



**HAL**  
open science

## Grouping maintenance strategy with availability constraint under limited repairmen

Phuc Do Van, Hai Canh Vu, Anne Barros, Christophe Bérenguer

► **To cite this version:**

Phuc Do Van, Hai Canh Vu, Anne Barros, Christophe Bérenguer. Grouping maintenance strategy with availability constraint under limited repairmen. SAFEPROCESS 2012 - 8th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, Aug 2012, Mexico city, Mexico. 10.3182/20120829-3-MX-2028.00194 . hal-00695786

**HAL Id: hal-00695786**

**<https://hal.science/hal-00695786v1>**

Submitted on 9 May 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Grouping maintenance strategy with availability constraint under limited repairmen

Phuc Do Van\* Hai Canh Vu\*\* Anne Barros\*\*  
Christophe Berenguer\*\*\*

\* *Lorraine university, CRAN, Nancy, France*

\*\* *Troyes university of technology, Troyes, France*

\*\*\* *Grenoble institute of technology, Grenoble, France*

---

**Abstract:** The paper deals with a development of the rolling horizon approach introduced recently for grouping maintenance strategies of multi-component systems by integrating two efficient optimization algorithms. The proposed approach can help to construct an optimal maintenance planning with a given availability constraint under limited repairmen. Thanks to the rolling horizon spirit, the maintenance planning can be updated to take into account short-term information which could be changed with time. A numerical example of a multi-component system is finally introduced to illustrate the use and the advantages of the proposed approach in the maintenance optimization framework.

Keywords: Grouping maintenance, availability, genetic algorithm, multi-unit system.

---

## 1. INTRODUCTION

In the maintenance optimization framework, grouping strategy has been recently introduced for multi-component system with positive economic dependence which applies that combining maintenance activities is cheaper than performing maintenance on components separately, see for example Cho and Parlar [1991], Galante and Passannanti [2009], Radouane et al. [2009], Bouvard et al. [2011].

A major challenge of the maintenance optimization then consists in joining the stochastic processes regarding to the components (time-dependent probabilities of failure) with the combinatorial problems regarding to the grouping of maintenance activities. While a long term or infinite planning horizon can be assumed to solve this problem in case of stable situations, dynamic models have been introduced in order to change the planning rules according to short-term information (e.g. failures and varying deterioration of components), using a rolling (finite) horizon approach Wildeman et al. [1997], Nicolai and Dekker [2007]. This approach is however limited to a class of problems with significant assumptions: maintenance durations are neglected and only one preventive maintenance for each component is preventively maintained only is considered in horizon interval. To solve these problems, an extension of the rolling horizon approach is recently introduced in Do Van et al. [2011]. Nevertheless in such papers, maintenance constraints have not yet been considered. From a practical point of view, it is often impossible to perform all the desirable maintenance actions due to the limitations on maintenance resources, such as maintenance budget, duration of maintenance time, and limited repairmen. The grouping maintenance problem remains widely open.

The aim of this paper is to propose an approach based on the rolling horizon spirit for grouping maintenance planning with availability constraint under limited repairmen. For example, according to a specific mission or a production campaign, the system may be required to operate with limited breaks durations, i.e. the maintenance duration time of the system have to be limited. With these constraints, to find optimal planning, dynamic programming presented in Wildeman et al. [1997], Do Van et al. [2011] is no longer usable since the combinatorial problem can be formulated as a set partitioning problem, which however can be NP-hard. To this end, the Genetic Algorithm (GA), recognized as a general search strategy which is often useful for solving combinatorial problems, see Holland [1975], Rahman and Ahmed [2009], is used. Moreover, the MULTIFIT algorithm introduced recently for bin-packing problems by Coffman et al. [1978] is also considered to determine the minimum maintenance duration time of each group under limited repairmen.

This paper is organized as follows. Section 2 is devoted to the description of general assumptions and maintenance policies. Section 3 focuses on the development of the rolling horizon approach under maintenance constraints. Two optimization algorithms (GA and MULTIFIT) are presented in Section 4. To illustrate the proposed grouping maintenance strategy, a simple numerical example is introduced in Section 5. In addition some numerical results are discussed here. Finally, the last section presents the conclusions drawn from this work.

## 2. PROBLEM FORMULATION

### 2.1 General assumptions

Consider a series system consisting of  $n$  independent components. With this kind of systems, a preventive/corrective maintenance of one or more components leads to the system being unavailable. We assume here that the failure rate of component  $i$  ( $i = 1, \dots, n$ ) is described by a Weibull distribution with scale parameter  $\lambda_i > 0$ , and shape parameter  $\beta_i > 1$ , then

$$r_i(t) = \frac{\beta_i}{\lambda_i} \left( \frac{t}{\lambda_i} \right)^{\beta_i - 1}. \quad (1)$$

We assume next that the deterioration of a component remains unchanged during the maintenance of other ones.

