RAIRO-Oper. Res. 51 (2017) 1–16
DOI: 10.1051/ro/2016002

RAIRO Operations Research
www.rairo-ro.org

APPROXIMATION NEIGHBORHOOD EVALUATION FOR THE DESIGN OF THE LOGISTICS SUPPORT OF COMPLEX ENGINEERING SYSTEMS

Chefi Triki[1,2], Abdallah Al-Alawin[3], Gianpaolo Ghiani[1] and Emanuele Manni[1]

Abstract. This paper deals with the problem of designing the logistics support of complex multi-indenture and multi-echelon engineering systems, with the aim of determining the spare parts stock and the maintenance resources capacity, as well as the level of repair. The problem is modeled as an integer program with a nonlinear probabilistic constraint on the expected availability, whose satisfaction can only be evaluated by means of very time-consuming simulation experiments. Thus, we use an optimization *via* simulation approach, in which the search space is efficiently explored through an approximated neighborhood evaluation mechanism, which makes use of several parameters estimated by means of simulation. Experimental results on a number of instances show the effectiveness of the proposed approach.

Mathematics Subject Classification. 90C90, 90B06.

Received April 15, 2015. Accepted January 14, 2015.

1. Introduction

Complex engineering systems are expensive and long-lived capital equipment (*e.g.*, commercial and military aircrafts and ships, power plants, radars, manufacturing plants, *etc.*) that, once failed, are not simply replaced, but should be repaired by using a variety of complex maintenance resources and highly skilled personnel. During their long life cycle they may fail several times and their repairing and downtime costs may be extremely high. While discarding is the normal decision in case of failure of cheap and highly demanded products, for complex systems discarding or repairing decisions are taken at the component or part level (rather than at the equipment level) and are driven by both economic and non-economic criteria. In this paper, our focus is on designing the logistics support system (LSS) of a given complex engineering system (equipment) with the aim of minimizing its life cycle cost (LCC), subject to minimum *expected availability* constraint (the *availability* is defined by Department of Defence, USA [9] as "a measure of the degree to which an item is in an operable and committable state at the start of a mission when the mission is called for at an unknown (random) time"). More precisely, in this paper we consider equipment availability as a measure of effectiveness, and the minimization of the LCC

Keywords. Inventory; maintenance; level of repair analysis; logistics support design; optimization *via* simulation; approximated neighborhood evaluation.

[1] Department of Engineering for Innovation, University of Salento, Via per Monteroni, 73100, Lecce, Italy.
[2] Department of Mechanical and Industrial Engineering, Sultan Qaboos University, Muscat, Oman. `chefi.triki@unisalento.it`
[3] Department of Business Administration, University of Dubai, Dubai, United Arab Emirates.

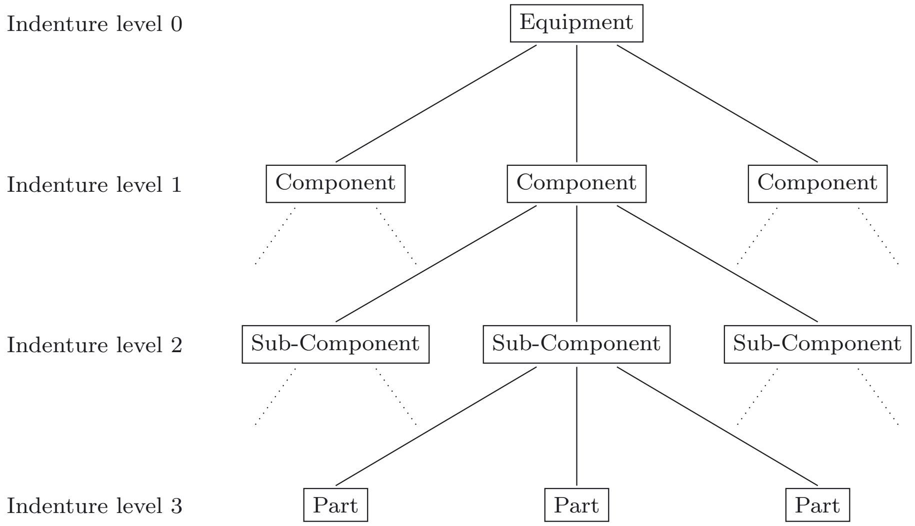

FIGURE 1. Multi-indenture structure of an equipment.

as a measure of efficiency in order to take: (i) *level Of repair analysis (LORA) decisions*: upon a failure of a component, determine whether it has to be discarded and replaced by a functioning component, or repaired. In the latter case, it has to be decided at which facility of the maintenance network it has to be repaired; (ii) *Spare parts decisions*: the number of spare parts that should be stocked for each component at each facility; and (iii) *Maintenance resources location and sizing decisions*: the number and capacity of the maintenance resources of various types to locate at each facility.

These three decisions are tightly related because whether repair or discard a component is influenced by the spare parts inventory as well as by the maintenance resources capacity. From one side, high spare stocks should result in reduced queues caused by scarce maintenance resources. From the other side, increasing the maintenance resources capacity should shorten the repairing processes, allowing to potentially reduce the spare parts at stock. Thus, simultaneously facing the aforementioned three types of decisions is crucial in ensuring a given target equipment availability while minimizing the overall expected cost.

Complex systems are usually represented as tree structures as shown in Figure 1. Indeed, each equipment is usually composed of several components linked each other through father-son relationships. Each level of the structure, called *indenture*, includes several components that once failed may be either discarded or repaired. In the first case, a disposal action is performed and the failed component is simply replaced with a functioning one taken from the stock. In the latter case, the component is removed and sent to be repaired, and a functioning component is put back into the equipment where the failure occurred. If such a component is not immediately available, then the equipment is down, until the failed component is repaired, or a functioning one becomes available from the maintenance network, which is made up by a number of facilities connected to each other at different levels, called *echelons* (Fig. 2). For this reason, we refer to this problem as multi-echelon. Facilities could be bases (sites where equipment operate), depots, workshops, or outsourcing institutions. Each facility can have maintenance resources (both machines and personnel) to repair the defective components, and can send/receive components to/from other facilities.

