Using a Novel Hierarchical Coloured Petri Net to Model and Optimise Fleet Spare Inventory, Cannibalisation and Preventive Maintenance

Jingyu Sheng[1], Darren Prescott[2]

1. 93582 Troops, Chinese Air Force, Shanxi, 038300, China
2. Resilience Engineering Research Group, University of Nottingham, Nottingham, NG7 2RD, UK

Abstract

Spare part availability is crucial to restoring inoperative platforms to the working state. Platforms failing during operation undergo corrective maintenance to replace failed components with spares. To reduce the frequency of this unplanned, corrective maintenance, platforms are inspected periodically and degraded components preventively replaced. Maintenance delays occur when spares are unavailable but cannibalisation can reduce these delays by allowing working components to be removed from inoperative platforms and used to restore other inoperative platforms. Fleets can be deployed across multiple bases that are served by one or more depots. Failed components that cannot be repaired at a base are sent to a depot for repair, along with associated requests for spares, which are satisfied by depot inventories.

The management of fleet corrective and preventive maintenance, cannibalisation, spare inventories, provision of spares to bases and depots, and response of the depot to spare requests is a complex problem for fleet maintenance managers and critical to ensuring acceptable fleet performance. This paper presents a novel hierarchical coloured Petri net (HCPN) model of a fleet spare inventory system, which accounts for these issues alongside fleet deployment and mission-oriented operation. The application of the model is demonstrated using case studies of two example fleets.

Keywords: Inventory, Fleet, Maintenance, Cannibalisation, Hierarchical Coloured Petri Net.

1 Introduction

Spare part supply is crucial to ensuring that non-mission capable (NMC) platforms can be maintained as required in order to meet fleet operational requirements. Critical component failures during operation lead platforms to be immediately removed from service for corrective maintenance to replace failed components by spares. Spare part unavailability can lead to mission cancellations, unwanted platform downtime and their associated costs but might be unavoidable, particularly if the cost of spares is high and maintenance budgets limited. It is therefore difficult for fleet managers to determine spare part inventory levels that maintain fleet performance at an acceptable cost.

Critical component failures on platforms can lead not only to unacceptable downtime or mission cancellations, but also to potentially catastrophic loss of platforms. Therefore, fleet maintenance crews perform preventive maintenance, periodically inspecting mission capable (MC) platforms and replacing any components that have degraded to an unacceptable condition. This condition-based maintenance (CBM) places new demands on spare supply; the platform inspection interval can adversely impact spare inventories and fleet performance if not properly determined. Frequent inspections may allow for quick identification and replacement of degraded components but this may place undue strain on the spare part inventory and negatively impact fleet performance. By contrast, less frequent inspections may allow greater fleet performance but may increase the risk of component [illegible] maintenance. When a [illegible] l preventive inspection to be performed alongside the required corrective maintenance. This opportunistic maintenance can help to reduce platform downtime and interruptions to the fleet operation schedule.

Fleet maintenance is often organised into three levels: organisation (O-), intermediate (I-) and depot (D-) level. NMC platforms are maintained at O-level, where failed components are replaced by spares or cannibalisation. Cannibalisation involves removing serviceable parts from a NMC platform and using them to repair other NMC platforms in the fleet when the required spares are unavailable. It can maintain fleet performance and reduce spare part demands by allowing the consolidation of spare shortages into a small number of NMC platforms. In some military organisations, cannibalisation is used as a regular supply method to reduce the maintenance turnaround for NMC platforms when spares are unavailable. For example, from 1996 to 2001, over 850000 cannibalisations are reported to have been performed in the U.S. Air Force and Navy [1].

Fleets may be deployed in multiple bases that are served by one or more depots. Bases and depots can have their own spare inventories and bases can have their own O-level and I-level maintenance organisations. Removed, failed components are sent for repair at I-level with any that are non-repairable at this I-level sent on to D-level for further repair. As a component that cannot be repaired at I-level is sent from a base to the depot for repair, a spare request from the base is also placed at the depot, which responds by sending back a spare of the required type. Once repaired, components are kept in stock at the depot and used to satisfy base spare requests. Operational requirements and sub-fleet (platforms deployed at a base) sizes may vary across bases so spares provision to bases and depots and the response discipline of depot spare inventories to conflicting base spare requests are important management problems in fleet spare inventory control.

The literature contains a number of fleet spare inventory management models. The most commonly used is the Multi-Echelon Technique for Recoverable Item Control (METRIC) model [2], an analytical model designed to optimise spare stock levels for fleets with a two-echelon supply system (base and depot). Spare demand follows a Possion distribution and the model presents a formula for the total spare shortages in the fleet. The MOD-METRIC model [3] extends METRIC by introducing two-indenture components (line- (LRU) and shop-replaceable units (SRU)). MOD-METRIC was implemented by the U.S. Air Force to determine F-15 weapon system stock levels. The original METRIC model was found to underestimate the expected number of spare shortages [4]. The VARI-METRIC model [4] attempts to address this for a two-echelon supply system, using a negative binomial distribution to describe base spare shortages. Sherbrooke [5] further extends the VARI-METRIC model to include two-indenture components.

Spare demands are often assumed to follow homogeneous Poisson processes in the METRIC models, making them independent of the number of operating platforms and the fleet's operational requirements. This assumption may not accurately describe the real demand in aircraft fleets, particularly in theatre. To overcome this, Lau et al. [6] build an analytical model to relate demand to the time-varying number of operating platforms for a two-echelon supply system. The model calculates fleet performance by computing the expected spare shortages and operational availability under spare provision scenarios. Wang and Ma [7] present an analytical model to estimate the operational availability of a fleet with dynamic missions. Hillestad [8] introduces the analytical Dyna-METRIC model, which includes time-dependent operational demands, for the U.S. Air Force.

It is a limitation of these analytical methods that maintenance policies such as preventive maintenance are ignored. A simulation modelling approach to spare inventory management allows the analysis of more operation and maintenance factors. Dyna-METRIC Version 5 [9] uses Monte Carlo sampling to accurately represent the uncertainties in spare demand and repair processes for aircraft fleets during wartime. However, unlike analytical models, simulation models cannot be used to directly compute optimal fleet spare provision scenarios. Finding optimal spare inventory levels would require a simulation of every possible spare provision scenario. This is achievable for finite research spaces that consist of only a relatively small number of scenarios. However, it is impractical or impossible if there are a large, potentially infinite, number of scenarios. In this case, metaheuristic algorithms such as a Genetic Algorithm (GA) [10-12], Scatter Search [13] and Simulated Annealing [14] can be used to find near-optimal scenarios of fleet spare provision based on the simulation models.

Marseguerra et al. [10] use a GA-based simulation approach to determine initial levels of spare component provision for a single manufacturing plant and maximise net profit. This work is extended [11] through the use of a multi-objective GA (MOGA) to simultaneously maximise net profit and minimise the total volume of spare components. The models in both papers are designed not for a fleet but for a single platform. Considering preventive maintenance, Ilgin and Tunali [12] use a single-objective GA to jointly optimise spare part inventories and platform periodic maintenance intervals for a fleet of manufacturing machines. Chen et al. [13] use Scatter Search to jointly optimise a component age-based preventive maintenance policy and spare provision in a multi-echelon supply network. Alrabghi et al. [14] study the joint simulation optimisation problem of component preventive maintenance and spare provisioning for a single multi-component manufacturing system.

There is little literature considering cannibalisation in spare inventory management models. The simulation-based Dyna-METRIC Version 6 [15] models cannibalisation but not spare inventory optimisation. To evaluate the effect of cannibalisation on fleet performance, Ormon and Cassady [16] and Salman et al. [17] build discrete event simulation models of fleet cannibalisation processes but the fleet is assumed to be deployed at one base and perform non-stop missions. Sheng and Prescott [18] build a HCPN model of a three-level fleet maintenance process considering cannibalisation and spares but only a single operational base is modelled. None of these simulation models includes preventive maintenance.

Existing fleet spare inventory models consider only limited aspects of fleet maintenance, rarely including factors such as mission-oriented fleet operation, preventive maintenance or cannibalisation, all of which can significantly affect demand for spares. Therefore, to support fleet managers in efficiently managing spare inventories, a comprehensive fleet spare inventory model is required, which accounts for factors related to fleet operation, maintenance and the provision of spares. A novel hierarchical coloured Petri net (HCPN) model is proposed in this paper to represent the spare inventory system of fleets with multiple operational bases, mission-oriented operation, three-level maintenance, preventive checks, opportunistic maintenance, repair and cannibalisation. The model is applied to two fleets, deployed at single and multiple bases, to investigate inventory management problems including: the provision of spares to inventories and the depot response to base spare requests. The joint optimisation of spare stock levels and the platform inspection interval is also performed using a GA.

2 Petri Nets

First introduced by Carl Adam Petri, Petri nets (PN) are powerful, graphical and mathematical tools for modelling complex, dynamic systems [21]. The reader is referred to [25] for a thorough review but the main concepts are introduced here for clarity. A PN is a directed graph with two types of nodes: *places*, drawn as circles; and *transitions*, drawn as bars, with nodes of different types connected by directed *arcs*. Places contain a discrete number of *tokens* whose distribution within the PN defines its *marking* and hence the system state. Transitions govern the dynamic behaviour of the PN, changing the system state when they *fire*. To fire, a transition is first *enabled*, meaning the number of tokens in each of its input places is no less than the weight of the arc linking it to the transition. Transitions can be *immediate*, drawn as solid bars, or *timed*, drawn as hollow bars. Immediate transitions fire as soon as they are enabled and timed transitions fire after a delay has elapsed following their becoming enabled. This firing delay can be either fixed or sampled from a known probability distribution.

Figure 1 shows a PN with an enabled, timed transition, and the marking that results once it fires after a firing delay *t*. Figure 1 also shows an *Inhibitor* arc, with a circle at its head, which would prevent the transition from being enabled if the number of tokens in the associated place were no less than the weight of the associated arc.

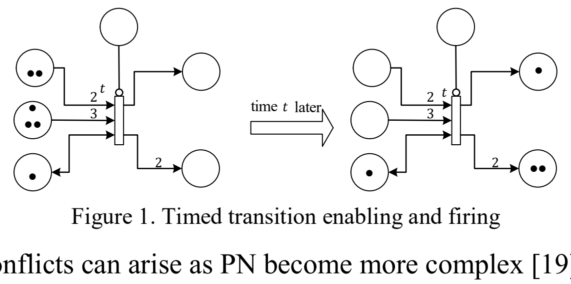

Figure 1. Timed transition enabling and firing

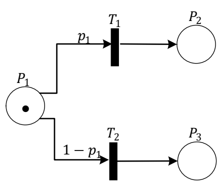

Figure 2. Conflicting transitions fire with probabilities

Conflicts can arise as PN become more complex [19]. For example:

- Two or more transitions with the same delay (maybe immediate) are enabled by the same input place;
- An enabled timed transition is disabled while waiting for its delay time to pass.

In the first case, if the number of tokens in the input place exceeds the sum of the weights of all arcs from it, the enabled transitions will fire simultaneously. If not, a probability check can be used to resolve the conflict and set the firing order [20]. Figure 2 shows two immediate transitions, T_1 and T_2, enabled by place P_1. T_1 fires first with probability p_1 if both T_1 and T_2 are enabled. As P_1 is marked, a uniform random variable p is sampled between 0 and 1. If $p \leq p_1$, T_1 fires, removing the token from P_1 and disabling T_2. Otherwise, the firing sequence is reversed.

The second case can be resolved using 'aging' tokens [19], where each token in an input place is assigned a 'counter' initially set to the firing delay of the related transition. The counter represents the remaining time that a token will be held by the transition's input place and decreases continuously while the transition is enabled, a process that stops when the transition is disabled. Aging tokens are useful for modelling platform operation processes. Using a timed transition to model a component's failure process allows a token counter to represent the component's remaining working life (time to failure), which decreases while the component is in use but remains static when it is not. The timed transition fires when the counter drops to 0, indicating component failure.