If a component fails, it is then immediately repaired. A corrective action restores the component involved into a state as good as before (minimal repair policy). Within this corrective policy, corrective maintenance durations are usually very small with respect to maintenance planning horizon and can be neglected. Therefore, when a corrective action is carried out for component  $i$ , it requires a corrective maintenance cost denoted  $C_i^c$ .

We assume that preventive maintenance actions can be planned in advance to prevent failures or/and to reduce maintenance costs. Herein, each component is preventively maintained after a fixed interval to be optimized. After a preventive maintenance action, the maintained component is considered as good as new. The preventive maintenance cost of the component  $i$  can be divided in three parts:

- a specific component cost  $c_i^p$ ;
- a setup-cost, denoted  $S$ , represents the preparation cost (or logistic cost) and can be shared when several maintenance activities are performed in the same maintenance occasion since execution of a group of maintenance activities requires usually only one set-up;
- an unavailability cost: since the system is unavailable during maintenance, therefore, if a preventive maintenance of component  $i$ , so-called activity  $i$ , leads the system to be unavailable during  $d_i$  units of time, an additional cost or unavailability cost  $c_i^d = d_i \cdot C^d$  relying on the production lost is incurred ( $C^d$  is the unavailability cost rate of the system). This additional cost can be also shared when several maintenance activities are simultaneously carried.

As consequence, if a preventive maintenance of component  $i$ , namely activity  $i$ , is separately performed, we have to pay the following preventive cost:

$$C_i^p = S + c_i^p + c_i^d. \quad (2)$$

Based on the preventive/corrective maintenance costs and the reliability parameters of components, it is possible to define for each individual component  $i$  a nominal preventive maintenance periodicity  $x_i^*$  that minimizes the induced long-term average cost Wildeman et al. [1997], assuming that the maintenance activities are separately performed.

### 2.2 Maintenance constraints

If we consider an mission interval of time  $PH$ , the average availability of the system can be then calculated by:

$$A = \frac{PH - D}{PH}, \quad (3)$$

where  $D$  is the total maintenance durations within this period or the total preventive maintenance durations since corrective maintenance durations are zero.

Assume now that the the average availability of the system is required to be higher a given level  $A_0$  for this interval of time, i.e.  $A \geq A_0$ . From Equation (3), we have

$$D \leq (1 - A_0) * PH \text{ or } D \leq D_0, \quad (4)$$

where  $D = (1 - A_0) * PH$  is the maximal total duration allocated for all preventive activities in the considered scheduling interval  $PH$ . To execute preventive activities, we suppose that that only  $m$  repairmen are available and each repairman can take only one preventive activity at a time.

### 2.3 Grouping maintenance

We are here interested in a grouping maintenance strategy according to two following reasons:

- (1) it is shown in a number of papers, see for example Cho and Parlar [1991], Wildeman et al. [1997], that grouping can save maintenance costs since the set-up cost and the unavailability cost can be shared when several maintenance activities are executed together. Note well however that when several maintenance activities are performed together, maintenance cost could be indirectly penalized
  - with the reduction of components useful life if the maintenance dates are advanced;
  - with the increasing of components failure probability which could imply a system immobilization if the maintenance dates are too late.
- (2) according to the availability constraint, we have to execute simultaneously preventive activities to reduce maintenance durations even this grouping action may lead to a higher maintenance cost in some cases. For example, when the set-up cost and unavailability cost rate are equal to zero, grouping is then more costly but without grouping the required availability level may not be reached.

In order to find the optimal groups which could balance to minimize the system maintenance costs under given maintenance constraints (total maintenance duration allowed under limited repairmen) on the scheduling horizon, the rolling horizon approach introduced recently in Wildeman et al. [1997], Do Van et al. [2011] will be developed and described in Section 3.

## 3. GROUPING MAINTENANCE STRATEGY

The developed grouping maintenance approach is divided into 4 phases:

- individual optimization;
- tentative planning;
- grouping optimization;
- update and decision.

### 3.1 Phase 1: Individual optimization

The objective is to find optimal individual preventive maintenance cycle based on an infinite-horizon maintenance model in which we assume an average use of components and in which the interactions between components are neglected.

Let  $M_i(x)$  denote the expected deterioration cost for component  $i$ , i.e. the expected costs incurred in  $x$  time units since the latest execution of activity  $i$ . According to the minimal repair policy,  $M_i(x)$  can be expressed as the following:

$$M_i(x) = C_i^c \cdot \int_0^x r_i(y) dy, \quad (5)$$

where  $r_i(\cdot)$  is the rate of occurrence of failure of component  $i$ . From equations (1) and (5), we obtain:

$$M_i(x) = C_i^c \cdot \left(\frac{x}{\lambda_i}\right)^{\beta_i}. \quad (6)$$

If component  $i$  is preventively maintained at  $x$ , the expected cost within interval  $[0, x + d_i]$  is:

$$\Gamma_i(x) = C_i^p + M_i(x) = C_i^p + C_i^c \cdot \left(\frac{x}{\lambda_i}\right)^{\beta_i}. \quad (7)$$

