Reliability Engineering and System Safety 251 (2024) 110335

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

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

Optimal attempt scheduling and aborting in heterogenous system performing asynchronous multi-attempt mission

Gregory Levitin [a,b], Liudong Xing [c,d], Yuanshun Dai [a,*]

[a] School of Computing and Artificial Intelligence, Southwest Jiaotong University, China
[b] NOGA- Israel Independent System Operator, Israel
[c] University of Massachusetts, Dartmouth, MA 02747, USA
[d] Department of Computer Science & Engineering, Graphic Era Deemed to be University, Dehradun, India

ARTICLE INFO

Keywords:
Multi-attempt mission
Activation scheduling
Attempt abort
Common abort command
Expected mission losses
Heterogenous system

ABSTRACT

Multi-attempt mission aborting systems have recently received significant attention from the reliability community. Existing models mostly assume parallel or sequential execution of multiple attempts, incurring great cost or low mission success probability (MSP), respectively. This paper advances the state of the art by considering a new model, where system components may be activated with certain delay allowing to activate next one before the previous component leaves the operation, balancing the expected cost of lost components (ECC) and MSP. Each component may abort its attempt according to an individual aborting policy defined by two parameters (the number of survived shocks and an operation time threshold) or upon receiving a common abort command. Because components may have different shock resistances and performance rates, their activation order can affect both MSP and ECC. Thus, we formulate and solve the optimal attempt scheduling and aborting policy (SAP) problem, which determines the vector of component activation times and the individual attempt aborting policy for each component to minimize the expected mission losses (EML). The EML, a function of MSP and ECC, is evaluated using a new numerical procedure. A detailed case study of a cloud data processing system is provided to demonstrate the proposed model.

1. Introduction

Controlling the risk of valuable asset losses is crucial for diverse safety-critical applications. A common means adopted in practice is to abort the mission task when a certain deterioration condition is met, followed by a rescue or return procedure to save the asset. The effectiveness of the mission aborting is dependent on the mission aborting policy (MAP) adopted. A too early abort would unnecessarily lower the mission success probability (MSP) while a too late abort would unnecessarily lower the asset survival probability. Thus, it is crucial to model and optimize the MAP to strike the balance between the MSP and the asset survivability. The MAP research has been applied to practical applications like aerospace [1], healthcare [2], battlefield [3], chemical reactor [4,5], marine [6], and transportation [7].

1.1. Related research

The MAP studies in the reliability community can be dated back to 1970s [8,9]. However, this research area did not receive significant attention until around 2018 [10,11]. Based on the number of attempts allowed during the mission, the MAP research can be classified into models for single-attempt missions and models for multi-attempt missions.

Different types of MAPs have been studied for single-attempt missions. For example, the single parameter MAP using the number of malfunctioned components was studied for k-out-of-n: F balanced systems [12], k-out-of-n: G systems [13], k-out-of-n multi-state systems [14], standby systems performing dynamic tasks [15], and unmanned aerial vehicles (UAVs) [16]. The single parameter MAP using the number of survived external shocks was studied for single-component systems [2], multi-state systems [17], systems having random rescue time [18], and UAV-truck systems [19]. The single parameter MAP using the completed mission was examined for warm standby systems [20], standby systems subject to failure propagation [21], standby systems with maintenance [22], and standby systems with condition-dependent loading [23]. Other decision variables used in the single parameter MAPs include the system degradation level or status [24–26], the

* Corresponding author.
E-mail address: 1125105129@qq.com (Y. Dai).

https://doi.org/10.1016/j.ress.2024.110335
Received 8 May 2024; Received in revised form 12 June 2024; Accepted 7 July 2024
Available online 10 July 2024

Acronyms

CAC	common abort command
ECC	expected cost of lost components
EML	expected mission losses
GA	genetic algorithm
HPP	homogeneous Poisson process
MAP	mission aborting policy
MSP	mission success probability
OP	operation phase
RP	rescue/return phase
SAP	scheduling and aborting policy
UAV	unmanned aerial vehicle
VM	virtual machine

Notation

T	maximum allowed mission time
N	total number of available system components
K	number of shocks that any component cannot survive with non-negligible probability
τ_j	duration of the OP performed by component j
$\mathscr{L}_j$	lifetime of component j in an attempt (i.e., the time between the start of the attempt and component loss)
$q_j(i)$	probability that component j survives the i th shock
Ω_j	probability that component j survives the first shock
ω_j	shock resistance deterioration factor for component j
c_j	cost incurred by the loss of component j
Λ_j, λ_j	shock arrival rates during OP, RP performed by component j
$\varphi_j(t)$	duration of the RP activated at time t from the beginning of an attempt performed by component j
g_j	performance rate of component j
t_j	activation time of component j
m_j	maximum number of shocks allowed in time interval $[0, \xi_j)$ of the attempt performed by component j
ξ_j	time from the start of an attempt performed by component j after which the attempt is never aborted
$\xi, \boldsymbol{m}, \boldsymbol{t}$	SAP $\xi = \{\xi_1, \ldots, \xi_N\}, \boldsymbol{m} = \{m_1, \ldots, m_N\}, \boldsymbol{t} = \{t_1, \ldots, t_N\}$
$a_j\left(\xi_j, m_j, t\right)$	probability that component j obeying the abort policy ξ_j, m_j remains in the OP till time t when it does not get the CAC
$d_j\left(\xi_j, m_j, t\right)$	probability that component j obeying the abort policy ξ_j, m_j remains in the OP till getting the CAC at time t and successfully completes the subsequent RP started at time t
$z_j\left(\xi_j, m_j, t\right)$	probability that component j obeying the abort policy ξ_j, m_j aborts the OP at time not later than t and successfully completes the subsequent RP
$P(t,i,\lambda)$	occurrence probability of i shocks in $[0,t]$ given that the shock rate is λ
Υ_m	random occurrence time of the m-th shock since the start of an attempt
Υ_{CAC}	random time of CAC since the beginning of the mission
C_F	penalty cost incurred by the mission failure
$C(\xi, \boldsymbol{m}, \boldsymbol{t})$	normalized EML associated with the SAP $\xi, \boldsymbol{m}, \boldsymbol{t}$
$H(\xi, \boldsymbol{m}, \boldsymbol{t})$	ECC during the mission performed under SAP $\xi, \boldsymbol{m}, \boldsymbol{t}$
$R(\xi, \boldsymbol{m}, \boldsymbol{t})$	MSP under SAP $\xi, \boldsymbol{m}, \boldsymbol{t}$
1(A)	logical function 1(FALSE)=0; 1(TRUE)=1.

number of times the system enters an unbalanced state [27] and the predictive reliability [28]. There also exist MAPs using two parameters to determine the condition of triggering the mission aborting. For example, the MAP using the number of malfunctioned components and system age was examined for self-healing systems [29] and standby systems [30]. The MAP using the degradation level or health state and system age was optimized for UAV systems [31] and transportation systems [32]. The MAP using the degradation level and the completed mission work was examined for multistate system with storage [33] and systems working in dynamic environments [34]. The MAP using the number of survived shocks and operation time elapsed was studied for resource constrained mission systems [35].

Examples of MAPs developed for multi-attempt missions [36] include the attempt-independent single parameter MAP using the number of survived shocks [37] and the attempt-dependent single parameter MAP using the degradation level [38,39]. The task-dependent dual-parameter MAP using the number of survived shocks and operation time was investigated for systems that perform multiple independent tasks for missions with unlimited [40] and restricted [41] time. All these multi-attempt models assumed sequential execution of multiple attempts by a single system (i.e., the next attempt cannot start before the preceding one is aborted and the system can be successfully rescued and maintained). When multiple functioning units are available, different attempts may be executed in parallel [42]. For example, the attempt-dependent dual-parameter MAP using the number of survived shocks and operation time was investigated for UAVs where each attempt is executed by two groups of UAVs adopting different MAPs [43].

The sequential execution of multiple attempts may lower the MSP while the parallel execution may incur high cost. To balance the MSP and cost, a consecutive, interval-based multi-attempt model was suggested to allow multiple attempts to start one by one with a predefined time interval; in the event of any attempt being successful (i.e., the mission succeeds), a common abort command (CAC) is issued to terminate all the ongoing attempts to save cost [44]. To further reduce the cost, the CAC may be issued when any attempt is close enough to complete [45]. While a constant activation delay is used in [44,45], the recent work in [46] may accommodate dissimilar activation delays.

1.2. Research contributions

This work considers a new consecutive, interval-based multi-attempt model with dissimilar activation delays. Different from the existing models that all assume statistically identical components [44–46], this work considers a heterogenous mission system with non-identical components characterized by different performance rates, shock resistances and costs. Consequently, the activation order of different system components can affect the MSP and mission cost, leading to a new optimization problem which jointly determines the attempt scheduling (i.e., the activation order and intervals) and attempt-dependent aborting policy to balance the MSP and cost. Specifically, under the new consecutive, multi-attempt mission model, this work makes the following contributions:

(1) Formulating the optimal attempt scheduling and aborting policy (SAP) problem, which determines the vector of component activation times and the individual attempt aborting policy for each component to minimize the expected mission losses (EML).

(2) Proposing a new numerical procedure of assessing the MSP, expected cost of lost components (ECC) and EML for the considered multi-attempt system subject to any SAP.

(3) Solving the proposed EML minimization problem using the genetic algorithm.

(4) Conducting a detailed case study of a cloud data processing software system to demonstrate the proposed model.

(5) Examining impacts of mission time, mission failure cost and individual component cost on the mission performance metrics and the optimal SAP solutions.

The proposed model has broad engineering applications and can be applied to any case where a mission is carried out by multiple agents. Refer to Section 2.4 for a UAV reconnaissance mission system and Section 7 for the cloud data processing system.

The structure for the rest of the paper: Section 2 describes the considered system and attempt aborting model, and formulates the EML optimization problem. A motivating and illustrative example is also given. Section 3 derives the attempt outcome probabilities for a single component. Section 4 derives the mission performance metrics of MSP, ECC and EML, followed by their numerical evaluation procedure in Section 5. Section 6 describes the optimization method. Section 7 conducts the case study and investigates the impacts of several model parameters. Section 8 gives conclusions and managerial suggestions, as well as future research directions.

2. System model and problem formulation

This section depicts the heterogenous, multi-attempt mission system considered in this work, the attempt aborting model as well as the formulation of the optimization problem addressed in this paper. A multi-UAV system is also presented to illustrate the proposed system model.