Coloured Petri nets (CPN) [21] extend PN to allow the concise modelling of the behaviour of complex systems such as concurrent systems by introducing:

- *token colour*: variables representing different data types, such as Boolean, integer or character strings;
- *place colour set*: all possible token colours of a specified type within a place;
- *arc expression*: the amount of tokens of specific colours that are removed from or added to the place linked by the related arc when a transition fires;
- *transition guard*: a Boolean expression representing a constraint on the transition enabling policy.

A CPN transition is enabled only if its input place markings satisfy their associated input arc expressions and its guard evaluates to be true. Tokens of different colours can fire in different ways. Figure 3 shows a CPN transition being enabled and firing. Each place has colour set, $CS = \{red, blue\}$, which holds two colour variables, cs_1 and cs_2. Arc expressions cs_1 and cs_2 represent one specifically coloured token from CS, while cs_1+cs_2 represents two coloured tokens. The transition guard, cs_1=cs_2, enables the transition if its input places contain tokens of the same colour. If the delay times for the blue and red tokens are t_1 and t_2 (t_1<t_2) respectively then the blue tokens switch to the output place at time t_1 and after a further delay time t_2-t_1 the red tokens switch to the output place.

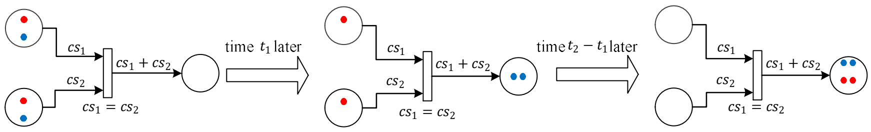

Figure 3. Transition enabling and firing in an example coloured Petri net

Hierarchical CPN (HCPN) are CPN that are organised as sets of modules, which can be non-hierarchical CPN or HCPN with their own sub-modules. Modules and sub-modules are related by *substitution transitions*, drawn here as rectangular boxes, which represent an abstract view of a sub-module to its parent module. Sub-modules give a detailed view of the functions of the related substitution transitions. HCPN bring a number of clear benefits [21]:

1. For complex systems single non-hierarchical nets can be unmanageable. Modification simplifies model amendment since new modules may be easily added and changes limited to only the necessary modules;
2. CPN modules behave as black boxes, allowing modellers to work at different levels of abstraction while disregarding higher or lower levels;
3. Modules of repeated system features, such as identical platform maintenance processes performed at different bases, may be defined once and used repeatedly, avoiding unnecessary module duplication and reducing the modelling workload when module changes are needed. A repeatedly-used module is a *parameter module*, with each replication being an independent *module instance*, each of which has its own marking.

The HCPN method is selected to model fleet spare inventories since it can provide a graphical, precise and concise representation of the operation and maintenance processes of large, complex fleets. Using token colours ensures that the size of the model will not increase with the size of the modelled fleet, since any number of platforms or components can be represented by an equivalent number of differently-coloured tokens within a single place. The HCPN model's modular structure means it can be easily modified and extended in future if further fleet operational or maintenance activities must be considered.

3 HCPN Model of Fleet Spare Inventory System

In this section an HCPN model of a fleet spare inventory system is presented, accounting for a variety of operational and maintenance factors including multiple bases, mission-oriented operation, preventive inspection, opportunistic maintenance, repair and cannibalisation. The model is designed for military aircraft fleets, but could be applied to other, similar fleets with similar maintenance processes and spare supply systems with little or no amendment. Abbreviations used for place names are given in Appendix 2.

3.1 Fleet Description

A fleet is assumed to consist of N_P identical and independent platforms which are deployed in N_B different bases and served by a single depot. The fleet is required to perform a number of missions each day. Platforms have two states: MC and NMC. Each MC platform is removed from service for preventive inspection after continuously performing a fixed number of missions, given for a platform p by $NM_{PC}(p)$, which may differ between bases and platforms. N_{bi} is the number of platforms deployed at base bi ($1 \leq bi \leq N_B$). A platform p at base bi is represented by (bi,pi) where pi is the index of the platform at the base and the check state of platform p is (p,wh) where wh represents the number of missions it must still perform before its next preventive inspection.

Each platform in the fleet can be thought of as a system containing a number of different repairable components, each of which are assumed to have three states, representing various states of degradation: working, PM due and failed. Components in the working state are fully operational and need no maintenance, while failed components can no longer perform their designated functions and need immediate replacement. In the PM due state the component is still operational but must be preventively replaced to avoid failure. A component's transition time between states follows a known independent distribution. A component c of type ct in platform p is represented by (p,ct,a) where a ($a \geq 0$) is its expected working life (in working hours), which is initialised by sampling from a known distribution and decreases continuously while component c is operational. A spare of type ct is represented by $(0,ct,a)$.

After performing a fixed number of missions, a MC platform joins the platform check queue (PCQ) where its components are inspected to see if any are failed or in the PM due state. If they are, the platform immediately joins the PMQ (platform maintenance queue) where its inoperative (failed or PM due) components are replaced by spares or cannibalisation. Otherwise, the platform successfully passes the check and returns to service. If opportunistic maintenance (OM) is allowed, a platform that fails during operation before its next scheduled check joins the PCQ where its components are inspected to see if any are in the PM due state. If OM is not allowed, the platform immediately joins the PMQ and only its failed components are replaced.

Each base has its own O- and I-level maintenance organisations. The O-level organisation restores NMC platforms to the MC state by replacing their failed components by spares or cannibalisation, which is performed when the required spares are unavailable. The maintenance of a NMC platform only begins if resources are available to replace all of its failed components. When a NMC platform joins the PMQ, maintenance crews remove its failed components and send them to I-level for repair, checking whether full restoration to the MC state is possible using existing spare and cannibalisation resources. If not, the platform joins the PWMQ (platform waiting maintenance queue) to wait for resources to become available. Otherwise, maintenance crews obtain the required parts from stock and cannibalisation before restoring the platform to a MC state.

Failed components are assumed to be repairable at I-level with a known probability. Components that are non-repairable at the base are sent to the depot for repair, along with associated requests for spares from the depot, which responds based on a one-for-one replenishment policy. If a spare is available from the depot inventory, one spare of the required type is transported to the base and the received, failed item stocked in the depot once repaired. All components are assumed to be repairable at the depot and both I-level and depot repairs are assumed to be perfect, returning failed components to the 'as good as new' state so they can be treated as brand new spares.

3.2 HCPN Model Hierarchy

Figure 4 shows the HCPN model, which consists of 21 modules, organised into 6 levels. Modules in the first two levels provide the main interfaces to model users, who can study their own fleets by setting numbers of bases and depots in the master module; base sub-fleet parameters including operational requirements, sub-fleet size, platform structure, cannibalisation policy, preventive inspection interval and spare stock levels in the base module; and depot spare inventories in the depot module. The model outputs to the base module the data needed for the calculation of fleet performance measures, which can be used by fleet managers to inform spare inventory related decisions. Tables 1 and 2 define the colour sets and colour variables applied in these modules.

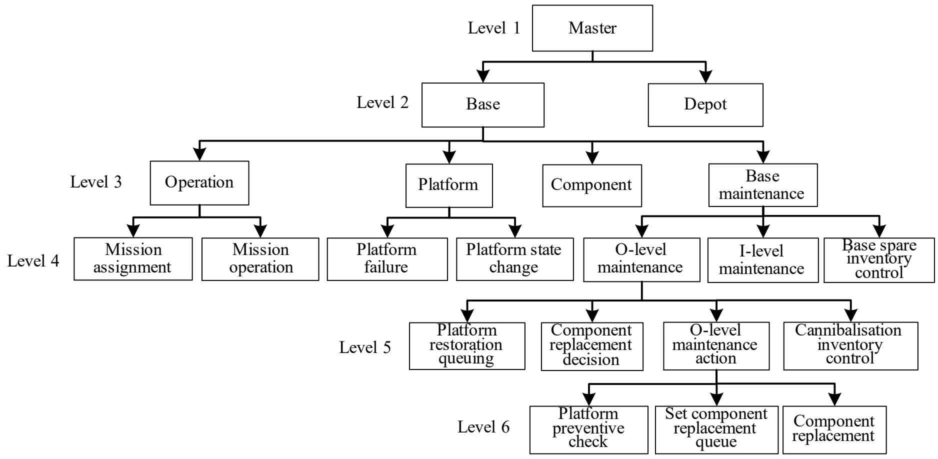

Figure 4. Model hierarchy

Table 1. Colour sets

Colour set	Colours	Meaning
BI	$\{1, ..., B\}$	Base index
PI	$\{1, .., N_P\}$	Platform index
P	$\{(bi, pi)\}$	Platform
CT	$\{1,2,\ldots,N_C\}$	Component type
PCS	$\{(p, wh)\}$	Platform check state
C	$\{(p, ct, a)\}$	Components: $a \geq 0$ is the expected working life of a component; $p=0$ signifies a spare component.
CRD	$\{(c, c_1)\}$ $\forall c, c_1 \in C$, $c(ct)=c_1(ct)$, $c(p)\neq 0$ and $c(p)\neq c_1(p)$	Component replacement decision: it is decided to replace the failed component c in platform $c(p)$ by component c_1. $c_1(p)=0$ means c will be replaced by a new spare component. $c_1(p)\neq 0$ means that the working component c_1 is cannibalised from platform $c_1(p)$.
M	$\{1,2,\ldots\}$	Missions
PM	$\{(p, m)\}$ $\forall p \in P$, $\forall m \in M$	Platform p is assigned to perform mission m
AS	(bi, c)	Spare components assigned to base *bi* by the depot

Table 2. Colour variables

Variable	Colour	Meaning
p, p_1	*P*	A platform
$c, c_1, \ldots, c_n$	*C*	A component
crd	*CRD*	A component replacement decision
ps	*PCS*	The check state of a platform
$ps(p)$	*P*	Platform with $ps(wh)$ missions to perform before next check
$c(p)$	*P*	Platform to which component c is fitted
$pm, pm_1, \ldots, pm_{Nm}$	*PM*	A platform assignment decision
$pm(p)$	*P*	A platform which is performing a mission
m	*M*	A mission
$pm(m), pm_1(m), \ldots, pm_{Nm}(m)$	*M*	A mission that is being performed
$c(ct)$	*CT*	The type of component c
as	*AS*	An assigned spare component

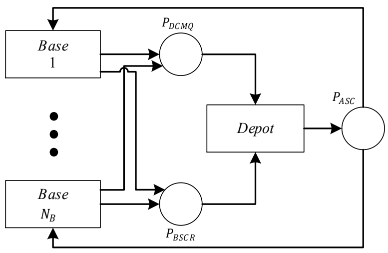

Figure 5. Master module

3.3 Master Module

The master module, shown in Figure 5, provides an abstract view of a fleet spare inventory system with multiple operational bases and a single depot. P_{BSCR} models a base spare request made when a failed component is non-repairable and P_{DCMQ} models the base sending the failed component to the depot. P_{ASC} models the depot assigning spares to the appropriate bases according to the one-for-one replenishment policy.

This module specifies the numbers of bases and depots in the modelled fleet, and hence the number of base and depot module instances required. If there were more depots, the spare supply relationship would be modelled by the exchange of tokens between the associated base and depot module instances through the related place instances of P_{BSCR}, P_{DCMQ} and P_{ASC}.

3.4 Base Module

The base module, shown in Figure 6, models the operations and maintenance performed at a base. It consists of 4 sub-modules:

- Operation module: models the operation of platforms that are deployed at a base;
- Platform module: models platform failures and changes of platform state;
- Component module: models degradation of multi-state components (working, PM due or failed);
- Base maintenance module: models corrective and preventive maintenance of NMC platforms and the repair at a base of PM due and failed components.

