AI·PROFICIENT

Artificial intelligence
for improved production efficiency,
quality and maintenance

Deliverable 3.3

D3.3: System-level proactive maintenance strategy

WP3: Platform AI analytics and decision-making support

T3.3: Proactive maintenance strategies at system/line level

Version: 1.0

Dissemination Level: PU

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957391.

Table of Contents

List of Figures

List of Tables

Disclaimer

This document contains a description of the AI-PROFICIENT project work and findings.

The authors of this document have taken any available measure for its content to be accurate, consistent and lawful. However, neither the project consortium as a whole nor the individual partners that implicitly or explicitly participated in the creation and publication of this document hold any responsibility for actions that might occur as a result of using its content.

This publication has been produced with the assistance of the European Union. The content of this publication is the sole responsibility of the AI-PROFICIENT consortium and can in no way be taken to reflect the views of the European Union.

The European Union is established in accordance with the Treaty on European Union (Maastricht). There are currently 28 Member States of the Union. It is based on the European Communities and the Member States cooperation in the fields of Common Foreign and Security Policy and Justice and Home Affairs. The five main institutions of the European Union are the European Parliament, the Council of Ministers, the European Commission, the Court of Justice and the Court of Auditors (http://europa.eu/).

AI-PROFICIENT has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 957391.

Title: D3.3: System-level proactive maintenance strategy

Lead Beneficiary:	UL
Due Date:	31/01/2023
Submission Date	31/01/2023
Status	Final ~~Preliminary Draft~~
Description	Description of system level scenario and algorithm for system-level proactive maintenance strategies optimization.
Authors	Phuc Do Van (UL), Chiara Franciosi (UL), Van Thai Nguyen (UL), Alexandre Voisin (UL)
Type	Report
Review Status	~~Draft~~ ~~WP Leader accepted~~ PC + TL accepted
Action Requested	~~Contribution from partners leading in each UC the feedback collection and management.~~ ~~To be revised by partners~~ ~~For approval by the WP leader~~ For approval by the Project Coordinator & Technical Leaders

VERSION	ACTION	OWNER	DATE
0.1	First Draft	UL	19/12/2022
0.2	Preliminary version	UL	10/01/2023
1.0	Final version	UL	30/01/2023

Executive Summary

The Deliverable D3.3 is a public document of AI-PROFICIENT project delivered in the context of Task 3.3: Proactive maintenance strategies at system/line level as a part of WP3: Platform AI analytics and decision-making support, regarding the description of the AI-PROFICIENT service that aim at optimizing maintenance scheduling at system level.

D3.3 incorporates, in the introduction, a reminder of the service description in the context of AI-PROFICIENT in relation to WP1 and the deliverables, and the problem statement of maintenance optimization. The second section summarizes the scientific contribution provided by the task on maintenance policy optimization; publications detailing these contributions are available on the AI-PROFICIENT web site and a link is provided to download them. The third section describes the contribution of the task to maintenance scheduling optimization that relies on a demonstration scenario that incorporate some other use case outcomes in addition to Continental Combiline components. For the sake of confidentiality, we change the real costs in euros to anonymized cost in arbitrary CU (Cost Unit).

1 Introduction

The goal of this deliverable is to gather the contribution that has been provided in Task 3.3 in relation with system level proactive maintenance strategy using AI technique in the context of AI-PROFICIENT project. The objective of the task is to provide, for system-level, maintenance decision-making using self-diagnostic, production anomaly detection (T2.3) and health assessment & prognostic (T2.4) services. This task will contribute to the development of opportune maintenance service (S_OPP) already presented in D1.5 (see Table 1).

Table 1- Excerpt of S_OPP service description from D1.5.

Service ID	S_OPP
Service input and dependency on other services:	The service should be implemented at the cloud and requires at least as input: • Logistic support information such as maintenance team/skill, spare parts, etc. • List of maintenance actions and related costs. • Production planning and constraints. It will also require some input from other services: • Health state evaluation if the service is available. • Component prognostics if the service is available. • Digital Twin if the service is available.
Service output:	Maintenance decision consists in planning maintenance actions, for each component, at the right time in a dynamic way, i.e., which can be adapted/updated in presence of short-term information such as a new maintenance opportunity, new information related to the components/system health state, logistic support or new impacting event available. Hence, the purpose of the service is to provide an optimal scheduling of the maintenance action to be performed on a line or system. The output of the service will be for a list of maintenance actions and date to be performed for each component/group of components. The estimation of system health state (predictive reliability or RUL) before and after maintenance execution will be also provided.
High level service description:	AI-based proactive maintenance approaches are emphasized. Two main steps are considered: • Step1 – Prognostics of health state at system level: The aim is to develop AI-based approaches allowing to predict the health state of the system considering only the prognostic results at component level (e.g., RUL of components), but also the dependence relationships between components under specific context associated to the systems missions/functions. To support this objective, several dependencies in terms of kinds of interactions between components (e.g., structural/functional, stochastic, informational dependence) should be first modelled and formulated. This helps to quantify the impact of one component/group of components on the health state of other components. Secondly, AI-based approaches (e.g., recurrent

	neural network) will be investigated to predict the health state at the system level from the prognostic results at component level and the formalized dependencies between components. The prognostics results of health state will be used for predictive maintenance decision-making at the second step. • Step 2 – Development of AI-based maintenance decision-making models[1]: The proposed models will be built on a set of appropriate decision rules and advanced AI algorithms (e.g., reinforcement learning) which allow to learn the most relevant decision rules to deal with the current condition of the system and its environment from observed data, the estimated health state at both components and system level. In that way, the proposed AI-based decision models should enable not only providing optimal maintenance planning considering both the requirements associated with the maintained system and its support one (e.g., spare parts, maintenance skill) but also to be able to update efficiently the maintenance planning in a dynamic context (e.g., structure changes on main or support system, occurrence of new maintenance opportunities).

