Level of Repair Analysis based on Genetic Algorithm with Tabu Search

Taoufik BOUACHERA[1], Mohammed KISHK[1] and Laurie POWER[2]

Abstract - Genetic algorithms and their hybrid schemes have shown a great efficacy in solving large scale combinatorial problems in which solutions are highly time-consuming. The level of repair analysis (LORA), mathematically formulised by an integer programming model (IP), is very difficult to optimize by means of traditional optimization techniques due to a large number of decision variables involved. In this paper, a hybridised Genetic Algorithm with Tabu Search is presented and its application to solve Level of repair analysis (LORA) problem is investigated. The LORA, considered as an important tool for strategic system maintenance decision making, seeks to determine the location in the repair network at which a failed component should be discarded or repaired. The proposed algorithm is developed in order to determine the best repair decision combination. The efficacy of the algorithm is investigated in the context of a case study. The maintenance costs of a structure of three-echelon repair and multi-indenture is optimised under the condition that repair decision should be taken for all system items. Typical results have shown that the algorithm can effectively handle a real industrial sized case study with adequate optimisation computational time.

Keywords*: Level of repair analysis, maintenance optimisation, Genetic Algorithms, Tabu Search.*

1. Introduction

1.1. Background

Over the last decade considerable emphasis has been put on whole life costing (Tysseland, 2007; Kleyner & al., 2008; Lindholm et al., 2004; Kishk et al., 2003). Traditional system acquisition has taken place on the purchase costs without explicit indication of operation costs. However, the decrease of company operational budgets has unveiled the need for techniques to forecast and to optimise ongoing cash flows over the system whole life. As result, a prerequisite for effective acquisition decision is the operational parameters such as: system availability, system reliability and the cost of the required support resources. Besides, it was found that operational readiness of expensive complex structures such as petroleum apparatus, aircrafts, ships and military equipments is very sensitive to the availability of spare components, maintenance resources and manpower.

[1]*The Scott Sutherland School, The Robert Gordon University, Garthdee Road, Aberdeen AB10 7QB, UK.*
[2]*School of Engineering, The Robert Gordon University, Schoolhill, Aberdeen AB10 1FR, UK.*

Since these support resources are very costly for systems containing thousands of components, there is always a trade-off between availability and maintenance costs. Ignoring these support issues in the early phase, however, may have costly consequences during the operation phase. This becomes in increasingly evident with fierce market conditions. As a result, companies are compelled to move from fragmented decision processes toward more integrated acquisition and operation decision in order to sustain their equipment use at low cost and required availability. Consequently, some industries become aware of the large potential for cost reductions by adopting whole life techniques in their acquisition process. Level of repair analysis LORA is one of the prescribed techniques in the military and maritime industries to achieve a system design with the minimum whole life maintenance cost (MIL.STD.1390D).

1.2. Level Of Repair Analysis (LORA)

The LORA approach was developed by military industry to evaluate the Integrated Logistic Support (ILS) factors contributing to the systems whole life cost. When a failure occurs, failed components are removed and repaired or replaced by new spare parts. In designing systems, Level of repair analysts considers all aspects of the system design and maintenance scenarios to achieve availability and cost balanced systems. As a result, they provide essential support requirements for the most effective maintenance strategy under predicted operational environments.

The basic of the LORA process is the following. Level of repair analysts have to decide for a given design which components to repair, which components to discard, where in the repair network to do this and finally where in the repair network to install the required maintenance resources. Thus, a number of reparation locations in which systems, subsystems and components have to be repaired or discarded is set up to satisfy maintenance requirements at minimum cost.

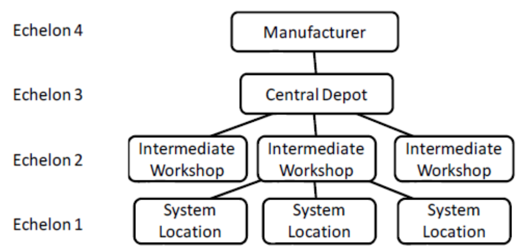

Figure 1 : A multi-echelon repair network

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

In the literature, various models of LORA have been discussed for a three echelon repair network (figure 2) and multi indenture system (figure 1). These models involve a large number of decision variables which makes LORA problem very difficult to optimize by means of traditional optimization techniques. For instance, the number of all possible combinations (part, repair and discard decision) for a system consisting of 32 parts spread between different indentures is 6.28×10^{10} (Kumar & al., 2006). Hence, techniques like integer programming and branch and bound method become difficult to use. Consequently, we focus on the opportunity of using Genetic Algorithms which are the most suitable to problems involving combinatorial optimisations.

