European Journal of Operational Research 197 (2009) 1166–1177

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Exact and heuristic methods for the selective maintenance problem

T. Lust [a], O. Roux [b,*], F. Riane [b]

[a] Faculté Polytechnique de Mons, Laboratory of Mathematics and Operational Research, 9, rue de Houdain, 7000 Mons, Belgium
[b] Facultés Universitaires Catholiques de Mons, Centre of Research and Studies in Industrial Management, Catholic University of Mons, 151, chaussée de Binche, 7000 Mons, Belgium

ARTICLE INFO

Article history:
Available online 10 April 2008

Keywords:
Selective maintenance
Reliability
Combinatorial optimization
Branch and bound
Tabu search

ABSTRACT

We present in this paper, new resolution methods for the selective maintenance problem. This problem consists in finding the best choice of maintenance actions to be performed on a multicomponent system, so as to maximize the system reliability, within a time window of a limited duration. When the number of components of the system is important, this combinatorial problem is not easy to solve, in particular because of the nonlinear objective function modeling the system reliability. This problem did not receive much attention yet. Consequently, rare are the effective resolution methods that are offered to the user. We thus developed heuristics and an exact method based on a branch and bound procedure, which we apply to various system configurations. We compare the obtained results, and we evaluate the best method to be used in various situations.

1. Introduction

In this paper, we are interested in preventive maintenance [1–4], and more particularly in the selective maintenance problem that consists in finding the best choice of maintenance actions to be performed on a multicomponent system, within a fixed time window. This limited time does not allow to carry out all maintenance actions. The idea is then to pick out a subset of actions to undertake whose total execution duration fits in the time window and that yields maximum reliability when the system is restarted after the maintenance period.

This kind of problem can be encountered for equipment that performs sequences of missions and can be repaired only between missions. This is the case for military equipment, production equipment on which maintenance actions are carried out the weekend, vehicles maintained between two deliveries, etc. This problem has been introduced by Rice et al. [5], where they consider systems presenting a particular architecture, constant components failure probabilities and only one type of maintenance actions (to repair a component).

Cassady et al. [6] extended the model of this problem, by considering components failure probabilities dependent on their age and multiple maintenance actions. They solve this problem for systems of specific configuration by a simple enumeration of all possible solutions. This enumeration gives the solution for only small size systems.

We thus propose new resolution approaches: a construction heuristic, which makes it possible to find a good solution very quickly, a tabu search based metaheuristic, which allows to improve the quality of the solution obtained by the construction heuristic (but always without guarantee of optimality) and an exact method, based on a branch and bound procedure for benchmarking purposes.

This paper is organized in the following way: we initially present the selective maintenance problem taken into account and its modeling. We then describe the new resolution methods developed and the numerical results of these methods on various systems.

2. Selective maintenance problem's statement

We define in this section the type of system that we propose to study, the various types of maintenance actions considered, and their effects on the system. We also present the calculation of the reliability of a system in series and/or parallel.

We consider that maintenance actions are carried out on a system, having to accomplish a given mission, and defined by a set of components connected to each other in series and/or parallel. An example of such a system is presented in Fig. 1. In this representation, the blocks correspond to the components. The failure of one of the components 3, 4, 5 or 10 placed in series involves the system failure. Components 6, 7, 8 or 9 being placed in parallel, the system fails only if the two parts of the parallel subsystem are not functioning. This arrives when the failure of one of the components 6 or 7 is in combination with the failure of one of the components 8 or 9.

* Corresponding author. Fax: +32 65323363.
E-mail address: roux@fucam.ac.be (O. Roux).

0377-2217/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2008.03.047

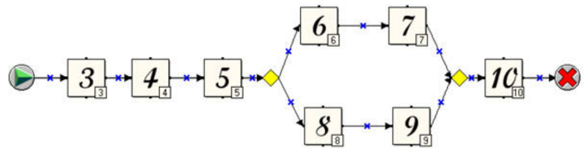

Fig. 1. Example of multicomponent system.

The system must perform a sequence of missions with breaks of known length between each mission, when the maintenance actions can be accomplished. During each mission, components can fail. Consequently, at the end of a mission, the components are either in operating condition or failed. As Cassady et al. [6], we consider two possible maintenance actions during the break:

- to replace (by new) a failed or a functioning component (the component age after the action is thus assumed to be to 0),
- to minimally repair a failed component, what restarts the component (the component age after the action is thus unchanged).

It should be noted that the system can fail before a programmed maintenance break. In this case, only minimal repairs on the components that caused the breakdown of the system will be carried out, so as to put back the system in operating condition. If many breakdowns of the system occur before the maintenance period, it should be worth revising the periodicity of the maintenance activities and the system design [7].

Once a maintenance break is started, the problem is to select a subset of actions to be performed at the fixed period of the break in order to maximize the system reliability while ensuring the respect of the limited duration of the maintenance break. It is the problem of reliability maximization under time constraint.

Given that the cost of the maintenance actions is considered negligible compared to the cost of a stop of the system during a mission, we do not take into account the costs of the maintenance actions in the modeling of the problem. We thus seek the maintenance actions that make it possible to maximize the system reliability, without constraint of budget.

We consider that the component failure probability for a given mission is dependent on its age, i.e. the higher the component age is, the higher its failure probability is. This probability is modeled through a reliability law [8,9], the most used probability law in maintenance is the Weibull distribution. Thanks to the components failure probabilities, we can determine the functioning probability of the system for a mission of a given duration, which depends on the functioning probabilities of the components and on the system architecture. Contrary to preceding modelings of the selective maintenance problem [5,6], we consider in this paper general systems in a series and/or parallel architecture.

We present below the way to compute the functioning probability of a component thanks to its reliability law of the Weibull type, and the functioning probability of a system according to the functioning probabilities of the components that compose it.

