Artificial Intelligence Methods Coursework 2

Name: Kah Yan Wong 201228822

Search algorithm

Item 1. Implement a population-based search algorithm

The population-based search algorithm used is a genetic algorithm. The hill climbing search in coursework 1 was used for 5 trials to extract the best chromosome. I mutated this chromosome to form a total of 50 chromosomes to be used as the initial population of the genetic algorithm.

For the selection operation, I used a modified roulette wheel selection to select the top 10 chromosomes. The initial selection calculates the probability of selecting a chromosome based on its proportion of the total fitness. I found that the probabilities of similar fitness levels chromosomes did not have significant difference (e.g. probabilities such as 0.142 and 0.141). The modified selection uses a scaling factor that calculates the proportion of probability more accurately, therefore increasing the chances of choosing a good chromosome.

Scaling factor = $\frac{1}{\sqrt{\left(1-\frac{x^2}{c^2}\right)}}$ where x = fitness of chromosome and c = ceiling fitness

For the crossover operation, I used a multi parent crossover method to perform a diagonal crossover on the selected chromosome. After 5 random parents are selected, parts of their genes are copied to replace parts of the first chromosome. The accuracy is compared with a 2-point crossover accuracy and the results show an improvement in the average accuracy by 10%. The operator becomes more explorative as there is more genetic information on the search space with more parents involved.

For the mutation operation, the chromosome is mutated if the random probability falls within the mutation probability of 0.1.

For the test condition, the test is validated if the desired fitness level was achieved. If the average accuracy of the generation was above 77 within the same number of evaluations as coursework 1, the test is accepted. Otherwise, it will proceed to the next generation.

Crossover, mutation and local search were applied with probabilities 0.6, 0.1 and 1.0 respectively.

Item 2. Incorporate local learning (memetic algorithms)

I applied a local search after the mutation operation to find the better solution in the offspring neighbourhood. A random solution is selected from the offspring neighbourhood and a distance is added to it to form a neighbouring chromosome. Neighbouring chromosomes will replace the current offspring if they perform better. The distance between the neighbouring chromosomes is calculated by $\frac{1}{|Maximum\ fitness - Average\ fitness|}$ multiplied with a random number from 0 to 1. The intensity of individual learning is applied until it reaches the limit of 10 iterations.

Item 3. Include some variant of differential search

The local search used is a single iteration of differential search. I used a second chance replacement method to accept better chromosomes that only shows significant improvement in accuracy. If the difference in accuracy was less than 0.5, it will keep the current chromosome rather than replace it. Based on the results, the average accuracy did not vary more than 1.0. This suggests that if the accuracy is a small difference, frequent replacements are rather unnecessary because it does not significantly affect the overall accuracy.

Performance evaluation

Item 1. Implement a clear performance measure

The number of evaluations was maintained across coursework 1 and 2 at 1750 evaluations. On average, the genetic algorithm search performed better with an increase in 10% mean accuracy within less or equal to the number of evaluations in stochastic hill climbing.

Item 2. Generate several simple descriptive results (e.g. mean, standard errors, boxplots, etc.)

Algorithm	Stochastic hill climbing			Genetic Algorithm		
	Min	Max	Average	Min	Max	Average
Mean Accuracy	10	75	66.7	71	76	74.2
Median Accuracy	65	71	68	68	72	70
Interquartile range	22	32	26	3	10	5
Std. Deviation	20.22	22.28	21.24	1.54	18.83	5.68

In diagram 1 and 2 in the appendix, the bar charts display the accuracy across trials and generations. The error bars correspond to the standard deviation error in each trial and generation. Since the standard error bars overlap in the genetic algorithm, the difference is not statistically significant. However, some of the standard error bars in the hill climbing algorithm do not overlap which indicates a very high difference in variability.

In diagram 3, the boxplots show a significant improvement in the interquartile range as well as median of accuracy.

Item 3. Run comparison statistics (e.g. unpaired two-samples Wilcoxon test)

Applying statistical tests to the results gives the following. Based on the Wilcoxon test, the value of z I -2.3444. The p-value is 0.1928. The result is significant at $p < 0.05$. Outliers and heavy tails in the hill climbing results might have highly influenced the t test.

References

N. Krasnogor, Jim Smith. A memetic algorithm with self-adaptive local search: TSP as a case study. Proceedings of the Genetic and Evolutionary Conference (GECC0-2000)

S. Kumar, V. Sharma, R. Kumari. Memetic search in differential evolutionary algorithm. International Journal of Computer Applications (0975-8887) Volume 90

A.E. Eiben, C.H.M. van Kemenade. Performance of Multi-Parent Crossover Operators on Numerical Function Optimization Problems.