2.1. System description

The system consists of N statistically different components. Each component j is characterized by performance rate g_j, shock survivability function $q_j(i)$ (see Section 3 for its definition) and cost c_j. The system must accomplish a mission within time T. Multiple attempts can be performed by the components to complete the mission. The components can start the attempts at different time instances such that t_j is the activation time of the j-th component. Two phases are engaged in each attempt: operation phase (OP) with a required amount of work W and rescue/return phase (RP). In the OP and RP phases, each component j is exposed to different random environments modeled by homogeneous Poisson processes (HPP) of shock arrivals with rates Λ_j and λ_j, respectively. The random arrival times of shocks are $Y_1 < Y_2 < \cdots$. The mission is successful if one of the components completes the OP, i.e., operates and survives all shocks during time needed to complete the work W.

2.2. Attempt aborting model

A component operating in a random environment may deteriorate more as more shocks happen to it, incurring a larger risk of losing the component [39]. To decrease the component loss probability, the OP may be terminated or aborted when component j has survived m_j shocks at time $Y_{m_j} < \tau_j = W/g_j$ (i.e., before the mission completion) implying a failed attempt. Following the abortion of OP, the RP is conducted with duration decided using function $\varphi_j(t)$, where t is the time elapsed from the start of the attempt/OP to the start of the RP. In the case of the OP being successfully accomplished, the RP is also conducted with duration of $\varphi_j(\tau_j)$. As Y_{m_j} increases, the time needed for accomplishing the remaining mission task decreases (i.e., being closer to the completion of the OP), making the abortion of OP less beneficial. Let $\xi_j \le \tau_j$ denote the time after which the OP should never be aborted. In other words, in the proposed attempt aborting model the OP continues if $Y_{m_j} \ge \xi_j$. The attempt aborting policy of individual component j is defined by parameters ξ_j and m_j. In addition, to reduce the expected number of component losses, a common aborting rule is implemented, where upon the OP completion by one component j (i.e., surviving all the shocks in the OP during time τ_j), all other already activated components get the CAC and immediately terminate the OP and begin the RP execution. No further components can be activated after the CAC is issued.

The mission fails if no component accomplishes the OP during time T. The failure of the mission incurs the penalty cost C_F. The loss of component j incurs cost c_j.

2.3. Problem formulation

When all the components are activated simultaneously at the beginning of the mission, the probability that at least one of them completes the OP within the time window T (i.e., MSP) is maximal. On the other hand, in this case all the components are exposed to shocks; thus, the expected cost of lost components (ECC) is also maximal.

When the components are activated one by one such that the next component is activated when the previous one leaves the OP (because of the attempt abort or component loss), the minimal overall components exposure to the shocks is achieved as no components are activated after one of them completes the OP and issues the CAC. On the other hand, such an activation schedule presumes the greatest time of the mission accomplishment, and some components may remain inactivated within the time window T even if the mission is not completed. Therefore, in this case both MSP and ECC are minimal.

To hit the balance between achieving the greatest MSP and the lowest ECC and achieve the minimal expected mission losses (EML), the components can be activated with some delay such that the next one is activated before the previous one leaves the OP. Moreover, because the components have different shock resistances and performance rates, the order of their activation affects both the MSP and ECC. The vector of component activation times $\boldsymbol{t} = \{t_1, \ldots, t_N\}$ and the vectors of the abort policy for each component $\xi = \{\xi_1, \ldots, \xi_N\}, \boldsymbol{m} = \{m_1, \ldots, m_N\}$ determine the attempt scheduling and aborting policy (SAP).

For any SAP $\xi, \boldsymbol{m}, \boldsymbol{t}$, the EML can be evaluated as $C(\xi, \boldsymbol{m}, \boldsymbol{t}) = H(\xi, \boldsymbol{m}, \boldsymbol{t}) + C_F(1 - R(\xi, \boldsymbol{m}, \boldsymbol{t}))$, where $H(\xi, \boldsymbol{m}, \boldsymbol{t})$ denotes the ECC during the mission and $R(\xi, \boldsymbol{m}, \boldsymbol{t})$ denotes the MSP. The optimization problem addressed in this work is to determine the SAP $\xi, \boldsymbol{m}, \boldsymbol{t}$ minimizing the EML, which is formulated as

$$\text{Minimize } C(\xi, \boldsymbol{m}, \boldsymbol{t}). \tag{1}$$

2.4. Motivating examples

Consider a reconnaissance mission system with N different unmanned aerial vehicles (UAVs). To accomplish the reconnaissance task, a UAV must cover a distance W from a base to a target and send photo images of the target to the base. The mission fails if no UAV completes the reconnaissance task within time T when the reconnaissance information remains relevant.

During the flight to the target (i.e., the OP), each UAV is exposed to random shocks caused by electromagnetic interference [40,47], which may destroy the UAV's control equipment and cause the UAV to crash. As the number of experienced shocks increases, the interference filter that protects the UAV deteriorates due to overheating, causing the reduction of its resistance to shocks. Each UAV j has a different speed (performance rate) g_j, which determines the OP flight time τ_j and can fly on a different altitude, which determines the shock rate Λ_j. Each UAV j also has a different interference filter deterioration rate, which determines the survival probability of the UAV after experiencing the i-th shock $q_j(i)$.

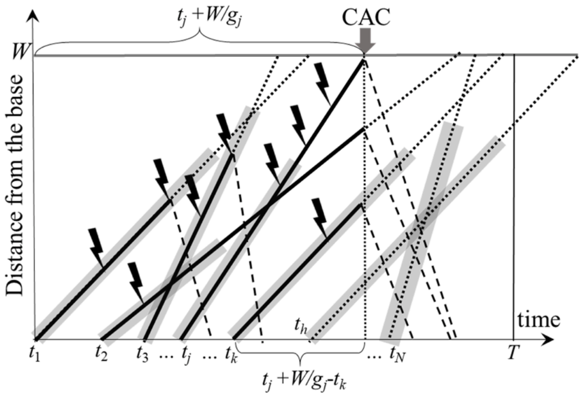

Fig. 1. Example of a mission realization by a heterogeneous system (no component loss case).

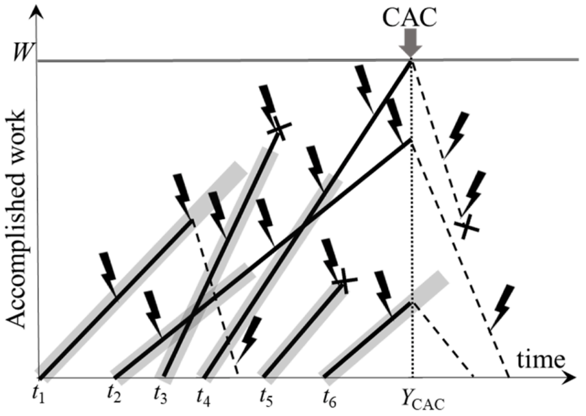

Fig. 2. Examples of different attempt outcomes ($m_j = 2$ for any j).

Each UAV j starts its reconnaissance attempt at time t_j from the beginning of the mission. When m_j shocks occur to the UAV j before time ξ_j, the attempt is aborted to reduce the UAV's loss probability. If one of the UAVs reaches the target and sends the images, the rest of the UAVs that are already on the way to the target abort their attempts immediately. If the attempt is aborted or accomplished at time t from the beginning of the attempt, the UAV flies back to the base (i.e., performs the RP). During the RP, the UAV can change the speed and altitude, which determines the RP time $\varphi_j(t)$ and shock rate λ_j during the RP.

Fig. 1 presents an example of a reconnaissance mission realization by a heterogeneous system of N components (UAVs). Solid and dashed lines correspond to OP (flight to the target) and RP (return flight), respectively. Dotted lines correspond to OP durations required for the attempt's completion (reaching the target). Grey rectangles indicate the parts of OP during which the OP aborts are allowed. In the example, $m_j = 2$ for any j. UAVs 1 and 3 abort their attempts upon the occurrence of the second shock and start the RP. UAV j experiences one shock before time ξ_j since its activation, two additional shocks after time ξ_j, survives these shocks and completes the attempt at time t_j+W/g_j. Immediately after the attempt completion, the CAC is issued, which causes attempt aborts and RP activation of UAVs 2 and k that still remain in the OP. The UAV N is not activated because t_N>t_j+τ_j and the mission is completed by UAV j before its activation. UAV h is not activated because t_h+τ_h>T, i.e., it cannot complete the mission within the required time window.

Refer to Section 6 for another example of a computational task performed by a group of virtual machines created on different servers in a cloud environment.

As another application example, consider a company that can use up to K offshore drilling rigs for underwater drilling in an oil-producing region under an oil proof contract. The period of work is limited by the contract. All drilling rigs operating simultaneously in the exploration area are subject to random storm impacts. A fixed time τ_j is needed for each rig j to reach the oil-bearing layer (depending on the specific rig construction and the layer's depth). Storms (shocks) can damage and destroy any rig, making further drilling impossible and causing losses. The rig fatal destruction probability increases with an increase in the number of experienced storms. Therefore, it is beneficial to interrupt the drilling mission and evacuate a rig if the number of storms it experiences exceeds some value. If one of the rigs completes its work, all rigs can be evacuated and no additional rigs are deployed. Storms can also impact the rigs during evacuation. The company's management problem is to find the rigs activation schedule and mission aborting policy balancing the losses associated with the damage to the rigs and penalty associated with the oil proof mission failure.