By using the renewal theory, the long-term average cost of component  $i$  can be determined as the following:

$$\phi_i(x) = \frac{\Gamma_i(x)}{x + d_i} = \frac{C_i^p + C_i^c \cdot \left(\frac{x}{\lambda_i}\right)^{\beta_i}}{x + d_i}. \quad (8)$$

The optimal interval length for the preventive maintenances of the component  $i$ , denoted  $x_i^*$ , can be obtained when it leads to the minimum value of  $\phi_i(x)$ .

$$x_i^* = \arg \min_x \phi_i(x) \implies \left. \frac{d\phi_i(x)}{dx} \right|_{x=x_i^*} = 0 \iff \frac{C_i^c \cdot (\beta_i - 1) \cdot (x_i^*)^{\beta_i} + C_i^c \cdot \beta_i \cdot d_i \cdot (x_i^*)^{\beta_i - 1} - C_i^p \cdot \lambda_i^{\beta_i}}{(x_i^* + d_i)^2} = 0 \quad (9)$$

Let  $\widetilde{x}_i^*$  be the optimal interval length for the preventive maintenances of component  $i$  when the maintenance duration is negligible ( $d_i = 0$ ),  $\widetilde{x}_i^*$  can be calculated by (see Wildeman et al. [1997]):

$$\widetilde{x}_i^* = \lambda_i^{\beta_i} \sqrt{\frac{C_i^p}{C_i^c(\beta_i - 1)}} \quad (10)$$

Let  $g(x) = C_i^c \cdot (\beta_i - 1) \cdot x^{\beta_i} + C_i^c \cdot \beta_i \cdot d_i \cdot x^{\beta_i - 1} - C_i^p \cdot \lambda_i^{\beta_i}$ . It is easy to show that  $g'(x) > 0, \forall \beta_i > 1, \forall x > 0$ ;  $g(0) = -C_i^p \cdot \lambda_i^{\beta_i} < 0$  and  $g(\widetilde{x}_i^*) = C_i^c \cdot \beta_i \cdot d_i \cdot (\widetilde{x}_i^*)^{\beta_i - 1} > 0$ . Then the equation (9) has only one solution ( $x_i^*$ ) in the interval  $(0, \widetilde{x}_i^*)$ . Moreover, it is not difficult to show that  $\phi_i''(x_i^*) > 0$ . Therefore, we can deduce that  $x_i^*$  exists and it is unique,  $x_i^* \in (0, \widetilde{x}_i^*)$ .

The minimal long-term average maintenance cost of component  $i$  can be calculated as follows:

$$\phi_i^* = \phi_i(x_i^*) \quad (11)$$

The optimal interval length  $x_i^*$ , which can be numerically calculated using Equation (9), represents a nominal preventive maintenance frequency of component  $i$  (with  $i = 1, \dots, n$ ) and can be used to define tentative execution times which help to identify optimal maintenance groups.

### 3.2 Phase 2: Tentative planning

The aim is to establish all tentative maintenance dates, assuming that maintenance activities are separately performed, within the interval of time  $PH$ .

Based on the nominal preventive maintenance frequencies, the first tentative maintenance execution time of component  $i$  ( $i = 1, \dots, n$ ) from the current date denoted  $t_0$  can be calculated by

$$t_{i1} = t_0 - t_i^e + x_i^* + d_i^\Sigma, \quad (12)$$

with  $t_i^e$  is the operational time elapsed from the last preventive maintenance of component  $i$  before  $t_0$ , and  $d_i^\Sigma$  is the cumulative maintenance durations before the execution of component  $i$ .  $d_i^\Sigma$  is added since the system is stopped during maintenance.

It is shown in Wildeman et al. [1997], Do Van et al. [2011] that to ensure that all components are take into account in the maintenance decision, the planning horizon should be chosen so that within this horizon interval each component is preventively performed at least one time. Therefore, if  $PH$  is shorter than  $(t_{j1} + d_j - t_0)$  with  $t_{j1} = \max_{i=1:n} t_{i1}$ , it is then set equal to  $(t_{j1} + d_j - t_0)$ .

Since useful lifetime of components may be different, several components may be preventively performed more than one time. Let  $i^j$  be the  $j$ th occurrence of maintenance activity  $i$  (or component  $i$ ) in the scheduling horizon, the tentative execution time of operation  $i^j$ , denoted  $t_{ij}$  ( $j \geq 2$ ), depends on the executed time  $t_{i^{j-1}}^*$  of operation  $i^{j-1}$ , the cumulative maintenance durations  $d_{i^{j-1}}^\Sigma$  from  $t_{i^{j-1}}^*$ , and the nominal periodicity  $x_i^*$ .

$$t_{ij} = t_{i^{j-1}}^* + d_{i^{j-1}}^\Sigma + x_i^*. \quad (13)$$

### 3.3 Phase 3: Grouping optimization

In the grouping spirit, each maintenance group is defined by three parameters:

- maintenance operations;
- group executed date;
- group maintenance duration.