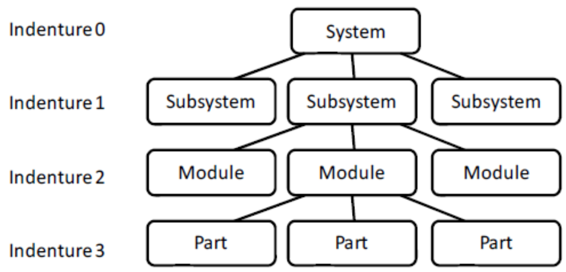

Figure 2 : A multi-indenture system

1.3. Objective & methodology

Based on the discussions above, this paper addresses the issue of minimising maintenance costs based on level of repair analysis. This optimisation is modelled as integer programming (IP) model in which the space solution is proportional to the number of echelons and the system structure size. In addition, not all maintenance features are studied, but the study is restricted only to model the effect of repair capacity on whole life maintenance cost. Optimal LORA approaches often make use of heuristic methods to optimize the IP model. In this sense, the main aim of this study is to show that Genetic Algorithms and Tabu Search can yield to optimal solution at an acceptable computing time with respect to solution space. The obtained algorithms can be seen as building blocks for more general maintenance optimisation model.

The remainder of the paper is organized as follows. Section 2 presents related works and a mathematical formulation of the LORA problem. Section 3 describes Genetic Algorithm, Tabu Search and their combination GATS algorithm used to solve the level of repair analysis (LORA) problem. Section 4 presents a numerical application of the proposed algorithm. Section 5 concludes the paper and offers some suggested directions for future research.

2. Related works

The most economical maintenance strategy for any component of system is to decide it is worth repairing or discarding it. Level of Repair Analysis (LORA) is an approach which examines the cost balance between repairing the component and discarding it. The framework of this approach is an iterative process ensuring the optimum maintenance planning. However, the LORA problem as combinatorial optimization is not the most widely studied in the literature (Basten & al., 2006). A limited works were devoted to solve the LORA problem. Barros et al. (2001), Saranga et al. (2006), Gutin et al. (2005) and Basten et al. (2009) modelled LORA as Integer Programming model in which all repair locations at the same echelon were aggregated. Besides, they all resolved their model under infinite capacity of resources. Brick et al. (2009) model LORA without aggregating data per echelon level for only 1 echelon and 2 indenture levels.

Barros & al. (2001) presented a mathematical framework as an Integer Programming (IP) model resolved by branch and bound algorithms. In this model the objective function has two elements: a fixed costFC of setting up maintenance facilities (test equipment, labour manpower and technical data); and a variable cost CV of ordering and holding spare parts. The goal of this IP problem is to find a subset of repair decisions that minimises the total maintenance costs while satisfying parts relationship and maintenance resource constraints. They have assumed that any installed maintenance capacity (fixed cost) performs reparation to all components belonging to the same indenture. Brik et al. (2009) work treated the applicability the location of facilities and installation of capacitated resources to LORA problems. They have proposed a mixed-integer problem MIP model for the discrete location of facilities and installation. Gutin et al. (2005) formulate the LORA problem as an optimization homomorphism problem on bipartite graphs and they have proved that the LORA problem a NP-hard problem. Saranga & al. (2006) adopt the same Barros Integer Programming model but with different fixed cost allocation. They considered that any component bears a specific fixed cost whereas in Barros model all components at the same indenture share the same fixed costs. Furthermore, Sarraga et al. have solved LORA problem by using the genetic algorithm software evolver. Basten & al., (2009) propose an Integer Programming model that generalizes the existing models (Barros model and Saranga model) by allowing a predefined set of components to share the same fixed costs. In addition, they modelled the LORA problem as a minimum cost flow problem with side constraints.

In this paper, we model the LORA problem as described in Barros, Sarranga and Basten models. We obtain the optimum repair decisions by minimising the maintenance cost given by equation (1) under the constraints (2, 3 and 4). The added value of this paper to the literature is that we used a hybrid Genetic and Tabu Search algorithms which are very suitable to NP-hard problems and combinatorial optimisation.