This service is intended to cover the _OPP requirement identified and detailed in the deliverable D1.4 (see Table 2).

Table 2 - Functionalities to be provided by the AI-PROFICIENT project (from D1.4).

AI-PROFICIENT Functionalities	***ID***
Monitoring	_MON
Diagnostic and anomaly detection	_DIA
Health state evaluation	_HEA
Component prognostics	_PRO
Hybrid models of production processes and digital twins	_HYB
Predictive Production quality assurance	_PRE
Root-cause identification	_ROO
Early anomaly detection	_EAR
Opportunistic maintenance decision-making	_OPP
Generative holistic optimization	_GEN
Future scenario based Lifelong self-learning system	_LSL
Human feedback	_HUM
Explainable and transparent decision making	_ETD

In the context of AI-PROFICIENT, 8 use cases have been selected to design, develop, and demonstrate the services provided by the project. During the elaboration of D1.3 (Pilot-specific demonstration scenarios) some use cases include Task 3.3 as potential contributor for system level proactive maintenance strategy (see Table 3). Nevertheless, when exploring more in details the use cases data availability and partners intention to support Task 3.5, no use case was dealing with line/system level scenario. Hence, we decided to build a scenario, based on Continental pilot site and more specifically Combiline data, to demonstrate the developed service. This mitigation measure has been presented in M18 review and approved. Furthermore, state-of-the-art techniques are at TRL level 3 and require further improvements. As such, the advances in this task are dealing with (1) advances and contribution related to reinforcement learning for maintenance of multi-component systems and (2) the proposal of a scenario to develop and demonstrate the service with respect to the Combiline system.

Table 3 - Original excerpt of expected partners involvement in T3.3 for each use case (from D1.3).

WP/Task	CONTI-2	CONTI-3	CONTI-5	CONTI-7	CONTI-10	INEOS-1	INEOS-2	INEOS-3
WP3- Platform AI analytics & decision-making support								
T3.3		UL	UL TEK IBE	INOS UL	INOS IBE			

For the above-mentioned reason, the service definition provided in D1.5, presented in Table 1 has been updated and is presented in Table 4.

Table 4 - Updated S_OPP service description.

Service ID	S_OPP
Service input and dependency on other services:	The service should be implemented at the cloud and require at least as input: • Logistic support information such as maintenance team/skill, spare parts, etc. • List of maintenance actions and related costs. • Production planning and constraints.
Service output:	The output of the service is the maintenance planning over the maintenance decision horizon. It consists of the group of components to be maintained together with the date of the operation.
High level service description:	The purpose of the service is to provide an optimal scheduling of the maintenance action to be performed on a line or system. This service aims at providing optimal dynamic maintenance planning considering both the requirements associated with the maintained system and its support one (e.g., maintenance crew) but also to be able to update efficiently the maintenance planning in a dynamic context (e.g., structure changes on main or support system, occurrence of new maintenance opportunities). Maintenance planning consists of the maintenance actions, for each component, and the time of operation. This planning is re-optimized in a dynamic way, i.e., which can be adapted/updated in presence of short-term information such as a new maintenance opportunity, new information related to the components/system health state, logistic support, or new impacting event available.

1.1 Problem statement

Maintenance involves technical and management tasks intended to sustain a component/system in, or restore it to, an operating state in which it can perform designated functions. Maintenance actions can be categorized into two broad groups which are corrective maintenance (CM) and preventive maintenance (PM). CM is also known as breakdown or run-to-failure maintenance, which repairs malfunctioned machines and is usually associated with high cost due to unexpected production losses. On the contrary, PM interventions are carried out on functioning machines to avoid their failures leading to reduce unplanned downtime cost. PM actions can be scheduled according to either age or state (degradation level) of machines. The later which is also known as condition-based maintenance (CBM) provides some advantages in comparison with the former. Its flexibly allows making maintenance decisions based on actual health condition of maintained machines instead of on a fixed calendar. Moreover, thanks to significant advances achieved in sensor technology recently which allows collecting rich degradation measurement information, CBM has become a popular approach for maintenance decision-making and optimization.

Maintenance optimization, given one or several objective functions, is related to 2 questions:

- Which maintenance actions should be performed? The goal of this optimization is to find the best maintenance actions to be performed on each component with respect to the state of the components/system.
- When should the maintenance actions take place? The goal of this optimization is to find the best time to perform the required maintenance actions.

The contributions of the task deal with both questions.

The contribution on the first question is related to scientific contribution, as the project is RIA project, and deals with maintenance policy optimization by considering the problem in the context of multi-components systems with dependencies.

The contribution related to the second question is the development of a service for maintenance scheduling optimization based on a proposed scenario with respect to the Combiline system for which only limited statical data is available.

The report is structured as follows. The second section summarizes the scientific contribution provided by the task on maintenance policy optimization. The publications detailing these contributions are available on the web site of AI-PROFICIENT. The third section describes the contribution of the task on the development of a maintenance scheduling optimization service. In that way, the mathematical formulation of the problem is reported in section 3.1, the dynamic grouping optimization algorithm is provided in section 3.2 and, finally, the application of algorithm to a case scenario concerning the packaging and cutting units of the Continental Combiline with the discussions/analyses on the obtained results is presented in sections 3.3 and 3.4) Section 4 draws some conclusions of the report.