The main idea is to find a grouping structure (or partition of all maintenance operations within the scheduling horizon) which minimizes the expected maintenance cost and scopes with an availability level constraint by using the Genetic Algorithm (GA). The MULTIFIT algorithm will be also used to determine the minimum maintenance duration of each group under limited repairmen, see Section 4.

### 3.3.1 Mathematical formulation

*Penalty cost calculation* If the  $j$ th ( $j = 1, \dots$ ) occurrence of maintenance activity  $i$  is actually executed at time  $t_{ij}^* = t_{ij} + \Delta t_{ij}$  instead of the tentative execution time  $t_{ij}$ , then the resulting additional cost is expressed by the penalty cost  $h_i(\Delta t_{ij})$  which is written as, see Cho and Parlar [1991], Wildeman et al. [1997].

$$h_i(\Delta t_{ij}) = \Gamma_i(x_i^* + \Delta t_{ij}) - \left( \Gamma_i(x_i^*) + \Delta t_{ij} \cdot \phi_i^* \right),$$

with  $\Delta t_{ij} > -x_i^*$ . By using Equation (7), we obtain:

$$h_i(\Delta t_{ij}) = M_i(x_i^* + \Delta t_{ij}) - M_i(x_i^*) - \Delta t_{ij} \cdot \phi_i^*,$$

Another kind of penalty cost function can be found in Bouvard et al. [2011].

From equations (6) and (9), we obtain finally:

$$h_i(\Delta t_{ij}) = C_i^c \cdot \left[ \left( \frac{x_i^* + \Delta t_{ij}}{\lambda_i} \right)^{\beta_i} - \left( \frac{x_i^*}{\lambda_i} \right)^{\beta_i} \right] - \Delta t_{ij} \frac{C_i^p \beta_i}{x_i^* (\beta_i - 1)}. \quad (14)$$

*Maintenance cost savings calculation* Assume now that several different maintenance operations  $i^j$  ( $i, j = 1, 2, \dots$ ) are performed in a group  $G^k$  with  $k \in N$ . Remember here that operations  $i^j$  and  $i^l$  ( $j \neq l$ ) are identical operations since they are respectively the  $j$ th and the  $l$ th occurrence of maintenance activity  $i$ . Let  $H_{G^k}(t)$  be the group penalty cost function at time  $t$ . The optimal execution time of the group  $t_{G^k}$  can be found when the  $H_{G^k}(\cdot)$  searches its minimal value  $H_{G^k}^*$ . That is:

$$H_{G^k}^* = H_{G^k}(t_{G^k}) = \min_t \left( \sum_{i^j \in G^k} h_i(t - t_{ij}) \right) \quad (15)$$

As mentioned above, the execution of a group of maintenance operations requires only one set-up cost, the group  $G^k$  yields a cost reduction:

$$V_1(G^k) = (\text{card}(G^k) - 1) \cdot S. \quad (16)$$

An additional cost saving relying on the reduction of maintenance duration when maintenance activities are simultaneously executed by  $m$  repairmen.

$$V_2(G^k, m) = \left( \sum_{i^j \in G^k} d_i - d_{G^k}(m) \right) C^d, \quad (17)$$

where  $d_{G^k}(m)$  is the total duration of the group  $G^k$ . Note well that  $d_{G^k}(m)$  depends on the duration of all group members and the number of repairmen  $m$ .  $d_{G^k}(m)$  will be optimally minimized by using the MULTIFIT algorithm, see Section 4.1.

The group savings is written as:

$$Q_{G^k} = V_1(G^k) + V_2(G^k, m) - H_{G^k}^*. \quad (18)$$

Based on all different groups, a grouping structure, namely  $GS$ , can be identified. The total maintenance duration can be determined by:

$$D_{GS} = \sum_{G^k \in GS} d_{G^k}(m), \quad (19)$$

and the total cost savings can be calculated as follows:

$$Q_{GS} = \sum_{G^k \in GS} Q_{G^k}. \quad (20)$$

According to availability constraint presented in Section 4, An optimal grouping structure can be searched by the following:

$$GS^{op} = \{GS, \max\{Q_{GS}\} | D_{GS} \leq D_0\}. \quad (21)$$

Note well that the maximum number of groups to be considered is here  $2^n - 1$  for  $n$  maintenance activities. The finding of optimal grouping structure becomes difficult since the combinatorial problem can be formulated as a set partitioning problem, which however can be NP-hard. To solve this problem, the GA algorithm will be used and described in Section 4.

### 3.4 Phase 4: Update and decision

Due to the previous phase, we have an optimal grouping planning within the finite planning horizon  $PH$ . However, with time new information by which the current planning can be impacted may occur, such as changing of maintenance constraints (increasing/decreasing of repairmen) or maintenance opportunities (see Do Van et al. [2011]), ... which could lead the current planning to be no longer optimal or even unusable. A new planning is needed. To this end, we simply go back to phase 2.