The following notations are used in our model.

- m = the number of the echelons in the reparation network.
- n = the total number of components for the system under consideration.

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Component i is the parent of the component j or component j is the child of the component i

r = repair options: repair, discard or move.

λ_i = Total number of maintenance tasks required in the whole life time of component i.

$CF_{r,e,i}$ = fixed cost related to repair option 'r ' at echelon e, for component i.

$VC_{r,e,i}$ = variable cost related to repair option 'r' at echelon e, for component i.

Let X, the repair decision, is 1 if repair decision has been chosen at the echelon e of the selected component i and 0 otherwise.

$$X_{r,e,i} = \begin{cases} 1 & \text{if repair option } r \text{ at echelon } e \text{ is selected for part } i \\ 0, & \text{otherwise} \end{cases}$$

Using the notation mentioned above, the total maintenance cost is:

$$\sum_{i=1}^{N} \sum_{r=1}^{3} \sum_{e=1}^{m} \left[VC_{r,e,i}\ \lambda_{it} + FC_{r,e,i} \right] X_{r,e,i} \tag{1}$$

Subject to

$$X_{r,\ 1,\ i} = 1 \qquad \text{for all parts} \tag{2}$$

$$X_{move,\ e,\ i} = \sum_{r=1}^{3} X_{r,\ e+1,\ i} = 1 \tag{3}$$

$$X_{r,\ e,\ j} = X_{r,\ e,\ i}$$
$$\forall\, e \text{ and (i is parent of j) where } r = \text{discard or move} \tag{4}$$

The objective function given in Eq. (1) sums the fixed and variable costs of performing repair and discard actions. The constraint given in Eq. (2) ensures that one repair option is chosen at the echelon one. If a move decision is taken at the echelon e, only one repair decision should be taken at the echelon e+1 (constraint given in Eq. 3). Otherwise, no repair option is chosen at the echelon (e+1).The equality constraint given in Eq. (4) requires that all the enclosed lower indentures of any subsystem have the same decisions of the subsystem itself with respect to the replace and move options at different echelons. The last constraint requires that there are only two repair decisions (repair or discard) at the last highest echelon. Figure (3) represents a sample of possible solution generated randomly by taking into consideration all the above constraints.

3. Hybrid Genetic & Tabu Search algorithms

Either the genetic algorithm GA or the tabu search TS are suitable tools for solving such problems. In the literature, however, several researchers have tried to combine these two algorithms to enhance their capabilities in solving combinatorial optimisation (Zdanski & al., 2002 and Hagemana & al., 2003). For instance, a GA speed is low for the huge size population and TS relies strongly on the initial solution. Consequently, GA and TS combination named GATS may overcome these limitations and maintain their advantages.

3.1. Genetic algorithms

Genetic algorithms are stochastic search techniques based on the theory of evolution for finding the global optimum solution. The genetic algorithm developed by Holland to optimise a function F(x), where x is a vector representing individual solutions (Gen & al. 2000). First of all, Genetic algorithms generate not only a single solution but a group of solutions, called a population. This population changes over time, but it always keeps its initial size. The population members are called strings or chromosomes from which a subset called parents is selected according to the best values of F(x). A fitness value in Genetic algorithms is a measure of goodness of a solution to the objective function, i.e., the fitness of an individual is directly related to its objective function value. At any iteration, a fitness value is calculated for each of the current individuals. The selection rule, called a survivability test, exclude from the population the strings which have the worst finesses. Second, new solutions called children (or offspring) are produced by genetic operators: crossover and mutation. Together parents and new children are grouped in a new population which will pass again through survival test. Thus, the population as a whole moves iteratively towards better solutions ideally to the global optimum.

Chromosome representation

The first step in implementing a genetic algorithm for a particular problem is to adopt a suitable chromosome representation. The representation scheme developed in this paper was a (n x d) binary matrix, where n is the number of all parts under consideration and d is the number of all the repair decisions throughout the repair network. A value of 1 in this representation implies that a repair, discard or move decision has been attributed to the component i and the echelon j. The binary representation of any chromosome or solution is visualised in Figure 3.

