AN ADVANCED EVOLUTIONARY ALGORITHM INTEGRATING GENETIC ALGORITHM AND BRANCH AND BOUND FOR JOINT MAINTENANCE GROUPING AND ROUTING OPTIMIZATION

Thi-Hoang-Giang Tran*, Ho-Si-Hung Nguyen

The University of Danang - University of Science and Technology, Vietnam

*Corresponding author: tthgiang@dut.udn.vn

(Received: May 01, 2025; Revised: June 16, 2025; Accepted: June 17, 2025)

DOI: 10.31130/ud-jst.2025.23(6A).264E

Abstract - In recent years, growing international trade and advancements in distributed control, information, and logistics have led to the development of geographically dispersed production systems (GDPS). The GDPS is a production network in which production is carried out across several sites located far apart from each other. Optimizing the maintenance plan for such a system requires considering the potential for grouping maintenance tasks and the impact of maintenance itineraries on transportation costs. A joint optimization between maintenance grouping and routing is essential to reduce both preparation and transportation costs. This paper proposes an advanced evolutionary algorithm integrating the Genetic Algorithm (GA) and Branch and Bound (BAB), called GaB. The proposed algorithm's effectiveness is demonstrated with a numerical example of a typical GDPS with five sites. Results show a 7.44% cost saving compared to individual maintenance and a 59% reduction in computational time compared to the comprehensive search approach.

Key words - Geographically dispersed production system; maintenance plan; branch and bound algorithm; genetic algorithm; advanced evolutionary algorithm.

1. Introduction

In the globalization process, clusters of customers tend to be located in a scattered way and form distributed market areas. In order to adapt geographical dispersion of customer areas, many manufacturing companies recently have changed their production models to Geographically Dispersed Production System (GDPS). The GDPS is a distributed manufacturing system including dispersed production sites and an administration center in charge of production and maintenance management. The production sites can be considered as subsystems composed of multiple components. For instance, Honda Corporation operates five engine production facilities located across Japan, China, Thailand, Italy, and the USA. These five divisions are managed by the Kumamoto headquarters, situated in Ozu-machi, Kikuchi-gun, Japan (see Figure 1). By adopting a distributed manufacturing model, Honda effectively reduces transportation and logistics costs, minimizes negative environmental impacts, enhances the quality of its products and services, and better satisfies customer demands. Additionally, the study referenced in [18] indicates that distributed mini-factory networks represent a growing trend for the future. This paper further highlights the application of the Geographically Dispersed Production System (GDPS) model in industries such as furniture manufacturing, glass production, and tire manufacturing. The GDPS often encounters significant challenges in adhering to standards and regulations, managing production efficiently, and most notably, planning and optimizing maintenance activities, primarily because of the physical separation between production locations.

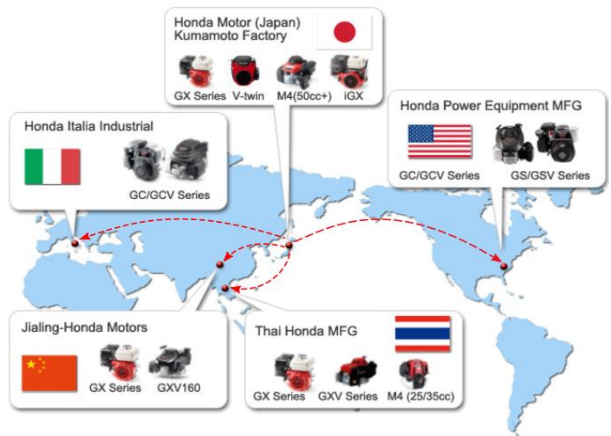

Figure 1. *The geographically dispersed production model of Honda corporation*

In the maintenance optimization framework of multicomponent systems, grouping maintenance has been recently recognized as a powerful approach allowing for reducing maintenance costs by jointly maintaining some components. Grouping maintenance activities to take advantage of profit from economic dependencies was first proposed by Dekkert [1]. Grouping strategies include stationary (fixed schedule and components) and dynamic (updated with short-term data) approaches [2]. The grouping maintenance is even more interesting in the case of GDPSs because it helps to reduce not only the preparation costs but also the travel distance, travel time, and transportation cost if the itinerary of the maintenance team or spare part travel is well defined. Consequently, numerous studies have focused on formulating maintenance grouping strategies tailored to such systems. These approaches aim to optimize the grouping of maintenance tasks for deteriorating components to enhance overall economic efficiency [3]. However, the maintenance routing has not been considered. Consequently, the maintenance has not been performed most effectively.