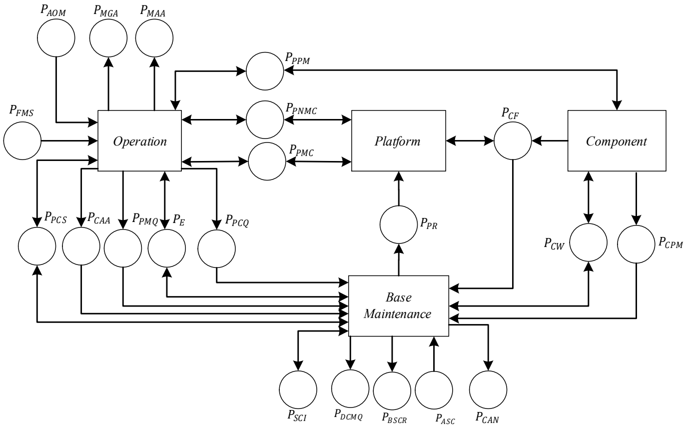

Figure 6. Base module

The operation module outputs platforms starting missions to P_{PPM}, which activates the failure processes of their working components (tokens in P_{CW}) in the component module. During operation, the component module outputs PM-due and failed components to P_{CPM} and P_{CF}. If the platform module finds that a component failure causes a platform failure it moves the failed platform to P_{PNMC}, which triggers the operation module to immediately remove the platform from service and send it to the base maintenance module through P_{PMQ}. The operation module uses P_{PCS} to monitor the operational state of MC platforms and sends platforms needing preventive checks to the base maintenance module through P_{PCQ}.

NMC platforms are maintained in the base maintenance module where their failed components are replaced by spares (tokens in P_{SCI}) or cannibalised components (tokens in P_{CW}). Through P_{PR}, the base maintenance module outputs restored platforms to the platform module, which changes their state from NMC to MC, meaning they can be assigned to scheduled missions by the operation module again. Once removed, failed components that are found to be non-repairable in the base maintenance module go through P_{DCMQ} to the depot for repair, with associated spare requests from the base placed in the depot spare inventories through P_{BSCR}.

3.4.1 Operation HCPN

The operation module, shown in Figure 7, models the mission-oriented operation of a base's platforms through:

- Mission assignment (MA) sub-module: assigns idle, MC platforms to missions at a base;
- Mission operation (MO) sub-module: models the detailed operational processes of assigned platforms.

Tokens in P_{FMS} represent the missions that must be performed by platforms at a base. When a mission is called, the MA sub-module assigns MC platforms to perform it if there are sufficient idle, MC platforms (represented by tokens in P_{POG} and P_{PMC}) available by outputting the assigned platforms to the MO sub-module through P_{PAM}. If not, it is aborted with a token placed in P_{MGA} by the MA sub-module.

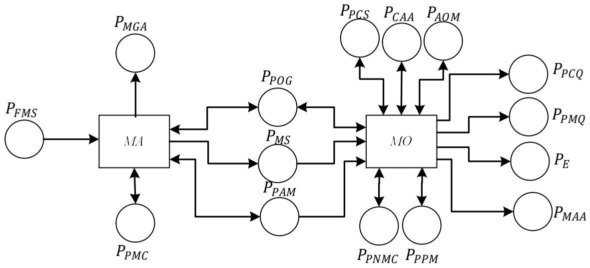

Figure 7. Operation module

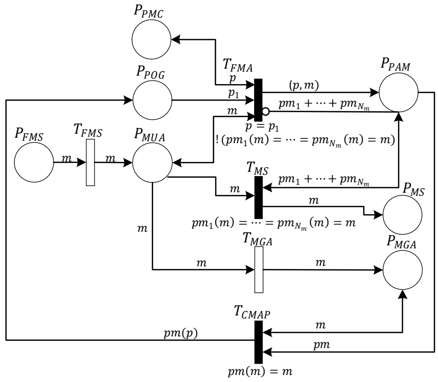

Figure 8. Mission assignment sub-module

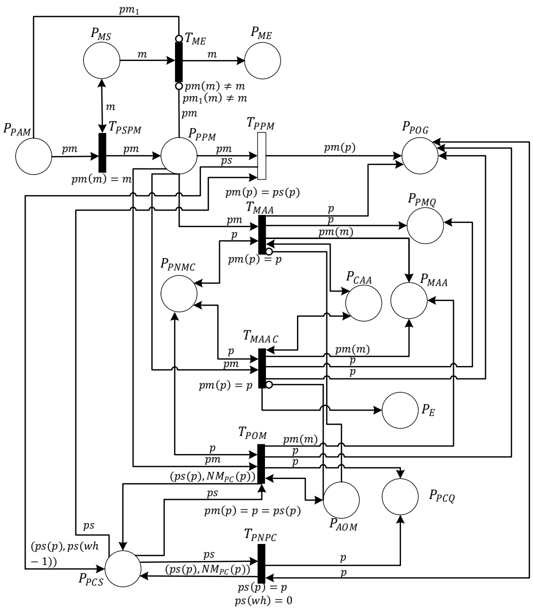

Figure 9. Mission operation sub-module

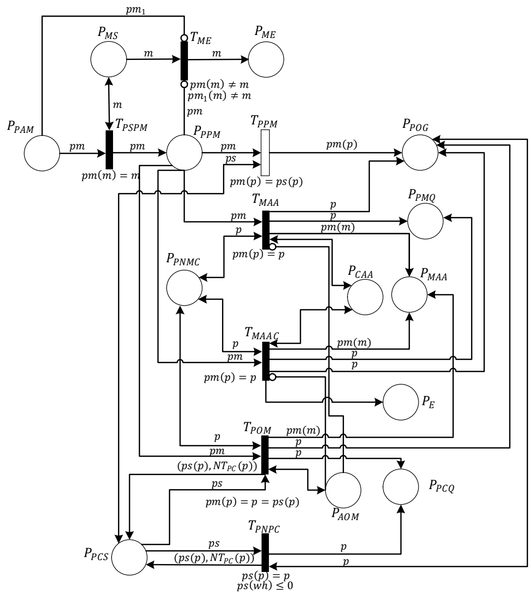

Figure 10. Mission operation sub-module under operating-hour based platform checks

The MO sub-module models platform performance during mission operation. If a platform becomes NMC during a mission (P_{PNMC} is marked), a mission air abort event occurs and a token is output to P_{MAA}. The NMC platform is immediately taken out of service and moved to P_{PMQ}, the platform maintenance queue. Simultaneously, a token is created in P_E if cannibalisation is allowed at the base (P_{CAA} is marked), since the NMC platform brings new cannibalisation resources. A MC platform is preventively checked following the completion of a set number of missions, with the MO sub-module moving it to P_{PCQ}, the platform check queue. If OM is allowed (P_{AOM} is marked), instead of P_{PMQ}, NMC platforms are added to P_{PCQ} where their PM-due components are replaced alongside the failed components.

Figure 8 shows the MA sub-module. T_{FMS} sets the scheduled mission start times given by the firing delays of tokens in P_{FMS} and adds tokens representing mission m to P_{MUA}. T_{FMA} assigns MC platforms to the scheduled mission, where N_m is the number of MC platforms required by mission m. If there are more than N_m idle, MC platforms when mission m is called, the fleet manager may apply a selection discipline in order to select N_m platforms. After N_m platforms have been assigned to mission m by T_{FMA}, T_{FMA} is inhibited and T_{MS} is enabled, immediately moving the token with colour m from P_{MUA} to P_{MS}, starting mission m. A mission must often start within a certain time of being scheduled and may be cancelled if insufficient platforms become available in this time. If this happens, T_{MGA} moves the token from P_{MUS} to P_{MGA}, indicating a mission ground abort event. T_{CMAP} then returns all platforms pre-assigned to the aborted mission to P_{POG} (the idle state) to await future missions.

Figure 9 shows the MO sub-module. When mission m starts (a token with colour m marks P_{MS}), transition T_{PSPM} immediately moves the platforms assigned to the mission from P_{PAM} to P_{PPM}, hence enabling T_{PPM}, which assigns a delay time representing the platforms' mission operation times to each token in P_{PPM}; this time may be fixed or randomly sampled from a distribution.

If OM is allowed (P_{AOM} is marked) and platform p becomes NMC during operation (P_{PNMC} marked by a token with colour p), T_{MAA} and T_{MAAC} are disabled and T_{POM} enabled. T_{POM} removes p from service (removing a token with colour pm ($pm(p)=p$) from P_{PPM}), adds p to the PCQ (creates a token with colour p in P_{PCQ}), records a mission air abort (places a token with colour $m=pm(m)$ in P_{MAA}) and refreshes the check state of p (changes the colour variable $ps(wh)$ of the related token in P_{PCS} to $NM_{PC}(p)$). If OM is not allowed (P_{AOM} is unmarked) when a platform fails, either T_{MAA} or T_{MAAC} is enabled depending on the cannibalisation policy. If no cannibalisation is allowed, T_{MAA} is enabled; this immediately removes the token with colour pm from P_{PPM}, creates tokens with colour $p=pm(p)$ in both P_{PMQ} and P_{POG} and places a token with colour $m=pm(m)$ in P_{MAA}, signifying the occurrence of an air abort event during mission m. If cannibalisation is allowed, T_{MAAC} is enabled and fires with the same result as T_{MAA} but for the addition of a token to P_E after T_{MAAC} fires, signifying that a new cannibalisation source is available.

The check state of platform p is represented by a token with colour ps ($ps(p)=p$) in P_{PCS}, where $ps(wh)$ represents the number of missions, initially $NM_{PC}(p)$, still to be performed by platform p before its next scheduled inspection. T_{PPM} updates the check state of platform p by decreasing $ps(wh)$ by 1 every time it fires and absorbs a token with colour pm ($pm(p)=p$) from P_{PPM}. When $NM_{PC}(p)$ missions have been performed without failure ($ps(wh)$ decreases to 0), T_{PNPC} is enabled and fires immediately, changing the check state $ps(wh)$ of platform p to $NM_{PC}(p)$ and adding p to the PCQ (creating a token with colour p in P_{PCQ}), signifying the inspection of platform p begins.

Figure 10 shows a variation of the MO sub-module that can be used if platform preventive inspection intervals are specified in terms of operating hours rather than number of missions. Differences include the modification of the two arcs between T_{PPM} and P_{PCS} to a double-headed arc with expression ps and the changing of check state wh of platform p to represent the remaining operating hours before its next check. Using the concept of aging tokens introduced in Section 2, the platform check state $ps(wh)$ continuously decreases while token ps in P_{PCS} enables T_{PPM} and stops when T_{PPM} is disabled. $NT_{PC}(p)$ is the inspection interval (in number of operating hours) for platform p. T_{PNPC} is enabled and fires immediately when the check state (wh) of platform p becomes non-positive (P_{PCS} contains a token with colour ps where $ps(p)=p$ and $ps(wh)\leq 0$) and p is not performing a mission (P_{POG} contains a token with colour p). After T_{PNPC} fires, the check state wh of platform p is changed to $NT_{PC}(p)$.

3.4.2 Platform HCPN

The platform module, shown in Figure 11, controls platform failures and changes of platform state, modelled respectively by the platform failure (PF) and platform state change (PSC) sub-modules. When component failures (tokens in P_{CF}) cause a platform failure, the PF sub-module outputs that platform to P_{PF}, activating the PSC sub-module, which immediately moves the platform from P_{PMC} to P_{PNMC}, changing its state from MC to NMC. After a platform is restored, the platform is placed in P_{PR} and the PSC sub-module changes the platform state back to MC.

A MC platform undergoing a check cannot perform missions and remains NMC in the PSC sub-module until the check ends. If the check is passed and no degraded component found, the platform is added to P_{PPC} and reverts to the MC state in the PSC sub-module. Otherwise, the platform joins the PMQ to await maintenance.