2.1. Functioning probability of a component

The reliability law gives the probability that one component achieves without failure a mission of length t. In the case of the Weibull distribution, the probability $R(t)$ is given by the following relation:

$$R(t) = e^{-\left(\frac{t}{\eta}\right)^{\beta}},$$

where β and η represent, respectively, the shape and scale parameters of the Weibull distribution. They are real numbers greater than zero.

If the component already carried out a mission of length T and is in a operating condition at the end of this mission, we use conditional reliability to determine the probability that the component successfully achieves a new mission of length t, defined by the following relation:

$$R(T,t) = \frac{R(T+t)}{R(T)} = \frac{e^{-\left(\frac{T+t}{\eta}\right)^{\beta}}}{e^{-\left(\frac{T}{\eta}\right)^{\beta}}}.$$

We have represented in Fig. 2 the comparison of the reliability law of a new component with that of a component 30 units of time aged, always with the Weibull law ($\beta = 4, \eta = 150$). We can notice that the functioning probabilities of the old compound are lower than those of the new component.

Hence, starting from the component age and the mission duration, it is easy, in the case of a reliability law of the Weibull type, to determine the probability that the component achieves the mission.

2.2. Functioning probability of a multicomponent system

We can consider a multicomponent system as a set of subsystems entirely in series or parallel. The decomposition of the system shown in Fig. 1 is given in Fig. 3. We have subsystem 1 in series composed of components 6 and 7 (SS1), subsystem 2 in series composed of components 8 and 9 (SS2) and subsystem 3 in parallel composed of subsystems 1 and 2 (SS3). The global multicomponent system (SG) in series is thus composed of components 3, 4, 5, 10 and subsystem 3. By using this decomposition, it is only necessary to determine the functioning probabilities of systems entirely in series, or functioning probabilities of systems entirely in parallel.

2.2.1. System in series

In order that a system in series functions after a mission of duration t, it is necessary that all the components i of the system function. The probability $RS(t)$ that the system functions after a mission of duration t is thus equal to the product of the functioning probabilities of the n components that compose it:

$$RS(t) = \prod_{i}^{n} R_i(t).$$

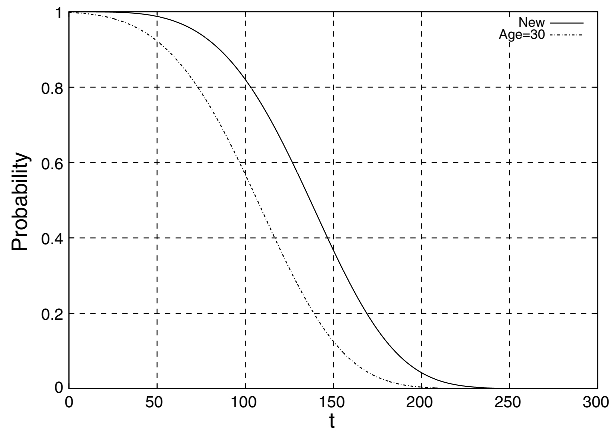

Fig. 2. Comparison of the reliability law of the Weibull type ($\beta = 4, \eta = 150$) for a new component and a component 30 u.t. aged.

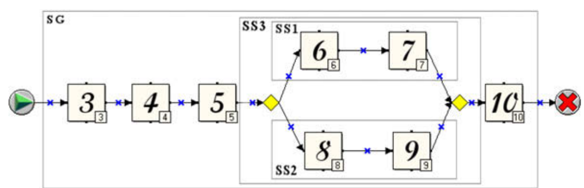

Fig. 3. Example of decomposed multicomponent system.

2.2.2. System in parallel

So that a system in parallel functions after a mission of duration t, it is necessary that at least one component of the system functions. As the probability that a component i fails is equal to $(1 - R_i(t))$, the probability $RP(t)$ that the system functions after a mission of duration t is equal to the complement with 1 of the product of the failure probabilities of the n components that compose it:

$$RP(t) = 1 - \prod_{i}^{n}(1 - R_i(t)).$$

3. Problem modeling

The modeling of the selective maintenance problem has already been approached by Cassady et al. [6]. We again partly take this modeling by extending it to any systems in series and/or parallel.

We indicate by, respectively, tmr_i, tr_i and trf_i the times necessary to carry out a minimal repair on a failed component i, to replace a failed component i and to replace a functioning component i. We suppose that $tmr_i < tr_i$ and $trf_i \leqslant tr_i$, i.e. times to replace a failed component are higher or equal to times of the other actions.

The system is composed of n components, functioning or failed at the end of mission k, depending on the state of the binary variable $Y_i(k)$:

$$Y_i(k) = \begin{cases} 1 & \text{if the component } i \text{ is functioning at the end of mission } k, \\ 0 & \text{otherwise.} \end{cases}$$

The minimal repair action on a failed component i at the end of mission k is symbolized by the binary variable $W_i(k)$, so that

$$W_i(k) = \begin{cases} 1 & \text{if minimal repair is performed between missions } k \text{ and } k+1, \\ 0 & \text{otherwise.} \end{cases}$$

In the same way, the action to replace a component i at the end of mission k is defined by the binary variable $V_i(k)$:

$$V_i(k) = \begin{cases} 1 & \text{if replacement is performed between missions } k \text{ and } k+1, \\ 0 & \text{otherwise.} \end{cases}$$

After the maintenance actions and before the mission $k+1$, the new state of a component is symbolized by the binary variable $X_i(k+1)$, so that

$$X_i(k+1) = \begin{cases} 1 & \text{if the component } i \text{ is functioning before the mission } k+1, \\ 0 & \text{otherwise.} \end{cases}$$