3. Attempt outcome probabilities for single component

Under the combined individual and common attempt aborting policy, any component j activated at time t_j can leave the OP in the following four cases (see examples in Fig. 2 and logic diagrams in Figs. 3 and 4):

1. If the component aborts the OP upon the CAC, which is issued at time Υ_{CAC}−t_j from the component activation by one of the previously activated components when $\Upsilon_{\text{CAC}} - t_j$<$Y_{m_j} \leq \xi_j$ (see component 6 in Fig. 2) or $Y_{m_j} > \xi_j$ and $\Upsilon_{\text{CAC}} - t_j$<$Y_{m_j}$<$\tau_j$ (see component 2 in Fig. 2);

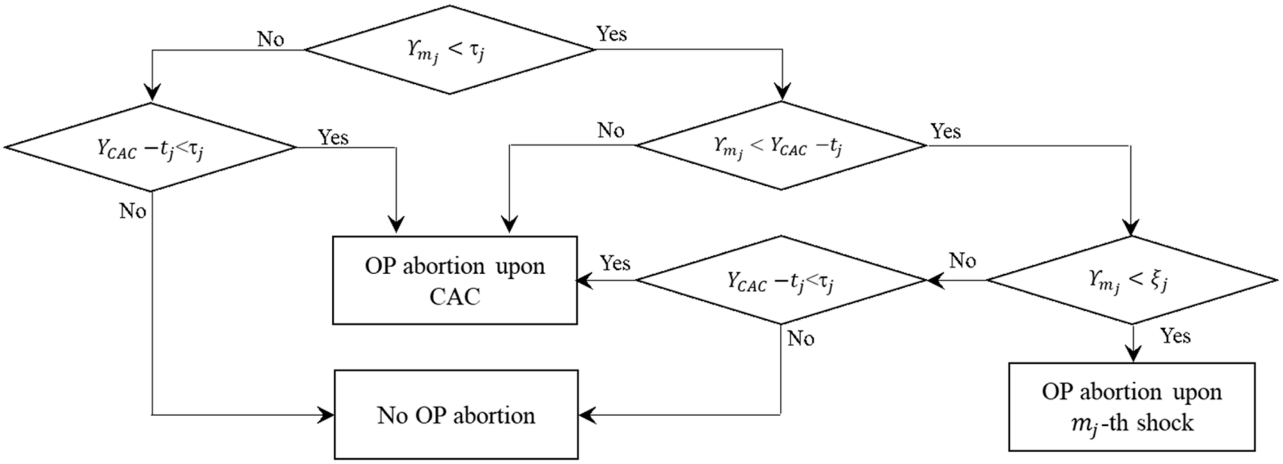

Fig. 3. Conditions of leaving the OP for surviving component.

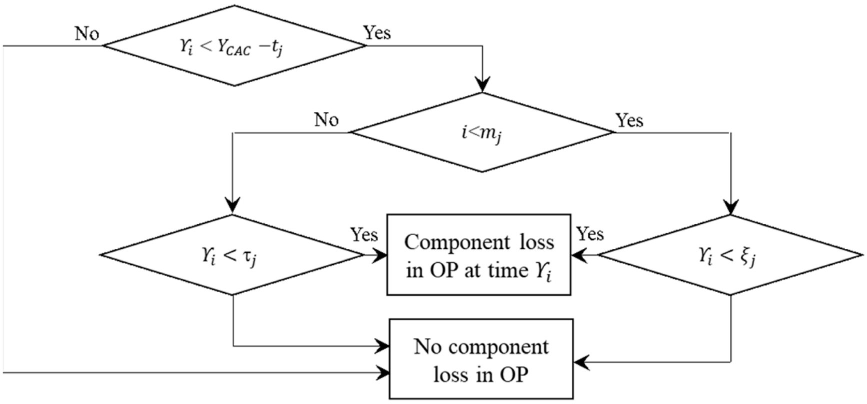

Fig. 4. Conditions of component loss in OP upon i th shock.

2. If the component aborts the OP upon the m_j-th shock, which occurs at time $Y_{m_j} \le \min\left(Y_{CAC} - t_j, \xi_j\right)$ (see component 1 in Fig. 2);
3. If the component completes the OP after operating during time τ_j, which happens when $Y_{m_j} > \xi_j$ and $Y_{CAC} - t_j > \tau_j$ (see component 4 in Fig. 2);
4. If the component is lost after experiencing the i-th shock where $0 < i \le m_j$ if $0 < Y_i \le \min\left(Y_{CAC} - t_j, \xi_j\right)$ (see component 5 in Fig. 2) or $0 < i < \infty$ if $\xi_j < Y_i < \min(Y_{CAC} - t_j, \tau_j)$) (see component 3 in Fig. 2).

Any component j that starts the RP at time t from the beginning of the OP can leave the RP either when it is lost after experiencing the i-th shock where $t < Y_i \le t + \varphi_i(t)$ (see component 4 in Fig. 2) or when it completes the RP after surviving all shocks (see components 1, 2 and 6 in Fig. 2).

Under the HPP with rate ρ, the probability that i (i=0, 1, 2,…) shocks occur to a system component during time interval [0,t) is

$$P(t, i, \rho) = e^{-\rho t}\frac{(\rho t)^i}{i!}. \tag{2}$$

In (2), $\rho = \Lambda$ for the OP and $\rho = \lambda$ for the RP of the attempt. Thus, $P(\xi, i, \Lambda)P(\tau - \xi, k, \Lambda)$ denotes the probability that i shocks happen in [0,ξ) and additional k shocks happen in [ξ,τ) during the OP of the attempt. $P(\xi, i, \Lambda)P(\tau - \xi, k, \Lambda)P(t, h, \lambda)$ denotes the probability that i shocks happen in [0, ξ), additional k shocks happen in $[\xi,\tau)$ during the OP, and h shocks happen in [0,t) since the start of the RP following the OP.

The shock survivability function $q_j(i)$ gives the probability that component j survives the i-th shock, where $q_j(0)\equiv 1$. In this work, $q_j(i)$ (for i>0) is defined as [48,49]

$$q_j(i) = \Omega_j \omega_j(i). \tag{3}$$

In (3), Ω_j is the survival probability of component j under the first shock, and $\omega_j(i) = \omega_j^{i-1}$ is a decreasing function of its argument ($0 < \omega_j < 1$). The shock model in (3) models the decreasing survival probability of a component at each shock as the number of survived shocks increases. Eq. (4) gives the probability that component j can survive I shocks.

$$\prod_{i=0}^{I} q_j(i) = \Omega_j^{I} \omega_j^{\frac{I(I-1)}{2}}. \tag{4}$$

Component j that does not get the CAC remains in the OP at time t since its activation if it experiences fewer than m_j shocks in interval [0, min(t,ξ_j)) and survives all the shocks in [0,t). The occurrence probability of such event is

$$a_j(\xi_j, m_j, t) = \Pr\left(\mathscr{L}_j > t, \mathrm{Y}_{m_j} > \min(t, \xi_j)\right)$$
$$= \sum_{i=0}^{m_j-1} P(\min(t, \xi_j), i, \Lambda_j) \sum_{k=0}^{\infty} P(t - \min(t, \xi_j), k, \Lambda_j) \prod_{h=0}^{i+k} q_j(h), \tag{5}$$

where $\mathscr{L}_j$ denotes the lifetime of the component. As $\xi_j \le \tau_j = W/g_j$, the probability that the component getting no CAC completes the OP (and the mission) is

$$a_j(\xi_j, m_j, \tau_j) = \sum_{i=0}^{m_j-1} P(\xi_j, i, \Lambda_j) \sum_{k=0}^{\infty} P(\tau_j - \xi_j, k, \Lambda_j) \prod_{h=0}^{i+k} q_j(h). \tag{6}$$

Component j operates during the OP till time t (when it accomplishes the OP or receives the CAC) and successfully performs the subsequent RP when fewer than m_j shocks happen in [0, min(t,ξ_j)). In addition, component j survives all shocks that happen during time t in the OP and during time $\varphi_j(t)$ in the RP. The occurrence probability of such attempt outcome is

$$d_j(\xi_j, m_j, t) = \Pr\left(\mathscr{L}_j > t + \varphi_j(t), Y_{m_j} > \min(t, \xi_j)\right) \tag{7}$$

$$= \sum_{i=0}^{m_j-1} P(\min(t, \xi_j), i, \Lambda_j) \sum_{k=0}^{\infty} P(t - \min(t, \xi_j), k, \Lambda_j) \sum_{h=0}^{\infty} P(\varphi_j(t), h, \lambda_j) \prod_{l=0}^{i+k+h} q(l).$$

The probability that component j completes the OP, but is lost in the subsequent RP is

$$a_j(\xi_j, m_j, \tau_j) - d_j(\xi_j, m_j, \tau_j). \tag{8}$$

Component j aborts the OP before time t since its start and survives the subsequent RP if it experiences the m_j-th shock at time $x<\min(t, \xi_j)$ from the start of the attempt and survives m_j shocks during the OP and all the shocks during the RP time $\varphi_j(x)$. The occurrence probability of the m_j-th shock in time interval [x, x+dx) during the OP is

$$P(x, m_j - 1, \Lambda_j)\Lambda_j dx, \tag{9}$$

where dx is infinitesimal.

The probability that component j survives all the shocks during the RP after surviving the m_j-th shock in [x,x+dx) is

$$\sum_{k=0}^{\infty} P(\varphi_j(x), k, \lambda_j) \prod_{i=0}^{k} q_j(m_j + i). \tag{10}$$

Hence, the probability that component j aborts the OP at time no later than t and survives the subsequent RP is

$$z_j(\xi_j, m_j, t) = \Pr\left(\mathscr{L}_j > Y_{m_j} + \varphi_j\left(Y_{m_j}\right), Y_{m_j} \le \min(t, \xi_j)\right) \tag{11}$$

$$= \Lambda_j \int_0^{\min(t,\xi_j)} P(x, m_j - 1, \Lambda_j) \sum_{k=0}^{\infty} P(\varphi_j(x), k, \lambda_j) \prod_{i=0}^{k+m_j} q_j(i) dx.$$

Component j that receives the CAC at time Υ_{CAC}-t_j=t since its activation can survive only if the attempt is aborted at time t (upon getting the CAC) or earlier (upon the occurrence of the m_j-th shock) and the RP is successful. Because these two survival cases are mutually exclusive, the conditional probability of the component loss given that component j receives the CAC at time t<τ_j is

$$1 - d_j(\xi_j, m_j, t) - z_j(\xi_j, m_j, t). \tag{12}$$

4. MSP and ECC evaluation

If component j is activated at time t_j, it can complete the mission at time

$$t_j + \tau_j = t_j + W/g_j. \tag{13}$$

When t_j+τ_j>T, component j cannot complete the mission within the required time window and is not activated.

If t_k>t_j+τ_j, then component k is not activated before the time when component j completes the mission. If t_k+τ_k<t_j+τ_j, then component k cannot remain in the OP when component j completes the mission (see Fig. 1) because the OP of component k must terminate earlier than the OP of component j. If

$$t_k < t_j + \tau_j < t_k + \tau_k \tag{14}$$

then component k can still remain in OP when component j completes the mission and get the CAC at time t_j+τ_j-t_k since its activation.