2 Maintenance policy optimization: scientific contributions on reinforcement learning for multi-component systems

The AI technique envisioned to address maintenance optimization problems at system level is the approaches based on the framework of multi-agent deep reinforcement learning (MADRL) which aims to tackle some important gaps toward industrial applications listed as bellows:

- The number of maintenance decisions needed to be optimized for a complex industrial system increase exponentially in the number of components as well as number of available maintenance decisions for each component causing computational expense for traditional optimization methods.
- Traditionally, the explicit cost structures at component level such as setup cost, spare part cost, maintenance labor cost, component maintenance costs are required to build the cost model at system level. From a practical point of view, maintenance actions are often grouped in each maintenance intervention due to the component economic dependence, which leads to the fact that such individual costs are not recorded separately, instead, only total cost is documented. As a result, the availability requirement of separately collecting individual maintenance costs to construct the cost model at system level in almost all existing maintenance decision-making optimization algorithms for multi-component systems is less practical.
- The economic dependence between components allows to get economic benefit when components are grouped to maintain. Therefore, it is necessary to incorporate such kind of component dependence into maintenance models.

To overcome these gaps, we proposed first an ANN-based predictor for estimating maintenance cost at system level which allows to get rid of demand of accessing individual costs at component level. After that the trained system cost predictor is integrated into MADRL framework that allows to optimize maintenance decisions for large-scale multi-component systems. An illustration of the proposed AI-based maintenance approach is given in Figure 1.

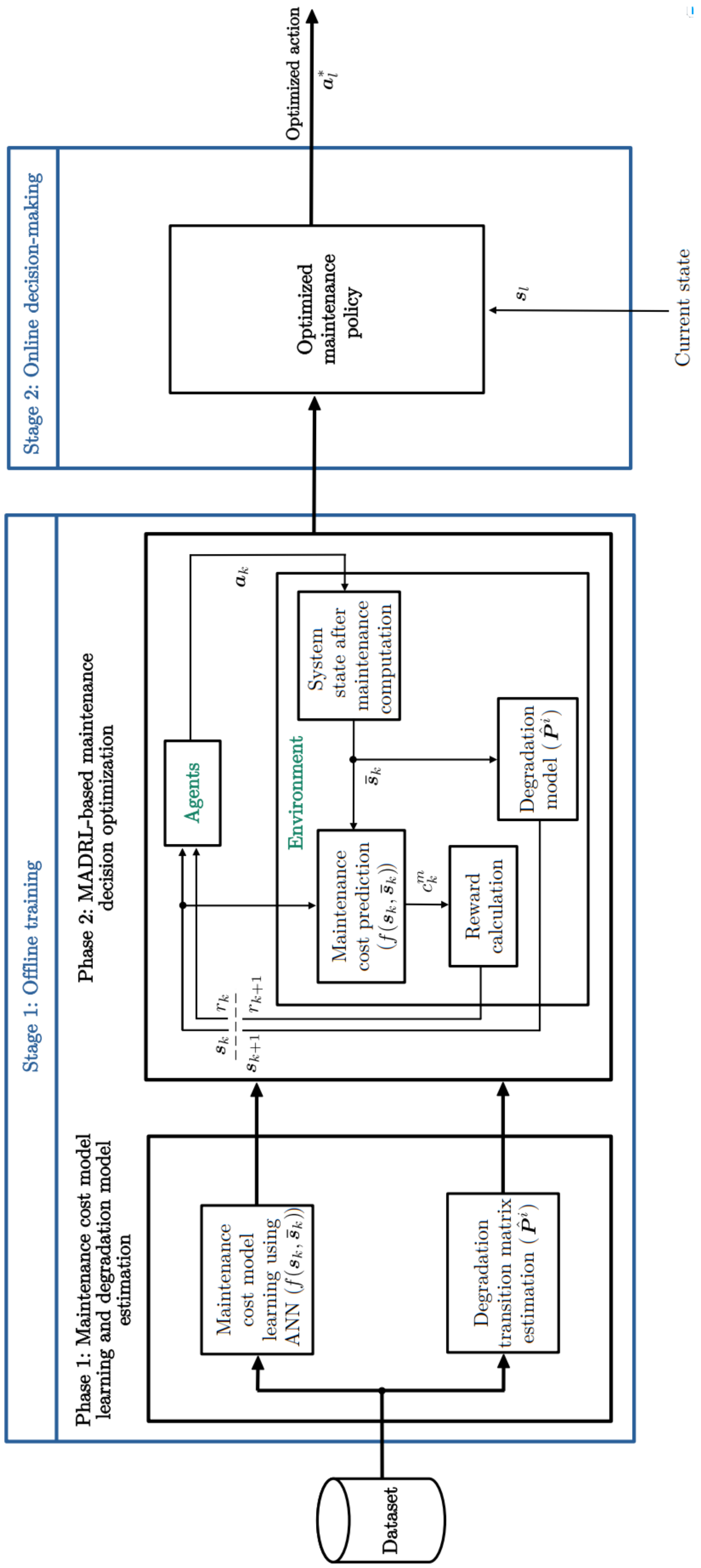

Figure 1- Illustration of AI-based maintenance approach for multi-component systems

The proposed maintenance approach consists of two main stages which are offline training and online decision-making. The first stage aims to optimize maintenance policy based on recorded data while the second one involves realizing optimized maintenance actions. It can be noticed that the first stage is a very important part of the proposed maintenance approach. Particularly, this stage is composed of two main phases. The first phase aims at learning system maintenance cost model using ANNs and at estimating component degradation probability transition matrices. The objective of the second phase is to construct an environment dedicated to MADRL algorithms that employs the trained cost model and the estimated matrices from the first phase, and to train learning agents to optimize maintenance policy by letting them interact with the constructed environment.