In this paper, we deal with the integrated problem (LORA, spare parts, and maintenance resources location and sizing decisions) which is solved by means of an optimization *via* simulation approach. In particular, we propose a heuristic procedure that efficiently explores the search space through an Approximated neighborhood evaluation (ANE) model, relying on the estimation (*via* simulation) of a number of parameters, in the spirit of the algorithm proposed by Ghiani *et al.* [12] for scheduling same-day couriers' shifts under probabilistic

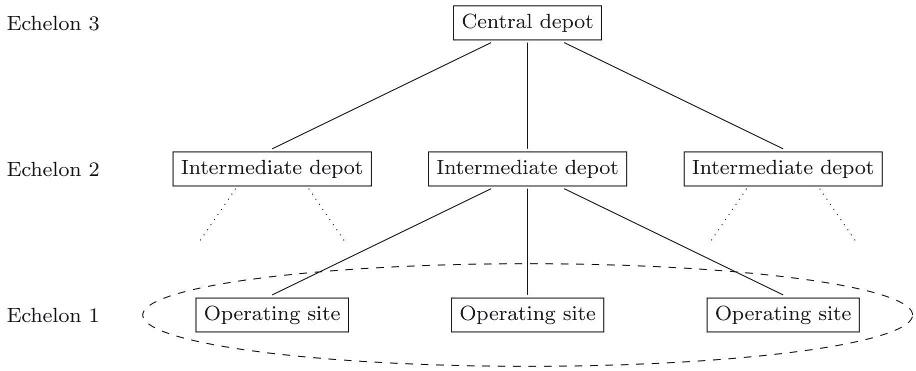

FIGURE 2. Multi-echelon structure of a network.

quality-of-service constraints. In contexts like this, the presence of probabilistic constraints does not allow to straightforwardly assess the feasibility of a solution, that can only be evaluated through time-consuming simulation runs. Thus, an approach that explicitly evaluates each neighbor of a given solution results to be very time-inefficient. On the other hand, a procedure that picks up a solution at random in the neighborhood of the current solution typically performs poorly in practice. Thus, trading off between these two extremes, we develop a neighborhood-search-based procedure that, when simulating a solution at a given iteration, collects some statistics. Such statistics are then used within an ANE framework in which the probabilistic constraints are approximated with deterministic linear functions of the problem variables.

The remainder of the paper is organized as follows. Section 2 summarizes relevant literature, whereas Section 3 provides a mathematical formulation of the problem with nonlinear probabilistic constraints on the expected availability, as well as a description of the general framework we propose. Section 4 gives the details of our ANE model. Section 5 reports our computational experiments. Finally, conclusions follow in Section 6.

2. Literature review

In the literature, two main streams may be identified. The first stream includes approaches that aim at determining only the slow-moving spare inventories required to achieve the target availability with minimum LCC [16, 17]. The second stream focuses on tackling the typical LORA decisions to manage the maintenance network [3]. Some other scholars tried to combine both strategies into an integrated framework [1, 7, 8].

Sherbrooke [16] presents an approach called METRIC (Multi-Echelon Technique for Recoverable Item Control), that is able to determine the inventories for a two-echelon model with the aim of minimizing the number of Expected Back Orders (EBO) over all the recoverable items. The METRIC model is discussed more thoroughly in [17], including the attractive extension VARI-METRIC that takes into account the variable EBO in order to a measure of the operational availability of the equipment.

Alfredsson [1] improves the way of determining the EBO by using queuing theory. In this work, he considers an equipment with a single indenture level, and a two-echelon maintenance network. He combines the LORA problem with the optimization of the spare parts under the METRIC model. The queuing approach used in Alfredsson's model defines the number of resources as variables of the problem affecting equipment's availability. His model results in a non-linear integer program solved by means of convexification techniques combined with a decomposition approach. His outcomes are based on some practical assumptions that, as also acknowledged by the author, may be restrictive, leading to a maintenance system that may be cost-inefficient.

Barros and Riley [3], Saranga and Dinesh Kumar [15] and Basten *et al.* [4] propose integer linear models to solve the pure LORA multi-indenture multi-echelon problem. However, they all assume an infinite capacity of the maintenance resources and aggregate the data per echelon in order to simplify the maintenance decisional process. More specifically, Barros and Riley [3] present an integer programming formulation to solve a multi-indenture and multi-echelon LORA problem by using a branch-and-bound-like method. Their model aims at minimizing the fixed and variable costs of a multiple location problem, relying on an earlier work due to Barros [2], where an integer programming model for the LORA problem is presented by considering the case of facilities with and without maintenance resources. Nevertheless, Barros [2] does not consider spare parts inventory decisions. Saranga and Dinesh Kumar [15] present an integer programming model and propose a methodology based on genetic algorithms for the solution of the multi-indenture and multi-echelon LORA problem. Basten *et al.* [4] generalize the previous works and develop a mixed-integer model for the multi-indenture multi-echelon case. However, even this work does not consider spare parts inventory.

Only few works take into account a limited capacity for the maintenance resources. Diaz and Fu [10] deal with limited repair facilities, and propose an approximation scheme that works well in the case of high facility utilization rates. Sleptchenko *et al.* [18, 19] study the effect of maintenance resources capacity on the multi-echelon multi-indenture problem, and the trade-off between spare parts inventory and maintenance capacity. They also present a procedure for the simultaneous optimization of spare parts and maintenance resources. Zijm and Avşar [21] analyze a two-indenture repairable item system, and propose an approximation model as well as a greedy optimization approach to meet a given target service level at minimal cost. Triki *et al.* [20] propose an optimization *via* simulation approach for a multi-indenture single-echelon problem. Basten *et al.* [5,6] further develop Basten *et al.* [4] by limiting the resources capacity, and ensuring that the components to be repaired at each location are less than the installed capacity of the resources. They develop mixed-integer models for the multi-indenture multi-echelon case, and propose an approach based on a minimum cost flow model. Basten *et al.* [7] then consider the combination of the LORA model with the spare parts inventory but focus only on a single-indenture two-echelon problem. A recent improvement with respect to the previous work has been proposed in Basten *et al.* [8] that consider a complete multi-indenture and multi-echelon structure and proposes an iterative algorithm for its solution. The authors report interesting results even when solving a real-life case related to naval sensor systems, but acknowledge two drawbacks in their work. The first is related to considering a symmetrical repair network (*i.e.*, making the same decisions at all locations of a given echelon for each component and resource). The second is related to the simplifying assumption of infinite repair capacities.

This paper attemps to fill this research gap. More specifically, it extends the definition of LORA maintenance and spare parts decisions to consider the optimal resources capacity while taking into account most of the aspects discussed in Basten *et al.* [7,8]. The three subproblems are integrated into a single optimization *via* simulation framework in order to achieve a target operational availability at minimum expected LCC. In addition, our approach exploits the capabilities of the simulation in order to explicitly consider the scheduling policy of the maintenance processes, the stochastic nature of the failures and maintenance times, the transportation resources, and the possibility of using outsourcing. To the best of our knowledge, no other work in the literature has considered these aspects all together within a unique framework.

3. General framework

We consider a set of identical equipment with a multi-indenture structure, which are located at several operating sites (bases) and are subject to different types of failures that occur according to given stochastic processes. For each component of the highest indenture level, we assume that there can be a single failure mechanism. When a failure occurs, the first decision is related to repairing or discarding the failed component. We assume that the action of discarding can take place only if there is at least a spare part available, which replaces the defective part. On the other hand, in case of a repair action, a second level decision involves at which facility of the maintenance network, organized according to a multi-echelon structure, the repair process will take place, according to a predetermined policy. For instance, the decision could be driven by factors like the number of spare parts available at the different facilities, or the expected time needed to repair the component. In particular, we assume here that the defective component can be repaired at the same facility where the failure occurred (if there is enough capacity), or can be sent to any other facility of the maintenance network.