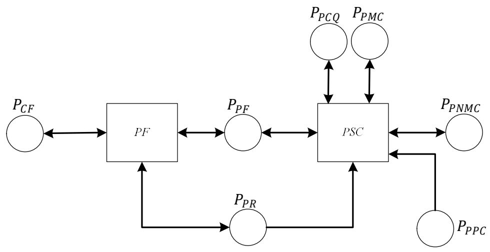

Figure 11. Platform module

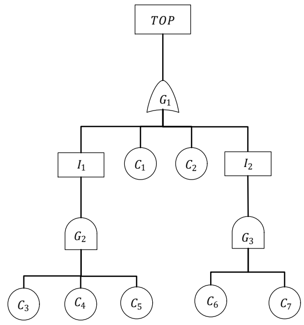

Figure 12. Example platform fault tree

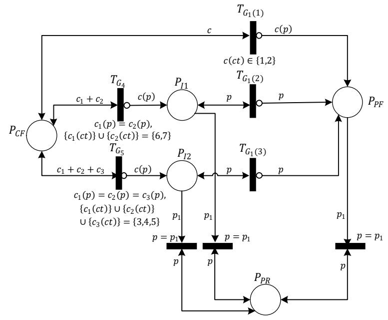

Figure 13. Example platform failure sub-module

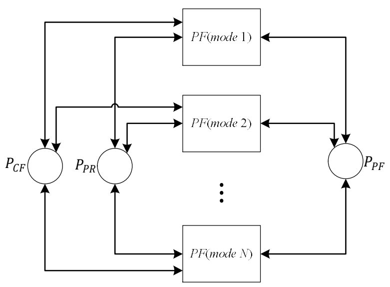

Figure 14. Platform failure sub-module with multiple failure modes

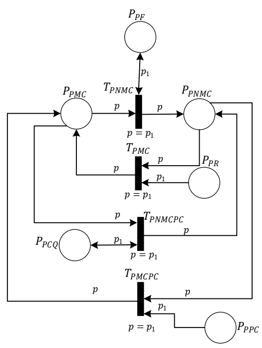

Figure 15. Platform state change sub-module

The PF sub-module models the platform failure process, which can be modelled using a fault tree (FT). The FT top event represents platform failure and basic events represent component failures. An example with seven basic events, C_1-C_7, three logic gates, G_1-G_3, and two intermediate events, I_1 and I_2, is shown in Figure 12. The PF sub-module is constructed by converting a FT to a CPN [18]:

- Basic events are represented by a single place;
- Each top event and intermediate event is represented by a place;
- An AND gate is converted to an immediate transition;
- An OR gate is converted to a number of immediate transitions equal to the number of converted places of the gate's input events.

Figure 13 shows the CPN that is equivalent to the example FT. A token of colour p is added to P_{PF} when component failures make platform p NMC. Restoring platform p to a MC state by marking P_{PR} with a token of colour p clears the places relating to the FT intermediate events and top event. Complex platforms, such as aircraft and ships, have multiple failure modes, each modelled by a FT with a related CPN. Each of these CPNs becomes a substitution transition when constructing the PF sub-module as shown in Figure 14.

The PSC sub-module, shown in Figure 15, models platform state changes. When platform p fails (a token with colour p in P_{PF}), T_{PNMC} immediately moves it from P_{PMC} to P_{PNMC}, meaning that MC platform p becomes NMC and needs maintenance. P_{PR} is marked by a token with colour p when the NMC platform p is restored to an operational MC state, enabling T_{PR} which immediately removes tokens with colour p from P_{PR} and P_{PNMC} and creates a token with colour p in P_{PMC}. T_{PNMCPC} and T_{PMCPC} control changes of platform state in the PCQ. When a MC platform p joins the PCQ (both P_{PCQ} and P_{PMC} contain a token with colour p), T_{PNMCPC} immediately moves the token with colour p from P_{PMC} to P_{PNMC}. If all components of platform p are found to be in the working state, T_{PMCPC} immediately moves the token with colour p from P_{PNMC} to P_{PMC}, indicating that platform p passes the check and is MC.

3.4.3 Component HCPN

The component module, shown in Figure 16, models failures of multi-state components, which in this paper are defined as those that can exist in a working state, a PM due state or a failed state.

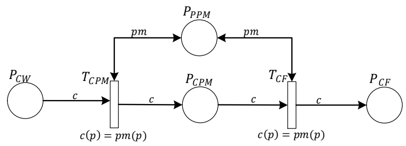

Figure 16. Component module

Each state, working, PM due and failed, is assigned its own place (P_{CW}, P_{CPM} and P_{CF} respectively) and the timed transitions T_{CPM} and T_{CF} model the component failure process. The two component failure stages are modelled independently. The remaining working life, $c(a)$, of a component c (represented by a token with colour c in P_{CW}) is initialised by sampling from a known probability distribution which models the time taken to degrade to the PM due state. T_{CPM} fires when $c(a)$ reaches 0, moving token c from P_{CW} to P_{CPM} and resetting $c(a)$ to a time sampled from the distribution which models the time taken to degrade from the PM due state to the failed state.

Though the component module is built for three-state components, it can be used to model the failures of binary-state components without modification by setting the firing delay of the T_{CF} to 0, meaning the failure times of two-state components are represented by the firing delays of T_{CPM}.

3.4.4 Base Maintenance HCPN

The base maintenance module, shown in Figure 17, models detailed O- and I-level maintenance actions that are performed at a base including: platform periodic checks, restoration of NMC platforms and repair of failed and PM due components. The base maintenance module consists of three sub-modules: O-level maintenance (OLM), I-level maintenance (ILM) and base spare inventory control (BSIC), which model corrective and preventive maintenance of NMC platforms performed at the base's O-level maintenance organisation, component repair performed at the base's I-level maintenance organisation, and the management of the base's spare inventories.

When a NMC platform is maintained in the OLM sub-module, its failed and PM due components are replaced by spares or cannibalised components. Restored platforms appear as tokens in P_{PR} and removed, inoperative components initially go through P_{ICMQ} to the ILM sub-module, where repairs to the 'as good as new' state are attempted at I-level. Through P_{IRC}, the ILM sub-module outputs repaired components to the BSIC sub-module where they are added to spare inventories. When a component is non-repairable at a base's I-level, the ILM sub-module sends it to the depot through P_{DCMQ} and requests a spare of the same type from the depot inventories.

Spare requests from a base enter through P_{BSCR} into the depot module, which outputs the requested spare components from all bases to P_{ASC}. The BSIC sub-module identifies and transports spares that are assigned by the depot to a specific base, and adds them to the base's spare inventories as they arrive. The base index is declared by the colour of any token in P_{PCS}.

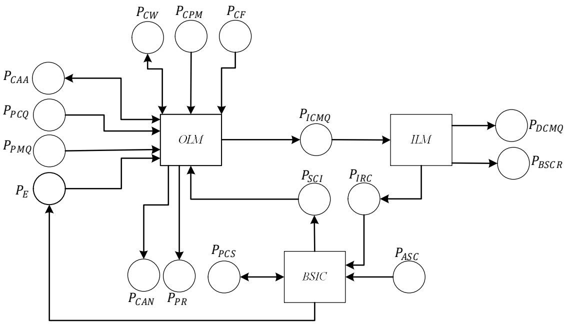

Figure 17. Base maintenance module

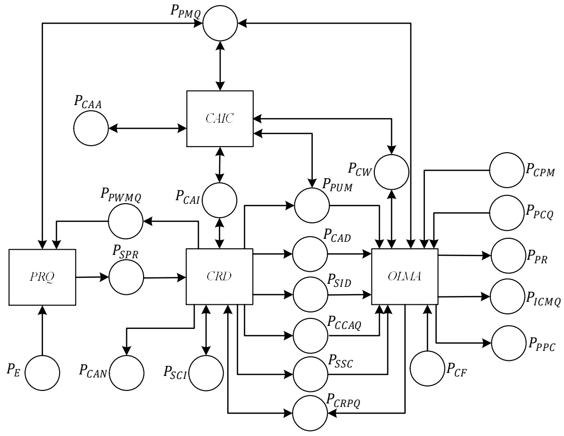

Figure 18. O-level maintenance module

3.4.4.1 O-level Maintenance

The OLM module, shown in Figure 18, is comprised of 4 sub-modules, shown in Figures 19-22:

- Platform restoration queuing (PRQ): schedules the restoration of NMC platforms in the PMQ;
- Cannibalisation inventory control (CAIC): manages available cannibalisation resources;
- Component replacement decision (CRD): identifies NMC platforms which can be fully restored to the MC state by existing spare and cannibalisation resources, and selects spares and cannibalisation resources to complete a platform restoration;
- O-level maintenance action (OLMA): carries out component replacement decisions and preventive checks.

The PRQ sub-module (Figure 19) schedules the maintenance of NMC platforms in the PMQ (tokens in P_{PMQ}). Through P_{SPR}, the PRQ sub-module outputs the platform with the highest maintenance priority to the CRD sub-module (Figure 21), which checks whether it can be fully restored using spares or by cannibalisation. If so, the platform moves to P_{PUM} and the necessary components are replaced in the OLMA sub-module according to the spare installation and cannibalisation decisions (represented by tokens in P_{SLID} and P_{LCAD}) from the CRD sub-module. The selected spares and cannibalised components are identified in P_{SSC} and P_{CCAQ}. The OLMA sub-module sends removed, failed components to P_{ICMQ} for I-level repair and outputs the platform to P_{PR} once all the necessary components have been replaced. Otherwise, if there are insufficient resources to fully restore the platform, the CRD sub-module puts it in P_{PWMQ} (platform waiting maintenance queue) to wait until resources are available. The PRQ sub-module checks all platforms in the PMQ for the possibility of full restoration in the CRD sub-module.