Furthermore, any technical system may be considered as collection of assemblies which are in turn considered as a collection of a set of subassemblies. This perspective on technical system is illustrated in figure 2. The number of levels, also referred as indenture levels, in the material breakdown structure of technical system is limited to the deeper detailed information needed for repair tasks and spare-part provision.

Fro a modelling perspective, the system breakdown structure is represented by a matrix, referred in the literature by

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

commonality matrix (figure 4), where the column represents parent items and in the row are child items. We start by assorting parts from the first indenture until the last but one indenture in the column as parent items. Then, we insert parts from the second indenture to the last one in the commonality matrix row. As shown, child parts 5, 6 and 7 belong to parent part 3 or parent part 3 is constituted of child parts 5, 6 and 7. According to this representation, whenever the parent part 3 is under discard or move decision, the child parts 5, 6 and 7 will have the same decision (constraint Eq. 4).

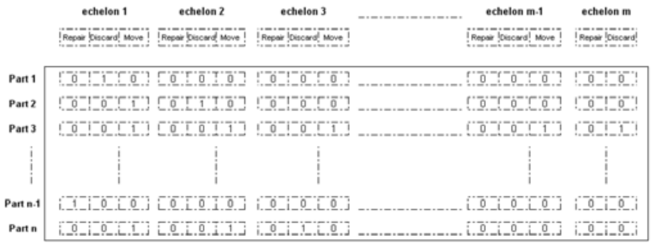

Figure 3: Sample of repair decision

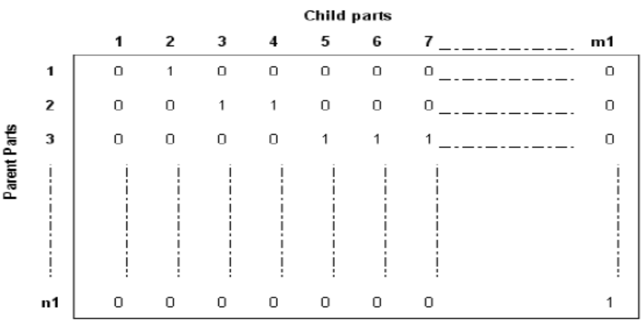

Figure 4: Matrix representation for system structure

Genetic Algorithm operators

The GATS algorithm uses fitness proportional selection with roulette wheel sampling for crossover operator. At each generation Elitism is applied in this study by replacing the worst solution by the best one with respect to total cost given in Eq. (1). After a pair of parents is selected, the crossover operator produces two new children or off springs. The crossover operator is applied on these two parent chromosomes by interchanging the information extracted from them. Since each parent's genetic code has the same structure, we apply one-point the crossover by considering the same crossover point selected at random. The children are generated by combining the left and right parts (figure 2); which is followed by adjusting the offspring repair decisions with respect to the constraint Eq. (4).

On the other hand, mutation is the other important element in genetic algorithms that creates randomly new children. This operator serves as a strategy to prevent solutions from being trapped in local optima. In this work, the mutation operator works by selecting randomly one chromosome outside the best solution list and replacing it by a new chromosome also generated randomly. In addition, we select one of the best solutions and we generate a repair decision for a component selected at random. Again, we adjust the new changes according to the constraint Eq. (4).

In our GATS algorithm, these two operators are applied for the individual generated by Genetic Algorithm and improved by tabu search.

3.2. Tabu Search

Tabu Search, concept based on the use of memory, tries to keep track of solution already visited. By leading the optimisation to new areas, TS is able to attain the global optimum instead of local minima. The framework of TS consists of generating some neighbouring solutions from an initial solution (Eswaramurthy & al., 2009). These solutions are evaluated by means of objective function and sorted. The tabu list is updated by the best solution according to its fitness. Afterwards, a new solution is identified and additional neighbouring searches are generated from it. When the best solution remains unchanged after a number of iterations, the optimum is achieved and the best solution will be returned.