We assume that the repair process, which is composed of a number of repair tasks, takes place at a single facility, which is the same facility where the spare part is available. Moreover, for each part type, there is a unique sequence of repair tasks, independently of the facility where the repair is performed, and a repair task involves using a unique maintenance resource, that can be shared among different repair processes. If no spare part is available for the failed component, then the equipment is down, until a component of the same type becomes available from a repair process. Downtime comprises the time needed to perform the repair processes, the time components are waiting for spare parts or maintenance resources, and the time needed to transport parts among the different facilities.

A general mathematical formulation of the problem, aiming at minimizing the LCC subject to availability restrictions on the operating equipment, is:

$$\text{Minimize} \quad \text{LCC}(\mathbf{x}) \tag{3.1}$$

$$\text{s.t.} \quad a(\mathbf{x}) \geq a_{\min} \tag{3.2}$$

where:

$$a(\mathbf{x}) = E[A(\mathbf{x}, \xi)].$$

Here, ξ is a vector including the random parameters of our model (*i.e.*, failures occurrences and repair processes durations), $A(\cdot,\cdot)$ represents the equipment availability over the planning horizon, and $a(\cdot)$ is its expected value. Moreover, $a_{\min}$ is the target availability, and $\mathbf{x}$ is a vector representing all the decision variables of the problem. This includes the number of spare parts and discarded components, and the number of required maintenance resources (machines, persons, *etc.*).

The availability function $a(\mathbf{x})$ is not known explicitly and may be only estimated through simulation. Since pure simulation approaches may be very computationally burdensome even for small systems, our attention is devoted to the use of an optimization *via* simulation approach that incorporates an ANE procedure for exploring the solution space, as recently proposed by Ghiani *et al.* [12].

To clarify this aspect, let $\mathbf{x}^{(k-1)}$ be the current solution at iteration $k-1$ of a generic neighborhood search procedure, and let $N(\mathbf{x}^{(k-1)})$ be its neighborhood, defined according to some criteria. In principle, the new solution $\mathbf{x}^{(k)}$ could be selected in $N(\mathbf{x}^{(k-1)})$ as the least cost feasible solution. This approach could be implemented by checking the feasibility (*e.g.*, the satisfaction of the availability constraint through simulation) of such a solution. If this control does not succeed, we should check the second least cost solution, and so on. Unfortunately, the least cost solutions in the neighborhood are likely to be infeasible, because they typically utilize a lower number of resources than $\mathbf{x}^{(k-1)}$. As a consequence, this approach might result in the examination of a huge number of solutions, requiring thus many time consuming simulation experiments. On the other hand, procedures picking up $\mathbf{x}^{(k)}$ at random in $N(\mathbf{x}^{(k-1)})$ perform poorly in practice. Thus, trading off between these two extremes, we propose a procedure that collects some statistics when simulating $\mathbf{x}^{(k-1)}$, and uses these statistics into an ANE procedure. More precisely, the basic idea consists in starting with an initial solution that must be simulated to assess its feasibility from the availability point of view. Then, in order to find a new solution in the neighborhood of the current one, all the statistics about the expected values (detailed in Sect. 4.1) collected at no additional computational cost during the simulation phase are fed into an optimization model, which is based on model (3.1) and (3.2). In particular, in such a model, the availability function $a(\cdot)$ is locally approximated by means of deterministic linear functions of the $\mathbf{x}$ variables. Then, the simulation phase is run again, and this procedure is iterated, until a satisfactory solution is obtained, or a time limit is reached (Fig. 3). It is worth underlying that, since some of the parameters used by the optimization model are estimated when simulating the current solution, the approximation of the availability function is valid only locally (*i.e.*, for neighbors "close" to the current simulated solution). Thus, while moving from one iteration to another for updating the current solution, the optimization phase may involve only small variations of the variables, namely the number of resources, the spare parts and/or the discarded components.

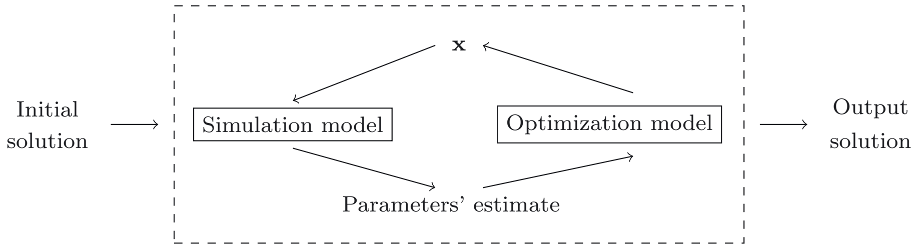

FIGURE 3. The solution approach.

Since our attention, in this paper, is not devoted to the development of complex heuristics, but rather to an efficient exploration of the search space, we embed our ANE model into a basic multi-start local search framework, whose pseudocode is depicted in Algorithm 1. Until a time limit is not exceeded, we first generate an initial solution and, in case it is infeasible, we recover feasibility by means of a MAKEFEASIBLE procedure. Then, given such a solution, we perform a local search phase in which the most promising neighbor of the current solution is obtained by solving the ANE optimization model (APPROXIMATEDNEIGHBORHOODEVALUATION procedure), until we reach a non-improving solution. An initial solution may be obtained in a number of ways. For instance, we could assign to each variable a value that is high enough to be sure that the target availability value is achieved, even if the cost will not be the least possible. Another way is to initialize all decision variables to zero (which obviously will result in an infeasible solution) and then gradually increase their values (by means of the MAKEFEASIBLE procedure). Alternatively, a METRIC-based approach may be used. In this paper, we use two different approaches, as described in Section 5.2.