In contrast to these works, some research efforts have primarily concentrated on addressing the maintenance routing problem, placing less emphasis on maintenance grouping. In these studies, the maintenance routing is usually considered as a vehicle routing problem (VRP) and solved by using meta-heuristic methods. For example, Liu et al. [4] applied adaptive large neighborhood search considering time windows and synchronized services; Moura [5] used mixed-integer programming with an efficient heuristic to reduce cost and time; and Soto et al. [6] combined multiple neighborhoods and Tabu search. However, these methods lack optimality guarantees. Therefore, exact algorithms like Exhaustive Search, Branch and Cut [7], and Branch and Bound [8] are often used, especially for systems with few production sites.

Recent studies have begun considering jointly both maintenance grouping and routing. For instance, Lopez-Santana et al. [9] proposed separate models for grouping (based on economic benefit) and routing (to minimize travel costs), solved sequentially using a heuristic inspired by planner intuition. Stalhane et al. [10] applied arc-flow and path-flow formulations to generate component groups and routing options. Dai et al. [11] addressed offshore wind farm maintenance using a MILP for route and schedule planning. Zhang et al. introduced Duo-ACO to solve joint scheduling and routing. López-Santana et al. [12] decompose the multi-period combined maintenance-and-routing optimisation into a column-generation framework. Delavernhe et al. [13] extend the idea by simultaneously selecting maintenance operations and routing technicians and demonstrate considerable cost reductions on benchmark instances. Moros et al. [14] propose a multi-timescale architecture that links long-term strategic plans. The maintenance routing and scheduling problem is addressed for tasks allocated to a specific time frame, as determined by the tactical planning model. The study confirms that routing decisions and stochastic operating conditions must be treated simultaneously if cost-effective maintenance is to be achieved. To the best of our knowledge, all the existing works solve the joint grouping and routing maintenance optimization sequentially. In addition, the GDPS has not been addressed in these studies. To address this, we propose the GaB algorithm, combining Genetic Algorithm and Branch and Bound to optimize maintenance planning in large-scale and multi-component systems.

2. The mathematical models

2.1. Definition of the problem and working assumptions

We consider a GDPS comprising n components distributed across m spatially separated sites. The failure behavior of component i $(0 < i \le n)$ is represented using the Weibull distribution. To maintain the long-term operational reliability of the GDPS, two types of maintenance activities are incorporated: preventive maintenance (PM) and corrective maintenance (CM). Preventive maintenance is scheduled at specific intervals to reduce the likelihood of frequent failures, restoring components to a "like-new" condition. In contrast, corrective maintenance is performed immediately following a failure, returning the component to its previous "used" condition. In this study, CM activities are handled locally by maintenance teams stationed at each site and are assumed to have negligible duration. Conversely, PM operations are carried out by a centralized maintenance team, which often requires significant travel from a maintenance hub to various production locations. This necessitates the transport of both personnel and essential resources (e.g., spare parts, tools) to and from the sites. Additionally, it is assumed that several components, grouped as G_k, are preventively maintained together. These components are located in m_s different sites. The itinerary of the maintenance team/spare part to the PM of all components of the group should contain all m_s sites. The maintenance itinerary is represented by the decision variable X_{jv} as follows:

$$X_{jv} = \begin{cases} 1, & if\ site\ j\ is\ the\ vth\ location\ visited \\ 0, & otherwise \end{cases} \quad (1)$$

Hence, the maintenance itinerary is determined by the following $(m_s \times m_{G_k})$ matrix X with m_s being the number of all sites of a system and m_{G_k} being the number of sites in group G_k:

$$X = \begin{bmatrix} X_{11} & \cdots & X_{1m_{G_k}} \\ \vdots & \ddots & \vdots \\ X_{m_s1} & \cdots & X_{m_s m_{G_k}} \end{bmatrix} \quad (2)$$

With

$$\sum_{j=1}^{m_s} X_{jv} = 1, \forall v = [1, 2 \dots m_{G_k}];$$

$$\sum_{v=1}^{m_{G_k}} X_{jv} \le 1, \forall j = [1, 2 \dots m_s]$$

As an example, a GDPS architecture consisting of 5 sites $(m_s = 5)$ and a maintenance itinerary passing through 4 locations $(m_{G_k} = 4)$ is fully defined by the following matrix X:

$$X = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

It means that the maintenance team has implemented an itinerary as follows 0-2-5-3-1-0, where 0 is a representation of the global maintenance center. Note that the maintenance itinerary can be transformed from a matrix of routing plans to a vector of routing plans. This vector, denoted I_{G_k} , includes a global maintenance center and sites containing maintained components as shown above. The following assumptions are considered. The following assumptions are considered:

- All maintenance itineraries begin and end at a global maintenance center

- A component failure results in the shutdown of its respective production site

- Disruption at one site does not affect the operation of other sites

- Maintenance team and necessary resources are assumed to be readily available at all times

- Each component requires a specific level of technical expertise for maintenance

- The time needed for corrective maintenance is considered negligible relative to the overall planning horizon

The following notations are used:

- t_{G_k}: Departure time of the maintenance team from the maintenance center;

- ω_i : Duration required to complete preventive maintenance for component *I*;

- C_i^{sp}: Cost associated with the spare parts needed for component *I*;

- C_i^{dt}: Financial loss incurred due to the downtime of component *I*;

- S_j^0: Site-preparation cost of components *i* at site *j*;

- $S_{i_j}^{tr}$: Transportation cost incurred when traveling from the central maintenance center to site *j*, where the component is serviced;

- R^{tr}: Rate applied to calculate transportation expenses;

- C_i^{lb}: Labor cost associated with performing a PM activity on component *I*;

- l_i: Required skill level of the maintenance personnel needed for component *I*;

- $R_i^{lb}(l_i)$: Labor cost rate corresponding to the technician's skill level l_i;

- t_{qj}^{tr}: Travel time required to move directly from site *q* to site *j*;

- L_{qj}: Physical distance between site *q* to site *j*.

Assume that the components of group G_k are tentatively maintained at their individual PM dates t_{i^m}. To be jointly maintained, the individual PM dates have to be adjusted, and the components of G_k are actually preventively maintained at t'_{i^m} instead of t_{i^m}, with $i \in G_k$. Let TLS_q denote the time that the maintenance team leaves site *q*, t_{qj}^{tr} is the travel time from site from site *q* to site *j*, the actual PM dates of components of group G_k located on site *j* are then $t'_{i^m} = TLS_q + t_{qj}^{tr}$. t'_{i^m} is also the grouped PM date of components of group G_k located at site *j*: $t'_{i^m} = TAS_j$. The maintenance team will leave site *j* at the time $TLS_j = TAS_j + \sum_{i\in G_k; i\in j} \omega_i$, and move to the next maintenance site.

2.2. Cost model and grouping economic profit model

This section aims to develop a cost model for components undergoing joint preventive maintenance. Using this model, an economic profit framework is constructed by comparing the cost of grouped maintenance activities with that of performing maintenance on each component individually.

2.2.1. Cost model

To perform a preventive maintenance action for group G_k, maintenance costs are paid for items:

- Spare part cost of a group ($C_{G_k}^{sp}$)

$$C_{G_k}^{sp} = \sum_{i\in G_k} C_i^{sp} \tag{3}$$

- Downtime cost of a group ($C_{G_k}^{dt}$)

$$C_{G_k}^{dt} = \sum_{i\in G_k} R_i^{dt}\, \omega_i \tag{4}$$

- Labor cost of a group ($C_{G_k}^{lb}$)

$$C_{G_k}^{dt} = R_{G_k}^{lb}(l_{max}). \sum_{i\in G_k} \omega_i \tag{5}$$

where: $R_{G_k}^{lb}(l_{max})$ depends on $l_{max} = \max_{i\in G_k} l_i$

- Site-preparation cost of a group ($S_{G_k}^0$)

$$S_{G_k}^0 = \sum_{j=1}^{m_s} \sum_{v=1}^{m_{G_k}} X_{jv} S_j^0 \tag{6}$$

Assume that the preparation-site cost at a site *j*, denoted S_j^0, is the same for the PM of a component.

- Transportation cost of a group ($S_{G_k}^{tr}$)

$$S_{G_k}^{tr} = R^{tr}. \left(L_{0v_1} + L_{v_{end}0} + \sum_{j=1}^{m_s} \sum_{v=2}^{m_{G_k}} \sum_{m=1, m\neq j} X_{jv}. L_{m(v-1)}. L_{mj} \right) \tag{7}$$

where: $L_{G_k}(I_{G_k})$ denote the total travel distance when the maintenance team follows the maintenance itinerary I_{G_k}.