Consider event E_j that component j completes the mission before time T and issues the CAC. Event E_j occurs at time t_j+τ_j since the beginning of the mission if $t_j + \tau_j \le T$, any component k for which t_k+τ_k≤t_j+τ_j fails to complete the mission and component j survives the OP during time τ_j. Therefore, the probability that component j completes the mission is

$$V_j = 1(t_j + \tau_j \le T) a_j(\xi_j, m_j, \tau_j) \prod_{k=1}^{N} (1 - a_k(\xi_k, m_k, \tau_k))^{1(t_k+\tau_k<t_j+\tau_j)}. \tag{15}$$

where 1() is a logical function: 1(FALSE)=0; 1(TRUE)=1.

As the events of the mission completion for different components are mutually exclusive, the MSP can be obtained as

$$R(\xi, \boldsymbol{m}, \boldsymbol{t}) = \sum_{j=1}^{N} V_j \tag{16}$$

$$= \sum_{j=1}^{N} 1(t_j + \tau_j \le T) a_j(\xi_j, m_j, \tau_j) \prod_{k=1}^{N} (1 - a_k(\xi_k, m_k, \tau_k))^{1(t_k+\tau_k<t_j+\tau_j)}.$$

If event E_j occurs, any component k for which t_k+τ_k≤t_j+τ_j gets no CAC during its OP and leaves the OP (because of abortion or loss) before time τ_k from its activation (does not complete the OP). The probability that such component k is lost is $1 - a_k(\xi_k, m_k, \tau_k) - z_k(\xi_k, m_k, \tau_k)$. The conditional probability of the component k loss given that event E_j occurs (i.e., component k leaves the OP before time τ_k) is

$$\frac{1 - a_k(\xi_k, m_k, \tau_k) - z_k(\xi_k, m_k, \tau_k)}{1 - a_k(\xi_k, m_k, \tau_k)}. \tag{17}$$

As components loss events are independent, the ECC among ones that are scheduled to complete the mission before component j in the case of event E_j occurring is

$$U_j = \sum_{k=1}^{N} c_k 1(t_k + \tau_k < t_j + \tau_j) \frac{1 - a_k(\xi_k, m_k, \tau_k) - z_k(\xi_k, m_k, \tau_k)}{1 - a_k(\xi_k, m_k, \tau_k)}. \tag{18}$$

Based on (8), we get the conditional probability that component j is lost given that event E_j takes place (i.e., component j completes its OP) as

$$\frac{a_j(\xi_j, m_j, \tau_j) - d_j(\xi_j, m_j, \tau_j)}{a_j(\xi_j, m_j, \tau_j)} \tag{19}$$

and the expected cost associated with the loss of component j when event E_j occurs as

$$F_j = c_j \frac{a_j(\xi_j, m_j, \tau_j) - d_j(\xi_j, m_j, \tau_j)}{a_j(\xi_j, m_j, \tau_j)}. \tag{20}$$

When event E_j takes place and t_k<t_j+τ_j<t_k+τ_k≤T, the time between the activation of component k and Υ_{CAC} is t_j+τ_j-t_k . According to (12), the ECC among the components with t_k<t_j+τ_j<t_k+τ_k in the case of event E_j occurring is,

$$S_j = \sum_{k=1}^{N} c_k 1(t_k < t_j + \tau_j < t_k + \tau_k \le T) \times (1 - d_k(\xi_k, m_k, t_j + \tau_j - t_k) - z_k(\xi_k, m_k, t_j + \tau_j - t_k)). \tag{21}$$

The ECC in the case where E_j occurs is

$$H_j = \sum_{j=1}^{N} V_j(U_j + F_j + S_j). \tag{22}$$

Let E_0 denote the event that all components that have been activated during the mission leave the OP before their completion (i.e., no CAC is issued during the mission). The occurrence probability of this event is

$$A_0 = \prod_{k=1}^{N} (1 - a_k(\xi_k, m_k, \tau_k)) 1(t_k + \tau_k \le T). \tag{23}$$

The conditional probability that component k is lost given that it has not completed the OP is given in (17). Thus, the conditional ECC given that event E_0 occurs is

$$U_0 = \sum_{k=1}^{N} 1(t_k + \tau_k \le T) c_k \frac{1 - a_k(\xi_k, m_k, \tau_k) - z_k(\xi_k, m_k, \tau_k)}{1 - a_k(\xi_k, m_k, \tau_k)} \tag{24}$$

and the ECC in the case of event E_0 is

$$H_0 = A_0 U_0. \tag{25}$$

The events E_0, E_1,..., E_N are mutually exclusive. Therefore, the overall ECC is

$$H(\xi, \boldsymbol{m}, \boldsymbol{t}) = \sum_{j=0}^{N} H_j. \tag{26}$$

With MSP $R(\xi, \boldsymbol{m}, \boldsymbol{t})$ and ECC $H(\xi, \boldsymbol{m}, \boldsymbol{t})$ derived, the EML is obtained as

$$C(\xi, \boldsymbol{m}, \boldsymbol{t}) = H(\xi, \boldsymbol{m}, \boldsymbol{t}) + C_F(1 - R(\xi, \boldsymbol{m}, \boldsymbol{t})) \tag{27}$$

5. Numerical EML evaluation procedure

On the basis of the derivation in Sections 3 and 4, we give the pseudocode of the numerical EML evaluation procedure for any given SAP ξ,$\boldsymbol{m}$, $\boldsymbol{t}$.

```
$U_0 = 0$; $A_0 = 1$;
For $k$=1,…,$N$: If $t_k + \tau_k \le T$ then
    Obtain $A_k = a_k(\xi_k, m_k, \tau_k)$, $D_k = d_k(\xi_k, m_k, \tau_k)$, $Z_k = z_k(\xi_k, m_k, \tau_k)$;
    $u_k = c_k(1 - A_k - Z_k)/(1 - A_k)$; $F_k = c_k(A_k - D_k)/A_k$; $U_0 = U_0 + u_k$; $A_0 = A_0 \times (1 - A_k)$;
$R$=0; $H$=$U_0 \times A_0$;
```

(continued on next page)

(*continued*)

```
6    For $j$=1,…,$N$: If $t_j + \tau_j \le T$
7        $V_j = A_j$; $S_j = 0$;
         $U_j = 0$;
8        For $k$=1,…,$N$:
9                $\mu = t_j + \tau_j - t_k$;
10               If $t_k + \tau_k < t_j + \tau_j$ then $V_j = V_j \times (1 - A_k)$, $U_j = U_j + u_k$;
11               If $t_k < t_j + \tau_j < t_k + \tau_k \le T$ then $S_j = S_j + c_k(1 - d_k(\xi_k, m_k, \mu) - z_k(\xi_k, m_k, \mu))$;
12       $R=R+V_j$; $H=H+V_j(U_j + F_j + S_j)$;
13   $C=(1-R)\ C_F + H$;
```

Step 3 obtains $a_k(\xi_k, m_k, \tau_k)$, $d_k(\xi_k, m_k, \tau_k)$, $z_k(\xi_k, m_k, \tau_k)$ using (6), (7) and (11) respectively. Since $\prod_{i=0}^{k} q_j(i)$ is a decreasing function of k, the infinite sum in (6), (7) and (11) can be substituted by the sum for k varying from 0 to K, where $\prod_{i=0}^{K} q_j(i)$ is negligible.

Refer to [48] for the computational aspects of obtaining these infinite sums and an example of determining the value of K in practice. As follows from (11), the computational complexity of step 3 is $O\left(K \times_{1 \le j \le N}^{\max} \tau_j /dx\right)$

Step 4 realizes Eq. (17) and obtains F_j, A_0 and U_0 according to (20), (23) and (24), respectively. Steps 6–11 consecutively obtain V_j, U_j and S_j for j=1,…,N according to (15), (18) and (21), respectively. Step 12 obtains the MSP and ECC according to (16) and (26), respectively. Step (13) computes the EML according to (27). It can be seen from the pseudo code that the computational complexity of Steps 5–13 is $O(N^2)$.

6. EML optimization

Finding the optimal SAP $\xi, \boldsymbol{m}, \boldsymbol{t}$ is a multidimensional optimization problem, where 3N parameters minimizing the EML function $C(\xi, \boldsymbol{m}, \boldsymbol{t})$ should be obtained. To solve the proposed EML minimization problem (1), the genetic algorithm (GA) is implemented in this work, which is one of the most applied optimization techniques in the reliability engineering field [50,51].

In GA, solutions need to be represented in strings. The SAP solution encoding for the proposed optimization problem is as follows. The string consists of 3N integer numbers ζ_{11},…,ζ_{13},…,ζ_{N1},…,ζ_{N3} ranging from 0 to 100 and is decoded as follows

$$\xi_j = 0.01\zeta_{j1}\tau_j;\ m_j = 1 + mod_{K-1}\zeta_{j2};\ t_j = 0.01\zeta_{j3}T$$

such that ξ_j can vary from 0 (corresponding to no abort policy) to τ_j (corresponding to abort allowed during the entire OP), m_j can vary from 1 to K, where K is the maximum number of shocks that any component can survive with non-negligible probability, t_j can vary from 0 (a component is activated at the beginning of the mission) to T (a component is never activated because $t_j + \tau_j > T$))

With the proposed string SAP solution representation, the standard mutation, crossover, and selection operations engaged in the GA optimization process [50,51] are implemented to solve the proposed SAP optimization problem (see the description of the GA in Appendix).

7. Case study

A detailed case study of a cloud data processing system is conducted to illustrate the proposed model and investigate impacts of several parameters on the mission performance and optimized solutions.

Table 1
Parameters of servers.

j	g_j	Λ_j	λ_j	Ω_j	ω_j	c_j	π_j	τ_j
1,2,3	6	0.40	1.20	0.93	0.99	5	0.27	20
4,5	8	0.20	0.97	0.87	0.96	7	0.45	15
6	15	0.65	1.40	0.99	0.97	11	0.32	10

7.1. Cloud system description

Consider a data processing software system that must complete a computational task consisting of W=120 mega-operations by time T=30. The software user has a contract on creating N=6 virtual machines (VMs) with a cloud service provider. The VMs are created on different cloud servers characterized by different protections and different processing speeds g_j. The time required for the VM created on server j to accomplish the task (i.e., the required OP time) is W/g_j. The servers hosting the VMs are exposed to random shocks in the form of hacker attacks with the aim to access and corrupt VM code and data. The success of an attack on server j prevents the computational task completion by the affected VM and causes temporary server shutdown, which incurs cost c_j. The attack rates are server specific such that Λ_j depends on server j protection and environment. Each attack, even when failing, may provide the attacker some information about the server protection, which could be exploited in future attacks. Thus, as the number of survived attacks increases, the attack success probability increases. Such shock resistance deterioration is modeled using (3) with parameters Ω_j and ω_j for each server j.