Total time $TMR(k)$ necessary to minimal repairs is given by

$$TMR(k) = \sum_{i=1}^{n} tmr_i \cdot W_i(k).$$

Total time $TR(k)$ devoted to replacements of functioning components is expressed by

$$TRF(k) = \sum_{i=1}^{n} trf_i \cdot V_i(k) \cdot Y_i(k).$$

Total time $TR(k)$ used to replacements of failed components is given by

$$TR(k) = \sum_{i=1}^{n} tr_i \cdot V_i(k) \cdot (1 - Y_i(k)).$$

We can thus express the total time $T(k)$ necessary to the realization of the maintenance actions $W_i(k)$ and $V_i(k)$ at the end of a mission k by

$$T(k) = TMR(k) + TRF(k) + TR(k).$$

We also consider that the length of mission $k+1$ is equal to $L(k+1)$, the maintenance time after the mission k is equal to $T_0(k)$, $B_i(k)$ and $A_i(k+1)$ are, respectively, the ages of the components after the mission k and before the mission $k+1$. Each component i follows a reliability law of Weibull type of shape parameter β_i and scale parameter η_i. The system reliability $R_S(k+1)$ before the mission $k+1$ is given by the function F (dependent on the system architecture), which receives in argument the vector resulting from the product between the vector R containing reliabilities of the components and the vector containing the state of the components at the beginning of the mission $k+1$. The probabilities $R_i(k+1)$ are given for each component i by the following relation:

$$R_i(k+1) = \frac{e^{-\left(\frac{L(k+1)+A_i(k+1)}{\eta_i}\right)^{\beta_i}}}{e^{-\left(\frac{A_i(k+1)}{\eta_i}\right)^{\beta_i}}}.$$

The model obtained for the selective maintenance problem, which consists in determining which components to replace (decisions represented by $V_i(k)$ variables) and which components to minimally repair (decisions represented by $W_i(k)$ variables) at the end of mission k, is given below. We can notice that we obtain a nonlinear combinatorial optimization problem, which can be considered as a non-separable, nonlinear knapsack problem [10]

$$\left[\begin{array}{ll} \max & R_S(k+1) = F(R(k+1) \cdot X(k+1)) \\ \text{s.t} & T(k) \leqslant T_0(k) \\ & W_i(k) + V_i(k) \leqslant 1 \forall i \\ & W_i(k) + Y_i(k) \leqslant 1 \forall i \\ & W_i(k), V_i(k) \in \{0,1\} \forall i \\ \text{with} & A_i(k+1) = B_i(k) - B_i(k) \cdot V_i(k) \forall i \\ & X_i(k+1) = Y_i(k) + W_i(k) + V_i(k) \cdot (1 - Y_i(k)) \forall i \end{array}\right].$$

The problem presents four sets of constraints: the first indicates that the execution time of the maintenance actions is limited, the second ensures that only one of the two maintenance actions can be carried out, the third stipulates that the W_i action is performed only if the component i is failed ($Y_i = 0$) and the fourth represents the $(0,1)$ constraint. The two last equalities make it possible to determine the components age and state after the maintenance break, necessary to the determination of the system reliability.

Remark. We are only interested by optimizing the reliability at the end of a given mission (single-mission problem). We are not interested in the problem, which is much more complex that consists in optimizing the maintenance actions so as to optimize reliability on a great number of missions [11].

4. New resolution methods

Three new methods were developed to solve the selective maintenance problem modeled above a construction heuristic, a heuristic based on the adaptation of the tabu search and an exact method based on a branch and bound procedure. We present these three methods hereafter.

4.1. Construction heuristic

The goal of the heuristic is to quickly provide a solution of good quality. The general functioning of the heuristic is as follows. Initially, if the system is failed after mission k, the method generates a starting solution, necessary to the application of the heuristic itself. The starting solution is obtained by applying minimal repairs on the failed components until the system is able to function, by considering first of all the most critical components, i.e. those located in less subsystems. We realize then in an iterative way the maintenance action that maximizes the ratio defined by the reliability of the system after action minus the reliability of the system before action, the whole divided by the time of the action, until no more maintenance actions are realizable.

So if we consider R_S as the system reliability, S as the current subset of selected maintenance actions, M as a maintenance action V or W performed on a component i, and T as the duration associated to M, an iteration consists of finding M that maximizes

$$\frac{R_S(S \cup \{M\}) - R_S(S)}{T(M)}.$$

Once an action is carried out, we actualize the system reliability, which grows at each iteration of the heuristic.

The main algorithm of the construction heuristic is given in Procedure 1, which includes the procedures **InitialSolutionGeneration** (Procedure 2), **SelectionAction** (Procedure 3) and **RealizationAction** (Procedure 4). In the procedures, the symbols $\downarrow, \uparrow$ and $\updownarrow$ specify, respectively, the transmission modes IN, OUT and IN OUT of a parameter to a procedure. The symbol --| marks the beginning of a comment line.

The entrance parameters of the main procedure are the number n of components, the duration T_0 of the maintenance time after the mission k (in the following description, the index k is omitted in a simplification purpose), the length L of the mission $k+1$, the β and η parameters of the Weibull laws of the components, the maintenance durations tmr, tr, trf of the different actions and the states Y and ages B of the components. The construction heuristic returns the system reliability R_S, the maintenance actions V and W performed and the states X and ages A of the components before the next mission. All these parameters are considered as global variables.

In the main procedure, X and Y as well as the total duration T of the maintenance actions carried out are initially initialized. Then, the components that constitute the system are divided in three sets:

- the set *SFU*: functioning components with no maintenance actions already performed,
- the set *SFA*: failed components with no maintenance actions already performed,
- the set *SW*: set of components having undergone a minimal repair.

After this, the initial reliability of the system R_S is computed by calculation of the vector R, which contains the reliability of each component. If R_S is equal to 0, the procedure **InitialSolutionGeneration** is executed.

The aim of this procedure is to generate a first solution that allows to obtain a system reliability different from zero. The generation of a starting solution, if the system is failed, is justified by the fact that in this particular case, nothing says that at least one action will be able to increase the system reliability. Indeed, if the breakdown of the system is due to more than one component, the choice of the action to be carried out would be done randomly (which would considerably reduce the performances of the heuristic), given that the maintenance actions are selected according to the reliability of the system after action minus the reliability of the system before action.

