Anthony Debruyn

Heuristic Optimisation

Implementation Exercise 1

Iterative improvement algorithms for the PFSP

Dr. F. Mascia, Dr. T. Stützle $\boxed{\text{INFO-H-413}}$

1 Introduction

The goal of this assignment was to...

2 Code Use

3 Exercise 1.1

3.1 The Results

We list here the average percentage deviation from the best solutions for each algorithm tested, along with the average computation time:

Algorithm	APD	ACT(ms)
–random–exchange–best	289	111
–random–exchange–first	295	38
–random–insert–best	313	100
–random–insert–first	277	133
–random–transpose–best	411	5
–random–transpose–first	416	12
–slack–exchange–best	241	110
–slack–exchange–first	255	34
–slack–insert–best	223	103
–slack–insert–first	200	120
–slack–transpose–best	299	16
–slack–transpose–first	308	16

Figure 1: The APD and ACT for each algorithm.

3.2 Difference between the solutions

We used the Student t-test to determine whether there there is a statistically significant difference between the solutions generated by the different perturbative local search algorithms. The obtained p-values are shown in figure 2. The algorithms corresponding to the numbers are the ones from figure 1, in the same order (alphabetically sorted).

The Student t-test can be used to test statistically the mean equality null hypothesis. The p-value corresponds to the probability that the null hypothesis is incorrectly rejected. If this value is below the significance level, the null hypothesis is rejected and thus the means are very different. If the value is above the significance level, the hypothesis is incorrectly rejected, thus accepted, and the means are approximately equal. The higher the value, the closer the means.

Here is the R code used to write the p-values to a file. The results matrix was created by taking the average percentage deviation of each algorithm and each instance (matrix of dimension 60x12).

```
a <- read.table("R-avRelPer--random--exchange--best.dat")$V1
...
l <- read.table("R-avRelPer--slack--transpose--first.dat")$V1
results <- c(a,b,c,d,e,f,g,h,i,j,k,l)

results <- array(results, dim=c(60,12))

x = 1
for (i in 1:11) {
  for (j in (i+1):12) {
    test[x] <- t.test (results[,i], results[,j], paired=T)$p.value
    x = x + 1
  }
}

```

```
write(test, file = "p values st test", ncolumns = 1)
```

We take a significance level of 0.05 ($\alpha = 0.05$).

	1	2	3	4	5	6	7	8	9	10	11	12
1		0.3946534	0.07607217	0.4263596	6.967038e-08	2.28503e-06	0.01718699	0.06069461	0.01642366	0.002925141	0.9075012	0.7670622
2			0.008214209	0.7082091	0.0001230069	0.0002814088	0.0003217527	0.0208894	0.0002224225	3.885381e-06	0.1482378	0.02733348
3				0.002881668	3.526205e-05	8.519584e-05	3.1477e-06	1.102071e-05	9.521436e-05	1.494252e-05	0.04240414	0.1129525
4					5.415206e-07	1.029794e-06	0.004716419	0.03777324	0.0126438	0.00134803	0.6283332	0.3083158
5						0.2608918	2.849944e-06	8.373585e-06	3.428071e-05	1.008478e-05	0.0005169061	0.0008130384
6							5.733163e-06	1.244147e-05	6.134921e-05	1.893243e-05	0.0007349604	0.001103452
7								0.004145256	0.1388542	0.002564425	1.704456e-09	5.077375e-10
8									0.01561275	0.0002874112	7.797309e-07	2.86032e-08
9										6.695723e-07	1.507685e-07	1.043949e-07
10											9.012818e-09	1.203368e-08
11												0.000153011
12												

Figure 2: P-values for each combination of algorithms (Student t-test). Paired test, with each possible pair of algorithms.

	1	2	3	4	5	6	7	8	9	10	11	12
1		0.8974896	0.01156519	0.4170737	2.895859e-11	2.212592e-10	1.084879e-07	0.0001979305	9.242308e-08	4.191557e-10	0.347911	0.0651303
2			0.008035984	0.9188293	4.431562e-11	1.941264e-11	9.260215e-07	5.47346e-05	1.075245e-06	2.531668e-09	0.8120446	0.3791478
3				0.001823353	2.374693e-10	2.373371e-10	1.959898e-10	6.103222e-10	1.755021e-11	2.449646e-11	0.03686592	0.1482729
4					1.755296e-11	2.374733e-11	7.345834e-06	0.0009992534	1.064014e-07	1.845123e-11	0.1229569	0.02358027
5						0.2256882	1.047611e-10	1.154039e-10	1.669757e-11	1.669234e-11	1.310162e-09	1.761984e-09
6							1.668972e-11	1.668972e-11	1.667664e-11	1.668972e-11	3.167558e-11	2.256913e-11
7								6.801048e-06	0.1766011	2.028964e-06	5.587946e-11	3.780468e-11
8									0.0002255686	2.658776e-08	3.07165e-10	6.522541e-11
9										3.068796e-08	1.665574e-11	1.665051e-11
10											1.66688e-11	1.668187e-11
11												8.19375e-06
12												

Figure 3: P-values for each combination of algorithms (Wilcoxon test). Paired test, with each possible pair of algorithms.

The Wilcoxon test does not assume that the population is normally distributed. We see on figure 3 that the p-values give the same idea of the difference between the solutions.

We can see on the figures that a lot of values are below the threshold α. We can thus conclude that most of the algorithms generate solutions of significantly different quality. We can check it in figure 1.

4 Exercise 1.2