Algorithm 1. MULTISTARTLOCALSEARCH Pseudocode.

procedure MULTISTARTLOCALSEARCH($t_{\max}$)
    $\mathbf{x}^\star \leftarrow$ null
    $\mathrm{LCC}(\mathbf{x}^\star) \leftarrow +\infty$
    while time $< t_{\max}$ **do**
        $k = 0$
        $\mathbf{x}^{(k)} \leftarrow$ INITIALSOLUTION()
        if $\mathbf{x}^{(k)}$ is infeasible **then**
            $\mathbf{x}^{(k)} \leftarrow$ MAKEFEASIBLE($\mathbf{x}^{(k)}$)
        end if
        repeat
            $k \leftarrow k + 1$
            $\mathbf{x}^{(k)} \leftarrow$ APPROXIMATEDNEIGHBORHOODEVALUATION($\mathbf{x}^{(k-1)}$)
            if $\mathbf{x}^{(k)}$ is infeasible **then**
                $\mathbf{x}^{(k)} \leftarrow$ MAKEFEASIBLE($\mathbf{x}^{(k)}$)
            end if
        until $\mathrm{LCC}(\mathbf{x}^{(k)}) > \mathrm{LCC}(\mathbf{x}^{(k-1)})$
        if $\mathrm{LCC}(\mathbf{x}^{(k-1)}) < \mathrm{LCC}(\mathbf{x}^\star)$ **then**
            $\mathbf{x}^\star \leftarrow \mathbf{x}^{(k-1)}$
            $\mathrm{LCC}(\mathbf{x}^\star) \leftarrow \mathrm{LCC}(\mathbf{x}^{(k-1)})$
        end if
    end while
    return $\mathbf{x}^\star$
end procedure

The MakeFeasible procedure (Algorithm 2) generates a feasible solution by iteratively adding resources and/or spare parts to an initial infeasible solution. Let I, L, O, and U be the sets of component types (type 0 represents equipment), maintenance resource types, facilities, and transportation resource types, respectively. Moreover, ρ_{lo} is the average utilization rate of resources of type $l \in L$ at facility $o \in O$, whereas ρ_u is the average utilization rate of transportation equipment of type $u \in U$. At each iteration, the MakeFeasible procedure adds resources, choosing the one with the highest utilization rate, among those whose utilization rate exceeds a given threshold $\rho_{\max}$. Analogously, the procedure adds spare parts of components such that the fraction of times in which unavailability is due to the lack of such components at certain facilities is greater than a maximum percentage $\pi_{\max}$. Then, the availability function is estimated again through simulation, and this procedure is iterated until $a(\mathbf{x}) \geq a_{\min}$.

Algorithm 2. MakeFeasible Pseudocode.

```
procedure MakeFeasible(x)
    while a(x) < a_min do
        if (∃(l ∈ L, o ∈ O) | ρ_lo > ρ_max) then
            Add 1 maintenance resource of type l ∈ L at facility o ∈ O
        end if
        if (∃(u ∈ U) | ρ_u > ρ_max) then
            Add 1 transportation resource of type u ∈ U
        end if
        //π_io : fraction of times in which unavailability is due to the lack of spare
        //parts of type i ∈ I at facility o ∈ O
        if (∃(i ∈ I, o ∈ O) | π_io > π_max) then
            Add 1 spare part of type i ∈ I at facility o ∈ O
        end if
        Update a(x) through simulation
    end while
    return x
end procedure
```

As stated before, the local search phase (Algorithm 1, line 12) is performed by feeding the current solution into the ANE model and, possibly, applying the MakeFeasible method to its output, in order to recover feasibility.

4. Approximated neighborhood evaluation model

The local approximation for the availability constraint is based on the interaction between a component and its father and son components from one side, and with components of the same type in other facilities from the other side. In particular, these relationships will be expressed as flow balance constraints at both the indenture and echelon levels. In addition, further constraints related to the capacity of maintenance and transportation resources are needed. In what follows, we first introduce some additional notation, and then present the mathematical model used within our ANE procedure.

4.1. Notation

In order to describe our mathematical model, we introduce the following additional notation. We note that all average values are obtained when assessing (*via* simulation) the feasibility of the current solution.

Sets and functions:

H: set of repair tasks (a repair task serves a unique component type and utilizes a unique resource type);

E: set of transportation processes;

$\alpha : H \to L$: function defining a mapping between a repair task and the maintenance resource used to perform it;

$\gamma : H \to I$: function defining a mapping between a repair task and the component type repaired by means of it;
$H_{io} = \{h \in H$: h is a repair task serving a component of type $i \in I$ at facility $o \in O\}$;
$H_{lo} = \{h \in H$: h is a repair task using a maintenance resource of type $l \in L$ at facility $o \in O\}$;
$E_u = \{e \in E$: e is a transportation process using transportation equipment of type $u \in U\}$;
$E_{io}^{sd} = \{e \in E$: e is a transportation process for sending a defective component of type $i \in I$ from facility $o \in O$ to any other facility$\}$;
$E_{io}^{sf} = \{e \in E$: e is a transportation process for sending a functional component of type $i \in I$ from facility $o \in O$ to any other facility$\}$;
$E_{io}^{rd} = \{e \in E$: e is a transportation process by which facility $o \in O$ receives a defective component of type $i \in I$ from any other facility$\}$;
$E_{io}^{rf} = \{e \in E$: e is a transportation process by which facility $o \in O$ receives a functional component of type $i \in I$ from any other facility$\}$;
$R_{io} = \{o' \in O$: o' is a facility that can send components of type $i \in I$ to be repaired at facility $o \in O\}$;
$B_{io} = \{o' \in O$: o' is a facility that can receive components of type $i \in I$ for reparing from facility $o \in O\}$;
$F_i = \{i' \in I$: i' is father of component type $i \in I\}$;
$S_i = \{i' \in I$: i' is son of component type $i \in I\}$.

Parameters:

N_o: number of equipment at facility $o \in O$;
W_{ho}: average workload for repair task $h \in H$ at facility $o \in O$;
n_{ho}: average number of components repaired by means of repair task $h \in H$ at facility $o \in O$;
n_e: average number of components transported by transportation process $e \in E$;
r_{io}: average stock of components of type $i \in I$ at facility $o \in O$;
$g_{ii'o}$: average percentage of components of type $i \in I$ at facility $o \in O$ that are not repaired because of a discarding decision involving a father component of type $i' \in I$;
$m_{io'o}$: average number of components of type $i \in I$ moved from facility $o' \in O$ to be repaired at facility $o \in O$;
$w_{ii'o}$: average number of components of type $i \in I$ waiting for components of type $i' \in I$ at facility $o \in O$;
t: number of years making up the planning horizon;
t_h: processing time for repair task $h \in H$;
t_e: transportation time for transportation process $e \in E$;
c_i: cost of spare parts of components of type $i \in I$;
c_l: cost to purchase resources of type $l \in L$;
c_u: fixed cost of transportation equipment of type $u \in U$;
c_e: variable cost of transportation process $e \in E$;
c_h: variable cost of repair task $h \in H$.

The decision variables are the following:

x_{io}^s: number of spare parts of components of type $i \in I$ at facility $o \in O$;
x_{lo}^r: number of resources of type $l \in L$ at facility $o \in O$;
x_{io}^d: number of discarded components of type $i \in I$ at facility $o \in O$;
x_u^t: number of transportation resources of type $u \in U$;
x_{ie}: number of components of type $i \in I$ to be transported by transportation process $e \in E$.

4.2. Mathematical model

In the following, we present a mathematical model, in which the probabilistic constraint on the expected availability is approximated by using estimates obtained by simulation.