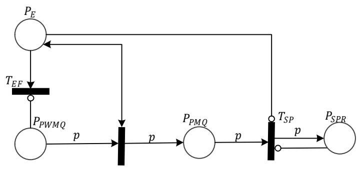

Figure 19. Platform restoration queuing sub-module

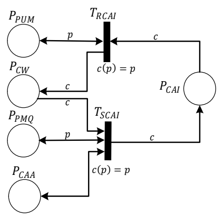

Figure 20. Cannibalisation inventory control sub-module

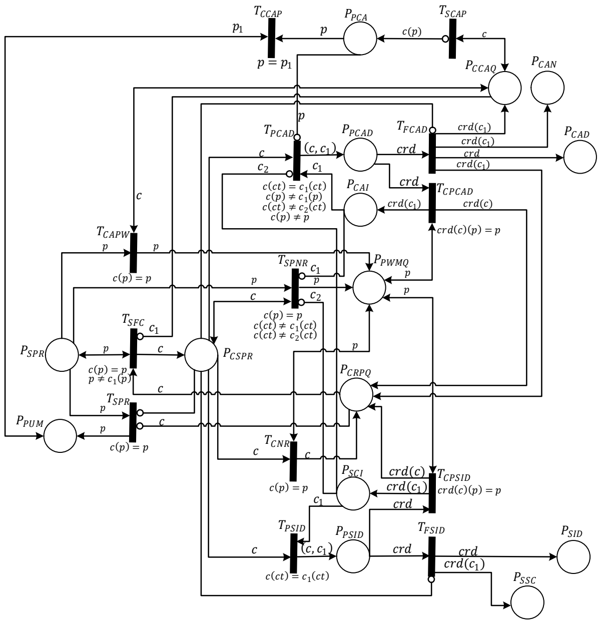

Figure 21. Component replacement decision sub-module

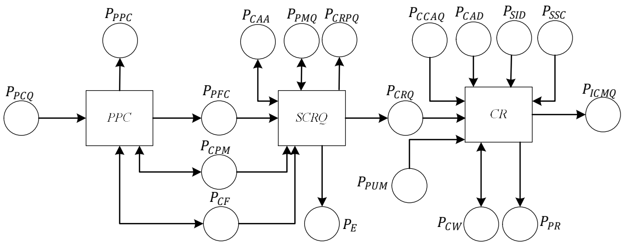

Figure 22. O-level maintenance action sub-module

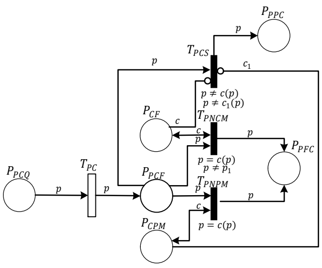

Figure 23. Platform preventive check sub-module

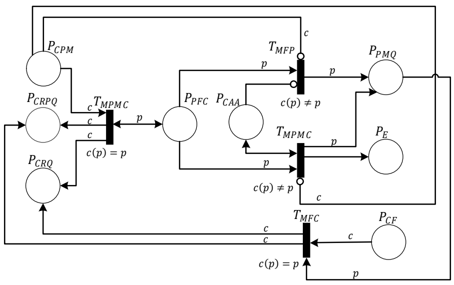

Figure 24. Set component replacement queue sub-module

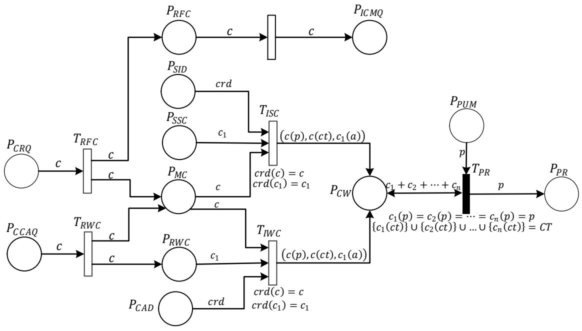

Figure 25. Component replacement sub-module

The CAIC sub-module (Figure 20) manages the available cannibalisation resources, which are the working components of NMC platforms. If cannibalisation is allowed (P_{CAA} is marked), a platform's cannibalisation candidates (tokens in P_{CW}) are added to the cannibalisation inventory (P_{CAI}) as it joins the PMQ and removed when restoration of the platform begins. MC Platforms in the platform check queue (P_{PCQ}) are checked in the OLMA sub-module. If failed or PM due components are found in a platform it is added to P_{PMQ} (platform maintenance queue). Otherwise, the platform passes the check and an associated token is placed in P_{PPC}.

The OLMA module (Figure 22), modelling the maintenance actions performed at a base's O-level maintenance organisation, comprises three sub-modules: platform preventive check (PPC), set component replacement queue (SCRQ) and component replacement (CR). When a platform joins a base's PCQ (P_{PCQ}), the PPC sub-module

(Figure 23) is activated and checks the platform for failed or PM due components. If it fails the check and needs maintenance, the platform is output to P_{PFC}. Otherwise, it passes and a token representing it is placed in P_{PPC}.

When a platform fails a periodic check, the SCRQ sub-module **(**Figure 24) adds its PM due components (if any exist) to the component removal (P_{CRQ}) and component replacement queues (P_{CRPQ}). After replacement tasks are identified for its PM due components, the platform joins the PMQ, which can be viewed as a replenishment of the cannibalisation resources. Hence, when a platform fails a check and joins the PMQ, a token is placed in P_E to enable the restoration check of platforms in the PWMQ. Failed components (tokens in P_{CF}) are identified in P_{CRQ} and P_{CRPQ} by the SCRQ sub-module once their parent platform enters the PMQ.

The CR sub-module (Figure 25) performs component replacement tasks. Once the CRD sub-module has placed component spare and cannibalisation replacement decisions in P_{SID} and P_{CAD}, the selected spares and cannibalised components are obtained from the inventory and their source platforms and installed in the destination platform after inoperative components have been removed. These inoperative components are sent to the base's I-level maintenance organisation for repair through P_{ICMQ}. When all inoperative components have been replaced, the platform undergoing maintenance (a token in P_{PUM}) is restored and output to P_{PR} by the CR sub-module.

3.4.4.2 I-level Maintenance

The ILM sub-module, shown in Figure 26, models component repair at a base's I-level maintenance organisation. After arriving at I-level, a failed component is first checked to see whether repair is possible. $p_{c(ct)}$ is the probability that component c is possible at I-level. When a token of colour c is added to P_{ICMQ} (failed component c arrives at I-level), a variable r is randomly sampled from the range 0 to 1. If $r<p_{c(ct)}$, T_{SICRQ} moves the token from P_{ICMQ} to P_{IFCRQ}, indicating the component is repairable at I-level. Otherwise, T_{SDCRQ} moves the token from P_{ICMQ} to P_{FCTD} and places a token representing an associated spare request in P_{BSCR}. Then T_{TID} adds the component to the depot component maintenance queue (P_{DCMQ}).

T_{IRFC}, whose associated firing delays represent component repair times, governs component repair at I-level, outputting tokens with colour $(0,ct,a)$ to P_{IRC}, where a is randomly sampled from a probability distribution relating to the component failure time. If there is no I-level maintenance, p_c, the I-level repair probability, is set to 0 for all components, meaning all inoperative components go directly to the depot after removal at O-level. Similarly, if there is no D-level maintenance, p_c is set to 1 for all components, meaning all component repair is at I-level.

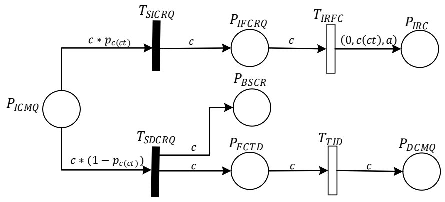

Figure 26. I-level maintenance sub-module

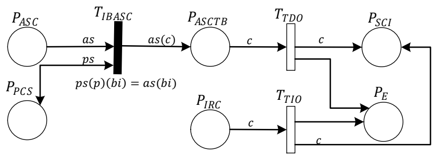

Figure 27. Base spare inventory control sub-module

3.4.4.3 Base Spare Inventory Control

The BSIC module, shown in Figure 27**,** identifies and transports base spare inventory replenishments, which are spare components assigned to a specific base by the depot and components repaired at the base's I-level. T_{IBASC} identifies spare requests from base bi, which can be satisfied by depot inventories, placing assigned spares in P_{ASCTB}. T_{IBASC} is enabled when tokens with colour as and ps mark P_{ASC} and P_{PCS} respectively and satisfy $ps(p)(bi)=as(bi)$, in which case it immediately removes the token with colour as from P_{ASC} and creates a token with colour $c=as(c)$ in P_{ASCTB}. This enables T_{TDO}, which moves the token from P_{ASCTB} to P_{SCI} after a delay representing the transportation time from the depot to base bi. Components repaired at I-level (tokens in P_{IRC}) are transported to the associated inventory by T_{TIO}. P_E is marked when T_{TDO} or T_{TIO} fire, indicating that new spares are available and the OLM module can therefore attempt to restore platforms in the PWMQ.

3.5 Depot Module

The depot module, shown in Figure 28, models the repair of inoperative components brought from bases and the response of depot inventories to base spare requests. Once requested, if a spare component of the required type is available in the depot stock, it is sent to the base through P_{ASC}. Otherwise, the spare request is backordered until the depot inventory is restocked by repaired components. T_{DRC} governs the depot repair of inoperative components from bases and outputs the perfectly repaired components to the depot spare inventories (P_{DSCI}). Spare requests arriving from bases (P_{BSCR} is marked) activate T_{ASC}, which satisfies the base requests with available spares.

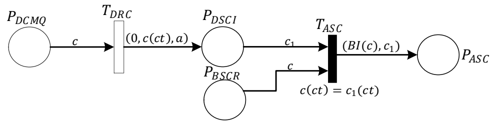

Figure 28. Depot module

3.6 Model Operation

Before using the HCPN model to investigate fleet spare inventory management problems, fleet operation and maintenance parameters must first be input by setting the initial marking, arc expressions and transition firing delays. For instance, cannibalisation policy is governed by P_{CAA} in the base module (Figure 6). P_{CAA} is unmarked if cannibalisation is not used at a base and marked otherwise.

The detailed model initialisation input parameters and associated places, transitions and arcs are listed in Table 3. In addition, platform restoration queuing and selection of cannibalisation resources are respectively represented by T_{SP} in the PRQ sub-module (Figure 19) and T_{PCAD} in the CRD sub-module (Figure 21), which switch tokens from their input places in an order determined by a selection discipline. The disciplines used to assign MC platforms to scheduled missions and to determine how depot spare inventories respond to spare requests from different bases are respectively governed by T_{FMA} in the MA sub-module (Figure 8) and T_{ASC} in the depot module (Figure 28). The numbers of bases and depots specify the numbers of instances of base and depot modules in the master module and spare supply relationships are modelled by how these instances transfer tokens.

Table 3. Input parameters represented by places, transitions and arcs

Place/transitions/arcs	Module	Input parameters represented by initial place marking/transition firing delays/arc expressions
P_{FMS}	*Base*	The total number of required missions at a base
P_{PMC}	*Base*	The total number and indices of platforms at a base
P_{CW}	*Base*	The total number and indices of components in platforms at a base
P_{CAA}	*Base*	The cannibalisation policy at a base
P_{SCI}	*Base*	The number of spares of each component type at a base
P_{DSCI}	*Depot*	The number of spares of each component type at a depot
P_{AOM}	*Base*	The opportunistic policy at a base
T_{FMS}	*MA*	The start time of each mission
T_{MGA}	*MA*	The assignment time window of each mission
T_{PPM}	*MO*	The length of each mission
T_{CF}	*Component*	Component failure time
T_{CPM}	*Component*	Component PM due time
T_{RFC}	*OLMA*	Removal time of failed and PM due components
T_{RWC}	*OLMA*	Removal time of cannibalised components
T_{ISC}	*OLMA*	Installation time of spare components
T_{IWC}	*OLMA*	Installation time of cannibalised components
T_{IRC}	*ILM*	Component repair time at I-level
T_{DRC}	*Depot*	Component repair time at a depot
T_{TOI}	*OLMA*	The transportation time from O-level to I-level
T_{TID}	*ILM*	The transportation time from a base to a depot
T_{TIO}	*BSIC*	The transportation time from I-level to O-level
T_{TDO}	*BSIC*	The transportation time from a depot to a base
T_{PC}	*PPC*	The time to perform a platform inspection
The *inhibitor* arc from P_{PAM} to T_{FMA}	*MA*	The required number of MC platforms to perform a scheduled mission
The *input* arc from P_{ICMQ} *to* T_{SICRQ}	*ILM*	The I-level repair probability of failed components
The *output* arc from T_{PNPC} *to* P_{PCS}	*MO*	The platform inspection interval

Once all input parameters are initialised, the HCPN model can be used as a framework for a Monte Carlo simulation analysis of the modelled fleet's performance. Monte Carlo simulation uses repeated random sampling and statistical analysis to obtain the required fleet performance [22]. Bespoke C# software was developed to encode the simulation of the HCPN model. The software takes in the required fleet parameters in Excel-file format and uses them to initialise the model and set the structure of the platform module within each module instance, before producing the required module instances and connecting them to produce the complete net. A specified number of simulations are performed with the marking during simulations used to gather information for the calculation of the following performance measures:

- Mission capable rate (MCR): proportion of time that platforms in the fleet are capable of performing missions in a given time interval;
- Mission abort rate (MAR): proportion of missions that are not performed successfully due to platform failures and cancelled due to a lack of MC platforms;
- Mission air abort rate (MAAR): proportion of missions aborted due to platform failures;
- Mission ground abort rate (MGAR): proportion of missions cancelled because insufficient MC platforms are available;
- Cannibalisation rate (CAR): number of cannibalisations performed per 100 missions completed by a fleet.

Table 4 lists the information needed to calculate these performance measures and the places where this information is collected. The developed software outputs related statistics to Excel files for analysis. The confidence level of the simulation results depends on the number of simulations performed, with convergence of results coming with more simulations [23]. In this paper, convergence is considered to occur when further simulations change the results by less than 1%, which generally happened after around 100 simulations.