To reduce the time of exposure to the attacks during the mission, the VMs are not created at the same time; instead, they are created at different time instances in different cloud servers and start their task execution immediately upon creation. An OP running by VM j is aborted if the number of attacks on the server hosting this VM reaches threshold m_j during time ξ_j from the creation of the VM. If any VM completes the task, the rest of the operating VMs get the common command to abort their task execution processes (i.e., the CAC). After the VM aborts the computation task (i.e., the OP) or successfully completes the task, the RP is activated to encrypt and transfer the resulting data from the hosting server. The amount of resulting data $\beta_j(t)$ is proportional to the amount of work completed by the VM by the moment t of RP activation: $\beta_j(t) = Bt/\tau_j$. The data transfer speed in server j is v_j. Therefore, in server j the time needed to complete the RP is $\varphi_j(t)=\beta_j(t)/v_j=Btg_j/(Wv_j)$, which can be represented as $\varphi_j(t)=\pi_j t$ with $\pi_j=Bg_j/(Wv_j)$ being a specific constant for each server. During the RP performed by server j, the shocks/attacks can also take place with rate λ_j depending on the protection of communication channel of the server.

The parameters of each server are presented in Table 1. Notice that several servers have identical parameters as they use identical processors and access protection. It is crucial to determine the SAP $\xi, \boldsymbol{m}, \boldsymbol{t}$ that minimizes the expected losses associated with the unaccomplished mission task and servers' shutdowns during the mission.

Figs. 5–7 present probabilities $a_k(\xi_k, m_k, t)$, $d_k(\xi_k, m_k, t)$, $z_k(\xi_k, m_k, \tau_k)$ obtained for servers 1, 4 and 6 for m_k=1. The probability $a_k(\xi_k, m_k, t)$ that VM k that does not get the CAC remains in the OP at time t in this case is equal to the probability that no attacks (shocks) occur before time t when $t \le \xi_k$. When $t > \xi_k$, $a_k(\xi_k, 1, t)$ becomes the probability that no attacks occur before time ξ_k, any number of attacks take place in time interval $[\xi_k, t)$ and the VM survives all these attacks. This explains the break of the curve in the point t=ξ_k. $\xi_k = 0$ means that no OP aborts are allowed, which corresponds to the greatest values of probability $a_k(\xi_k, 1, t)$. When $\xi_k = \tau_k$, the VM can abort the OP during the entire OP, which corresponds to the minimal values of probability $a_k(\xi_k, 1, t)$.

The probability $d_k(\xi_k, 1, t)$ behaves similar to $a_k(\xi_k, 1, t)$ because the VM has to survive in the OP during time t (which corresponds to $a_k(\xi_k, 1, t)$) and then to survive the RP (which makes $d_k(\xi_k, 1, t)$ always

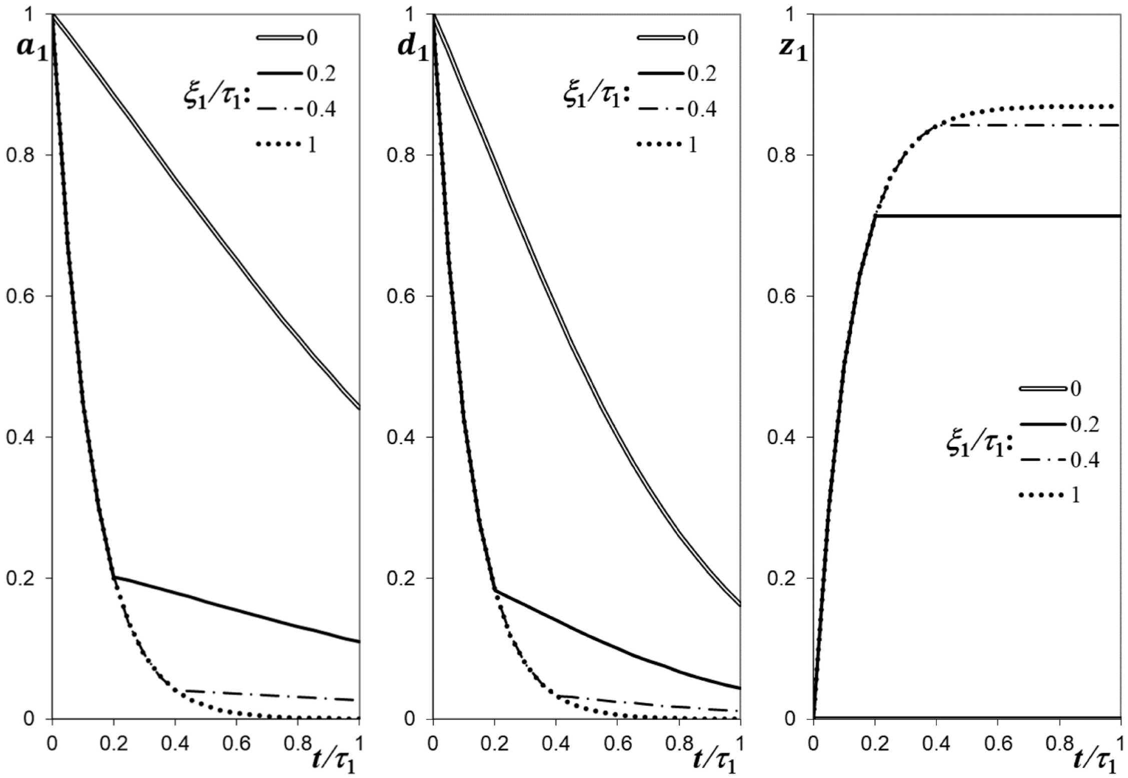

Fig. 5. Probabilities $a_1(\xi_1, 1, t)$, $d_1(\xi_1, 1, t)$ and $z_1(\xi_1, 1, t)$.

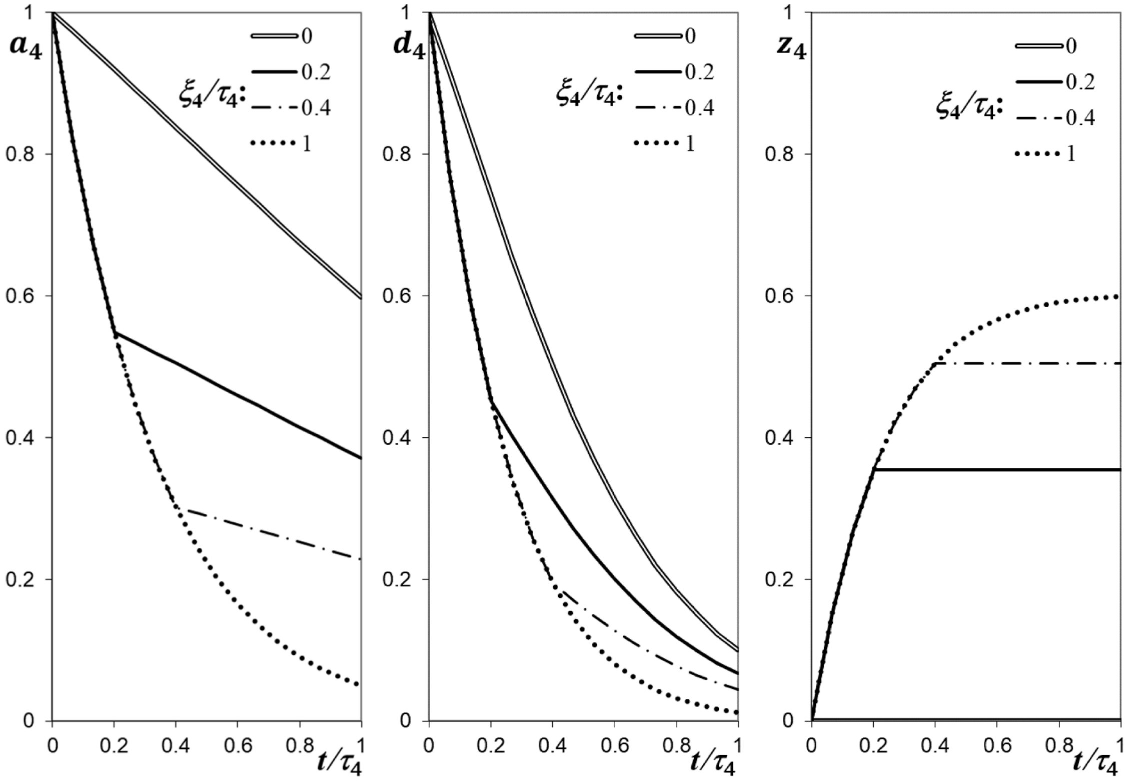

Fig. 6. Probabilities $a_4(\xi_4, 1, t)$, $d_4(\xi_4, 1, t)$ and $z_4(\xi_4, 1, t)$.

lower than $a_k(\xi_k, 1, t)$).