4.2.1. Components balance constraints

When a failure occurs, components at different indenture levels may be involved. At the same time, because components may flow between facilities at different echelons, this flow must be balanced. Thus, the first set of constraints aim at balancing father-son interactions as well as inter-facility flows. It is worth noting that equipment (type $i = 0$) are treated by means of a different constraint. Intuitively, the balance equation for a given component type $i \in I \setminus \{0\}$ and a given facility $o \in O$ can be expressed as:

average no. of components of type i being (repaired + transported + at stock) $-$ initial stock of components of type i + average no. of components of type i that are down because waiting for son components + average no. of components of type i that have been sent to be repaired from o to other facilities

=

average no. of components of type $i' \in F_i$ that are down because are waiting for components of type i + average no. of components of type i that have been sent from other facilities to o for repairing.

Mathematically, the balance constraint, for each component type $i \in I \setminus \{0\}$ and for each facility $o \in O$, is expressed as:

$$\sum_{h \in H_{io}} n_{ho} + \sum_{e \in E^{sd}_{io} \cup E^{sf}_{io}} n_e + r_{io} - x^s_{io} + \sum_{i' \in S_i} w_{ii'o} + \sum_{o' \in B_{io}} m_{ioo'} = \sum_{i' \in F_i} w_{i'io} + \sum_{o' \in R_{io}} m_{io'o}, \quad \forall i \in I \setminus \{0\}, \forall o \in O. \tag{4.1}$$

4.2.2. Equipment balance constraint

As mentioned before, balance constraints should be written differently for the specific case of the equipment (root node in Fig. 1, component type $i = 0$ according to our notation), since from one side an equipment does not have any father component and, from the other side, the target availability refers, obviously, to the equipment and not to components. Thus, the flow balance constraint for the equipment ($i = 0$) is:

$$\sum_{o \in O} \left(\sum_{h \in H_{0o}} n_{ho} + \sum_{i' \in S_0} w_{0i'o} \right) \leq (1 - a_{\min}) \sum_{o \in O} N_o. \tag{4.2}$$

In the left-hand side we consider the average number of equipment that are down, obtained by summing over all the facilities the average number of equipment that are being repaired and the average number of equipment that are waiting for other components to be repaired. This value must not exceed the overall number of equipment available at all the facilities, multiplied by $(1 - a_{\min})$. For instance, if $a_{\min} = 0.9$ and the overall number of equipment is 20, then the average number of down equipment can be at most 2.

4.2.3. Components-facilities balance constraints

These constraints are modeled as classical flow balance constraints, and ensure that the number of defective and functional components of each type sent out of all facilities matches the number of components of the same type entering all facilities:

$$\sum_{o \in O} \sum_{e \in E^{sd}_{io} \cup E^{sf}_{io}} x_{ie} = \sum_{o \in O} \sum_{e \in E^{rd}_{io} \cup E^{rf}_{io}} x_{ie}, \quad \forall i \in I. \tag{4.3}$$

4.2.4. Maintenance resources capacity constraints

These constraints aim at ensuring that the maintenance resources are capacitated enough to face all the workload generated during the planning horizon. The whole workload of each resource type at each facility should be less than its capacity expressed in terms of duration of the planning horizon, its utilization rate, and the related sizing decision variable x^r_{lo}. Thus, the maintenance resources capacity constraints are as follows:

$$\sum_{i \in I} \sum_{h \in H_{lo} \cap H_{io}} t_h \left(W_{ho} - \sum_{e \in E^{sd}_{io}} x_{ie} + \sum_{e \in E^{rd}_{io}} x_{ie} \right) \leq t \rho_{lo} x^r_{lo}, \quad \forall o \in O, \forall l \in L. \tag{4.4}$$

4.2.5. Transportation resources capacity constraints

Analogously, the transportation resources should respect the following capacity limitations:

$$\sum_{i\in I}\sum_{e\in E_u} t_e x_{ie} \le t\rho_u x_u^t, \quad \forall u \in U. \tag{4.5}$$

4.2.6. Objective function

Minimizing the LCC without violating the constraint on the target availability is the goal of this model. There are five types of costs to be considered: the costs of spare parts, the fixed costs of the resources, the fixed costs to start a maintenance process, the variable costs for repairing a component, and the fixed and variable transportation costs. The expression of the LCC to be minimized is:

$$\begin{aligned}\sum_{i\in I}\sum_{o\in O} c_i\left(x_{io}^s + y_{io}^d\right) + \sum_{l\in L}\sum_{o\in O} c_l x_{lo}^r + \sum_{u\in U} c_u x_u^t + \sum_{i\in I}\sum_{e\in E} c_e x_{ie} \\ + \sum_{i\in I}\sum_{o\in O}\sum_{h\in H_{io}} c_h\left(W_{ho} - x_{io}^d - \sum_{i'\in F_i} g_{i'io}x_{io}^d - \sum_{e\in E_{io}^{sd}} x_{ie} + \sum_{e\in E_{io}^{rd}} x_{ie}\right).\end{aligned} \tag{4.6}$$

4.2.7. Neighborhood restriction

In addition to the previously defined constraints, the model used by the ANE procedure includes a constraint that restricts the neighborhood size, by imposing that the distance between the current solution and the new solution is lower than a given threshold D. If we denote by $\mathbf{x}^{(k-1)}$ the solution at iteration $k-1$, and by $d(\cdot,\cdot)$ the function defining the distance between two solutions, the distance constraint can be generically written as:

$$d(\mathbf{x}, \mathbf{x}^{(k-1)}) \le D. \tag{4.7}$$

4.3. Estimation of the number of components being repaired

A maintenance process can be modeled as a queuing system in which a failed component enters for being repaired by means of a number of repair tasks. We assume that during a repair task $h \in H$ a unique resource $l \in L$, such that $l = \alpha(h)$, is needed, and, moreover, a unique type of component $i \in I$, such that $i = \gamma(h)$, is repaired. If the appropriate maintenance resource is busy, the component waits until such a resource becomes available. The average number of components being repaired by means of repair task $h \in H$ at facility $o \in O$ at iteration k is given by:

$$n_{ho}^{(k)} = \frac{\rho_{lo}^{(k)}}{1-\rho_{lo}^{(k)}}, \tag{4.8}$$

where $\rho_{lo}^{(k)}$ is the utilization rate at iteration k of resources of type $l \in L$, such that $l = \alpha(h)$, needed to perform repair task $h \in H$.