The **Classification** procedure, not described in this paper, simply classifies the components by decreasing order of criticity. The criticity depends on the number of systems entirely in series or parallel in which the component is located. For example, for the system shown in Fig. 3, the criticity of components 3, 4, 5 and 10 is larger than that of the other components. Once that the components are classified, a minimal repair is carried out on the most critical component of *SFA* until the reliability of the system becomes different from zero. After the execution of the minimal repair, the set *SW* is updated by addition of the component having undergone the minimal repair, and this component is withdrawn from the set *SFA*.

Then, the maintenance actions to be carried out are picked out thanks to the procedure **SelectionAction** (Procedure 3), which determines the maintenance action that maximizes the ratio defined by the reliability profit generated by the action divided by the time of the action.

Procedure: 1. Construction heuristic

Parameters ↓: n, T_0, L (integers); $\beta, \eta, tmr, tr, trf, Y, B$ (vectors)
Parameters ↑: R_S (real); W, V, X, A (vectors)
$X \leftarrow Y$
$A \leftarrow B$
--| Initialization of the total duration T of the maintenance actions performed
$T \leftarrow 0$
--| Initialization of *SFU* (set of functioning components with no maintenance actions already performed), *SFA* (set of failed components with no maintenance actions already performed) and *SW* (set of components having undergone a minimal repair)
$SFU \leftarrow \{\}$
$SFA \leftarrow \{\}$
for each component i **do**
  if $Y(i) = 1$ **then**
    $SFU \leftarrow SFU + \{i\}$
  else
    $SFA \leftarrow SFA + \{i\}$
$SW \leftarrow \{\}$
--| Calculation of the initial reliability R_S of the system
for each component i **do**
  $R(i) = \dfrac{e^{-\left(\frac{L+A(i)}{\eta(i)}\right)^{\beta(i)}}}{e^{-\left(\frac{A(i)}{\eta(i)}\right)^{\beta(i)}}}$
$R_S = F(R \cdot X)$

--| Initial solution generation
if $R_S = 0$ **then**
  InitialSolutionGeneration $(R \downarrow, T \updownarrow, SW \updownarrow, SFA \updownarrow)$
--| Main loop
repeat
  $RatioMax \leftarrow -1$
  --| Determination of the maintenance action of maximal ratio
  SelectionAction $(T \downarrow, R \downarrow, SFU \downarrow, SFA \downarrow, SW \downarrow, IndexMax \uparrow,$
  $ActionMax \uparrow, RatioMax \updownarrow,)$
  if $(RatioMax \neq -1)$ **then**
    --| Realization of the maintenance action of maximal ratio
    RealizationAction $(IndexMax \downarrow, ActionMax \downarrow, RatioMax \downarrow, T \updownarrow,$
    $R \updownarrow, SFU \updownarrow, SFA \updownarrow, SW \updownarrow)$
until $(T = T_0)$ or $(RatioMax = -1)$
$R_S = F(R \cdot X)$

For the components belonging to the set *SFU*, the ratio is equal to the reliability profit generated by a replacement (action V, the reliability of the component in question thus becomes equal to the reliability of a new component) of one of the components of *SFU*, divided by the time of the action V (trf) of the component.

For the components belonging to the set *SFA*, given that the components of *SFA* are failed components, two ratios are computed: the reliability profit generated by a replacement (action V) divided by the time of the action $V(tr)$ of the component and the reliability profit generated by a minimal repair (action W) divided by the time of the action $W(tmr)$ of the component.

Procedure: 2. InitialSolutionGeneration

Parameters ↓: R (vector)
Parameters ↕: T (integer); SW, SFA (sets)
--| Classification of the components of SFA by decreasing order of criticity
Classification (SFA ↕)
--| Realization of minimal repairs in order of criticity
repeat
  if $T + tmr(SFA(0)) \leqslant T_0$ **then**
    $T \leftarrow T + tmr(SFA(0))$
    $W(SFA(0)) \leftarrow 1$
    $X(SFA(0)) \leftarrow 1$
    $SW \leftarrow SW \cup \{SFA(0)\}$
    $SFA \leftarrow SFA \setminus \{SFA(0)\}$
    $R_S = F(R \cdot X)$
  else
    $SFA \leftarrow SFA \setminus \{SFA(0)\}$
until $(R_S \neq 0)$ or $(SFA = \{\})$

In this heuristic method, we consider the possibility to revise choices previously made. It can not be then considered as a greedy method. Indeed, for the failed components of the set *SFA*, there are two choices: to carry out an action V or an action W. If the choice goes to an action W, we always give the opportunity to call into question this decision, and to carry out an action V in the place of the action W. This is carried out via the set *SW*, which contains the failed components that have already undergone a minimal repair.

We thus evaluate the ratio given by this possibility in the following way: reliability profit generated by a replacement (action V) of one of the components of *SW* divided by the time of the action $V(tr)$ minus the time of the action $W(tmr)$ of the component.

This case generally occurs at the end of the heuristic, when it remains enough time to carry out an action V in the place of an action W, which can only increase the system reliability.

The procedure **SelectionAction** returns the ratio of the best action (*RatioMax*), the index of the component on which the action must be performed (*IndexMax*) and the action to be carried out (*ActionMax*). Then, if the best ratio is different from -1 (the realization of a maintenance action remains still possible), the **RealizationAction** procedure executes the maintenance action identified by the variable *ActionMax* on the component identified by the variable *IndexMax*. According to the maintenance action carried out, the sets SFU, SFA and SW and the variables V, W, X, A and R are updated and the total duration T of the maintenance actions performed is increased by the time of the selected action.