The probability $z_k(\xi_k, 1, t)$ that the VM aborts the OP before time t and survives the subsequent RP increases as ξ_k increases because the time interval during which the OP can be aborted increases. When $\xi_k = 0$, no OP aborts are allowed and, therefore $z_k(0, 1, t) = 0$. For any $t > \xi_k$, the OP can be aborted only in time interval [0, ξ_k). Therefore, $z_k(\xi_k, 1,$

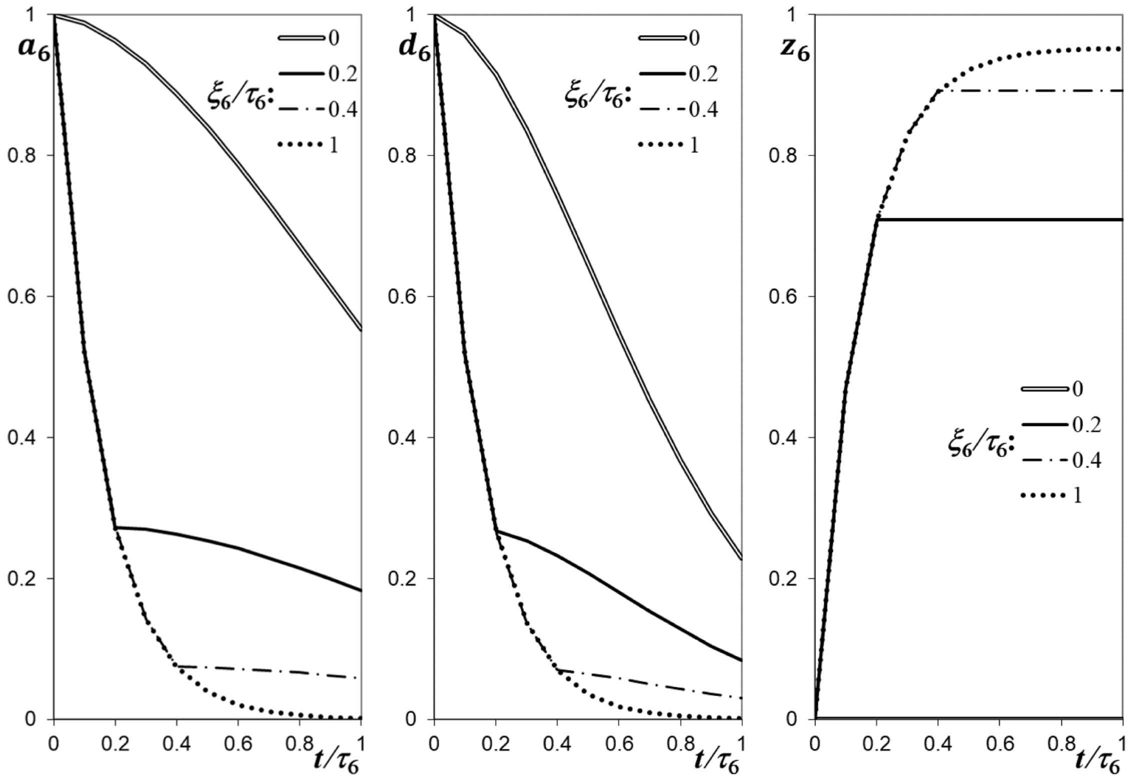

Fig. 7. Probabilities $a_6(\xi_6, 1, t)$, $d_6(\xi_6, 1, t)$ and $z_6(\xi_6, 1, t)$.

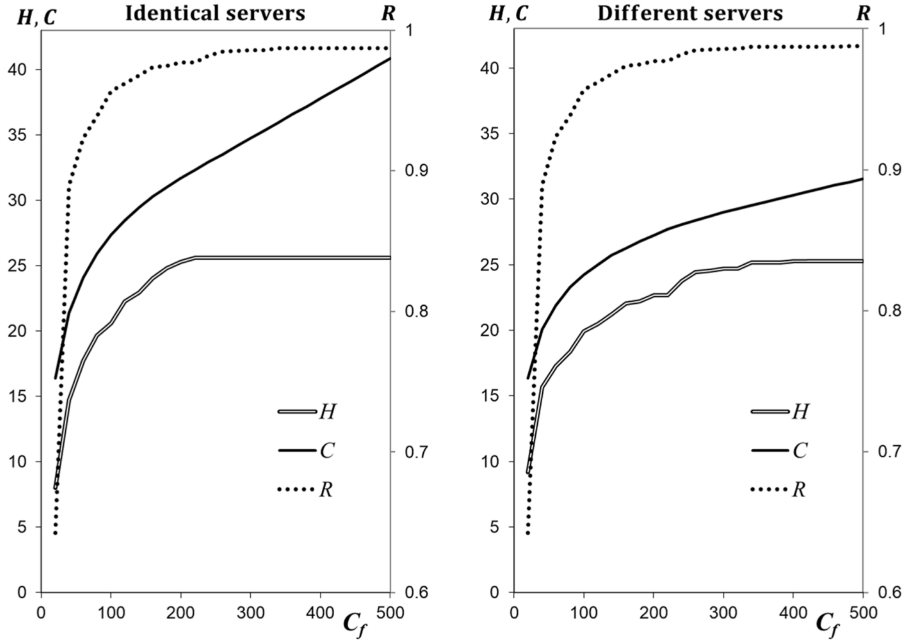

Fig. 8. Parameters of the best obtained solutions as functions of the mission failure cost C_F for identical and different servers.

$t)$=$z_k(\xi_k, 1, \xi_k)$ for t>ξ_k.

7.2. SAP optimization and impact of mission failure cost C_F

Fig. 8 presents the mission performance metrics *R, H* and *C* corresponding to the best obtained SAP as functions of the mission failure cost C_F. Two cases are compared: six different available servers with parameters presented in Table 1 and six identical servers with parameters corresponding to the first row of the table. Some of the SAP solutions are presented in Tables 2 and 3. SAP is represented in the following format *k* (m_k, ξ_k, τ_k) for any server *k* on which a VM is activated during the mission. $k(-\ , 0, \tau_k)$ corresponds to a server on which individual OP aborting is not allowed. Tables 4–8 present the OP schedules for some SAP solutions.

With an increase in C_F, the mission success becomes more important, the aborting policy becomes riskier and more VMs are activated during

Table 2
The best obtained SAP and corresponding mission metrics for system with different servers.

	SAP	R	H	C
20	1(1,0.05,0);2(2,0.05,10);6(1,0.3,0)	0.642	9.201	16.353
40	3(1,0.05,9);4(-,0,15);5(1,0.067,0);6 (1,0.2,15)	0.889	15.674	20.100
80	1(4,0.05,9);4(-,0,0);5(-,0,15);6(1,0.1,15)	0.939	18.368	23.280
100	1(1,0.05,0);2(3,0.05,9);4(2,0.067,0);5 (-,0,15);6(1,0.1,15)	0.957	19.924	24.235
220	1(4,0.05,0);2(6,0.05,9);4(-,0,0);5 (9,0.33,15);6(3,0.1,15)	0.977	22.652	27.706
240	1(2,0.05,0);2(3,0.05,8);3(5,0.05,9);4 (4,0.067,0);5(-,0,15); 6(2,0.2,15)	0.982	23.761	28.067
500	1(-,0,0);2(-,0,8);3(-,0,9);4(-,0,0);5(-,0,15);6 (-,0,15)	0.987	25.287	31.569

Table 3
The best obtained SAP and corresponding mission metrics for system with identical servers.

	SAP	R	H	C
20	1(1,0.1,0);2(1,0.1,1);3(1,0.05,10)	0.581	7.987	16.364
40	1(1,0.05,0);2(1,0.05,1);3(2,0.1,9);4 (3,0.05,10)	0.833	14.737	21.396
80	1(1,0.05,0);2(2,0.1,1);3(2,0.05,8);4 (3,0.05,9);5(6,0.1,10)	0.922	19.657	25.925
100	1(1,0.05,0);2(3,0.05,1); 3(5,0.1,8);4 (6,0.1,9);5(7,0.05,10)	0.933	20.607	27.346
220	1(3,0.05,0);2(5,0.1,1);3(4,0.05,7);4 (5,0.05,8);5(-,0,9);6(-,0,10)	0.969	25.571	32.297
240	1(3,0.05,0);2(6,0.1,1);3(7,0.1,7);4(-,0,8);5 (-,0,9);6(-,0,10)	0.969	25.577	32.908
500	1(-,0,0);2(-,0,1);3(-,0,7);4(-,0,8);5(-,0,9);6 (-,0,10)	0.969	25.608	40.818

the mission, which causes the increase of MSP by the price of increasing ECC. From a certain level of C_F, all six VMs are activated and no individual OP aborting is allowed. This corresponds to the greatest possible MSP and ECC. A further increase in the cost C_F cannot affect the MSP and ECC; thus, the MSP R and ECC H remain constant whereas the EML increases as a linear function of C_F. It can be seen that using different servers allows achieving lower EML than using identical servers.

As it can be seen from the OP schedules presented in Tables 4–8, on identical servers the VMs are activated one by one with a certain delay. On the contrary, when the servers have different characteristics, the VM activation schedule becomes more complex and some groups of VMs can be activated simultaneously.

On identical servers the no aborting policy is used for the last activated VMs for which the activation probabilities are relatively low (because the previously activated VMs have a good chance to complete the mission). This allows to hit the balance between the MSP and the ECC because no aborting policy is used only as a last chance to complete the mission when the previous attempts fail. On different servers, the complex interplay of the servers' parameters leads to a more complex aborting policy in which the VMs created on some servers use no aborting policy though they are not activated last. For example, the VMs on servers 4 and 5 can use the no aborting policy because the shock rates on these servers are minimal, which guarantees the low loss probability of the VMs created on these servers even when no aborts are allowed.

7.3. *Impact of individual aborting policy and mission time T*

If the shocks are unobservable, the individual shock-based aborting is impossible (corresponding to $\xi = 0$ in the proposed model) and only CAC can cause VMs to abort the OP. Fig. 9 presents the comparison of the mission performance metrics R, H and C corresponding to the best obtained SAP with and without individual OP aborting as functions of the allowed mission time T for $C_F = 100$. Six different servers with parameters from Table 1 are considered. Some of the SAP solutions are presented in Tables 9 and 10.

Intuitively, the EML decreases as the mission time increases because fewer VMs must operate simultaneously being exposed to attacks. When individual shock-based OP aborting is not allowed, the minimal value is achieved when T=40 and a further increase in the mission time does not affect the optimal SAP and EML. Five out of the six VMs are activated during the mission because activating the sixth VM on the costliest server without individual aborting option is too risky and causes an increase in the ECC. When the individual OP aborting is allowed, the EML for T<50 is very close to the EML achieved without individual aborting. Indeed, when T is small, the overlapping of OPs for different VMs is high and the CAC has greater influences on the OP aborting than shocks. On the contrary, when T increases and overlapping decreases, the contribution of the individual OP aborting to the ECC reduction increases, which allows a further increase in the EML. All six VMs are activated and the balance between the MSP and ECC is achieved by choosing the individual OP aborting policy.

Table 4
OP schedule for C_F = 20 (different servers).

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

Table 5
OP schedule for C_F = 220 (different servers).

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

Table 6
OP schedule for $C_F = 500$ (different servers).

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

Table 7
OP schedule for $C_F = 20$ (identical servers).