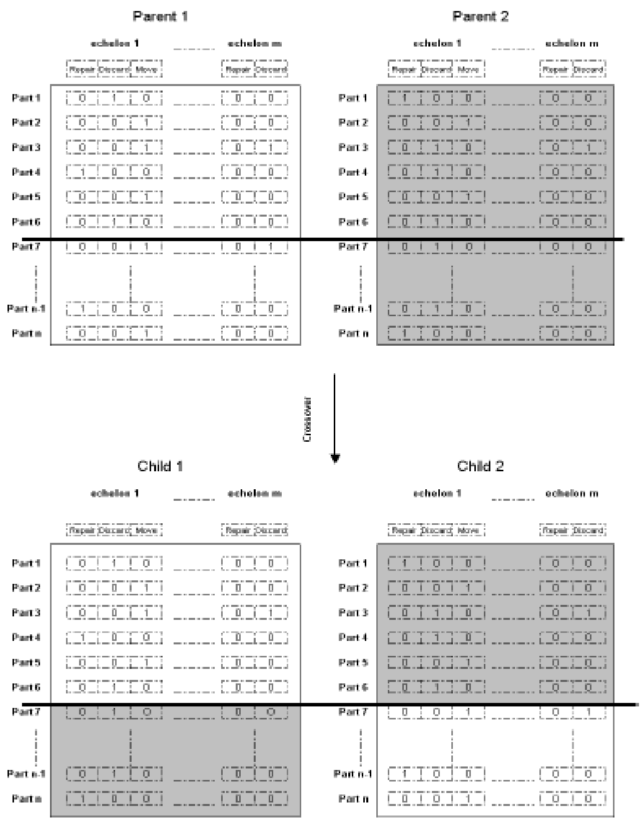

Figure 5: An example use of the crossover operator

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

The procedure of TS consists of the following steps which are depicted in the figure (6). First, a number of neighbourhood solutions that can be produced from an initial solution are examined. Then, a solution with the best fitness value and it is not in the tabu list is selected from the explored neighbourhood. This way, tabu search tries to assure that the method does not re-examine a solution previously generated. Finally, TS procedure iterates the previous step until no more neighbours are present (all are tabu), or when during a predetermined number of iterations no improvements are found.

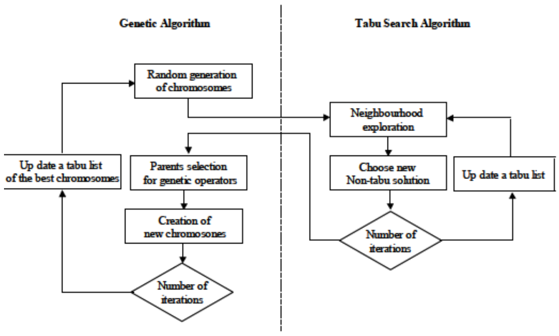

Figure 6: General flowchart of the GATS algorithm

3.3. Genetic & Tabu Search Algorithm

This approach, widely used in the literature, combines the advantages and mitigates the disadvantages of the two algorithms. Tabu search relies only on one solution and miss information of a larger set of solutions, however, Genetic Algorithms lead to lower solution quality with increasing problem size (Zdanski & al., 2002). In this study the GATS algorithm starts by generating N initial possible solutions (figure 6). A tabu search, as an iterative process, is then used for upgrading theses solutions through neighbouring exploration. Afterwards, the flow returns to the Genetic Algorithm which is again an iterative process. By means of the genetic operators new off springs are produced. Then, a tabu list of the best solutions is updated by the new off springs according to the fitness value. The stopping criteria for the GATS algorithm are a predefined number of consecutive iterations attaining the same best solution is reached.

The mean steps of the algorithm are shown in figure 6 and described as follows:

1. Generate randomly a set of solutions (20 solutions) verifying the equations 2, 3 and 4.
2. Refine each solution by the neighbourhood routine with respect to fitness value. A neighbourhood solution is obtained only by modifying the value of one element from the solution under consideration to 1 or 0. Besides, the neighbourhood solutions are not accepted until they verify the constraint equations 2, 3 and 4. Then, a tabu list is updated containing all the fitness values of the solutions that have been explored. After, a new neighbourhood is explored only when its fitness value does not exist to the tabu list.
3. Repeat step 2 until there is no improvement of the best fitness value.
4. Replace the solution by its best neighbourhood.
5. Choose two solutions to produce new chromosomes using genetic operators: parent selection and crossover. These new solution are accepted when they verify the constraint equations 2, 3 and 4.
6. Create a new chromosomes using genetic operator: mutation.
7. Update a tabu list of the best chromosomes.
8. Repeat step 1 until there is no improvement of the best chromosome.

The proposed algorithm has been implemented into a computer routine using the MATLAB® programming environment (The MathWorks, 2008).