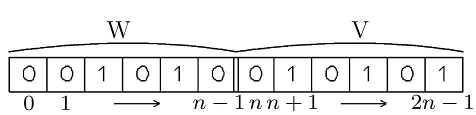

Fig. 4. Solution coding.

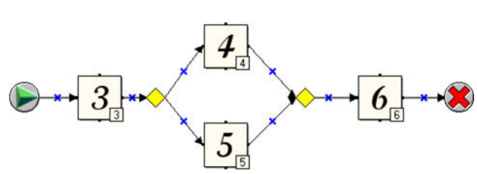

Fig. 5. Elementary system E.

Table 1
Characteristics of the components of the elementary system

Component	β	η (days)	*tmr* (hour)	*tr* (hour)	*trf* (hour)	$B(k)$(days)	$Y(k)$
3	3	120	3	5	1	30	1
4	4	150	2	4	2	60	0
5	2.5	130	1	3	2	28	1
6	4	180	2	6	3	56	0

Procedure: 3. SelectionAction

Parameters ↓: T (integer); R (vector); SFU, SFA, SW (sets)
Parameters ↑: $IndexMax$ (integer); $ActionMax$ (string)
Parameters ↕: $RatioMax$ (real)
for each component $i \in SFU$ **do**
  if $T + trf(SFU(i)) \leqslant T_0$ **then**
    $R' \leftarrow R$
    $R'(SFU(i)) = e^{-\left(\frac{L}{\eta(i)}\right)^{\beta(i)}}$
    $Ratio \leftarrow \frac{F(R' \cdot X) - R_S}{trf(SFU(i))}$
    if $Ratio > RatioMax$ **then**
      $RatioMax \leftarrow Ratio$
      $IndexMax \leftarrow SFU(i)$
      $ActionMax \leftarrow$ VSFU
for each component $i \in SFA$ **do**
  if $T + tr(SFA(i)) \leqslant T_0$ **then**
    $R' \leftarrow R$
    $R'(SFA(i)) = e^{-\left(\frac{L}{\eta(i)}\right)^{\beta(i)}}$
    $Ratio \leftarrow \frac{F(R' \cdot X) - R_S}{tr(SFA(i))}$
    if $Ratio > RatioMax$ **then**
      $RatioMax \leftarrow Ratio$
      $IndexMax \leftarrow SFA(i)$
      $ActionMax \leftarrow$ **VSFA**

  if $T + tmr(SFA(i)) \leqslant T_0$ **then**
    $X' \leftarrow X$
    $X'(SFA(i)) \leftarrow 1$ {Minimal repair of the component $SFA(i)$}
    $Ratio \leftarrow \frac{F(R \cdot X') - R_S}{tmr(SFA(i))}$
    if $Ratio > RatioMax$ **then**
      $RatioMax \leftarrow Ratio$
      $IndexMax \leftarrow SFA(i)$
      $ActionMax \leftarrow$ **WSFA**
for each component $i \in SW$ **do**
  if $T + tr(SW(i)) - tmr(SW(i)) \leqslant T_0$ **then**
    $R' \leftarrow R$
    $R'(SFA(i)) = e^{-\left(\frac{L}{\eta(i)}\right)^{\beta(i)}}$
    $Ratio \leftarrow \frac{F(R' \cdot X) - R_S}{tr(SW(i)) - tmr(SW(i))}$
    if $Ratio > RatioMax$ **then**
      $RatioMax \leftarrow Ratio$
      $IndexMax \leftarrow SW(i)$
      $ActionMax \leftarrow$ VSW

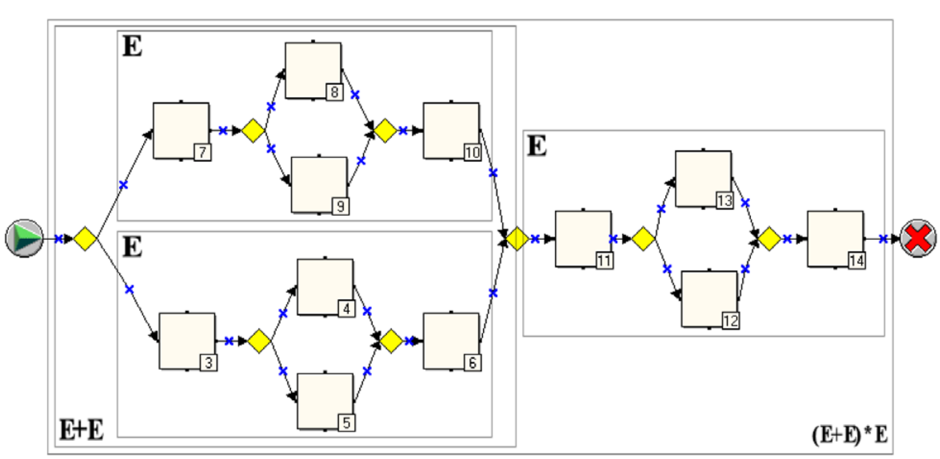

Fig. 6. Sample system (E + E) * E.