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

Table 8
OP schedule for $C_F \geq 220$ (identical servers).

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

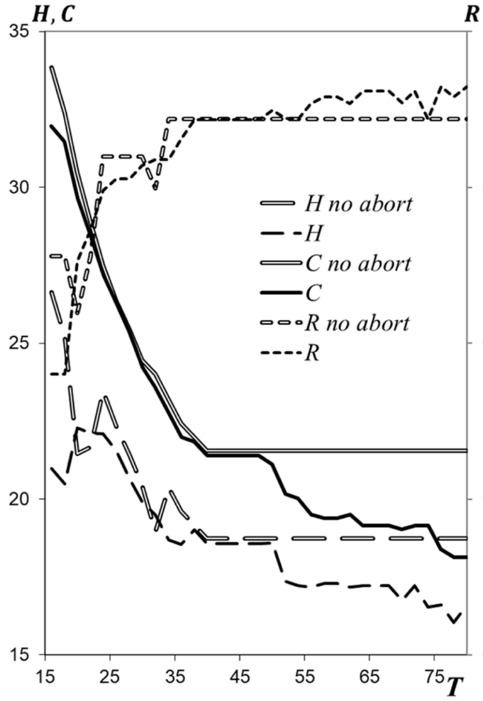

Fig. 9. Parameters of the best obtained solutions as functions of the allowed mission time T for missions with and without individual OP aborting.

Table 9
The best obtained SAP and corresponding mission metrics for different values of mission time T (individual OP aborting is allowed).

T	SAP	R	H	C
20	1(6,0.05,0);4(3,0.067,0);5(-,0,4);6(1,0.2,0)	0.926	22.275	29.651
30	1(3,0.05,9);2(1,0.05,0);4(2,0.067,0);5(-,0,15);6(1,0.1,15)	0.957	19.924	24.235
40	1(-,0,20);2(3,0.05,1);3(3,0.05,0);4(1,0.067,19);5(-,0,20);6(1,0.2,20)	0.972	18.573	21.399
50	1(2,0.05,1);2(-,0,30);3(2,0.05,0);4(-,0,20);5(2,0.067,19);6(1,0.2,20)	0.975	18.597	21.115
60	1(-,0,21);2(3,0.05,0);3(3,0.05,20);4(8,0.267,40);5(-,0,39);6(1,0.2,40)	0.977	17.175	19.495
70	1(-,0,32);2(-,0,31);3(5,0.1,10);4(-,0,50);5(9,0.267,51);6(1,0.2,0)	0.977	16.739	19.041
80	1(5,0.05,15);2(-,0,36);3(-,0,37);4(1,0.133,0);5(-,0,58);6(5,0.2,56)	0.982	16.614	18.390

Table 10
The best obtained SAP and corresponding mission metrics for different values of mission time T (no individual OP aborting is allowed).

T	SAP	R	H	C
20	1(-,0,0);4(-,0,4);5(-,0,0)	0.910	21.445	30.466
30	1(-,0,0);4(-,0,15);5(-,0,0);6(-,0,15)	0.960	20.432	24.456
≥40	1(-,0,1);2(-,0,0);3(-,0,20);4(-,0,20);5(-,0,19)	0.972	18.725	21.540

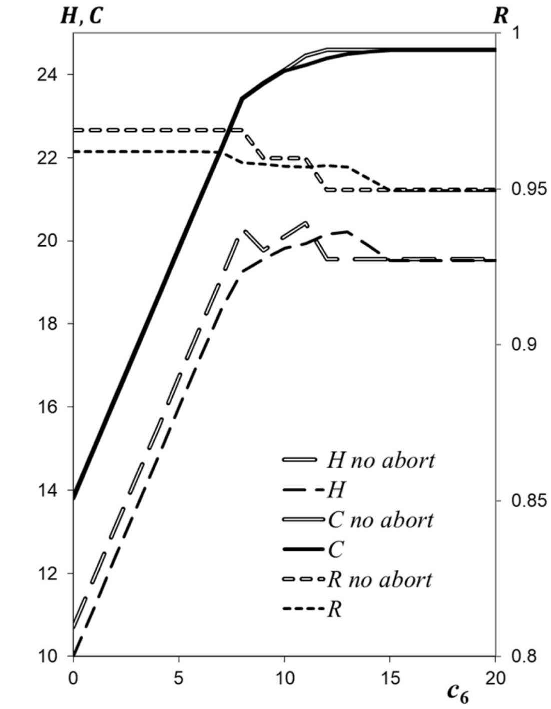

Fig. 10. Parameters of the best obtained solutions as functions of server's 6 shutdown cost for missions with and without individual OP aborting.

Table 11
The best obtained SAP and corresponding mission metrics for different values of server's 6 shutdown cost (individual OP aborting is allowed).

c_6	SAP	R	H	C
2	1(-,0,9);2(-,0,10);3(4,0.05,8);5(1,0.067,10);6 (-,0,0)	0.962	12.388	16.198
7	1(4,0.05,8);2(-,0,9);3(8,0.1,10);5 (1,0.067,10);6(5,0.2,0)	0.962	18.353	22.185
10	2(2,0.05,0);4(3,0.067,0);5(-,0,15); 6(3,0.1,15)	0.957	19.824	24.093
14	1(4,0.05,9);3(2,0.05,8);4(8,0.133,15);5 (4,0.067,0);6(1,0.3,15)	0.953	19.869	24.557
15	1(3,0.05,8);2(4,0.05,9);4(4,0.067,0);5 (9,0.2,15)	0.949	19.525	24.588

Table 12
The best obtained SAP and corresponding mission metrics for different values of server's 6 shutdown cost (no individual OP aborting is allowed).

c_6	SAP	R	H	C
2	1(-,0,8);2(-,0,9);3(-,0,10);4(-,0,10);6(-,0,0)	0.969	13.107	16.233
7	1(-,0,10);2(-,0,9);3(-,0,8);4(-,0,10);6(-,0,0)	0.969	19.099	22.225
10	2(-,0,0);4(-,0,0);5(-,0,15);6(-,0,15)	0.959	20.100	24.124
$\geq$14	1(-,0,8);2(-,0,9);4(-,0,0);5(-,0,15)	0.950	19.550	24.590

7.4. *Impact of individual aborting policy and component loss cost*

Fig. 10 presents the mission performance metrics *R*, *H* and *C* corresponding to the best obtained SAP with and without individual OP aborting as functions of server's 6 shutdown cost c_6 for T=30 and C_F = 100. The shutdown costs of the rest of the servers remains the same as in Table 1. Some of the SAP solutions are presented in Tables 11 and 12 and some of the OP schedules are presented in Tables 13–15 (these schedules for SAPs with and without individual OP aborting coincide).

When $c_6<5$, the VM on server 6 is activated at the beginning of the mission having no chance to be aborted by the CAC and the no shock-based OP abort is allowed for this VM. When $5\leq c_6\leq 7$, the VM on server 6 is still activated at the beginning of the mission, but the individual OP aborting is allowed. A further increase in c_6 leads to a late activation of the VM (at time 15) such that it becomes affected by the CAC and a more cautious OP aborting policy (increase of ξ_6 and decrease of m_6). When c_6 exceeds 14, the VM on server 6 is not activated during the mission and the value of c_6 does not affect the mission metrics anymore.

When no individual aborts are allowed, the VM on server 6 is not activated when c_6 exceeds 11 because the risk of the server shutdown cannot be reduced by the cautious individual OP aborting policy. Therefore, for $11<c_6<14$ the EML in the case where no individual aborting is allowed becomes greater than in the case where the individual aborting is allowed. For the rest of values of c_6 the difference in EML is negligible.

8. Conclusion and future research directions

This paper has formulated a new optimization problem, the optimal attempt scheduling and aborting policy (SAP) problem for a heterogeneous multi-attempt mission system with *N* components characterized by different performance rates, shock resistances and costs. Specifically, a vector of component activation times $\boldsymbol{t}=\{t_1,\dots,t_N\}$ and vectors of the abort policy for each component $\xi=\{\xi_1,\dots,\xi_N\},\boldsymbol{m}=\{m_1,\dots,m_N\}$ are determined to minimize the EML, striking the balance between the MSP and the ECC. A new numerical procedure has been put forward to evaluate the mission performance metrics of MSP, ECC and EML. Based on the SAP solution representation in strings, the GA has been realized to solve the proposed optimization problem. The proposed system model is demonstrated by a multi-UAV reconnaissance mission system. A detailed case study of a cloud data processing software system is also provided to demonstrate the proposed methodology and impacts of several model parameters on the mission performance and the optimal SAP solutions.

Some managerial suggestions taken out of the case study include (1) as the mission failure cost increases, it becomes more important to accomplish the mission, thus it is beneficial to use riskier aborting policies and activate more components during the mission; (2) heterogeneous systems allows achieving lower EML than homogeneous systems; (3) as the mission time proceeds, the EML tends to decrease and eventually stabilizes at the minimal value when individual shock-based attempt aborting is not allowed; (4) a component with a lower loss cost tends to be activated earlier; a component with a higher loss cost tends to be activated later and adopt a more cautious attempt aborting

Table 13
OP schedule for c_6 = 7.

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

Table 14
OP schedule for $c_6 = 10$.

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

Table 15
OP schedule for $c_6 = 15$.

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

policy.

The proposed mission system model assumes a single task carried out during the mission. One future direction is to extend the model by considering missions engaging multiple tasks. Both component-dependent and task-dependent aborting policies may be examined. Another future direction is to address component maintenance during the RP and allow maintained components to re-attempt the task. In the considered model it was assumed that the components are affected by independent shock processes. Further research should consider the case where the same shock process can affect a group of components. The assumption that the mission is completed if any component completes its OP also can be relaxed by considering that each component completing its OP contributes to the mission success and the success probability depends on the number of successful OPs in the mission. The common aborting policy in this case can be based on the number of completed OPs and the number of components lost so far.

CRediT authorship contribution statement

Gregory Levitin: Writing – original draft, Software, Methodology, Conceptualization. **Liudong Xing:** Writing – original draft, Validation, Formal analysis. **Yuanshun Dai:** Project administration.

Declaration of competing interest

There is no conflict of interests associated with this paper.

Data availability

No data was used for the research described in the article.

Acknowledgment

The work of L. Xing was partially supported by the National Science Foundation under Grant No. 2302094.