If we assume that, during our iterative approach, the expected workload of a resource does remain unchanged from one iteration to another, the utilization rate at the generic iteration k should respect the following equation:

$$\rho_{lo}^{(k)} = \frac{\rho_{lo}^{(k-1)} x_{lo}^{r(k-1)}}{x_{lo}^{r(k)}} \tag{4.9}$$

that relates the utilization rates at iterations $(k-1)$ and k. In (4.9), $x_{lo}^{r(k)}$ denotes the value of variable x_{lo}^r at iteration k. Thus, the average number of components being repaired at iteration k, namely $n_{ho}^{(k)}$, is a function of $x_{lo}^{r(k)}$ and can be easily obtained by combining (4.8) and (4.9):

$$n_{ho}^{(k)} = f\left(x_{lo}^{r(k)}\right) = \frac{\rho_{lo}^{(k)}}{1-\rho_{lo}^{(k)}} = \frac{\frac{\rho_{lo}^{(k-1)} x_{lo}^{r(k-1)}}{x_{lo}^{r(k)}}}{1-\frac{\rho_{lo}^{(k-1)} x_{lo}^{r(k-1)}}{x_{lo}^{r(k)}}} = \frac{\frac{n_{ho}^{(k-1)}}{1+n_{ho}^{(k-1)}} x_{lo}^{r(k-1)}}{x_{lo}^{r(k)} - \left(\frac{n_{ho}^{(k-1)}}{1+n_{ho}^{(k-1)}}\right) x_{lo}^{r(k-1)}}.$$

Given that such an expression is nonlinear, in order to linearize it we make the assumption that while moving from iteration $(k-1)$ to iteration k the value of $x_{lo}^{r(k)}$ can assume one of the following three values: $x_{lo}^{r(k-1)}-1$, $x_{lo}^{r(k-1)}$, or $x_{lo}^{r(k-1)}+1$. Thus, the function $f\left(x_{lo}^{r(k)}\right)$ can be split into three pieces as follows:

$$f\left(x_{lo}^{r(k)}\right)=\lambda_{1lo}^{(k)}f\left(x_{lo}^{r(k-1)}-1\right)+\lambda_{2lo}^{(k)}f\left(x_{lo}^{r(k-1)}\right)+\lambda_{3lo}^{(k)}f\left(x_{lo}^{r(k-1)}+1\right),$$

where $\lambda_{1lo}^{(k)}$, $\lambda_{2lo}^{(k)}$, and $\lambda_{3lo}^{(k)}$ are binary variables associated with the possible values that $x_{lo}^{r(k)}$ can assume at iteration k (*i.e.*, $x_{lo}^{r(k-1)}-1$, $x_{lo}^{r(k-1)}$ or $x_{lo}^{r(k-1)}+1$, respectively). Thus, $x_{lo}^{r(k)}$ can be replaced in the model by:

$$x_{lo}^{r(k)}=\lambda_{1lo}^{(k)}\left(x_{lo}^{r(k-1)}-1\right)+\lambda_{2lo}^{(k)}\left(x_{lo}^{r(k-1)}\right)+\lambda_{3lo}^{(k)}\left(x_{lo}^{r(k-1)}+1\right),$$

with:

$$\sum_{j=1}^{3}\lambda_{jlo}^{(k)}=1.$$

4.4. Effect of discarding components

An important aspect to take into account is that, when deciding to discard a component rather than repairing it, this decision will affect both n_{ho} and the workload W_{ho}, and, consequently, constraints (4.1), (4.2) and (4.4). Specifically, the workload W_{ho} should be reduced by the number of discarded components that, as a consequence of such a decision, do not need to be repaired. Moreover, when a component is discarded, its son components will not be repaired. Thus, the workload for repairing task $h\in H$ at facility $o\in O$ shoud be further reduced by the number of components of type $i\in I$, such that $i=\gamma(h)$, that will not be repaired because of a discarding action involving a father component $i'\in F_i$. Hence, the resulting new workload, denoted as $W_{ho}^{'}$, is:

$$W_{ho}^{'}=W_{ho}-x_{io}^{d}-\sum_{i'\in F_i}g_{i'io}x_{i'o}^{d}, \tag{4.10}$$

where, again, i is such that $i=\gamma(h)$.

The updated workload $W_{ho}^{'}$ should be used in place of W_{ho} within constraints (4.4), but should be also used to update the average number n_{ho} of components being repaired by means of repair task $h\in H$ at facility $o\in O$ and, consequently, constraints (4.1) and (4.2). In order to obtain the modified expression of $n_{ho}^{(k)}$, in the fashion of (4.10), the difference between the discarded components resulting at iteration k and those at iteration $(k-1)$ should be subtracted from $n_{ho}^{(k)}$:

$$n_{ho}^{\prime(k)}=n_{ho}^{(k)}-\left(x_{io}^{d(k)}+\sum_{i'\in F_i}g_{i'io}x_{i'o}^{d(k)}-x_{io}^{d(k-1)}-\sum_{i'\in F_i}g_{i'io}x_{i'o}^{d(k-1)}\right)\frac{t_{ho}}{t}. \tag{4.11}$$

4.5. Outsourcing

Our model includes the possibility to outsource activities to any other facility of the maintenance network. However, an outsourcing facility is characterized by some peculiar features: (i) its own repairing processes are of no interest to our model; (ii) a lead time that is independent from the task to be achieved; (iii) has an infinite number of spare parts; (iv) transferring spare parts to any other facility takes an amount of time that equals the delay time. Finally, the objective function should be modified in order to accommodate the repairing costs at the outsourcing facilities and the corresponding transportation costs.

4.6. The ANE model

The whole ANE model for our multi-echelon multi-indenture problem can thus be stated as:

$$\text{minimize} \quad \sum_{i\in I}\sum_{o\in O} c_i\left(x^s_{io}+y^d_{io}\right)+\sum_{l\in L}\sum_{o\in O} c_l x^r_{lo}+\sum_{u\in U} c_u x^t_u+\sum_{i\in I}\sum_{e\in E} c_e x_{ie}$$
$$+\sum_{i\in I}\sum_{o\in O}\sum_{h\in H_{io}} c_h\left(W'_{ho}-x^d_{io}-\sum_{i'\in F_i} g_{i'io}x^d_{io}-\sum_{e\in E^{sd}_{io}} x_{ie}+\sum_{e\in E^{rd}_{io}} x_{ie}\right)$$

s.t.

$$\sum_{h\in H_{io}} n'_{ho}+\sum_{e\in E^{sd}_{io}\cup E^{sf}_{io}} n_e+r_{io}-x^s_{io}+\sum_{i'\in S_i} w_{ii'o}+\sum_{o'\in B_{io}} m_{ioo'}=\sum_{i'\in F_i} w_{i'io}+\sum_{o'\in R_{io}} m_{io'o}, \quad \forall i\in I\setminus\{0\},\forall o\in O$$

$$\sum_{o\in O}\left(\sum_{h\in H_{0o}} n'_{ho}+\sum_{i'\in S_0} w_{0i'o}\right)\le(1-a_{\min})\sum_{o\in O} N_o$$

$$\sum_{o\in O}\sum_{e\in E^{sd}_{io}\cup E^{sf}_{io}} x_{ie}=\sum_{o\in O}\sum_{e\in E^{rd}_{io}\cup E^{rf}_{io}} x_{ie}, \quad \forall i\in I$$