Procedure: 4. RealizationAction

Parameters ↓: *IndexMax* (integer); *RatioMax* (real); *ActionMax* (string)
Parameters ↕: *T* (integer); *R* (vector); *SFU*, *SFA*, *SW* (sets)
if $ActionMax = \text{VSFU}$ **then**
  $V(IndexMax) \leftarrow 1$
  $SFU \leftarrow SFU \setminus \{IndexMax\}$
  $T \leftarrow T + trf(IndexMax)$
  $A(IndexMax) \leftarrow 0$
  $R(IndexMax) = e^{-\left(\frac{L}{\eta_i}\right)^{\beta_i}}$
if $ActionMax = \text{VSFA}$ **then**
  $V(IndexMax) \leftarrow 1$
  $SFA \leftarrow SFA \setminus \{IndexMax\}$
  $T \leftarrow T + tr(IndexMax)$
  $A(IndexMax) \leftarrow 0$
  $X(IndexMax) \leftarrow 1$
  $R(IndexMax) = e^{-\left(\frac{L}{\eta_i}\right)^{\beta_i}}$
if $ActionMax = \text{WSFA}$ **then**
  $W(IndexMax) \leftarrow 1$
  $SFA \leftarrow SFA \setminus \{IndexMax\}$
  $SW \leftarrow SW \cup \{IndexMax\}$
  $T \leftarrow T + tmr(IndexMax)$
  $X(IndexMax) \leftarrow 1$
if $ActionMax = \text{VSW}$ **then**
  $V(IndexMax) \leftarrow 1$
  $W(IndexMax) \leftarrow 0$
  $SW \leftarrow SW \setminus \{IndexMax\}$
  $T \leftarrow T + tr(IndexMax) - tmr(IndexMax)$
  $A(IndexMax) \leftarrow 0$
  $R(IndexMax) = e^{-\left(\frac{L}{\eta_i}\right)^{\beta_i}}$

4.2. Exact method

The exact approach is based on a branch and bound (B&B) procedure, which is an arborescent method proceeding by an intelligent enumeration of the solutions space. The enumeration is reduced thanks to pruning, which consists to eliminate subsets of solutions by calculation of bounds on their evaluation functions.

We present below the elements necessary to the development of a B&B procedure [12], i.e.

- the separation rule of the solutions: how to create the subsets of solutions.
- the evaluation function: how to evaluate the subsets of solutions.
- the exploration strategy: how to direct the research in the tree structure.

4.2.1. Separation rule

The separation rule is implicit: a component i is selected and two subsets are created if the component is in a functioning state $((W_i = 0, V_i = 0)$ and $(W_i = 0, V_i = 1))$, and a division in three subsets is realized if the component is failed $((W_i = 0, V_i = 0)$; $(W_i = 0, V_i = 1)$ and $(W_i = 1, V_i = 0))$.

4.2.2. Evaluation function

A subset of solutions is evaluated by relaxation of the time constraint of the selective maintenance problem: a replacement of the components for which no decision was already undertaken is carried out, i.e. the reliability of these components i is regarded as equal to $e^{-\left(\frac{L}{\eta_i}\right)^{\beta_i}}$ with $X_i = 1$. The reliabilities of the components of the subset of solutions depend on the maintenance actions already taken. In this way, the obtaining of an upper limit for the evaluation of a subset of solutions is guaranteed.

4.2.3. Exploration strategy

Two different strategies of exploration were considered, which lead to two alternatives of the B&B [12]:

- Depth search, where the last node created is separated in priority. This method, called Depth-First Branch and Bound (DFBB), has the advantage of being not very greedy in memory.
- Breadth search, where the selected node is the node of maximum evaluation. We call this method Best-First Branch and Bound (BFBB). This alternative presents the disadvantage to consume more memory than the DFBB, but, in general, makes it possible to improve the initial solution rather quickly.

Also, with an aim of accelerating the B&B procedure, it is interesting to have a good initial solution. So, we use the solution found by the construction heuristic as the initial solution.

In addition, the component on which the separation is carried out is, as in the construction heuristic, the component that presents the best ratio reliability of the system after action minus the reliability of the system before action, the whole divided by the time of the action. In this way, we hope to quickly improve the current solution and to eliminate a great number of solutions subsets.

4.3. Tabu search

The tabu search [13], metaheuristic based on the evolution of only one solution, was adapted to the selective maintenance problem, with an aim of improving the solution obtained by the construction heuristic, while maintaining a reasonable resolution time.

We present here the main adaptations necessary to the resolution of the selective maintenance problem:

4.3.1. Solution coding

A solution is coded with an array of size $2 \cdot n$ representing the maintenance actions carried out on the n components. The n first boxes, subscripted of 0 to $n-1$, correspond to the actions W and the n following, subscripted of n to $2n-1$, to the actions V. Fig. 4 represents a solution for $n=6$ components. We carry out the action W on components 2 and 4, and the action V on components 1, 3 and 5.

4.3.2. Starting solution definition

The initial solution of the algorithm is the solution generated by the construction heuristic.

4.3.3. Neighborhood definition

From the current solution X, we randomly draw an index z ranging between 0 and $2n-1$. If $0 \leqslant z \leqslant n-1$ then an action W on the component z is undertaken. If $n \leqslant z \leqslant 2n-1$ then an action V on the component $z-n$ is realized. The neighbor X' is obtained by doing or removing the action corresponding to the index z of the array of X, while being attentive to not carry out an action W on a component in a functioning state, nor carrying out an action V and an action W on the same component. In this last case, the neighbor is generated by permutation of the actions V and W.

Once the neighbor X' is generated, it should also be checked that the total duration of the maintenance actions performed by the neighbor solution does not exceed the maintenance break of length T_0.

4.3.4. Evaluation function

The evaluation of a solution is given by the system reliability obtained with the different maintenance actions considered in the solution.

4.3.5. Management of the tabu list

The movement is characterized by the index of the modified action(s). We then forbid during m (tabu tenure) iterations the choice of this(these) action(s) for the generation of a new neighbor.

4.3.6. Aspiration criterion

A traditional aspiration criterion is also used: if a solution is tabu but is better than the best solution found by the algorithm, the solution is accepted.

5. Numerical results

5.1. Data sets

We apply the construction heuristic, the tabu search and the two versions of the B&B to systems of different sizes and configurations. We also implement for benchmarking purposes an exact method proceeding by a simple enumeration of all the acceptable solutions (which is the method used by Cassady et al. [6]).

The elementary system, noted E, which was used as a basis for the creation of the data sets, is given in Fig. 5. It is composed of four elements, in series and parallel.