Appendix

The basic steps of the GA include:

1. Generate an initial population of K_{pop} randomly constructed solutions (strings) and evaluate their fitness values equal to EML C using the numerical procedure proposed in Section 5.
2. Select two solutions randomly, and produce a new solution (offspring) using a crossover procedure which first copies the entire string of the first solution to the offspring and then copies a randomly chosen string fragment from the second parent to the same positions of the offspring.
3. Allow the offspring to mutate with probability p_{mut}. The mutation randomly increments or decrements the number in a randomly chosen string position.
4. Evaluate the offspring fitness (EML) and apply a selection procedure that compares the new offspring with the worst solution in the population and selects the one that is better. The better solution joins the population, and the worse one is discarded. If the population contains equivalent solutions following the selection process, redundancies are eliminated, and the population size decreases.
5. Generate new randomly constructed solutions to replenish the population after repeating steps 2–4 K_{cross} times.
6. Terminate the GA after repeating the genetic cycle (steps 2–5) K_{cycle} times or when no the best solution improvement is achieved after 10 cycles of steps 2–5.

In this work, the parameters K_{pop}=50, p_{mut}=0.8, K_{cross}=2000 and K_{cycle}=100 are chosen. Running the GA 10 times for solving the same problems with different randomly generated initial populations of solutions shows the difference of the obtained EML not exceeding 1.2 %.

References

[1] Dong T, Luo Q, Han C, Xu M. Parameterized design of abort trajectories with a lunar flyby for a crewed mission. Adv Space Res 2023;71(6):2550–65.

[2] Levitin G, Finkelstein M. Optimal Mission abort policy for systems operating in a random environment. Risk Anal 2018;38(4):795–803.

[3] Zhao X, Fan Y, Qiu Q, Chen K. Multi-criteria mission abort policy for systems subject to two-stage degradation process. Eur J Oper Res 2021;295(1):233–45.

[4] Cheng G, Li L, Shangguan C, Yang N, Jiang B, Tao N. Optimal joint inspection and mission abort policy for a partially observable system. Reliab Eng Syst Saf 2023;229:108870.

[5] Levitin G, Xing L, Dai Y. Mission aborting in n-unit systems with work sharing. IEEE Trans Syst Man Cybern Syst 2022;52(8):4875–86.

[6] Thompson F, Guihen D. Review of mission planning for autonomous marine vehicle fleets. J Field Robot 2019;36(2):333–54.

[7] Mayrhofer M, Wächter M, Sachs G. Safety improvement issues for mission aborts of future space transportation systems. ISA Trans 2006;45(1):127–40. https://doi.org/10.1016/S0019-0578(07)60072-X.

[8] Filene RJ, Daly WM. The reliability impact of mission abort strategies on redundant flight computer systems. IEEE Trans Comput 1974;C-23(7):739–43. https://doi.org/10.1109/T-C.1974.224023.

[9] Hyle CT, Foggatt CE, Weber BD, Gerbranchtt RJ, Diamant L. Abort planning for apollo missions. In: Proceedings of the 8th aerospace sciences meeting; 1970. https://doi.org/10.2514/6.1970-94.

[10] Rodrigues A, Cavalcante C, Alberti A, Scarf P, Alotaibi N. Mathematical modelling of mission-abort policies: a review. IMA J Manag Math 2023. https://doi.org/10.1093/imaman/dpad005.

[11] Xing L. Reliability and resilience in the internet of things. chapter 7. Elsevier; 2024.

[12] Wu C, Zhao X, Qiu Q, Sun J. Optimal mission abort policy for k-out-of-n: f balanced systems. Reliab Eng Syst Saf 2021;208:107398.

[13] Myers A. Probability of loss assessment of critical k-out-of-n: g systems having a mission abort policy. IEEE Trans Reliab 2009;58(4):694–701.

[14] Zhao X, Zhang J, Wu C, Wang X. Optimization partial mission abandonment strategy for k-out-of-n multi-state system. Comput Ind Eng 2024;187:109842.

[15] Zhao X, Liu H, Wu Y, Qiu Q. Joint optimization of mission abort and system structure considering dynamic tasks. Reliab Eng Syst Saf 2023;234:109128.

[16] Zhao X, Lv Z, Qiu Q, Wu Y. Designing two-level rescue depot location and dynamic rescue policies for unmanned vehicles. Reliab Eng Syst Saf 2023;233:109119.

[17] Levitin G, Finkelstein M, Xiang Y. Optimal inspections and mission abort policies for multistate systems. Reliab Eng Syst Saf 2021;214:107700.

[18] Levitin G, Xing L, Dai Y. Optimal aborting policy for shock exposed missions with random rescue time. Reliab Eng Syst Saf 2023;233:109094.

[19] Yan R, Zhu X, Zhu XN, Peng R. Optimal routes and aborting strategies of trucks and drones under random attacks. Reliab Eng Syst Saf 2022;222:108457.

[20] Levitin G, Xing L, Dai Y. Mission Abort Policy in Heterogeneous non-repairable 1-out-of-N warm standby systems. IEEE Trans Reliab 2018;67(1):342–54.

[21] Levitin G, Xing L, Luo L. Influence of failure propagation on mission abort policy in heterogeneous warm standby systems. Reliab Eng Syst Saf 2019;183:29–38.

[22] Levitin G, Xing L, Dai Y. Joint optimal mission aborting and replacement and maintenance scheduling in dual-unit standby systems. Reliab Eng Syst Saf 2021;216:107921.

[23] Levitin G, Xing L, Dai Y. Co-optimization of state dependent loading and mission abort policy in heterogeneous warm standby systems. Reliab Eng Syst Saf 2018;172:151–8.

[24] Liu B, Huang H, Deng Q. On optimal condition-based task termination policy for phased task systems. Reliab Eng Syst Saf 2022;221:108338.

[25] Yang L, Wei F, Qiu Q. Mission risk control via joint optimization of sampling and abort decisions. Risk Anal 2024;44(3):666–85.

[26] Zhao X, Wang X, Dai Y, Qiu Q. Joint optimization of loading, mission abort and rescue site selection policies for UAV. Reliab Eng Syst Saf 2024;244:109955.

[27] Fang C, Chen J, Qiu D. Reliability modeling for balanced systems considering mission abort policies. Reliab Eng Syst Saf 2024;243:109853.

[28] Cheng G, Shen J, Wang F, Li L, Yang N. Optimal mission abort policy for a multi-component system with failure interaction. Reliab Eng Syst Saf 2024;242:109791.

[29] Qiu Q, Cui C, Wu B. Dynamic mission abort policy for systems operating in a controllable environment with self-healing mechanism. Reliab Eng Syst Saf 2020;203:107069.

[30] Levitin G, Xing L, Dai Y. Mission Abort policy for systems with observable states of standby components. Risk Anal 2020;40(10):1900–12.

[31] Liu L, Yang J. A dynamic mission abort policy for the swarm executing missions and its solution method by tailored deep reinforcement learning. Reliab Eng Syst Saf 2023;234:109149.

[32] Liu L, Yang J, Yan B. A dynamic mission abort policy for transportation systems with stochastic dependence by deep reinforcement learning. Reliab Eng Syst Saf 2024;241:109682.

[33] Levitin G, Xing L, Dai Y. Optimal mission aborting in multistate systems with storage. Reliab Eng Syst Saf 2022;218:108086. Part A.

[34] Zhao X, Li R, Cao S, Qiu Q. Joint modeling of loading and mission abort policies for systems operating in dynamic environments. Reliab Eng Syst Saf 2023:108948.

[35] Levitin G, Xing L, Dai Y. Optimizing time-varying performance and mission aborting policy in resource constrained missions. Reliab Eng Syst Saf 2024;245:110011.

[36] Levitin G, Xing L. Mission aborting policies and multiattempt missions. IEEE Trans Reliab 2024;73(1):51–2.

[37] Levitin G, Finkelstein M, Xiang Y. Optimal mission abort policies for repairable multistate systems performing multi-attempt mission. Reliab Eng Syst Saf 2021;209:107497.

[38] Qiu Q, Kou M, Chen K, Deng Q, Kang F, Lin C. Optimal stopping problems for mission-oriented systems considering time redundancy. Reliab Eng Syst Saf 2021;205:107226.

[39] Zhao X, Dai Y, Qiu Q, Wu Y. Joint optimization of mission aborts and allocation of standby components considering mission loss. Reliab Eng Syst Saf 2022;225:108612.

[40] Levitin G, Xing L, Dai Y. Optimal task sequencing and aborting in multi-attempt multi-task missions with a limited number of attempts. Reliab Eng Syst Saf 2023;236:109309.

[41] Levitin G, Xing L, Dai Y. Optimal task aborting and sequencing in time constrained multi-task multi-attempt missions. Reliab Eng Syst Saf 2024;241:109702.

[42] Levitin G, Xing L, Dai Y. Optimizing partial component activation policy in multi-attempt missions. Reliab Eng Syst Saf 2023;235:109251.

[43] Levitin G, Xing L, Dai Y. Using kamikaze components in multi-attempt missions with abort option. Reliab Eng Syst Saf 2022;227:108745.

[44] Levitin G, Xing L, Dai Y. Optimal task aborting policy and component activation delay in consecutive multi-attempt missions. Reliab Eng Syst Saf 2023;238:109482.

[45] Meng S, Xing L, Levitin G. Optimizing component activation and operation aborting in missions with consecutive attempts and common abort command. Reliab Eng Syst Saf 2024;243:109842.

[46] Meng S, Xing L, Levitin G. Activation delay and aborting policy minimizing expected losses in consecutive attempts having cumulative effect on mission success. Reliab Eng Syst Saf 2024;247:110078.

[47] Xing L, Johnson BW. Reliability theory and practice for unmanned aerial vehicles. IEEE Internet Things J 2023;10(4):3548–66. https://doi.org/10.1109/JIOT.2022.3218491.

[48] Levitin G, Finkelstein M. Optimal mission abort policy for systems in a random environment with variable shock rate. Reliab Eng Syst Saf 2018;169:1–17.

[49] Cha J, Finkelstein M. On new classes of extreme shock models and some generalizations. J Appl Probab 2011;48:258–70.

[50] Goldberg D. Genetic algorithms in search optimization and machine learning. Addison Wesley Reading, MA; 1989.

[51] Levitin G. Genetic algorithms in reliability engineering. Guest Editorial, Reliab Eng Syst Saf 2006;91(9):975–6.