4. Computational experience

In this section, we present the results of numerical experiments to test the effectiveness of our GATS algorithm. For comparison sake, we conduct first the same case study that was carried out in Saranga's work (Saranga & al., 2006).

Table 1: Case study data used in Saranga's work, maintenance costs

Part	Echelon 2						Echelon 3				Lampda
	VC repair	VC discard	VC move	FC repair	FC discard	FC move	VC repair	VC discard	FC repair	FC discard	
1	400	1500	40	150	80	360	620	1560	720	25	0,110
2	1000	2500	45	1000	85	200	1050	560	1150	34	0,164
3	1000	3000	60	150	90	900	1100	2020	1250	40	0,041
4	900	1000	20	200	64	1000	950	1050	1900	40	0,199
5	200	1200	30	200	22	600	700	1250	750	28	0,217
6	1500	1800	50	2500	40	200	1604	500	2700	50	0,193
7	1200	2000	60	1800	40	600	1250	2005	2000	42	0,294
8	500	1000	30	2000	100	50	505	600	2100	50	0,075
9	500	2000	50	1200	30	200	600	2050	1250	31	0,036
10	400	1000	40	1500	20	360	450	1000	1800	22	0,066
11	400	900	10	700	20	360	300	902	720	25	0,067
12	400	600	10	700	20	360	340	601	720	25	0,043
13	300	1000	20	1000	85	50	250	501	1150	34	0,068
14	250	800	20	1000	80	50	200	201	1150	34	0,058
15	150	700	20	1000	80	50	100	201	1150	34	0,038
16	800	1500	30	100	30	900	500	1502	1250	40	0,004
17	300	500	10	120	30	900	300	501	1250	40	0,037
18	100	400	20	200	64	1000	75	405	1900	40	0,081
19	75	300	10	200	34	1000	60	302	1900	40	0,066
20	250	300	10	200	34	1000	200	301	1900	40	0,052
21	500	1000	20	200	22	600	400	401	750	28	0,109
22	100	200	10	150	22	600	75	150	750	28	0,108
23	500	1200	40	2500	40	50	450	240	2700	50	0,066
24	400	600	30	2500	40	50	300	130	2700	40	0,127
25	400	600	50	1800	40	600	300	604	2000	42	0,063
26	200	400	40	1800	40	600	190	405	2000	42	0,037
27	300	360	30	1800	40	600	300	351	2000	42	0,037
28	250	250	20	1800	40	600	200	250	2000	42	0,055
29	100	200	10	1800	40	600	100	202	2000	42	0,040
30	75	200	10	1800	40	600	50	201	2000	42	0,062
31	500	1000	30	2000	46	40	505	500	2100	30	0,075
32	500	2000	50	1200	30	400	600	2050	1250	31	0,036

Table 2 : Case study data used in Saranga's work, material structure

Commonality matrix

	Part 11	Part 12	Part 13	Part 14	Part 15	Part 16	Part 17	Part 18	Part 19	Part 20	Part 21	Part 22	Part 23	Part 24	Part 25	Part 26	Part 27	Part 28	Part 29	Part 30	Part 31	Part 32
Part 1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Part 2	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Part 3	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Part 4	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Part 5	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
Part 6	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
Part 7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0
Part 8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Part 9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Part 10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

In this experiment, the settings are chosen as described in (Saranga & al., 2006) on two echelon repair network for an aircraft engine with three-indenture structure. Different costs (CF: fixed costs and CV: variable costs) for various repair

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

options at different echelons for all items are listed in Table (1). Besides, the commonality matrix shows the relationship between first indenture parts (from 1 to 10) and the second indenture parts (from 11 to 32). The optimal or near optimal solution obtained by Saranga' work and our GATS algorithm are similar, only part 5 got different repair decision (table 3). The total maintenance costs incurred are respectively 4255.274 and 4216.274. In addition, the maximum maintenance cost achieve by the simulation is 42 759 representing more than ten times the optimum solution. This witnesses the benefit of adopting LORA in maintenance planning and support provision.