## 4. OPTIMIZATION ALGORITHMS

### 4.1 MULTIFIT algorithm

Consider now a group consisting of  $k$  maintenance activities. To perform the maintenance of the group, only  $m$  ( $m \leq k$ ) repairmen are available. The objective is to find the minimum execution duration of this group. To this end, the MULTIFIT algorithm is used. A short description of this algorithm is the following:

- Step 0 Without loss generality,  $k$  maintenance activities are renumbered by increasing order according to their decreasing duration, i.e.  $d_1 \geq d_2 \geq \dots \geq d_k$ . Next, let  $dg_{up} = \max(d_1, 2 \sum_{i=1}^k d_i / m)$  and  $dg_{low} = \max(d_1, \sum_{i=1}^k d_i / m)$ . And finally, set  $w = 0$ .
- Step 1 Set  $dg = \frac{1}{2}(dg_{up} + dg_{low})$  and  $w = w + 1$ .
- Step 2 Assign successively maintenance activities to repairmen with time limited  $dg$ .
- Step 3 If all activities can be assigned to  $m$  repairmen, then set  $dg_{up} = dg$  and go to step 4, otherwise set  $dg_{low} = dg$  and go to step 4.
- Step 4 If  $w = w_{max}$  then stop, otherwise go to step 1.

It is shown that the worst-case performance ratio is:

$$\alpha = \frac{dg}{dg_{optimal}} = \frac{13}{11} + 2^{-w_{max}}$$

The iterations number can be fixed in advance, as recommended in Coffman et al. [1978],  $w_{max} = 7$ .

### 4.2 Genetic algorithm

The Genetic Algorithm has been recognized as a general search strategy which is often useful for solving combi-

natorial problems Holland [1975]. Recently, GA has been widely studied, experimented and applied in many fields in engineering worlds. A GA starts by creating an initial population of solutions. Each solution is next evaluated using an objective function. During each iteration step, genetic operations (crossover and mutation operations) are applied in order to search potential better solutions. The GA structure is in presented in Fig. 1.

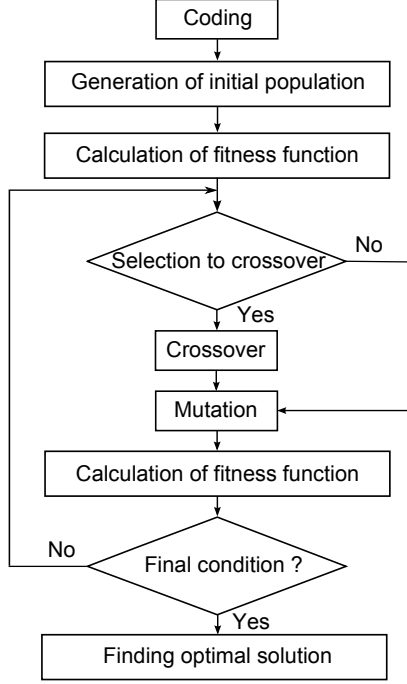


Fig. 1. Structure of GA

*Coding* A coding step determines how the problem is structured in the algorithm. In our problem, each solution is represented by an matrix, namely  $X$ , with  $ngmax$  rows and  $n$  columns. Where,  $ngmax$  is the maximum number of groups in a feasible solution.  $ngmax$  depends on the requested availability level  $A_0$ , or the allowed total maintenance duration  $D_0$ . If  $N$  ( $N \geq n$ ) preventive maintenance activities with durations  $d_1 \geq d_2 \geq \dots \geq d_N$  are considered in the scheduling interval, the maximum number of groups can be then determined as the following:

$$ngmax = \{l + 1 | D_0 - d_1 \geq \sum_{i=1}^l d_{N-i+1}\}. \quad (22)$$

If maintenance activity  $j$  is in group  $i$ ,  $X_{ij} = 1$  otherwise  $X_{ij} = 0$ . Here,  $X$  corresponds to a grouping structure.

Example: considering 7 preventive activities with  $d_1 = 5, d_2 = 1, d_3 = 1, d_4 = 1, d_5 = 1, d_6 = 1, d_7 = 1$ . If the total maintenance duration allowed is 8, by applying (22), the maximum number of groups is then 4. A feasible solution containing 4 groups  $G_1 = \{1, 2, 7\}, G_2 = \{3\}, G_3 = 4, G_4 = \{5, 6\}$  is represented by matrix  $X$  as follows:

$$X = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

*Generation of initial population* GA makes randomly an initial population of solutions. If there are too few solutions in the population, GA have a few possibilities to perform crossover and only a small part of search space is explored. On the other hand, if there are too many solutions, GA slows down. The population size is usually chosen between 60 and 100 solutions. To generate an initial solution, the number of groups in the solution is randomly chosen between 1 and  $ngmax$ . Next, all maintenance activities are randomly put into the chosen groups.