For more details about the proposed maintenance approach, please take a visit to the AI-PROFICIENT web site (following this link: https://ai-proficient.eu/resources/), the following articles are available for download:

- Van-Thai Nguyen, Phuc Do, Alexandre Voisin, Benoît Iung. Reinforcement learning for maintenance decision-making of multi-state component systems with imperfect maintenance. 31st European Safety and Reliability Conference, ESREL 2021, Sep 2021, Angers, France. ⟨10.3850/978-981-18-2016-8 304-cd⟩.
- Nguyen, V.-T., Do, P., Voisin, A., & Iung, B. (2022). Weighted-QMIX-based Optimization for Maintenance Decision-making of Multi-component Systems. PHM Society European Conference, 7(1), 360–367. https://doi.org/10.36001/phme.2022.v7i1.3319
- Van-Thai Nguyen, Phuc Do, Alexandre Voisin, Benoit Iung, Artificial-intelligence-based maintenance decision-making and optimization for multi-state component systems, Reliability Engineering & System Safety, Volume 228, 2022, 108757, ISSN 0951-8320, 10.1016/j.ress.2022.108757.

3 Maintenance scheduling optimization: demonstration scenario in Continental

In this section we develop a service, i.e., an algorithm for maintenance scheduling that considers constraints that are based on realistic scenario built in relation to the Continental pilot site. The optimization function is based on cost minimization while the constraints deal with components dependencies and the number of maintenance people available.

Chapter 3 is divided in three sections: the first section 3.1 provides the problem formulation, in which costs considered in the model and assumptions done for the simulation are reported; section 3.2 describes the dynamic grouping maintenance algorithm, including formulation of the economic profit considered in the algorithm and the genetic algorithm developed for maintenance scheduling optimization; the section 3.3 presents the built demonstration scenario based on the Continental pilot line, i.e. the Combiline, on which the algorithm was applied; finally, section 3.4 provides the results and discussion of the application.

3.1 Problem formulation

We consider a **series-system** consisting of N_C components. It is assumed that each component is preventively maintained **at least once** during a **short-term planning horizon** starting from t_{begin} to t_{end}. The planning horizon might be one week or one month, which depends on the company's actual situations.

The preventive cost of maintaining a component i is given by $C_M^i = C_S + C_P^i + C_U^i$ in which:

- C_S is the **setup cost** which is the same for all components which **can be shared** if several components are maintained together.
- C_P^i is the **component-specific preventive cost** which **cannot be shared.**
- C_U^i is the **unavailability cost** caused by the maintenance of component i since the system is unavailable during its maintenance duration. Specifically, if the preventive maintenance of component i is implemented during d^i time units, it leads to the unavailability cost $C_U^i = d^i.C_D$ where C_D is a positive constant representing downtime cost rate related to production loss. The unavailability cost **can be shared** if several components are grouped to maintain.

It is supposed that at time t_{begin} the maintenance duration (d^i) and execution time (t^i) of each component are known. In addition, there are N_{RM} repairman available to implement these preventive activities and **each repairman can only maintain one component at a time.** From a practical point of view, **the number of repairmen might be changed overtime** due to economic or technical reasons.

The objective is to find **optimal groups of maintenance activities** to implement as well as **optimal execution times for each group.**

3.2 Dynamic grouping maintenance algorithm

The proposed solution for the problem mentioned above is the dynamic grouping maintenance approach which consists of two steps: (1) the first step aims to evaluate the economic benefit if preventive maintenance activities are grouped to be implemented; (2) the objective of second step is to optimize maintenance groups and maintenance execution time for each group.

3.2.1 Economic profit formulation

Consider a group G_k of N_{G_k} preventive maintenance activities that are jointly executed, the economic benefit of G_k consists of the following parts:

- **Setup cost saving**

  Implementing a group of several maintenance activities requires only one setup cost. Therefore, the setup cost saving of executing group G_k is given by:

$$B_S^{G_k} = (|G_k| - 1).C_S = \left(N_{G_k} - 1\right).C_S$$

- **Unavailability cost saving:** The unavailability cost due to the maintenance of a component causing a system shutdown can be shared if several components are maintained together by N_{RM} repairmen. Particularly, this reduction cost is computed as follows:

$$B_U^{G_k} = \left(\sum_{i \in G_k} d^i - d^{G_k}(N_{RM})\right).C_D$$

  where $d^{G_k}(N_{RM})$ is the total maintenance duration of group G_k which depends on the number of repairmen N_{RM} as well as durations of all maintenance activities belonging to this group. It should be noted that if there is only one repairman available, all maintenance activities must be implemented sequentially. As a result, $d^{G_k}(N_{RM}) = \sum_{i \in G_k} d^i$, in other words, $B_U(G_k, N_{RM}) = 0$. For $N_{RM} > 1$, MULTIFIT algorithm is employed to find minimal values of $d^{G_k}(N_{RM})$.