Table 4. Information for calculating fleet performance measures

Indicator	Information	Places	Module
CAR	Number of cannibalisations performed	P_{CAN}	*Base*
	Number of missions performed	P_{ME}	*MO*
MAR, MGAR, MAAR	Number of missions required	P_{FMS}	*Base*
	Number of mission ground aborts	P_{MGA}	*Base*
	Number of mission air aborts	P_{MAA}	*Base*
MCR	Total MC hours of each platform	P_{PMC}	*Base*

3.7 Model Capability

The proposed HCPN models can concisely model operation and maintenance systems of complex fleets and provide an ideal framework for Monte Carlo simulation, hence delivering a tool for fleet managers to:

- Estimate and predict fleet performance. Given all the fleet characteristics including fleet deployment, operational requirements at different bases, platform and component failure characteristics, time relating to maintenance actions, available maintenance resources and the applied maintenance policies, the HCPN models can be used to obtain the fleet performance measures that concern fleet managers;
- Manage the spare inventory system. Due to the flexibility of the HCPN models, different spare provision scenarios and inventory control policies such as the depot response policies to conflicting base spare requests and base stocking rules can be easily compared based on their effects on fleet performance, which can be used to direct the provision and management of spare inventories.
- Select maintenance strategies. The HCPN modules allow fleet managers to study and compare the effects of different maintenance policies such as cannibalisation, opportunistic maintenance and different preventive check intervals on fleet performance, and address their strengths and weaknesses without modification of the whole models. This feature ensures that the selection of maintenance policies could be easily done by fleet managers according to their objectives of maintenance management.
- Optimise fleet maintenance system performance. Through employing intelligent optimisation algorithms, maintenance strategy and spare provision can be jointly optimised, which can support fleet managers in selecting the most effective way to improve and optimise the fleet maintenance performance.
- Direct the fleet maintenance system design. Since the HCPN model can represent different fleet maintenance systems that vary according to the number of depots, base maintenance capability and distance between depots and base maintenance shops, it can help fleet managers determine the organisation and locations of the available maintenance facilities through the comparisons of fleet performance under different scenarios.

4 Model Application

To demonstrate how the HCPN model can be used to assist with spare inventory management, it is first applied to an example fleet with a single base. The effect of spares on fleet performance and the impact of platform inspection intervals on spare inventory performance are studied. The joint optimisation of spare stock levels and platform inspection intervals are also studied using a GA to solve the optimisation problem. A further example fleet with two operational bases is studied to demonstrate the use of the HCPN model in fleet spare inventory management for fleets with multiple operational bases. Two specific problems are studied: the allocation of spares to bases and the depot, and depot spare inventory response discipline when conflicting base spare requests arise.

4.1 Application to a Fleet with a Single Base

The fleet consists of 20 identical, independent platforms, each comprising 10 three-state components connected in series. There are sufficient maintenance crews to complete all required maintenance at any time. Table 5 shows the distributions governing component state changes, removal/installation times, repair times and spare costs. The fleet must perform 18 identical missions per day for one year. The duration of each mission in hours is assumed to follow a Uniform (16,24) distribution. Platform inspection duration (hours) follows a Triangle (4,6,7) distribution. The time taken in hours to transport components between the base's O- and I-level maintenance organisations and between the base and depot follow Triangle (0.1,0.15,0.25) and Triangle (20,22,24) distributions respectively.

4.1.1 Effects of Cannibalisation and Opportunistic Maintenance (OM)

In order to determine appropriate cannibalisation and OM policies for the fleet, their effects on fleet performance are first investigated through four scenarios, which differ according to the applied maintenance polices:

- Scenario 1: no cannibalisation, no OM
- Scenario 2: with cannibalisation, no OM
- Scenario 3: no cannibalisation, with OM
- Scenario 4: with cannibalisation, with OM

Table 5. Example fleet parameters

Component	PM time	Failure time	Remove time (hour)	Repair time (hour)	Install time (hour)	Base repair rate	Spare price	Spare holding cost
1	Weibull (β=1.4, η =1500)	Weibull (β=1.4, η =600)	Uniform (4.5,5.5)	Triangle (225,230,240)	Uniform (3.5,5)	0.8	92000	1300
2	Weibull (β=1.8, η =1700)	Weibull (β=1.8, η =450)	Uniform (5.5,6.5)	Uniform (150,160)	Uniform (6,8)	0.95	100000	1600
3	Weibull (β=1.7, η =2000)	Weibull (β=1.7, η =500)	Triangle (4,5,6)	Uniform (170,190)	Triangle (6,6.5,7)	0.9	44000	760
4	Normal (μ=1900, σ=18)	Normal (μ=350, σ=18)	Uniform (3.8,4.5)	Triangle (150,156,160)	Uniform (3.8,4.5)	0.75	70000	1400
5	Weibull (β=1.5, η =1850)	Weibull (β=1.5, η =750)	Triangle (3.5,4.5,6)	Uniform (195,230)	Triangle (3,4,4.5)	0.85	120000	1040
6	Weibull (β=1.3, η=2300)	Weibull (β=1.3, η=230)	Triangle (3,4,5)	Triangle (230,240,250)	Uniform (4,6)	0.85	50000	1000
7	Weibull (β=1.3, η=2300)	Weibull (β=1.3, η=230)	Triangle (3,4,5)	Triangle (230,240,250)	Uniform (4,6)	0.9	50000	1000
8	Lognormal (μ=7.6, σ=0.8)	Lognormal, (μ=6.2, σ=0.8)	Uniform (2.8,4.2)	Triangle (170,178,190)	Uniform (3.6,4)	0.9	80000	800
9	Weibull (β=1.6, η =1950)	Weibull (β=1.6, η =360)	Triangle (3,4,4.5)	Uniform (180,210)	Triangle (3,3.5,4.5)	0.8	86000	1080
10	Weibull (β=1.9, η =1780)	Weibull (β=1.9, η =450)	Triangle (3.5,4.5,6)	Triangle (195,210,230)	Triangle (3,4,4.5)	0.75	40000	600

None of the scenarios allow platform preventive checks or spares. A random discipline is applied to platform restoration and cannibalisation queuing. Convergence of results for all performance indicators was observed by 100 simulations. This was the case for all scenarios analysed in this paper but to guarantee accuracy of results, 200 simulations are performed in all cases. Table 6 summarises fleet performance in the four scenarios. The results show that, for this fleet:

1. Allowing only cannibalisation increases MCR and reduces MAR (compare values between scenarios 2 and 1). Note the reduction in MAR is due to a huge reduction in MGAR, however MAAR increases and 6.23 cannibalisations are required every 100 missions.
2. Allowing only OM also increases MCR and reduces MAR (compare scenarios 3 and 1). In contrast to when performing only cannibalisation (scenario 2), the reduction in MAR is due not just to reduced MGAR but also to reduced MAAR, although more missions are performed if OM is applied. This happens because PM due components in NMC platforms are replaced alongside failed components under OM, helping to reduce the number of platform failures experienced during operation.
3. Allowing both OM and cannibalisation (scenario 4) gives the best fleet performance in terms of MCR, MGAR and MAR. Using OM in addition to cannibalisation reduces MAAR and CAR (compare scenarios 2 and 4). This may be due to OM reducing the number of platform failures, resulting in fewer mission air aborts, limiting available cannibalisation resources and the demand for cannibalisation.

Therefore, for this fleet, these results suggest that it may be better to use both cannibalisation and OM.

Table 6. Effects of cannibalisation and opportunistic maintenance on fleet performance

Fleet Performance		Scenario 1	Scenario 2	Scenario3	Scenario 4
MCR	Average	62.5%	81.4%	76.1%	83.0%
	Maximum simulated	64.2%	82.6%	77.6%	84.3%
	Minimum simulated	60.4%	80.2%	74.1%	81.6%
	Standard deviation	0.68%	0.50%	0.64%	0.48%
MAR	Average	35.6%	16.3%	18.7%	11.5%
	Maximum simulated	38.3%	18.1%	21.2%	12.9%
	Minimum simulated	33.5%	14.8%	17.2%	10.2%
	Standard deviation	0.80%	0.60%	0.76%	0.51%
MGAR	Average	30.7%	9.5%	15.7%	8.3%
	Maximum simulated	33.2%	11.1%	18.1%	9.7%
	Minimum simulated	28.8%	8.1%	14.3%	7.0%
	Standard deviation	0.73%	0.56%	0.69%	0.48%
MAAR	Average	4.9%	6.8%	3.0%	3.2%
	Maximum simulated	5.1%	7.3%	3.2%	3.5%
	Minimum simulated	4.7%	6.3%	2.7%	2.9%
	Standard deviation	0.08%	0.18%	0.09%	0.10%
CAR	Average	0	6.23	0	4.54
	Maximum simulated	0	7.47	0	5.41
	Minimum simulated	0	4.87	0	3.46
	Standard deviation	0	0.48	0	0.33

4.1.2 Effect of Spares

To investigate the effect of spares on fleet performance and the effect of platform inspection intervals on spare inventories, 10 scenarios, each assuming a fixed number of spares (ranging from 1 to 10) for each type of component are studied under three platform inspection scenarios, which differ according to the inspection interval: 5 missions, 25 missions, and infinite (i.e. no preventive inspection). Both cannibalisation and OM are allowed and all spares are stocked at the base.

Figures 29-31 summarise average fleet performance under the three platform inspection scenarios. The results show that, for this fleet, increasing the provided number of spares:

1. Can increase MCR and reduce MGAR (see Figure 29). However, the additional benefit decreases with increasing number of spares. The flattening of these rates happens because platform NMC time has three elements: preventive inspection time, time waiting for available spare or cannibalised resources, and actual restoration time; of these elements, increasing the number of spares can only reduce waiting time.
2. Can reduce CAR (see Figure 30), with virtually no cannibalisations performed if 6 or more spares are stocked for each component type no matter the platform inspection policy. This is expected since cannibalisations will not be required when sufficient spares are available.
3. May have no significant impact on MAAR (see Figure 31). This is particularly true for shorter inspection intervals – observe how average MAAR is unchanged no matter how many spares are stocked for an inspection interval of 5 missions with little change also seen in the other scenarios. This may be because

critical operational requirements require most platforms in the fleet to perform missions daily (18 missions per day for a fleet of 20 platforms), resulting in a similar number of platform failures under a specified inspection policy as the increase in the spare inventories.

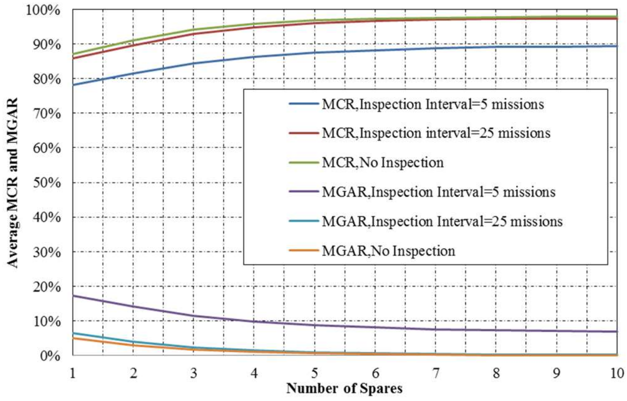

Figure 29. Change of mission capable rate and mission ground abort rate with number of spares

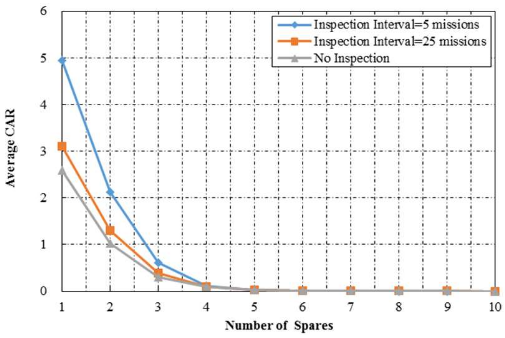

Figure 30.Change of cannibalisation rate with number of spares

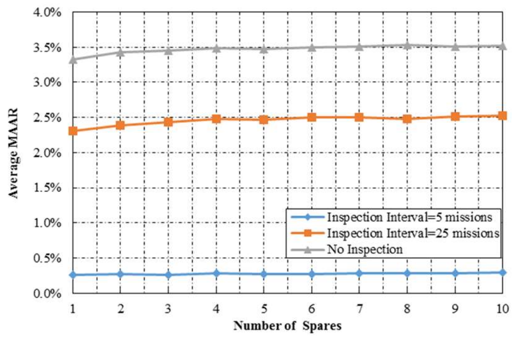

Figure 31. Change of mission air abort rate with number of spares

The results also show that, for this fleet:

1. If the number of spares is unchanged, reducing the inspection interval can decrease MCR (see Figure 29). The MCR drops from 97.3% to 89.5% when 10 spares are provided and the inspection interval is reduced from 25 to 5 missions. This shows that a poorly-specified inspection interval could negatively impact MCR even when sufficient spares are provided.
2. Providing more spares will not necessarily increase the MCR (see Figure 29). The MCR is higher when stocking 2 spares and the inspection interval is 25 missions than it is when stocking 10 spares and the inspection interval is 5 missions.