*Calculation of fitness function* Each solution is evaluated by its fitness function which is defined as the following:

$$f(X) = \begin{cases} Q_X & \text{if } D_X \leq D_0 \\ -\infty & \text{otherwise} \end{cases} \quad (23)$$

*Selection to crossover* The goal is to select pairs of parent solutions to Crossover phase. It is divided into 2 steps:

- *Solution selection*: the selection process is based on fitness. Array of parent solutions are sorted according to their fitness values in ascending order. The population is categorized into  $s$  groups. Next, a parent solution is randomly chosen from these  $s$  groups with  $s$  groups probabilities  $p_1 \leq p_2 \leq \dots \leq p_s$  ( $\sum_{i=1}^s p_i = 1$ ) respectively. Thus, the fitter a solution is, the more chance it have to be chosen for a parent in the next generation. This combinatorial grouping was found most effective;
- *Selection to Crossover*: each pair of selected solutions is now randomly selected for Crossover step with Crossover probability that is introduced to leave some part of population survive to next generation. Crossover probability is usually chosen between [70% ÷ 90%]. If a pair of selected solutions is not selected for Crossover phase, it is moved directly to Mutation step.

*Crossover* The objective is to combine selected solutions to generate next better generation solutions for Mutation step by preserving their characteristics. Single point crossover is used in this algorithm which randomly chooses a locus and exchanges the subsequences before and after that locus between two selected solutions to create two children. For example, in Fig. 2, given two parent solutions we choose randomly maintenance activity 4 as the crossover point, i.e. solution 1= {Ia,Ib}, solution 2={IIa,IIb}. As consequence, two new solutions are {Ia,IIb} and {IIa,Ib}.

*Mutation* Mutation is made to prevent falling GA into local extreme, but it should not occur very often, because then GA will in fact change to random search. So, we usually choose Mutation probability in a interval [0.2% ÷ 1.5%].

For each selected solution, a random maintenance activity in a group is next moved to another group in order to generate a new solution. An illustration of the mutation process is shown in Fig. 3.

*Final condition* Final condition is introduced to stop the algorithm process. Herein, limited generations number is

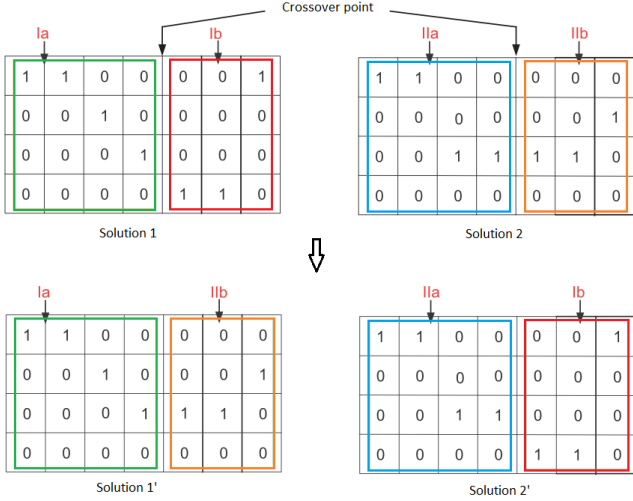


Fig. 2. Illustration of Crossover step

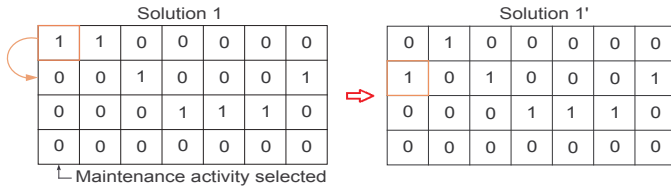


Fig. 3. Illustration of Mutation step

used as a criterion, according to specific problems, other kinds of final conditions can be used, see Rahman and Ahmed [2009].

*Finding optimal solution* the optimal solution with highest fitness value is searched among the last generation solutions.

## 5. NUMERICAL EXAMPLE

The purpose of this section is to show how the proposed grouping strategy can be used in preventive maintenance optimization with availability constraint under limited repairmen through a simple of 20 series components system in which each component is preventively replaced. When a component fails, it is immediately maintained according to a minimal-repair policy. A failure repair restores the component involved into a state as good as before. We assume failure rate of a component  $i$  ( $i = 1, \dots, 20$ ) is described by a Weibull distribution with scale parameter  $\lambda_i > 0$ , and shape parameter  $\beta_i > 1$ . Table 1 reports the random data for five components. For set-up cost and unavailability cost rate, we take  $S = 10$  and  $C^d = 5$  respectively.

By assuming that all components are individually maintained, nominal maintenance periodicity  $x_i^*$ , minimum average maintenance cost  $\phi_i^*$  and the next preventive replacement date  $t_{i1}$  (with  $i = 1, \dots, 20$ ) are calculated by substitution of the input data in Equations (9) and (12). The results are given in Table 2.

Table 1. Data of 20 components system.