- **Penalty cost**

If several components are grouped to be maintained, the preventive maintenance execution time of some components might be either anticipated or delayed. These changes might increase downtime cost or waste components' useful life. Therefore, if the preventive maintenance of component i is shifted Δ_t from its planned time t^i, we must pay something for it. In this scenario, we will use a penalty function given as below:

$$P^i(\Delta_t) = P^i(t - t^i) = \begin{cases} \alpha^i.(\Delta_t)^2 \; if \; \Delta_t \leq 0 \\ \beta^i.(\Delta_t)^2 if \; \Delta_t > 0 \end{cases}$$

It should be noted that $P^i(0) = 0$ when preventive maintenance execution time of component i remains unchanged.

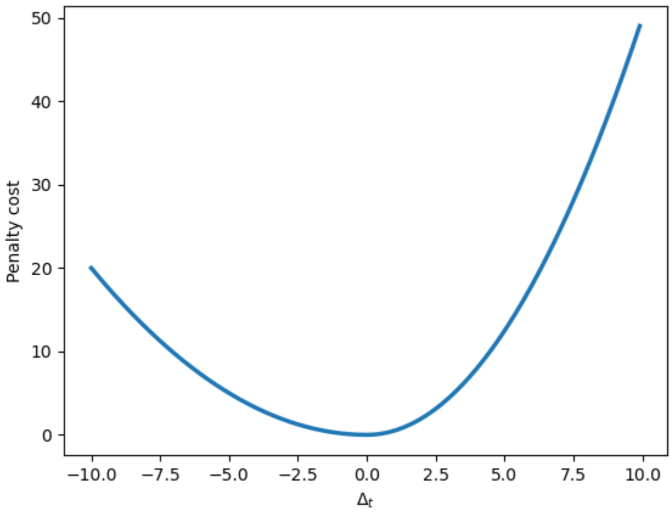

Figure 2 – Example of penalty cost function $P^i(\Delta_t)$ with $\alpha^i = 0.2$, $\beta^i = 0.5$

The penalty cost of group G_k is a function of time as:

$$P^{G_k}(t) = \sum_{i \in G_k} P^i(t - t^i)$$

The optimal execution time of group G_k denoted as t^{G_k} can be obtained numerically by optimizing $P^{G_k}(t)$.

Based on the above analysis, the cost benefit of implementing a group of maintenance activities G_k is given by the sum of the setup and unavailability cost savings, minus the penalty cost:

$$EB^{G_k} = B_S^{G_k} + B_U^{G_k} - P^{G_k}$$

3.2.2 Grouping optimization using Generic Algorithm (GA)

A grouping solution or grouping structure denoted by GS is a partition of N_{PM} preventive maintenance activities. A partition of $\{1, 2, \dots, N_{PM}\}$ is a collection of K mutually exclusive groups G_1, G_1, …, G_K which cover all N_{PM} maintenance activities, i.e., $G_j \cap G_l = \emptyset \; \forall \; j \neq l$ and $G_1 \cup G_2 \cup \dots \cup G_K = \{1, 2, \dots, N_{PM}\}$.

The grouping optimization problem is to find an optimal grouping structure $GS^* = \{G_1^*, G_2^*, \dots, G_K^*\}$ with corresponding optimal execution maintenance times for each group $t_{GS}^* = \{t_{G_1}^*, t_{G_2}^*, \dots, t_{G_K}^*\}$. Generic algorithm will be used to find GS^* and t_{GS}^*.

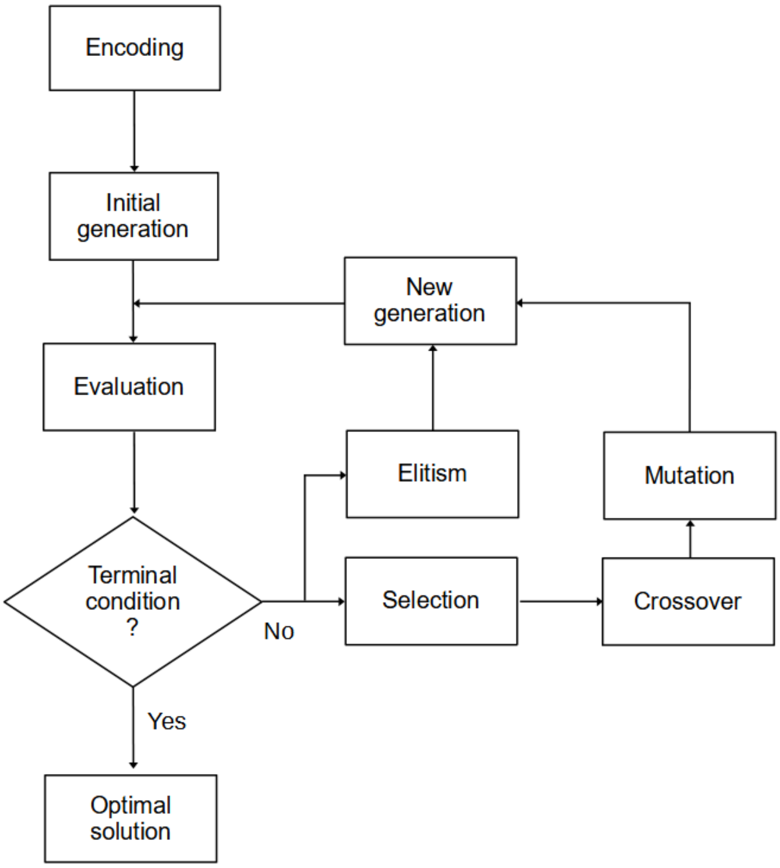

Figure 3 - Generic algorithm procedure

The general procedure to apply the GA to obtain optimal grouping solution is illustrated in Figure 3. The steps reported in the flowchart are described below:

- **Encoding:** Each individual of the GA population, i.e., a potential solution, is represented by an array denoted as S which consists of N_{PM} elements corresponding to N_{PM} preventive maintenance activities. It should be noted that if $S(i) = S(j)$, maintenance activity i and j are executed together. Considering, for example, an array S with 7 elements, $S = [1, 2, 3, 4, 2, 3, 1]$, that encodes the execution time of the maintenance intervention for each of the 7 elements: the 7 preventive maintenance activities are going to be grouped in 4 groups of maintenance activities, which are $G_1 = \{1, 7\}$, $G_2 = \{2, 5\}$, $G_3 = \{3, 6\}$, $G_4 = \{4\}$.
- **Initial population:** The population size used for GAs is usually chosen between 60 and 100. To generate an initial solution, each element in an array used to encode a grouping solution is randomly chosen between 1 and N_{PM}.
- **Evaluation:** Firstly, the array encoding each solution is decoded to get groups of maintenance activities $G_1, \dots, G_k$. After that, the economic benefit of each group is computed. The fitness of a grouping solution is the sum of the economic benefit of all groups.

- **Elitism:** Two best solutions in current population are directly copied to the next generation to protect the best solutions from the high level of disruption.
- **Selection:** Pairs of parent solutions used in the crossover step are selected by using the “linear ranking” selection. Particularly, the population is first sorted according to the fitness values of its elements in ascending order and then is categorized into several groups. Next, a parent solution is randomly chosen from these s groups according to group probabilities.
- **Crossover:** Each pair of parent solutions is randomly selected for crossover operator with a self-adaptive probability computed as bellows:

$$p_c = \begin{cases} p_c^{max} - \dfrac{(p_c^{max} - p_c^{min})(f_c - f_{avg})}{f_{max} - f_{avg}}, if\ f_c > f_{max} \\ p_c^{max} \qquad , otherwise \end{cases}$$

  in which:

  - p_c^{min} and p_c^{max} are respectively the lower and upper bound of crossover probability.
  - f_{avg} and f_{max} are respectively the average and the maximal fitness of all solution in the population.
  - f_c is the higher fitness of the two parent solutions’ fitness.

  The two-point crossover mechanism will be used to produce two children from two selected parents. Specifically, two points are first randomly chosen to divide each parent solution into three parts. The elements between these two points are then exchanged to create new child solutions.

- **Mutation**: The objective of mutation step is to prevent GA optimizer falling into local optimal solution. Each child solution produced by crossover has a mutation probability (mutation rate) computed by:

$$p_m = \begin{cases} p_m^{max} - \dfrac{(p_m^{max} - p_m^{min})(f_{max} - f_m)}{f_{max} - f_{avg}}, if\ f_m > f_{avg} \\ p_m^{max} \qquad , otherwise \end{cases}$$

  in which:

- p_m^{min} and p_m^{max} are respectively the lower and upper bound of mutation probability.
- f_{avg} and f_{max} are respectively the average and the maximal fitness of all solution in the population.
- f_m is the fitness of the parent solution.

  If the mutation occurs in a child solution, a random maintenance activity in a group will be randomly moved to another group.

3.3 Demonstration scenario in Continental

In this demonstration scenario, we consider the grouping maintenance problem for the cutting unit (CONTI-5) and the packaging system (CONTI-7) which are two of the most important units of the Combiline.

Specifically, the core of the cutting system is the circular blade which performs a considerable amount of cuts each day. These repetitive cuts make the blade to be worn out and it needs to be replaced frequently to be in optimal condition to perform accurate cuts that will not compromise the final quality of the tires. Moreover, if the blade is worn, its’ capability to produce good cuts is compromised leading to quality issues downstream of the production line. Therefore, when this kind of problem is detected, the treads are segregated without going to the next step in the process.

The tread after being cut needs to be packed automatically on trolleys in the packing unit to go to tire building process. The packing unit is made of multiple belts and the wear of some of them can create misalignment on the trolley. As the treads are managed via robots in the next steps, the alignment needs to be as perfect as possible. However, due to the wear condition of the belts it can happen that the treads are not correctly placed onto the leaf. Hence, also in this case, the problem is detected, and the treads are segregated.

Based on the data provided by the company, we decided to fix the length of the planning horizon equal to be one month (30 days). The hyper-parameters of the components of the two subsystems used in this demonstration are reported in Table 5. The alpha and beta parameters are related to the penalty function previously presented in section 3.2.1. It should be noted that parameters in green color are computed from data provided by the company, whereas parameters in red color are hypothesized due to the lack of data. We also hypothesize that the setup cost C_S is 500 CU, and the downtime cost rate C_D is 100 CU per hour.

Table 5 - Data of the considered components of the Combiline. Green data are real data from Combiline and red data are hypothesized.

ID	Component	Alpha (α^i)	Beta (β^i)	Replacement time(s) (t^i) (hours)	Maintenance duration (d^i) [hours]	Replacement cost (C_P^i) [CU]
0	Circular blade	5	20	8, 512	2	423.81
1	Standard roller	5	16	10	1.5	228.33
2	Adjustable shock absorber	5	8	15	0.5	88.91
3	Flat belt	5	6	34, 538	0.25	37.46
4	Round vacuum cup	5	7.2	120	0.4	20.7
5	Tige-Fourche 1	5	6	125, 509	0.25	10.71
6	Tige-Fourche 2	5	6	340, 684	0.25	2.6
7	Suction cup	5	6	40, 376, 712	0.25	21.08
8	Cylinder clevis 1	5	8	30, 366, 702	0.5	2.6
9	Cylinder clevis 2	5	6	50, 386	0.25	12.24
10	Cylinder clevis 3	5	6	60, 396	0.25	2.6

There are 21 preventive maintenance activities within the considered planning horizon that are listed in Table 6. Particularly, each preventive maintenance activity is represented by a tuple of two elements that are respectively the execution time and the ID of the component on which maintenance is carried out. For example, maintenance activity encoded by (10, 1) is carried out at time 10 on component 1 (standard roller).