Table 2
Symbols and configuration of the systems generated

System	Configuration
4	E
8*	E * E
8+	E + E
12*	E * (E + E)
12+	E + (E * E)
16*	E * (E + (E * E))
16+	E + (E * (E + E))
20*	E * (E + (E * (E + E)))
20+	E + (E * (E + (E * E)))
24*	E * (E + (E * (E + (E * E))))
24+	E + (E * (E + (E * (E + E))))
28*	E * (E + (E * (E + (E * (E + E)))))
28+	E + (E * (E + (E * (E + (E * E)))))

Table 3
Results of the exact methods of resolution of the selective maintenance problem

Systems	$T_0(k)$(h)	Optimal reliability	Time of the exact methods (second)		
			DFBB	BFBB	Enumeration
4	6	0.874	0	0	0
8*	12	0.784	0.01	0	0.02
8+	12	0.987	0.01	0	0.02
12*	18	0.918	0.05	0.05	0.83
12+	18	0.983	0.07	0.04	0.78
16*	24	0.925	1.51	0.88	37.92
16+	24	0.994	0.56	0.49	37.62
20*	30	0.949	29.81	125.65	1893.09
20+	30	0.995	18.22	68.47	1727.90
24*	36	0.954	423.59	–	57360
24+	36	0.997	237.48	–	55920
28*	42	0.957	7765.17	–	–
28+	42	0.998	3483.51	–	–

The components characteristics of this system are given in Table 1. We can notice that two of the four components are failed at the end of mission k. The failure of these components involves the failure of the system, since the component 6 in series is failed.

The problem thus consists in finding the maintenance actions to be undertaken during the maintenance break so as to maximize the system reliability for the next mission.

To generate more complex systems, we take again the elementary system, which is multiplied by arranging it in series and/or parallel. The serialization is represented by the symbol *, and the parallelization by the symbol +. For example, the system (E + E) * E (see Fig. 6) is composed of two elementary systems put in parallel, the whole put in series with another elementary system. So, this system is composed of 12 components, including 6 failed components. We have on the whole generated 13 systems of dimension going from $n = 4$ to $n = 28$. The symbols used for the representation of these systems (which also indicates the number of components of the systems) as their configuration are given in Table 2.

The duration of the next mission $L(k+1)$ is fixed at 40 days for all the systems considered. The maintenance break $T_0(k)$ increases according to the complexity of the system (the more there are failed components, the more we attribute a maintenance break of high duration).

5.2. Results of the exact methods

The results of the application of the exact methods (Depth First Branch & Bound, Best First Branch & Bound and the enumeration) for the various systems generated starting from the elementary system are given in Table 3. They were obtained on a 2.4 GHz Pentium IV having 480Mo of memory.

Table 4
Results of the heuristics of resolution of the selective maintenance problem

Systems	Construction heuristic			Tabu search		
	Reliability	Gap (%)	Time (second)	Reliability	Gap (%)	Time (second)
20*	0.924	2.63	0	0.949	0	0.581
20+	0.961	3.42	0.01	0.994	0.10	0.581
24*	0.924	3.14	0.01	0.953	0.10	0.821
24+	0.961	3.61	0.01	0.995	0.20	0.841
28*	0.924	3.45	0.01	0.955	0.21	1.102
28+	0.961	3.71	0.02	0.998	0	1.122

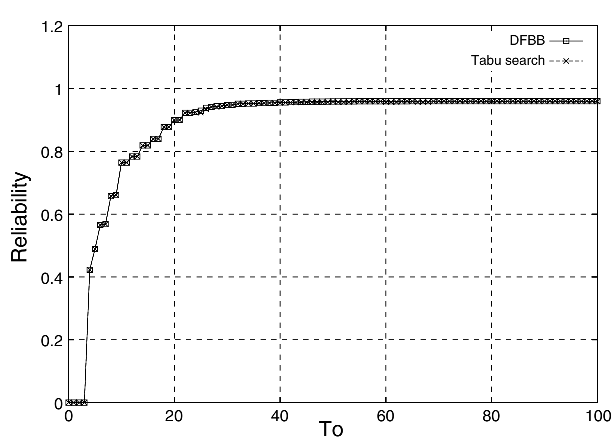

Fig. 7. Comparison between the DFBB and the tabu search for the system noted 28* and for various duration of the maintenance break T_0.

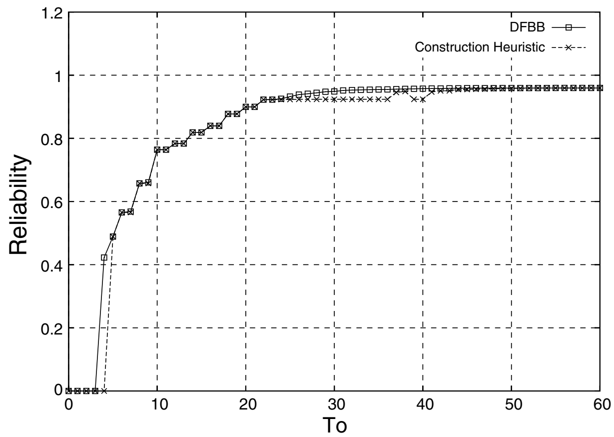

Fig. 8. Comparison between the DFBB and the construction heuristic for the system noted 28* and for various duration of the maintenance break T_0.

We remark that the BFBB version is faster than the DFBB, for systems of size lower or equal to $n = 16$. On the other hand, this difference in the execution time is not very significant. Beyond $n = 20$, the DFBB becomes faster than the BFBB, what can be explained by the great memory place requested by the BFBB, which is necessary to the memorization of all the nodes of the structure tree that are being explored. From $n = 24$, the memory place requested by the BFBB is such as the method is not more applicable.

The complete enumeration of the acceptable solutions quickly becomes not exploitable for systems of dimension greater than $n = 16$. Indeed, for $n = 20$, the execution time is of approximately 30 minutes, and for $n = 24$ this time becomes equal to approximately 15 h. We also notice that the resolution time of the DFBB starting from 20 components does not become negligible any more and that it can be interesting to apply heuristics if the system includes at least 20 components.