4.1.3 Joint Optimisation of Platform Inspection Interval and Spare Inventories

The results in the previous section demonstrate the complicated relationship between spare inventories and inspection intervals but cost is also a major factor. Repairable spares can be expensive, meaning inventories often cannot stock enough of them due to budget constraints. Mission cancellations and air aborts can also induce huge penalty costs. Therefore, it is important to jointly determine spare inventory stocks and platform inspection intervals to minimise the total cost of spares and penalties due to mission aborts.

4.1.3.1 Problem Formulation

The objective of this optimisation problem is to minimise the total cost of spares and penalties due to platform failures and mission ground aborts.

The cost of spares includes purchase and holding costs. Let s_j, PC_j, HC_j and HT_j respectively represent the initial amount, the price per unit, the holding price per hour and the total hours held in the inventory of the purchased spare of type j ($1 \leq j \leq 10$). HT_j is calculated during HCPN model simulations. The spare cost, SC, is given by:

$$SC = \sum_{j=1}^{10} (s_j \times PC_j + HC_j \times HT_j) \tag{1}$$

The cumulative number of mission air aborts (platform failures), N_{MAA}, and total ground aborts, N_{MGA}, are obtained from the base module (Figure 17) during simulations. The platform failure and mission cancellation costs, FTC and $MCTC$, are given by:

$$FTC = N_{MAA} \times FC \tag{2}$$

$$MCTC = N_{MGA} \times MCC \tag{3}$$

where FC and MCC are the penalty cost per platform failure and per ground abort. The total cost, TC, is therefore:

$$TC = \sum_{j=1}^{10} (s_j \times PC_j + HC_j \times HT_j) + N_{MAA} \times FC + N_{MGA} \times MCC \tag{4}$$

The decision variables are $s_1,\ldots,s_{10}$, which represent the initial number of spares of each type in the inventory and NM_{PC}, the platform inspection interval. Thus, the joint optimisation of the fleet spare inventory and platform inspection interval is represented by:

$$\text{Objective: Min } TC \tag{5}$$

$$\text{Subject to: } s_j \geq 0, \text{ integer}, 1 \leq j \leq 10 \tag{6}$$

$$NM_{PC} > 0, \text{ integer} \tag{7}$$

The penalty costs of air and ground aborts are assumed to be 30000 and 3000 respectively and it is assumed that the inspection interval (NM_{PC}) and the number of spares for each component type (s_j) are integers which vary between 5 and 30 missions and between 0 and 3 spares respectively. This yields over 2.7×10^8 possible scenarios, making it impractical to exhaustively simulate all scenarios in order to find an optimal solution.

4.1.3.2 Genetic Algorithm

The Genetic Algorithm (GA) approach is used in this paper to search for an optimal solution due to its capability of solving a wide range of optimisation problems with a variety of decision variables. First introduced by Holland in 1975, it is a computational technique based on the Darwinian principle of 'survival-of-the-fittest' [24]. Rather than searching the solution space from a single solution, the search is performed on a population of solutions, each of which is assigned a fitness (in this case the TC calculated using the HCPN). The fitter individuals in a population are then more likely to be chosen as parents for the next generation, with the GA operators of selection, crossover and mutation applied to produce a new population. Since the GA approach is not the emphasis of this research it is not discussed in detail here. The reader is instead referred to [10-12] and [24] although this is just a small subset of the available literature. The use of the GA demonstrates how fleet maintenance decision making can be supported by the HCPN model in conjunction with an optimisation technique.

4.1.3.3 Problem Solution

To obtain a satisfactory solution for the considered fleet, 60 generations are produced, each with a population of 30 individuals. The crossover and mutation rates are set to be 0.9 and 0.05 respectively. An elitism policy is applied where the 5 best scenarios in a population directly enter the next generation.

Figure 32 shows the evolution of minimal and average costs from generation to generation. The solution with the lowest total cost of 4222000 is identified in the 32nd generation, and requires $(s_1,s_2,s_3,s_4,s_5,s_6,s_7,s_8,s_9,s_{10})$ = (1,0,0,2,0,0,0,0,0,1) and a platform inspection interval of 8 missions. To demonstrate the efficiency of the GA-based approach, this optimal scenario is compared to the four boundary scenarios:

- Scenario 1: min. inspection interval (5 missions), min. number of spares (0 for each type)
- Scenario 2: min. inspection interval (5), max. number of spares (3)
- Scenario 3: max. inspection interval (30), min. number of spares (0)

- Scenario 4: max. inspection interval (30), max. number of spares (3)

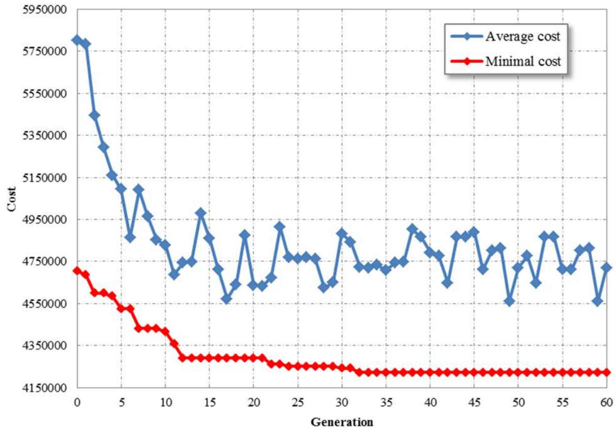

Figure 32. Minimal and average cost

Table 7 summarises the average fleet performance for these four scenarios and the optimal scenario. The optimal scenario performs best in terms of cost and seems acceptable in other terms, never providing the worst performance for any particular measure.

Table 7. Boundary scenario and optimisation results

Fleet performance	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Optimal Scenario
Average MCR	74.6%	84.5%	82.2%	93.3%	79.1%
Average MAR	21.0%	11.8%	12.1%	4.8%	15.5%
Average MGAR	20.8%	11.5%	9.6%	2.1%	15.0%
Average MAAR	0.2%	0.3%	2.5%	2.7%	0.5%
Average CAR	7.68	0.61	4.89	0.37	6.42
Average cost(000)	4576	5014	6832	8008	4222

4.2 Application to a Fleet with Two Bases

The application of the HCPN model to fleets with multiple operational bases is demonstrated through the study of two inventory management problems:

1. How to allocate purchased spares to depot and base inventories?
2. How should the depot respond to conflicting base spare requests?

An inefficient allocation of spares may lead to their unavailability when required due to being in transit between bases and the depot. The depot response discipline might be chosen to either guarantee the performance of a specified base or balance the performance of all bases.

The fleet used to study these problems consists of 55 identical and independent platforms, 2 bases and 1 depot. The platforms are identical to those investigated in Section 4.1 as are the mission start time and the time taken to inspect a platform. The two bases have different inspection policies and operational requirements, shown in Table 8. All inoperative components are sent to the depot for repair and the time in hours to transport a component between bases and the depot follows a Triangle(20,22,24) distribution.

Table 8. Fleet parameters

	Deployed platforms	Mission requirement (missions per day)	Mission length (hours)	Inspection interval (missions without failure)
Base 1	30	25	Uniform (16,18)	10
Base 2	25	22	Uniform (14,16)	12

The following sections illustrate how to apply the HCPN model to these problems through the comparison of different scenarios; optimisation could also be performed using a method similar to that presented in Section 4.1.3. For all scenarios studied, 200 simulations are performed and both OM and cannibalisation are allowed.

4.2.1 Spare Allocation

4 spares are assumed for each component type and these are allocated between the depot and bases as follows:

- Scenario 1: depot: 4, base 1: 0, base 2: 0
- Scenario 2: depot: 2, base 1: 1, base 2: 1
- Scenario 3: depot: 0, base 1: 2, base 2: 2

The depot responds to base spare requests using a FIFO (first in first out) policy. Table 9 summarises the average overall fleet performance, and the performances of the sub-fleets at each of the bases under the three scenarios.

Table 9. Simulation results under different spare allocation scenarios

Fleet performance	Scenario 1 (4, 0, 0)			Scenario 2 (2, 1, 1)			Scenario 3 (0, 2, 2)		
	Fleet	Base 1	Base 2	Fleet	Base 1	Base 2	Fleet	Base 1	Base 2
Average MCR	83.2%	82.6%	83.8%	83.1%	82.5%	83.9%	82.4%	81.0%	84.1%
Average MAR	6.2%	5.3%	7.1%	6.2%	5.3%	7.1%	6.8%	6.7%	7.0%
Average MGAR	5.5%	4.6%	6.5%	5.5%	4.6%	6.5%	6.1%	5.9%	6.4%
Average MAAR	0.7%	0.7%	0.6%	0.7%	0.7%	0.6%	0.7%	0.7%	0.6%
Average CAR	5.14	5.59	4.63	5.13	5.56	4.61	4.11	5.04	3.04

The results suggest that, for sub-fleet 1:

1. Performance increases as the number of spares in the depot inventory increases, as does CAR.
2. Stocking all spares at the depot (scenario 1) gives the best MCR and MAR for this sub-fleet but also the highest CAR.
3. By contrast, allocating spares equally between bases (scenario 3) gives the lowest CAR but the worst MCR and MAR.
4. Scenario 2 gives similar MCR, MAR and CAR to those seen when all spares are stocked at the depot (scenario 1).

For sub-fleet 2:

1. As for sub-fleet 1, CAR decreases as the number of spares allocated to its base inventory increases.
2. CAR decreases by about one-third if the number of spares of each type at the base increases from 0 to 2.
3. In contrast to the change in MCR for sub-fleet 1, the MCR for sub-fleet 2 slowly decreases as the depot's share of spares increases.

For the overall fleet:

1. The changes in performance across the scenarios mirror those of sub-fleet 1.
2. Stocking all spares at the depot (scenario 1) gives the best MCR and MAR, but also the highest CAR.
3. Allocating spares to base inventories (scenarios 2 and 3) lowers the CAR, but can negatively affect fleet performance according to other measures.

The transportation time between bases and the depot is relatively short (about 24 hours) meaning that stocking spares at the depot benefits fleet performance since the depot inventories are within reach of both sub-fleets, therefore benefiting each base. The fleet is unequally deployed across the two bases, bases have different operational requirements and all failed components go to the depot for repair. Allocating more spares to the depot could help to maintain the performance of the larger base 1 sub-fleet, which experiences more failures during operation, but may be detrimental to the smaller sub-fleet with fewer failures, because the less frequent base 2 spare requests may be dominated by those from base 1 under the FIFO depot response policy. Considering that the negative effect on the base 2 sub-fleet performance is slight when using more depot spares, it may be better to stock all spares at depot inventories to maximise fleet MCR and minimise MAR. However, if the objective of the spare inventory management is to minimise the use of cannibalisation, it may be better to stock all spares at bases.

4.2.2 Depot Response Discipline

In considering depot response discipline, three new policies are studied in addition to the FIFO policy:

- Random selection,
- Base 1 priority,
- Base 2 priority.

There are again 4 spares for each component type and the depot stocks them all. Table 10 summarises overall fleet and sub-fleet performance under the three policies, with performance under the FIFO policy given in Table 9.