Unit	$\lambda_i$	$\beta_i$	$c_i^p$	$C_i^c$	$d_i$	$t_i^e$
1	288	1.9334	168	42	1	638.98
2	281	1.9797	159	47	2	508.91
3	276	1.9965	155	47	6	486.87
4	264	1.9930	111	36	2	405.53
5	288	1.9244	142	38	3	538.48
6	270	1.9209	108	59	4	252.64
7	265	1.9417	127	46	3	313.84
8	278	1.9271	180	31	5	580.86
9	294	1.9992	180	50	4	378.94
10	265	1.9098	159	58	3	259.66
11	265	1.9223	106	57	2	161.57
12	286	1.9731	123	51	5	239.35
13	268	1.9246	123	38	5	275.31
14	291	1.9777	117	41	1	206.62
15	252	1.9480	103	57	5	53.79
16	291	1.9196	146	44	2	248.10
17	256	1.9585	158	53	5	129.09
18	277	1.9358	134	58	5	56.08
19	267	1.9859	171	58	4	15.83
20	266	1.9509	187	31	6	224.36

Table 2. Values of  $x_i^*$ ,  $\phi_i^*$ , and  $t_{i1}$ .

Unit	$x_i^*$	$\phi_i^*$	$t_{i1}$	Unit	$x_i^*$	$\phi_i^*$	$t_{i1}$
1	638.98	0.5932	0	11	417.57	0.6289	289
2	557.91	0.6483	50	12	514.35	0.6229	310
3	563.87	0.6929	80	13	585.31	0.5619	350
4	506.53	0.5191	110	14	531.62	0.5023	370
5	647.48	0.5369	120	15	407.79	0.6954	400
6	438.64	0.6562	200	16	607.10	0.5708	410
7	505.84	0.6196	210	17	506.09	0.7792	430
8	789.86	0.5658	230	18	498.08	0.7019	500
9	602.94	0.6969	250	19	502.83	0.8052	550
10	509.66	0.7578	280	20	757.36	0.6149	600

### 5.1 Maintenance planning without availability constraint

Consider firstly a grouping maintenance planning without availability constraint using the rolling horizon approach. Only one repairman is considered for all maintenance activities. To define the scheduling horizon, the current date is set equal to 0 ( $t_{begin} = 0$ ) and the ending horizon date corresponds to  $t_{201} + d_{20} = 606$  ( $t_{end} = 606$ ), i.e.  $PH = 606$ . Herein, 20 individual preventive activities are therefore considered in this interval of time. To find the optimal grouping planning, the dynamic programming presented in Do Van et al. [2011] was used and the results are reported in Table 3.

Table 3. Grouping without constraint.

Group units	Optimal date $t_G$	duration $d_G$
{1,...,5}	67.16	14
{6,...,12}	242.96	26
{13,...,17}	389.25	18
{18,...,20}	538.56	15

The total maintenance cost savings is  $Q_{\Sigma}^* = 148.64$ . And the total maintenance durations is  $D = 14 + 26 + 18 + 15 = 73$ , the average availability of the system is hence  $A = (PH - D)/PH = 0.8795$ .

## 5.2 Maintenance planning with availability constraint

We are interested now to construct a grouping maintenance planning with a given availability level under limited repairmen. To find optimal groups, the dynamic programming approach is no longer applicable and the GA algorithm presented in Section 4 is used. GA simulation was generated with 1500 generations from 60 initial solutions. The MULTIFIT algorithm is also used to find the minimum duration of each maintenance group with limited repairmen. Different values of the availability level constraint  $A_0$  and the number of repairmen  $m$  are considered for the scheduling interval  $[0, 606]$ . The results presented in Table 4 show that:

- to establish a maintenance planning under a given availability level constraint, the number of necessary repairmen is bounded. The lower bound corresponds to the minimum number of repairmen with that a maintenance planning can be constructed. The upper bound corresponds to the minimum number of repairmen that leads to a maintenance planning with highest cost savings. For example, when  $A_0 = 0.9719$  (the system availability is required to be not lower than 0.9719), the lower and upper bound are 5 and 7 respectively. If the number of repairmen is lower than 5 ( $m < 5$ ), no maintenance planning can be then established. Maintenance planning and cost savings remain unchanged when  $m \geq 7$ . Note well that the upper bound of repairmen is lower than the maximum number of maintenance activities in a group, see for example, when  $A_0 = 0.9901$  we need only 13 repairmen for executing a group of 20 maintenance activities with a minimum execution duration;
- when the number of repairmen is in the bounded interval, an increasing of repairmen leads to a maintenance planning with an increasing of maintenance cost savings.

The results show also that when only single repairman is considered for preventive actions and the availability level constraint is low enough, the GA algorithm and the dynamic programming lead to the same grouping maintenance planning, see for example the case with  $A_0 = 0.8795$  and  $m = 1$ .

## 6. CONCLUSIONS

In this work the rolling horizon approach introduced recently is developed by integrating both GA and MULTIFIT algorithm in order to establish an optimal maintenance planning of multi-component systems under an availability level constraint with limited repairmen. The proposed approach can also help (i) to determine the minimum number of repairmen to ensure that a establishable maintenance planning, which scopes with a given availability level constraint, can be constructed; (ii) to determine the minimum number of repairmen that leads to an optimal maintenance planning satisfying the availability constraint with highest cost savings. Moreover, thanks to the rolling horizon spirit, the grouping maintenance planning can be easily updated when short-term information is available, e.g an increasing/decreasing of repairmen or opportunities.