$$\sum_{i\in I}\sum_{h\in H_{lo}\cap H_{io}} t_h\left(W'_{ho}-\sum_{e\in E^{sd}_{io}} x_{ie}+\sum_{e\in E^{rd}_{io}} x_{ie}\right)\le t\rho_{lo}x^r_{lo}, \quad \forall o\in O,\forall l\in L$$

$$\sum_{i\in I}\sum_{e\in E_u} t_e x_{ie}\le t\rho_u x^t_u, \quad \forall u\in U$$

$$W'_{ho}=W_{ho}-x^d_{\gamma(h)o}-\sum_{i'\in F_{\gamma(h)}} g_{i'\gamma(h)o}x^d_{i'o}, \quad \forall h\in H,\forall o\in O$$

$$n'_{ho}=n_{ho}-\left(x^d_{\gamma(h)o}+\textstyle\sum_{i'\in F_{\gamma(h)}} g_{i'\gamma(h)o}x^d_{i'o}-x^{d(k-1)}_{\gamma(h)o}-\sum_{i'\in F_{\gamma(h)}} g_{i'\gamma(h)o}x^{d(k-1)}_{i'o}\right)\frac{t_{ho}}{t},$$
$$\forall h\in H,\forall o\in O$$

$$d(\mathbf{x},\mathbf{x}^{(k-1)})\le D.$$

5. Computational results

In this section we describe the computational experiments we have performed to validate our approach and to measure its efficiency. As a benchmark, we consider the widely used VARI-METRIC (VM) method. All the algorithms are coded in Java, the optimization models are solved by means of IBM ILOG CPLEX 12.3, and the experiments are run on a computer with an Intel Core i5 processor clocked at 2.53 GHz with 4 GB of RAM.

5.1. Test instances and simulation framework

Since no benchmark test problems are available, we consider a test case, resembling the maintenance of complex equipment made up of four indenture levels, namely the equipment at level zero, eight components at level one, 40 sub-components at level two, and 225 parts at the last indenture level. For each part at the last level, we suppose that failures are generated according to a Poisson process with randomly generated rates in $[0.5, 3]$ failures per year. Moreover, we consider a network with four facilities, consisting of two bases (having $N_1 = 10$ and $N_2 = 8$ equipment, respectively), one intermediate facility, and one depot. Furthermore, the depot can make use of outsourcing to face high workloads. The maintenance resources consist in four types of devices ($L = \{1, \ldots, 4\}$). We assume that the cost for purchasing each type of resource (in M€) is $4, 5, 1$, and 3, respectively. The cost (in M€) of purchasing one spare part of each component is generated uniformly in $[0.2, 0.4]$ for components at level one, in $[0.02, 0.04]$ for sub-components at level two, and in $[0.002, 0.004]$ for parts at the last level. The target availability is chosen to be $a_{\min} = 0.9$, and the LCC is estimated over a 10 years planning horizon. The transportation resources between the facilities consist in trucks having a cost of 1 M€ each, and having variable transportation costs per component (in M€) and inter-facility transportation times (days) uniformly generated in $[0.001, 0.004]$ and $[3, 7]$, respectively. Finally, repairing a component through outsourcing costs 10% of its cost and takes 100 days.

In each simulation experiment, we compute the 95% confidence interval of the expected equipment avalability. The number of samples needed to assess the feasibility of a solution is then determined in such a way that the lower bound of such a confidence interval is greater than or equal to $a_{\min}$.

5.2. Results

In the first part of our experiments, we compare our approach, under different settings, to VM. As pointed out in the previous sections, VM mainly deals with the determination of the spare parts at stock, assuming that the repair decisions are known *a priori*. Moreover, it does not allow any flexibility in the maintenance network, does not deal with outsourcing as a decision variable, does not include the possibility of discarding components, and considers an unlimited capacity for the resources. Thus, in order to allow a fair comparison of VM and our approach, the VM solution concerning spare parts has been embedded into our simulator, in order to determine the type and quantity of resources needed to meet the target availability.

Then, we have considered a number of ANE-based variants, in which we vary the way we obtain the initial solution and/or the policy we use for taking maintenance decisions. The details for each of such variants, along with the relative acronym we use, are reported in Table 1, whereas Table 2 shows the results of our comparisons over 100 runs, reporting the LCC and the expected availability (AVAIL) of the best solution, as well as the average number of MakeFeasible iterations needed to obtain the first feasible solution (MF_ITER), the number of local search restarts (LS), the average number of samples used to declare the feasibility or infeasibility of the solutions generated in the search process (SAMPLES), and the overall number of neighborhood search iterations (ITER).

The data reported in Table 2 show that the best results, in terms of LCC, are obtained for the ANE_VM_LT_FULL variant, where the VM solution is used in order to initialize the procedure, and discarding and outsourcing decisions are allowed. However, it is worth noting that the case in which the initial solution is obtained by assigning zero to all the decision variables, followed by the MakeFeasible procedure, has generated good quality solutions, ensuring an availability level of 0.97, even better than ANE_VM_LT_FULL. In general, not surprisingly, it can be observed that all ANE-based variants outperform VM. This somehow expected behaviour can be explained as follows: VM's decisions are related to spare parts only. Thus, the simulator tends to use a great number of maintenance resources in order to meet the target availability. On the other hand, our approach aims at finding an adequate trade-off between spare parts and maintenance resources, generating thus better quality solutions.

The second part of our experiments concerns the comparison between VM and a restricted number of ANE settings, by using the Taguchi approach [14] which utilizes an orthogonal array to optimize the amount

TABLE 1. Details about initial solutions and maintenance policies for the compared methods.

Approach	Initial solution	Policy description
ANE_VM	VM	Discarding and outsourcing are not allowed. 50% of the failed components are repaired where the failure occurs, and the remaining 50% is sent to the upper echelon.
ANE_VM_LT	VM	Discarding and outsourcing are not allowed. The facility where to repair a failed component is chosen to guarantee the minimum lead time to perform the operation.
ANE_VM_SP	VM	Discarding and outsourcing are not allowed. The facility where to repair a failed component is chosen on the basis of the number of spare parts available.
ANE_VM_LT_FULL	VM	Discarding and outsourcing are allowed. The facility where to repair a failed component is chosen to guarantee the minimum lead time to perform the operation.
ANE_0_LT_FULL	0 is assigned to each decision variable	Discarding and outsourcing are allowed. The facility where to repair a failed component is chosen to guarantee the minimum lead time to perform the operation.

TABLE 2. Experimental results considering different settings (LCC in M€).