Table 10. Simulation results under various depot response policies

Fleet performance	Random			Base 1 priority			Base 2 priority		
	Fleet	Base 1	Base 2	Fleet	Base 1	Base 2	Fleet	Base 1	Base 2
Average MCR	83.2%	82.5%	84.0%	82.9%	84.9%	80.5%	82.8%	78.9%	87.6%
Average MAR	6.1%	5.5%	6.9%	7.0%	4.1%	10.3%	6.7%	9.1%	4.0%
Average MGAR	5.4%	4.7%	6.3%	6.3%	3.4%	9.7%	6.0%	8.3%	3.3%
Average MAAR	0.7%	0.8%	0.6%	0.7%	0.8%	0.6%	0.7%	0.7%	0.7%
Average CAR	5.16	5.56	4.70	5.01	5.61	4.28	4.85	4.83	4.86

The results show that:

1. Of the four polices, random selection gives the best fleet MCR and MAR but also the highest CAR.
2. Under FIFO, the fleet performs similarly to how it does under random selection.
3. The priority response policies guarantee the sub-fleet with priority performs best, as expected.
4. Overall fleet performance, especially MAR, tends to be worse under a priority response policy than under the other policies.

Therefore, overall fleet performance may be maximised by employing the random or FIFO policy at the depot to satisfy base spare requests. By contrast, a need to guarantee sub-fleet performance at a specified base requires a priority depot response policy. It should be noted that in practice, the response policy may also depend on factors such as the nature of the missions at each base.

5 Summary and Conclusions

A novel HCPN model of a fleet spare inventory system has been developed, considering a wide range of fleet factors including multiple operational bases, mission-oriented operation, three-level maintenance, preventive checks, opportunistic maintenance and cannibalisation. The multi-level model comprises modules, which can be instanced to represent repeatedly-used activities and facilities such as the operation and maintenance processes performed at different bases. This facilitates its modification and extension to study fleets with varying operation and maintenance factors.

The application of the HCPN model is first demonstrated by studying an example fleet with a single operational base. The necessity of performing cannibalisation and opportunistic maintenance for the modelled fleet is analysed, as are the impacts of platform preventive inspections and spare inventories on fleet performance. The simulation results suggest that using cannibalisation, opportunistic maintenance and spares can help to improve fleet MCR and reduce the number of mission aborts, and the platform preventive inspection interval can have a significant impact on the performance of fleet maintenance and spare inventory system. Then a GA is applied to jointly optimise spare inventories and the inspection interval with the objective of minimising the total cost of spares and penalties related to mission aborts. The GA-based approach is demonstrated for a bounded horizon which contains a huge number of solutions, making exhaustive simulation impractical. The performance of the derived optimal solution found is verified through its comparison to a number of boundary scenarios.

The HCPN model is further applied to a fleet of platforms that are deployed at two bases and served by a single depot in order to demonstrate its application to managing spare inventories for fleets with multiple operational bases. Two inventory management problems are studied: the allocation of spares to depot and base inventories and the discipline of how the depot responds to conflicting base spare requests. To provide a guide of how to use the HCPN model to solve these problems for practical fleets, a number of solutions are simulated and compared. Through comparisons, fleet managers can choose the best solution according to a specific management objective. These case studies demonstrate the capability of the HCPN model to help fleet managers design, manage and optimise fleet spare inventory and maintenance systems.

Biographies

Jingyu Sheng is a technical Major in the Chinese Air Force. He was awarded a PhD degree by the University of Nottingham in 2015, having focussed on modelling the maintenance of aircraft fleets. His research interests include complex system modelling, spare inventory and maintenance management, and multi-objective optimisation.

Darren Prescott is Assistant Professor in Risk and Reliability Engineering in the Resilience Engineering Research Group at the University of Nottingham. His current research interests include aircraft fleet maintenance modelling, the application of reliability modelling techniques to support decision making in autonomous systems and the development of asset management models. He has published around 50 papers in the areas of risk, reliability and maintainability. He is chair of the ESRA (European Safety and Reliability Association) Technical Committee on Aeronautics and Aerospace.

References

1. Curtain, N.P., 2001. *Military aircraft: cannibalizations adversely affect personnel and maintenance*. Washington D.C.: United States General Accounting Office.
2. Sherbrooke, C.C., 1966. *Metric: a multi-echelon technique for recoverable item control*. Santa Monica: The RAND Coporation.
3. Muckstadt, J., 1973. Model for a multi-item, multi-echelon, multi-indenture inventory system. *Management Science*, 20(4), pp.472-481.
4. Slay, F.M., 1984. *VARI-METRIC: an approach to modelling multi-echelon resupply when the demand process is poisson with a gamma prior*. Washington: Logistics Management Institute.
5. Sherbrooke, C.C., 1986. VARI-METRIC: Improved approximation for multi-indenture, multi-echelon availability models. *Operations Research*, 34(2), pp.311-319.
6. Lau, H.C., Song, H., See, C.T. and Cheng, S.Y., 2006. Evaluation of time-varying availability in multi-echelon spare parts systems with passivation. *European Journal of Operational Research*, 170(1), pp.91-105.
7. Wang, N. and Ma, L., 2011. Estimation and analysis of time-varying availability under the drive of mission scenarios. In: IEEE, *9th International Conference on Reliability, Maintainability and Safety*. Guiyang, China, 12-15 June 2011. New York: IEEE.
8. Hillestad, R.J., 1982. *DYNA-METRIC: dynamic multi-indenture technique for recoverable item control*. Santa Monica: The RAND Coporation.
9. Isaacson, K.E. and Boren, P., 1988. *Dyna-METRIC version 5: a capability assessment model including constrained repair and management adaptations*. Santa Monica: The RAND Corporation.
10. Marseguerra, M., Podofilini, L. and Zio, E., 2001. Use of genetic algorithm for the optimisation of spare parts inventory. In: Zio, E., *12th European Safety and Reliability Conference*. Torino, Italy, 16-20 September 2001. Torini: Politecnico di Torino.
11. Marseguerra, M., Zio, E. and Podofilini, L., 2005. Multiobjective spare part allocation by means of genetic algorithm and monte carlo simulation. *Reliability Engineering & System Safety*, 87(3), pp.325-335.
12. Ilgin, M. and Tunali, S., 2007. Joint optimisation of spare parts inventory and maintenace policies using genetic algorithm. *International Journal of Advanced Manufacturing Technology*, 34, pp.594-604.
13. Chen, M., Hsu, C. and Chen, S., 2006. Optimising joint maintenance and stock provisioning policy for a multi-echelon spare part logistics network. *Journal of the Chinese Institute of Industrial Engineers*, 23(4), pp.289-302.
14. Alrabgni, A. and Alabdulkarim, A., 2001. Simulation based optimisation of joint maintenance and inventory for multi-component manufacturing systems. In: Zio, E., *12th European Safety and Reliability Conference*. Torino, Italy, 16-20 September 2001. Torini: Politecnico di Torino.
15. Isaacson, K.E. and Boren, P., 1993. *DYNA-METRIC Version 6: an advanced capability assesment model*. Santa Monica: The RAND Corporation.
16. Ormon, S.W. and Cassady, C.R., 2004. Cannibalization policies for a set of parallel machines. In: IEEE, *2004 Annual Reliability and Maintainability Symposium*. Los Angeles, USA, 26-29 Janannuary 2004. New York: IEEE.

17. Salman, S., Cassady, C.R., Pohl, E.A. and Ormon, S.W., 2007. Evaluating the impact of cannibalization on fleet performance. *Quality and Reliability Engineering International*, 23(4), pp.445-457.
18. Sheng, J. and Prescott, D.R., 2017. A hierarchical coloured petri net model of fleet maintenance with cannibalisation. *Reliability Engineering & System Safety*, 168, pp.290-305.
19. Volovoi, V., 2007. System-level maintenance policies via stochastic petri nets with aging tokens. In: IEEE, *2007 Annual Reliability and Maintainability Symposium*. Orlando, USA, 22-25 Janananuary 2007. New York: IEEE.
20. Chew, S., Dunnett, S.J. and Andrews, J.D., 2008. Phased mission modelling of systems with maintenance free operating periods using simulated petri nets. *Reliability Engineering & System Safety*, 93(7), pp.980-994.
21. Jensen, K. and Kristensen, L.M., 2009. *Coloured petri nets: modelling and validation of concurrent systems*. Heidelberg: Springer.
22. Raychaudhuri, S., 2008. Introduction to Monte Carlo simulation. In: Scott J.M., et al., *2008 Winter Simulation Conference*. Miami, USA, 7-10 December 2008. New York: IEEE.
23. Le, B., 2013. *Modelling railway bridge asset management*. PhD. University of Nottingham.
24. Mitchell, M., 1999. *An introduction to genetic algorithms*. Cambridge: MIT Press.
25. Murata, T., 1989. Petri nets: properties, analysis and applications. *Proceedings of the IEEE*, 77(4), pp.541-580.

Appendix: Meaning of Places

Place	Meaning	Colour Set	Place	Meaning	Colour Set
P_{DCMQ}	Depot component maintenance queue	*C*	P_{SSC}	Selected spare component	*C*
P_{BSCR}	Base spare component request	*C*	P_{PCA}	Cannibalised platforms that are not restored	*P*
P_{ASC}	Assigned spare component	*AS*	P_{CSPR}	Failed components of the selected platform	*C*
P_{FMS}	Fleet missions	*M*	P_{PSID}	Primary spare installation decisions	*CRD*
P_{PCS}	Platform check state	*PCS*	P_{PCAD}	Primary component cannibalisation decisions	*CRD*
P_{CAA}	Cannibalisation policy	-	P_{PCF}	Finished platform checks	*P*
P_{PMC}	Mission-capable platforms	*P*	P_{PFC}	Platforms which failed a check	*P*
P_{PNMC}	Non-mission-capable platforms	*P*	P_{CRQ}	Component removal queue	*C*
P_{CF}	Failed components	*C*	P_{MC}	Missing components in platforms	*C*
P_{PPM}	Platforms that are performing missions	*PM*	P_{RFC}	Removed, failed components	*C*
P_{PR}	Restored platforms	*P*	P_{RWC}	Removed, working components	*C*
P_{PMQ}	Platform maintenance queue	*P*	P_{IFCRQ}	I-level component repair queue	*C*
P_{PCQ}	Platform check queue	*P*	P_{FCTD}	Components that cannot be repaired at I-level	*C*
P_{E}	Event	-	P_{ASCTB}	Spares assigned to a specific base	*C*
P_{AOM}	Opportunistic maintenance policy	-	P_{PUM}	Platforms undergoing maintenance	*P*
P_{CW}	Working components	*C*	P_{CAI}	Component cannibalisation inventory (working components that can be cannibalised)	*C*
P_{MGA}	Mission ground abort	*M*	P_{PWMQ}	Platform waiting maintenance queue	*P*
P_{MAA}	Mission air abort	*M*	P_{SPR}	Platform selected for restoration	*P*
P_{CCAQ}	Component cannibalisation queue (working components selected for cannibalisation)	*C*	P_{CRPQ}	Component replacement queue (components to be replaced during maintenance)	*C*
P_{SCI}	Base spare inventories	*C*	P_{CPM}	PM due components	*C*
P_{CAN}	The number of cannibalisation performed	*C*	P_{SID}	Spare installation decision	*CRD*
P_{MS}	Mission starts	*M*	P_{CAD}	Cannibalisation decision	*CRD*
P_{PAM}	Platforms that are assigned to a mission	*PM*	P_{PPC}	Platforms that pass checks successfully	*P*
P_{POG}	Platforms on ground	*P*	P_{ICMQ}	I-level component maintenance queue	*C*
P_{MUA}	Mission under assignment	*M*	P_{IRC}	I-level repaired components	*C*
P_{ME}	Missions end	*M*	P_{DCMQ}	Depot component maintenance queue	*C*
P_{PF}	Platform failures	*P*			