MAINTAINING NETWORK CONNECTIVITY

► STUDENTS / UNIVERSITIES

Güniz Irmak KÖKSALAN / Sabancı University
Deniz ŞAHİN / Sabancı University

► SUPERVISOR(S)

Tonguç ÜNLÜYURT

ABSTRACT

In a repairable system, failure diagnosis is the first step in repair. The sequence in which the components are tested can affect the cost of diagnosis. In previous works, the testing cost was independent of sequence. This paper considers the case when the testing cost depends on the previous test conducted. This may have applications for instance when the physical location of costs are different. Local search algorithms are implemented and related expected cost analysis shows that starting with cost/probability sequence with swapping and picking the best swap after several trials performs better than randomly generated sequences. In total 9 test case scenarios with 3 different component sizes (10-20-30) and different costs are used in the analysis.

OBJECTIVE

The objective of this project is by using different heuristics (local search algorithms-swap), determining a sequence minimizing the cost of finding the failing component of the given series system.

PROJECT DETAILS

- A repairable system
- Series network consists of n components
- Functions when all the components function
- Cost is associated with performing a test on each component
- Cost depends on the identity and on the order in which the component is tested
- The lifetime of components are independent random variables
- The tests are considered to be perfect (100% confidence)
- Optimal strategy is minimizing the expected cost (mean cost)

$$E[C(S)] = c_{[1]} + c_{[2]}(1-P_{[1]}) + c_{[3]}(1-P_{[1]}-P_{[2]}) + \dots + c_{[N-1]}\left(1-\sum_{j=1}^{N-2} P_{[j]}\right) \quad (4\text{-}1)$$

Figure 1. Under perfect testing mean cost of using any diagnostic strategy (Nachlas)

```
10

0.0483648 0.138223 0.148569 0.0938067 0.0527946 0.135349 0.0353754 0.156389 0.108931 0.082198

2.24335 7.05078 6.87622 0.159912 0.520935 8.32855 4.65302 2.17529 7.90985 7.43866
8.74664 9.86206 2.23083 0.474854 7.20795 5.06927 4.935 7.09137 4.04816 1.84113
0.652771 8.11096 7.66418 7.39502 5.50812 6.82495 8.18665 6.32324 5.76935 9.29657
7.82013 5.82733 3.39142 3.98346 3.22998 4.80194 0.0161743 5.98389 4.04572 4.57764
5.99213 6.23016 6.53931 4.13757 5.97595 0.931702 6.54022 7.61841 7.68005 1.58661
0.653687 6.90918 5.32471 0.870667 9.84009 7.16095 4.36096 4.71985 4.42139 8.01727
9.02649 5.91522 4.57611 8.38196 9.29413 9.48395 5.3833 7.53326 8.7793 5.2301
0.482483 0.39917 0.164795 7.7066 3.40637 9.80347 9.71008 8.94196 0.654297 3.49915
7.40295 4.23096 5.56488 8.79822 4.31976 6.30615 7.05414 2.61261 3.75427 8.20129
2.27539 9.71588 7.67883 0.117493 7.27173 7.04254 5.10071 0.605774 4.67682 6.83258
```

Figure 2. Data set 1 with n=10 components (1st row), probability values(2nd row), nxn cost matrix (starting with 3rd row)

- Generation code is used to create 9 data sets with 3 different component sizes, with randomly generated cost and probability values
- Local Search Algorithms:
  1. n=0, start with initial feasible solution (random/ “cost/prob” sequence)
  2. Swap randomly selected elements and calculate the cost
  3. If cost(x^n+1) > cost(x^n) set n= n+1, if not stop

EXPERIMENTS & RESULTS

1	Randomly generated sequence	• To start the problem an initial sequence is required, permutation is used to generate a sequence
2	Random + Best Swap Sequence	• Depending on number of swaps best solution is stored
3	Random + Latest Swap Sequence	• The last swap made is stored
4	Cost/Probability (c/p) Sequence	• When cost only depends on identity this method gives the optimal
5	Cost/Probability + Best Swap Sequence	• Adding swap to the problem improves the solution

- All 5 methods are tested in the 9 data sets
- 2 different sequences (randomly generated & cost/probability) are tested via swapping
- Number of swaps were 100-500-1000

Table 1. Best performing heuristic comparison for each data set

component size	Test Set	After 100 Swap Best	After 500 Swap Best	After 1000 Swap Best	c/p +100 Swap	c/p + 500 Swap	c/p + 1000 Swap
10	1	13.1367	9.54257	9.75684	13.3214	10.9019	**8.30225**
10	2	15.4566	11.3969	10.8871	17.1833	12.2486	**9.40173**
10	3	12.4396	9.99028	9.42955	12.5706	8.01171	**8.00984**
20	4	28.8696	28.3317	**21.8241**	31.2149	25.8348	25.3857
20	5	35.6422	29.5995	**25.0703**	29.7686	28.7577	25.6805
20	6	36.5622	29.6552	28.6159	38.9671	32.014	**23.7908**
30	7	60.3893	47.7856	50.8825	48.2616	48.2616	**40.7205**
30	8	56.7176	56.5569	**43.8948**	49.8825	45.1629	46.9933
30	9	64.841	47.2052	50.6122	50.5248	50.5248	**46.7298**

```
test 1
10
initial sequence
5 1 8 4 3 7 10 2 9 6
initial solution
32.8121

after swap best sequence
8 2 7 4 1 3 6 9 10 5
after swap best solution
13.1367

after random swap sequence
10 4 5 2 6 9 8 1 7 3
after random swap solution
16.9613

cost/probability sequence
10 2 8 5 1 7 4 6 3 9
cost/probability solution
13.3214

cost/probability + swapping sequence
10 2 8 5 1 7 4 6 3 9
cost/probability + swapping solution
13.3214
```

Figure 3. Result txt file of data set 1, n=10 and test 1 with 100 swaps

CONCLUSIONS

- After Best Swap and Cost/Probability with Best Swap performed better than the remaining 3 algorithms tested
- Swap number of 1000 gave the best result
- To get better results using more advanced heuristics such as local search algorithms with diversification is recommended

REFERENCES

Nachlas, J. A. (1991). Diagnostic-strategy selection for series systems. Microelectronics Reliability, 31(1), 205-206. doi:10.1016/0026-2714(91)90421-3