Approach	LCC	AVAIL	MF_ITER	LS	SAMPLES	ITER
VM	237 652	0.90				
ANE_VM	181 874	0.91	51.32	9.15	8.31	318.26
ANE_VM_LT	122 124	0.92	118.74	9.02	9.77	274.53
ANE_VM_SP	177 563	0.95	12.12	7.12	9.41	311.82
ANE_VM_LT_FULL	109 965	0.93	31.65	8.15	11.98	267.65
ANE_0_LT_FULL	123 574	0.97	45.74	7.27	10.75	296.87

of information obtained from a limited number of experiments by varying the levels of some key input parameters. In our case, the input parameters that affect the LCC are grouped into sets involving costs, times, and failure rates. To each parameter is assigned either a low (L) or a high (H) randomly generated value. Such values are then combined in several ways generating, thus, 64 test problems. In order to make easier the presentation of these results we then devided the 64 instances into four rows clustered on the basis of two of the most critical parameters: repairing times and failure rates. The four rows correspond to the combinations H-H, H-L, L-H and L-L of such parameters and represent the average values over 14, 20, 14, and 16 instances, respectively, obtained by varying the values of the remaining parameters. The details of this experiment, involving VM, ANE_VM, and ANE_VM_LT_FULL are reported in Table 3. More specifically, for each cluster of test problems Table 3 shows the LCC achieved by the three approaches, as well as the average percentage relative deviation (DEV) of the best solution provided by ANE_VM on VM, and by ANE_VM_LT_FULL on ANE_VM. The availability values are not reported, because the $a_{\min}$ target is always achieved.

TABLE 3. Comparison between VM, ANE_VM, and ANE_VM_LT_FULL (LCC in M€).

Combination of parameters	VM	ANE_VM		ANE_VM_LT_FULL	
	LCC	LCC	DEV (%)	LCC	DEV (%)
H-H	581 021	509 538	−12.88	337 972	−34.32
H-L	343 364	277 657	−19.03	152 866	−40.87
L-H	350 930	236 530	−28.25	154 603	−35.98
L-L	271 277	175 960	−34.34	105 791	−38.16
AVERAGE			−23.63		−37.33

The results of Table 3 confirm the superiority of both our ANE variants with respect to VM. Specifically, ANE_VM produces solutions which on average are about 24% better than VM. Moreover, ANE_VM_LT_FULL clearly outperforms both ANE_VM and VM. In particular, ANE_VM_LT_FULL is, on average, approximately 37% better than ANE_VM. As one may expect, better results are obtained for low values of the critical parameters. Specifically, it is clear in column four how better results are obtained for low values of the repairing time. Similarly, the last column shows slightly better results when low values of failures rates are involved.

6. Conclusions

This paper deals with the problem of designing the logistics support of complex multi-indenture and multi-echelon engineering systems, in order to determine the spare parts stock and the maintenance resources capacity, as well as to perform a level of repair analysis. The goal is to minimize the equipment's expected LCC, while ensuring a target operational availability. The problem has been approached through an optimization *via* simulation approach employing a heuristic procedure that explores the search space through an ANE method, which is based on the estimation *via* simulation of a number of parameters. The experimental results reported in the paper have shown the superiority of our approach with respect to the widely used VM method, with average improvements up to 53%, which could result in very consistent LCC savings and, most importantly, our methods finds an adequate trade-off between spare parts and maintenance resources, as expected.

Possible directions of future research investigations include solving practical real-life applications related to the design of complex engineering systems.

References

[1] P. Alfredsson, Optimization of multi-echelon repairable item inventory systems with simultaneous location of repair facilities. *Eur. J. Oper. Res.* **99** (1997) 584–595.

[2] L.L. Barros, The optimization of repair decisions using life-cycle cost parameters. *IMA J. Math. Appl. Business Ind.* **9** (1998) 403–413.

[3] L.L. Barros and M. Riley, A combinatorial approach to level of repair analysis. *Eur. J. Oper. Res.* **129** (2001) 242–251.

[4] R.J. Basten, J.M.J. Schutten and M.C. van der Heijden, An efficient model formulation for level of repair analysis. *Ann. Oper. Res.* **172** (2009) 119–142.

[5] R.J. Basten, M.C. van der Heijden and J.M.J. Schutten, A minimum cost flow model for level of repair analysis. *Int. J. Prod. Econ.* **133** (2011) 233–242.

[6] R.J. Basten, M.C. van der Heijden and J.M.J. Schutten, Practical extensions to a minimum cost flow model for level of repair analysis. *Eur. J. Oper. Res.* **211** (2011) 333–342.

[7] R.J. Basten, M.C. van der Heijden and J.M.J. Schutten, Joint optimization of level of repair analysis and spare parts stocks. *Eur. J. Oper. Res.* **222** (2012) 474–483.

[8] R.J. Basten, M.C. van der Heijden, J.M.J. Schutten and E. Kutanoglu, An approximate approach for the joint problem of repair analysis and spare parts stocking. *Ann. Oper. Res.* **224** (2015) 121–145.

[9] Department of Defence, USA, *Definitions of terms for reliability and maintainability.* Technical Report MIL-STD-721C (1981).

[10] P. Diaz and M.C. Fu, Models for multi-echelon repairable item inventory systems with limited repair capacity. *Eur. J. Oper. Res.* **97** (1997) 480–492.

[11] I.H. Garbie, An analytical technique to model and assess sustainable development index in manufacturing enterprises. *Int. J. Prod. Res.* **52** (2014) 4876–4915.

[12] G. Ghiani, E. Manni and A. Quaranta, Shift scheduling problem in same-day courrier industry. *Trans. Sci.* **44** (2010) 116–124.

[13] D. Lagana, G. Mirabelli and T. Pizzuti, Adaptation of the multi-layer allocation problem for warehouse layout optimization: A case study. In *Proc. of IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)*. Berlin, Germany (2013) 167–172.

[14] D.C. Montgomery, *Design and analysis of experiments*, 7th edition. John Wiley & Sons (2009).

[15] H. Saranga and U. Dinesh Kumar, Optimization of aircraft maintenance/support infrastructure using genetic algorithms – level of repair analysis. *Ann. Oper. Res.* **143** (2006) 91–106.

[16] C.C. Sherbrooke, Metric: A multi-echelon technique for recoverable item control. *Oper. Res.* **16** (1968) 122–141.

[17] C.C. Sherbrooke, *Optimal Inventory Modeling of Systems: Multi–Echelon Techniques*, 2nd edn. John Wiley& Sons (2004).

[18] A. Sleptchenko, M.C. van der Heijden and A. van Harten, Effects of finite repair capacity in multi-echelon, multi-indenture service part supply systems. *Int. J. Prod. Econ.* **79** (2002) 209–230.

[19] A. Sleptchenko, M.C. van der Heijden and A. van Harten, Trade-off between inventory and repair capacity in spare part networks. *J. Oper. Res. Soc.* **54** (2003) 263–272.

[20] C. Triki, A. Alalawin and G. Ghiani, Optimizing the performance of complex maintenance systems. In *Proc. of the 5th International Conference on Modelling, Simulation and Applied Optimization*. Hammamet, Tunisia (2013) 1–6.

[21] W.H. Zijm and Z.M. Avşar, Capacitated two-indenture models for repairable item systems. *Int. J. Prod. Econ.* **81-82** (2003) 573–588.