As a second important issue related to the optimisation problem is the computational time. The algorithms GATS is written in the MATLAB language and implemented on a Pentium 4 CPU 2.60 GHZ with 512 Mo RAM. The computing time required to solve the LORA problem varies with system structure (total number of items) and the repair network. Figure 7 gives the computing time taken to solve the problems for the data sets created randomly for 3 echelon network. For problem that has been discussed above, it took an average time of 21 seconds to solve the problems. As was previously mentioned, the solution representation is a (n x d) binary matrix, where n is the number of all parts under consideration and d the number of all repair decision throughout the repair network. The solution has for a system with n parts with m echelons and ri repair options at echelon i, the number of possible solutions is equal to 2^{n*s}.

Where: $s = \sum_{i=1}^{m} r_i$, r_i is the number of repair options at echelon i.

For a case study with 40 parts, the size of the solution space will be for 3 echelons as high as $2{,}14 \times 10^{96}$.

Table 3 : Best LORA solution for Saranga's case study

	echelon 1			echelon m	
	Repair	Discard	Move	Repair	Discard
Part 1	1	0	0	0	0
Part 2	0	0	1	0	1
Part 3	1	0	0	0	0
Part 4	0	1	0	0	0
Part 5	0	1	0	0	0
Part 6	0	0	1	0	1
Part 7	0	1	0	0	0
Part 8	0	0	1	0	1
Part 9	0	1	0	0	0
Part 10	0	1	0	0	0
Part 11	0	1	0	0	0
Part 12	0	1	0	0	0
Part 13	0	0	1	0	1
Part 14	0	0	1	0	1
Part 15	0	0	1	0	1
Part 16	0	1	0	0	0
Part 17	0	1	0	0	0
Part 18	0	1	0	0	0
Part 19	0	1	0	0	0
Part 20	0	1	0	0	0
Part 21	0	1	0	0	0
Part 22	0	1	0	0	0
Part 23	0	0	1	0	1
Part 24	0	0	1	0	1
Part 25	0	1	0	0	0
Part 26	0	1	0	0	0
Part 27	0	1	0	0	0
Part 28	0	1	0	0	0
Part 29	0	1	0	0	0
Part 30	0	1	0	0	0
Part 31	0	0	1	0	1
Part 32	0	1	0	0	0

Total maintenance costs = **4255.274**

Best solution obtained by Saranga's work

	echelon 1			echelon m	
	Repair	Discard	Move	Repair	Discard
Part 1	1	0	0	0	0
Part 2	0	0	1	0	1
Part 3	1	0	0	0	0
Part 4	0	1	0	0	0
Part 5	1	0	0	0	0
Part 6	0	0	1	0	1
Part 7	0	1	0	0	0
Part 8	0	0	1	0	1
Part 9	0	1	0	0	0
Part 10	0	1	0	0	0
Part 11	0	1	0	0	0
Part 12	0	1	0	0	0
Part 13	0	0	1	0	1
Part 14	0	0	1	0	1
Part 15	0	0	1	0	1
Part 16	0	1	0	0	0
Part 17	0	1	0	0	0
Part 18	0	1	0	0	0
Part 19	0	1	0	0	0
Part 20	0	1	0	0	0
Part 21	0	1	0	0	0
Part 22	0	1	0	0	0
Part 23	0	0	1	0	1
Part 24	0	0	1	0	1
Part 25	0	1	0	0	0
Part 26	0	1	0	0	0
Part 27	0	1	0	0	0
Part 28	0	1	0	0	0
Part 29	0	1	0	0	0
Part 30	0	1	0	0	0
Part 31	0	0	1	0	1
Part 32	0	1	0	0	0

Total maintenance costs = **4216,274**

Best solution delivered by our GATS algorithm

A comparison between two and three echelon network computational time that takes GATS algorithm to come out with the optimal solution is shown in (figure 7). The computational time increases exponentially with system structure size and the bigger the number of echelon is the higher the computing time is. Thus, researchers consider three echelon repair network is enough in practice to handle maintenance activities and to be modelled by acceptable computational time.

