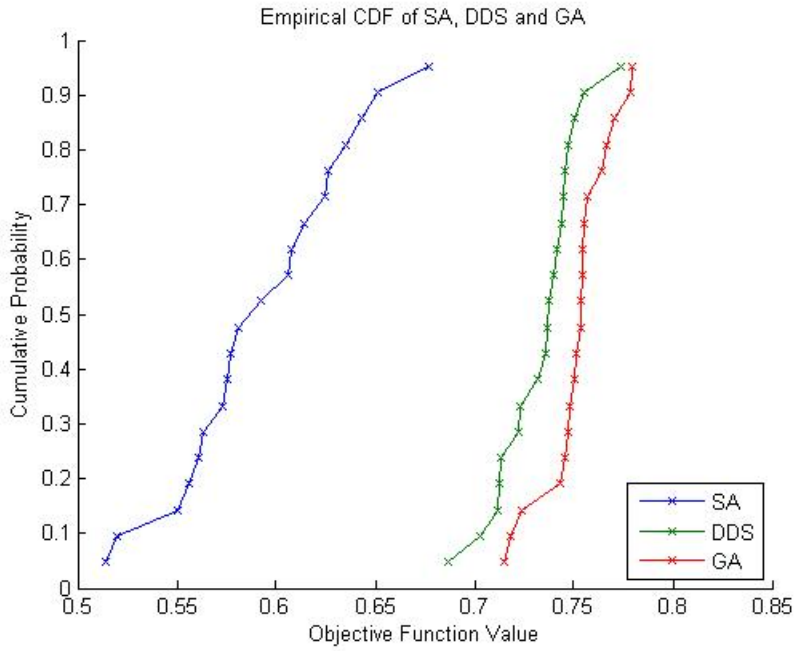
From the graph alone, it is clear that both GA and DDS dominate SA. It also appears that GA performs slightly better than DDS over the course of the run, and may dominate at the end.

d

Comparison 1: Box Plot



Comparison 2: CDF



Comparison 3: Two sample t test

The null hypothesis for each pair is that the mean value of the two search heuristics is the same.

SA/DDS:

Statistics: T Statistic: -13.509848
P value for Two Sided: 0.00000
P value for One Sided: 0.000000

For comparison at $\alpha = 0.05$, $t_{\alpha/2,v} = 2.052042$, $t_{\alpha,v} = 1.703421$. As $t < -t_{\alpha/2,v}$, we reject the null hypothesis and accept that the two mean values are different. Then, since $t <= -t_{\alpha,v}$, we can accept the alternate hypothesis that $\mu_x - \mu_y < \Delta_0$, indicating that the mean of SA is statistically significantly less than the mean of DDS.

As the means were so far off, it can easily be seen that SA performed much worse than DDS. This confirms what was seen in the graph in part c, the box plot above, and in the cdf, where DDS dominated SA.

SA/GA:

Statistics: T Statistic: -15.615549
P value for Two Sided: 0.000000
P value for One Sided: 0.000000

For comparison at $\alpha = 0.05$, $t_{\alpha/2,v} = 2.058272$, $t_{\alpha,v} = 1.707344$. As $t < -t_{\alpha/2,v}$, we reject the null hypothesis and accept that the two mean values are different. Then, since $t <= -t_{\alpha,v}$, we can accept the alternate hypothesis that $\mu_x - \mu_y < \Delta_0$, indicating that the mean of SA is statistically significantly less than the mean of GA.

As the means were so far off, it can easily be seen that SA performed much worse than GA. This confirms what was seen in the graph in part c, the box plot above, and in the cdf, where GA dominated SA.

DDS/GA:

Statistics: T Statistic: -3.172571
P value for Two Sided: 0.003015
P value for One Sided: 0.001508

For comparison at $\alpha = 0.05$, $t_{\alpha/2,v} = 2.025410$, $t_{\alpha,v} = 1.686598$. As $t < -t_{\alpha/2,v}$, we reject the null hypothesis and accept that the two mean values are different. Then, since $t <= -t_{\alpha,v}$, we can

accept the alternate hypothesis that $\mu_x - \mu_y < \Delta_0$, indicating that the mean of SA is statistically significantly less than the mean of GA.

The means were significantly closer together in this pairing, but since both standard deviations were so low, it makes sense that we can still confirm that GA performed better than DDS. This confirms what was seen in the graph in part c, the box plot above, and in the cdf, where GA dominated DDS.

2

a

For a single iteration (a single generation) we can figure out the wall clock time. We first have 50 seconds of serial calculations, then 10 seconds of communication, then each processor is able to handle the calculation of a single fitness function which takes 50 seconds, and finally we must perform communication again for 10 seconds. For 100 generations we just need to multiply. Therefore the total wall clock time is:

$$\begin{aligned} T(20) &= 100(50s + 10s + 50s + 10s) \\ &= 100 * 120s = 200m = 3hr \text{ and } 20m \end{aligned}$$

In serial the GA algorithm would take $100(50s + 20 * 50s) = 105000s = 1750m = 29hr$ and $10m$. Therefore, the speedup is:

$$S(20) = \frac{T(1)}{T(20)} = \frac{105000s}{12000s} = 8.75$$

Which is fantastic!

b

Everything is exactly the same with 22 processors instead of 20. Because there are only 20 offspring in each generation and we're not allowed to split the fitness calculation for one offspring between processors, we will always have 2 processors that are idle.

c

First we'll write speed up in terms of T_F:

$$\begin{aligned} S(20) &= \frac{100(50s + 20T_F)}{100(50s + 10s + T_F + 10s)} \\ &= \frac{50s + 20T_F}{70s + T_F} \end{aligned}$$

So efficiency with 20 processors is:

$$E(20) = \frac{S(20)}{20} = \frac{50s + 20T_F}{20(70s + T_F)}$$

And we can then solve for all values of T_F which will lead to at least 80% efficiency:

$$\begin{aligned}
\frac{50s + 20T_F}{20(70s + T_F)} &\geq 0.8 \\
50s + 20T_F &\geq 16(70s + T_F) \\
4T_F &\geq 1070s \\
T_F &\geq 267.5s
\end{aligned}$$

Meanwhile efficiency for 22 processors is:

$$E(22) = \frac{S(22)}{22} = \frac{50s + 20T_F}{22(70s + T_F)}$$

And we can again solve for T_F that leads to at least 80% efficiency:

$$\begin{aligned}
\frac{50s + 20T_F}{22(70s + T_F)} &\geq 0.8 \\
50s + 20T_F &\geq 17.6(70s + T_F) \\
2.4T_F &\geq 1182s \\
T_F &\geq 492.5s
\end{aligned}$$

The value of T_F for 22 processors to be 80% efficient is much higher than the value for 20 processors. This is because 2 processors do no work at all and only serve to lower the efficiency compared to the 20 processor case. We can imagine that each processor needs to handle a certain proportion of the speed up, but if 2 processors contribute none at all everything else needs to pick up the slack. If we put the same $T_F = 492.5s$ into the 20 processor case this actually leads to 88% efficiency!