5.3. Results of the heuristics

We compare in this section the performances of the construction heuristic and the tabu search and we evaluate the gap between the reliabilities obtained by these methods and the optimal reliability. The number of iterations of the tabu search has been fixed at 1000 given that no more significant improvements have been noted beyond this number. The size of the tabu list is fixed at 7, and the neighborhood is completely explored, which makes the tabu search deterministic. Thus, only one execution of the tabu search is realized.

The results of the construction heuristic and the tabu search are given in Table 4, for the systems having at least 20 components, since it has been shown that starting from this size, it is preferable to apply heuristics rather than an exact method.

We note that the results of the construction heuristic are relatively good, since the maximum gap compared to the optimal solution is 3.71%. The execution time of the heuristic is moreover very low, which makes it possible to instantaneously obtain the solution. The tabu search makes it possible to improve the results of the heuristic and to be very close to the optimal solutions. The maximum gap is indeed 0.21% and the execution time is of about one second.

We also represent in Fig. 7 the evolution of the reliability obtained by the tabu search compared to the reliability obtained by the DFBB according to the duration T_0 of the maintenance break. We used the system noted 28*. We remark that whatever the duration T_0, the reliability of the tabu search is practically equal to the reliability obtained by the exact method. In Fig. 8, we represent the evolution of the reliability obtained by the construction heuristic compared to the reliability obtained by the DFBB. We note that the results of the heuristic are also very close to the optimal reliability (except for the maintenance break $T = 4$ where the construction heuristic obtains a reliability equal to 0 whereas optimal reliability is equal to 0.42).

6. Conclusion

We proposed in this paper new resolution methods for the selective maintenance problem, extended to general architecture systems in series and/or parallel. We highlighted that the construction heuristic gives good results, and moreover very quickly. The exact methods that have been developed, based on a branch and bound procedure, make it possible to considerably reduce the execution time of a complete enumeration of Cassady et al. [6]. The computational time of the exact methods based on the branch and bound becomes, however, important starting from $n = 20$, and we showed that the use of metaheuristics, such as the tabu search, made it possible to significantly improve the results of the construction heuristic and to reach results very close to the optimal solutions in a time remaining completely acceptable.

These resolution methods could be at the base of a new strategy of maintenance of multicomponent systems. This strategy would take the duration T_0 of the maintenance break and the period T at which the system is stopped as parameters. It would thus prove necessary to determine the optimal duration T_0 (if T_0 is too short, few maintenance actions could have been undertaken and the system will present few chances to achieve the next mission and if T_0 is too high, the unavailability of the system will increase) and the optimal period T (if this period is too long, the system is likely to fail). For this, a model of simulation should be developed (reproducing the dynamic of the system during the missions), as well as optimization methods for continuous problems (to determine the continuous parameters T and T_0). The methods developed in this paper would intervene at the time of the maintenance break, for the determination of the maintenance actions to be undertaken. Other actions, such as imperfect replacements, could also be integrated.

A criterion that has not been considered in this study, but that could be added, is the cost of the maintenance actions. It could be integrated as a constraint in the selective maintenance problem (we dispose of a budget for the achievement of the maintenance actions) or like a new criterion, in more of the reliability. This problem has already been tackled by Lust and Teghem [14]: a multicriteria combinatorial optimization problem is obtained, because it is necessary to maximize reliability and to minimize the cost (two contradictory criteria) at the same time. That would imply the intervention of the decision maker, since it will have to choose the solution corresponding best to his preferences.

References

[1] W. Pierskalla, J.A. Voelker, A survey of maintenance models: The control and surveillance of deteriorating systems, Naval Research Logistics Quarterly 23 (3) (1976) 353–388.
[2] Y. Sherif, M. Smith, Optimal maintenance models for systems subject to failure: A review, Naval Research Logistics Quarterly 28 (1981) 47–74.
[3] D. Cho, M. Parlar, A survey of maintenance models for multiunit systems, European Journal of Operational Research 51 (1991) 1–23.
[4] H. Wang, A survey of maintenance policies of deteriorating systems, European Journal of Operational Research 139 (3) (2002) 469–489.
[5] W.F. Rice, C.R. Cassady, J.A. Nachlas, Optimal maintenance plans under limited maintenance time, in: Proceedings of the Seventh Industrial Engineering Research Conference, 1998.
[6] C.R. Cassady, W.P. Murdock, E.A. Pohl, Selective maintenance for support equipment involving multiple maintenance actions, European Journal of Operational Research 129 (1) (2001) 252–258.
[7] J.H. Zhao, Z. Liu, M.T. Dao, Reliability optimization using multiobjective ant colony approaches, Reliability Engineering and System Safety 92 (2007) 109–120.
[8] I. Gertsbakh, Reliability Theory: With Applications to Preventive Maintenance, Springer, 2000.
[9] J.J. Patton, Preventive Maintenance, third ed., ISA – The Instrumentation, System, and Automation Society, 2004.
[10] K.M. Bretthauer, B. Shetty, The nonlinear knapsack problem – Algorithms and applications, European Journal of Operational Research 138 (2001) 459–4472.
[11] L. Maillart, R. Cassady, C. Rainwater, K. Schneider, Selective maintenance decision-making over extended planning horizons, Technical Report 807, Department of Operations, Weatherhead School of Management, Case Western Reserve University, 2005.
[12] F. Hillier, G. Lieberman, Introduction to Operations Research, seventh ed., McGraw-Hill, 2000.
[13] F. Glover, M. Laguna, Tabu Search, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.
[14] T. Lust, J. Teghem, Multicriteria maintenance problem resolved by Tabu search, in: Volumes des Preprints du 12th IFAC Symposium on Information Control Problems in Manufacturing (INCOM'2006), Saint-Etienne, 2006, p. 6.