Table 4. Grouping planning with constraints.

$A_0$	$m$	$Q_{GSup}$	Optimal solution	$t_G$	$d_G$
0.8795 ( $D_0 = 73$ )	$m = 1$	148.64	$G1 = \{1, \dots, 5\}$	67.16	14
			$G2 = \{6, \dots, 12\}$	242.96	26
			$G3 = \{13, \dots, 17\}$	389.25	18
			$G4 = \{18, \dots, 20\}$	538.56	15
	$m = 2$	323.76	$G1 = \{1, \dots, 5\}$	67.16	7
			$G2 = \{6, \dots, 12\}$	235.96	13
			$G3 = \{13, \dots, 16\}$	359.69	7
	$m = 3$	374.29	$G1 = \{1, \dots, 5\}$	67.16	6
			$G2 = \{6, \dots, 12\}$	234.96	9
	$m \geq 7$	421.10	$G1 = \{1, \dots, 5\}$	67.16	6
$G2 = \{6, \dots, 12\}$			234.96	5	
$G3 = \{13, \dots, 20\}$			411.45	6	
$G1 = \{1, \dots, 8\}$			116.33	7	
0.9719 ( $D_0 = 17$ )	$m = 5$	413.95	$G1 = \{1, \dots, 7\}$	108.15	6
			$G2 = \{8, \dots, 14\}$	269.69	5
			$G3 = \{15, \dots, 20\}$	431.05	6
	$m \geq 7$	421.10	$G1 = \{1, \dots, 5\}$	67.16	6
			$G2 = \{6, \dots, 12\}$	234.96	5
			$G3 = \{13, \dots, 20\}$	411.45	6
0.9835 ( $D_0 = 10$ )	$m < 8$	No solution			
	$m = 8$	305.63	$G1 = \{1\}$	0	1
	$m = 9$	373.56	$G2 = \{2, \dots, 20\}$	276.39	9
	$m \geq 10$		$G1 = \{1, 2, 4, \dots, 7, 9\}$	131.11	4
0.9901 ( $D_0 = 6$ )	$m < 13$	No solution			
	$m \geq 13$	304.28	$G1 = \{1, \dots, 20\}$	265.03	6

Our future research work will focus on the development of the proposed approach for systems with inter-component dependencies. Furthermore, condition based maintenance will be also developed in this grouping maintenance framework.

## ACKNOWLEDGMENTS

This work has been supported and partly financed by the European project MoDe - Maintenance on Demand (<http://fp7-mode.eu>) funded from the European Community's 7th Framework Program (Project SCP8-GA-2009-233890 - FP7 Sustainable Surface Transport).

## REFERENCES

- A. Bouvard, S. Artus, C. Berenguer, and V. Cocquemot. Condition-based dynamic maintenance operations



- planning and grouping. application to comercial heavy vehicles. *Reliability Engineering and System Safety*, 2011.
- D. Cho and M. Parlar. A survey of maintenance models fo rmulti-unit systems. *European Journal of Operational Research*, 51, 1991.
- Jr. E.G. Coffman, M.R. Garey, and D.S. Johnson. An application of bin-packing to multiprocessor scheduling. *SIAM Journal on Computing*, 7:1, 1978.
- P. Do Van, K. Bouvard, F. Brissaud, A. Barros, and C. Berenguer. Dynamic grouping maintenance strategy with time limited opportunities. In *Avance in Safety, Reliability and Risk Management - Proc.ESREL 2011, 18-22 September 2011, Troyes, France.*, 2011.
- G. Galante and G. Passannanti. An exact algorithm for preventive maintenance planning of series-parallel systems. *Reliability Engineering and System Safety*, 94 (10):1517 – 1525, 2009.
- J.H. Holland. *Adaptation in natural and artificial systems*. The University of Michigan Press, Ann Arbor, 1975.
- R.P. Nicolai and R. Dekker. A review of multi-component maintenance models. In Terje Aven & Jan Erik Vinne, editor, *Risk, Reliability and Societal Safety - Proc.ESREL 2007, 25-27 June 2007, Stavanger, Norway*, pages 289–296. Taylor & Francis, 2007.
- L. Radouane, C. Alaa, and A. Djamil. Opportunistic policy for optimal preventive maintenance of a multi-component system in continuous operating units. *Computers & Chemical Engineering*, 9(9):1499–1510, 2009.
- K.M. Rahman and S.I. Ahmed. Performance analysis of genetic algorithm for solving the multiple-choice multi-dimensional knapsack problem. *International Journal of Recent Trends in Engineering*, 2(2):103–105, 2009.
- R.E Wildeman, R. Dekker, and A.C.J.M. Smit. A dynamic policy for grouping maintenance activities. *European Journal of Operational Research*, 99, 1997.