Table 6 – Example of preventive maintenance activities scheduled in the time horizon.

ID	Maintenance activity	ID	Maintenance activity	ID	Maintenance activity
1	(8, 0)	8	(125, 5)	15	(30, 8)
2	(512, 0)	9	(509, 5)	16	(366, 8)
3	(10, 1)	10	(340, 6)	17	(702, 8)
4	(15, 2)	11	(684, 6)	18	(50, 9)
5	(34, 3)	12	(40, 7)	19	(386,9)
6	(538,3)	13	(376, 7)	20	(60,10)
7	(120, 4)	14	(712, 7)	21	(396, 10)

3.4 Result and discussion

In the following we present, first, the result of the service with the scenario described in the above section. We then study the influence of some specific parameters that can vary according to the use case considered.

3.4.1 Result and discussion from the standard scenario

We run the developed GA algorithm to find the optimal grouping solutions for three values of the number of available repairmen. It should be noted that GA does not guarantee the finding of global optimal solution. Therefore, we need to run the algorithm several times to obtain the best results which are given in the Table 7, Table 8 and Table 9.

Table 7 – Scenario with one repairman

Number of repairmen	Saving cost [%]	Unavailability period [hours]	Optimal group	Optimal execution time [hours]	Maintenance duration [hours]
1	43.02%	11.15	(1, 3, 4)	9.5	4.0
			(5, 12, 15)	33.889	1.0
			(18, 20)	54.545	0.5
			(7, 8)	122.049	0.65
			(10, 16)	351.818	0.75
			(13, 19, 21)	385.375	0.75
			(2, 6, 9)	515.613	2.5
			(11, 14, 17)	698.375	1.0

Table 8 – Scenario with 2 repairmen

Number of repairmen	Saving cost [%]	Unavailability period [hours]	Optimal groups	Optimal execution time [hours]	Maintenance duration [hours]
2	46.49%	6.65	(1, 3, 4)	9.5	2.0
			(5, 12, 15)	33.889	0.5
			(18, 20)	54.545	0.25
			(7, 8)	122.049	0.4
			(10, 16)	351.818	0.5
			(13, 19, 21)	385.375	0.5
			(2, 6, 9)	515.613	2.0
			(11, 14, 17)	698.375	0.5

Table 9 – Scenario with 3 repairmen

Number of repairmen	Saving cost [%]	Unavailability period [hours]	Optimal groups	Optimal execution time [hours]	Maintenance duration [hours]
3	48.49%	6.15	(1, 3, 4)	9.5	2.0
			(5, 12, 15, 18, 20)	41.133	0.5
			(7, 8)	122.049	0.4
			(10, 16)	351.818	0.5
			(13, 19, 21)	385.375	0.25
			(2, 6, 9)	515.613	2.0
			(11, 14, 17)	698.375	0.5

As can be seen from the results reported in Table 7, Table 8 and Table 9, the maintenance grouping algorithm allows saving 46.49% of cost even if there is only one repairmen available. In this case the cost reduction is mainly due to the saving of set-up costs when several maintenance activities are carried out together.

In the case where there is more than one repairman available, cost can be saved more by saving setup cost as well as by reducing the system unavailability. Particularly, the percentage of money economized if there are two and three repairmen available is 46.49% and 48.49% respectively. It can be noticed that the system's unavailability is reduced significantly by (11.15 - 6.65) / 11.15 = 40.36 % and (11.15 - 6.15) / 11.15 = 44.84 % when there are 2 and 3 repairmen available respectively.

It can be also noticed that the optimal maintenance groups in the case where the number of repairmen is fixed to 1 and 2 are the same, however, the duration needed to carry out the maintenance intervention on each group are different. It is worth noting that more maintenance activities can be grouped to implement when there are more repairmen available. Particularly, in the case where there are 1 and 2 repairmen available, maintenance activities numbered 5, 12, 15, 18, 20 are put into two different groups to carry out, i.e., group (5, 12, 15) and (18, 20), while they are in the same group when the number of repairmen is 3.

3.4.2 Sensitivity analysis

This section aims to get some insights on the impact of several hyper-parameters on the characteristics of optimal grouping solutions. Particularly, the number of repairmen and the setup cost are considered as the factors used for the analysis in the following subsections.

3.4.2.1 Number of repairmen

We first study the impact of the number of repairmen on the saving cost and the system unavailability by running the grouping algorithm for different values of N_{RM} range from 1 to 6 (the other parameters are the same as used in the above section). The result of this study is illustrated in Figure 4 and Figure 5.

It can be noticed that the evolution of the saving cost and the system unavailability is opposite because the increase of saving cost in fact requires the decrease of system unavailability. Moreover, it is interesting to see that the optimal policy is unchanged when there are enough repairmen available to carry out maintenance activities ($N_{RM} \geq 3$). This can be explained by the assumption that an increase in the number of repairmen do not lead to an extra cost as well as by the optimization mechanism of MULTIFIT algorithm which assumes that a maintenance activity can be carried out by only one repairman. For example, considering the case where, there is one maintenance intervention needed to implement, while there are two repairmen available. The intervention can be done by only one of the two repairmen; thus, the maintenance duration is not reduced.