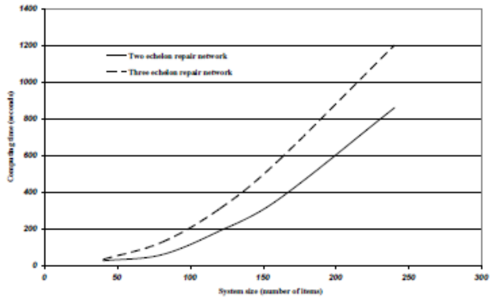

Figure 7 : GATS computational time

5. Conclusion and future works

A typical level of repair analysis includes multiple repair facilities and thousands of system items, is formalised by Integer Programming (IP) model. Traditional optimisation techniques can not be effectively applied to solve LORA models for real-world applications in which systems may enclose millions of parts. In this paper, a hybrid algorithm of Genetic Algorithm and Tabu Search (GATS) has been developed and implemented into a computational algorithm in MATLAB code to solve this mathematical formulation. The algorithm adopts a matrix representation for the system breakdown structure to handle the constraint linking parent items and children items. The efficacy of the algorithm has been validated in the context of an example application. The repair decision of all system items has been optimised for a structure of three-echelon repair network and multi-indenture system. The results have shown that quite large LORA optimisation can be obtained in realistic times, demonstrating that the algorithm is practical.

There are some practical issues that need to be addressed, however. This LORA problem optimises maintenance costs based only on repair facilities. This should be extended to include other maintenance costs such as spare part provision. Further research in this area will include studying the impact of both spare part provision and repair facilities on LORA problems. Besides, spare part optimisation under finite repair capacity is being integrated into the development of the algorithm and will be reported in a future paper.

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

References

1. Alfredsson, P., 1997, “Optimization of multi-echelon repairable item inventory systems with simultaneous location of repair facilities”. European Journal of Operational Research, 99:584
2. Barros, L. and Riley, M., 2001, “A combinatorial approach to level of repair analysis”, European Journal of Operational Research
3. Basten, R. J. I., Schutten, J. M. J., and Van der Heijden, M.C., 2009, ”An efficient model formulation for level of repair analysis”. Annals of Operations Research.
4. Brick, E.D. and Uchoa, E., 2009, “A facility location and installation of resources model for level of repair analysis” European Journal of Operational Research, 192(2):479-486.
5. Eswaramurthy, V. P. and Tamilarasi, A., 2009, “Hybridization of Ant Colony Optimization Strategies in Tabu Search for Solving Job Shop Scheduling Problems”, International Journal of Information and Management Sciences
6. Gen, M. and Cheng., R.,2000, “Genetic Algorithms and Engineering Optimization”. John Wiley & Sons, Inc.
7. Glover, F. and Laguna, M., 1997, “Tabu search”, Kluwer Academic Publishers, 1997
8. Goldberg, DE., 1989, “Genetic algorithms in search, optimization and machine learning”. Reading, MA: Addison-Wesley.
9. Gutin, G., Rafiey, A., Yeo, A., & Tso, M., 2005, “Level of repair analysis and minimum cost homomorphisms of graphs”. Discrete Applied Mathematics.
10. Hagemana, J.A.; Wehrens R.; Van Sprang H.A. and Buydens L.M.C, 2003, “Hybrid genetic algorithm–tabu search approach for optimising multilayer optical coatings, Analytica Chimica Acta 490
11. Jones, JV., 2006, “Integrated Logistics Support Handbook”, McGraw-Hill; NewYork, 3rd edition.
12. Kishk, M.; Al-Hajj, A.; Pollock, R.; Aouad, G.; Bakis, N.; and Sun, M., 2003, “Whole life costing in construction A state of the art review”, Published by RICS Foundation, London, UK.
13. Kleynera, A. and Sandborn, P., 2008, “Minimizing life cycle cost by managing product reliability via validation plan and warranty return cost”, International Journal of Production Economics
14. Lindholm A., and Suomala P., 2005: ‘Present and Future of Life Cycle Costing: Reflections from Finnish Companies’, Canadian Journal of Civil Engineering 32: 250–259.
15. MIL.STD.1390D, 1988, “US Military Standard: Level of repair analysis”. US department of defence.
16. Saranga, H. and Dinesh Kumar, U., 2006, “Optimization of aircraft maintenance/support infrastructure using genetic algorithms - level of repair analysis”. Annals of Operations Research.
17. Tysseland, B. E., 2007, “Life cycle cost based procurement decisions A case study of Norwegian Defence Procurement projects”, International Journal of Project Management.
18. Zdanski, M. And Pozivily, J., 2002, “Combination genetic/tabu search algorithm for hybrid flowshops optimization”, Proceedings of Algoritmy, Conference on Scientific Computing.
19. The MathWorks (2008) Using MATLAB®. The MathWorks, Inc., version 7.6.