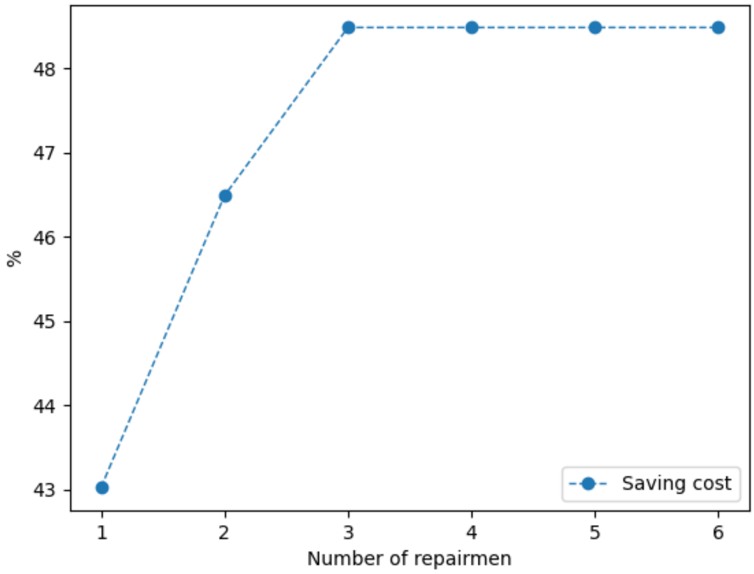

Figure 4 – Impact of number of repairmen on saving cost.

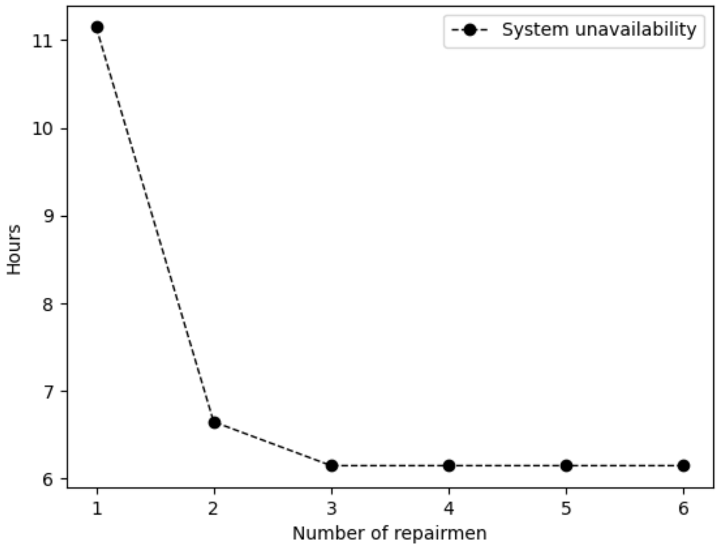

Figure 5 – Impact of number of repairmen on system unavailability.

3.4.2.2 Setup cost

In this section, we investigate how the setup cost affects the way maintenance interventions are grouped to implement by running the grouping algorithm for different values of C_S. The result of this simulation is illustrated in Figure 6.

It can be noticed that maintenance activities tend to be carried out in groups when setup cost is high, whereas there are more maintenance activities which are implemented separately when setup cost is low. This can be explained by the fact that the higher the setup cost is, the higher economic benefit we can get when maintenance activities are implemented together.

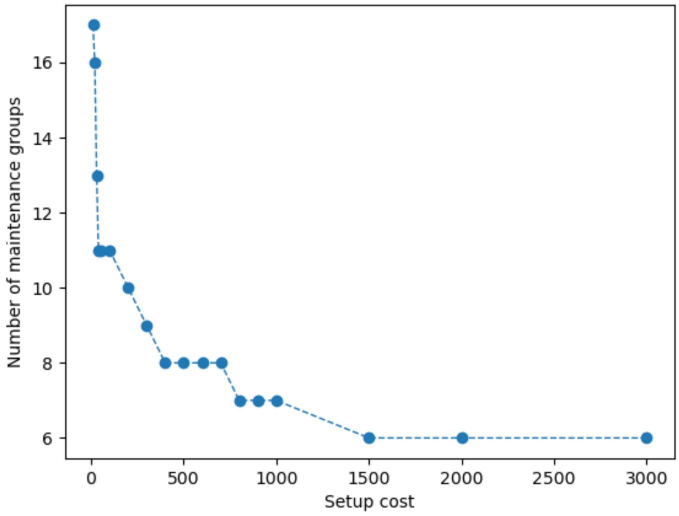

Figure 6 – Impact of setup cost on number of maintenance groups

4 Conclusions

The Deliverable reports the work developed during Task 3.3 dealing with maintenance optimization at the line level (considering several components and subsystems).

The two main outcomes of this task detailed in the documents are:

- A scientific contribution to maintenance decision-making optimization of multi-component systems considering the impact of component dependencies based on MADRL framework;
- A service for dynamic opportunistic maintenance scheduling, namely an algorithm for the scheduling of maintenance activities that considers constraints based on a realistic scenario built in relation to the Continental pilot site. The algorithm was, therefore, applied on a demonstration scenario based on Continental pilot, i.e., the Combiline, and different analysis were performed varying the parameters (as number of repairmen, setup cost) in input to the model. The application of the algorithm shows that it is possible to reduce unavailability periods and save costs proper scheduling of maintenance activities on critical components of the line.

Acknowledgements

This project has received funding from the European Union’s Horizon 2020 research and innovation program under grant agreement No 957391.