The total cost of preventive maintenance tasks during one maintenance itinerary is expressed as follows:

$$C_{G_k}^p = C_{G_k}^{sp} + C_{G_k}^{dt} + C_{G_k}^{lb} + S_{G_k}^0 + S_{G_k}^{tr}$$
$$= \sum_{i\in G_k} C_i^{sp} + \sum_{j\in G_k} \left(R_{jG_k}^{dt}. \sum_{i\in G_k; i\in j} \omega_i \right) + R_{G_k}^{lb}.\omega_{G_k} + \sum_{j=1}^{n_s} \sum_{v=1}^{m_s} X_{jv} S_j^{sp}$$
$$+ R^{tr}. \left(L_{0v_1} + L_{v_{end}0} + \sum_{j=1}^{m_s} \sum_{v=2}^{m_{G_k}} \sum_{m=1, m\neq j} X_{jv}. L_{m(v-1)}. L_{mj} \right) \tag{8}$$

2.2.2. Grouping economic profit model

To assess the efficiency of maintenance grouping strategies, the concept of grouping economic profit is commonly employed. This metric represents the cost savings achieved by performing maintenance on components as a group rather than individually. The grouping economic profit of group G_k , denoted *EPG*, can be expressed as follows:

$$EPG_{G_k}(t_{G_k}, I_{G_k}) = \sum_{i\in G_k} C_i^p - C_{G_k}^p - \Delta H_{G_k}(t_{G_k}, I_{G_k})$$
$$= \left(\sum_{i\in G_k} S_i^0 - S_{G_k}^0 \right) + \left(\sum_{i\in G_k} S_i^{tr} - S_{G_k}^{tr} \right)$$

$$-\left(C_{G_k}^{lb}-\sum_{i\in G_k}R_i^{dt}.\omega_i\right)-\Delta H_{G_k}\left(t_{G_k},I_{G_k}\right)$$
$$=\Delta S_{G_k}^{0}+\Delta S_{G_k}^{tr}-\Delta C_{G_k}^{lb}-\Delta H_{G_k}\left(t_{G_k},I_{G_k}\right) \quad (9)$$

where,

$\Delta S_{G_k}^{0}$ is site-preparation cost saving due to reduction preparation cost of and be calculated as follows:

$$\Delta S_{G_k}^{0}=\sum_{i,j\in G_k}S_{i_j}^{0}-\sum_{j=1}^{m_s}\sum_{v=1}^{m_{G_k}}X_{jv}S_j^{0} \quad (10)$$

$\Delta S_{G_k}^{tr}(I_{G_k})$ is transportation cost saving due to the reduction of travel distance and is determined by:

$$\Delta S_{G_k}^{tr}(I_{G_k})=R^{tr}.\left[\sum_{j\in G_k}n_j^{G_k}.\left(L_{0j}+L_{j0}\right)-\left(L_{0v_1}+L_{v_{end}0}+\sum_{j=1}^{m_s}\sum_{v=2}^{m_{G_k}}\sum_{m=1,m\neq j}X_{jv}.L_{m(v-1)}.L_{mj}\right)\right] \quad (11)$$

$\Delta C_{G_k}^{lb}$ is the labor-loss cost and is calculated as Equation:

$$\Delta C_{G_k}^{lb}=R_{G_k}^{lb}.\omega_{G_k}-\sum_{i\in G_k}R_i^{lb}.\omega_i \quad (12)$$

$\Delta H\left(t_{G_k},I_{G_k}\right)$ is the incurred penalty cost when components are jointly maintained. The penalty cost $\Delta H_{G_k}(t)$ of a group G_k is rewritten as follows:

$$\Delta H\left(t_{G_k},I_{G_k}\right)=\sum_{I_v\in G_k}\sum_{i\in G_k;i\in I_v}h_i\left(t_v(I_v)-t_i\right) \quad (13)$$

From Equation (9), the optimal maintenance date ($t_{G_k}^{*}$) and itinerary ($I_{G_k}^{*}$) can be determined as

$$\left(t_{G_k}^{*},I_{G_k}^{*}\right)=\underset{t_{G_k},I_{G_k}}{\arg\max}\,EPG_{G_k}(t_{G_k},I_{G_k}) \quad (14)$$

In summary, the economic profit model is represented by the total economic profit of group G_k as shown in Equation (9).

In a short-term planning horizon, there are many PM activities. Therefore, a grouping solution of this horizon may contain several groups. A grouping solution or grouping structure can be defined as a collection of mutually exclusive group $SG=\{G^1,G^2,\dots,G^e\}$ with $G^h\cap G_k=\emptyset$ and $G^1\cup G^2\cup\dots\cup G^e$ covers all maintenance activities in the planning horizon. In a grouping structure, the minimum number of groups is one group (when all maintenance components are maintained in a maintenance itinerary, and the maximum number of groups can equal to the number of components (when each component is maintained individually). The total economic profit of grouping structure GS can be calculated as:

$$EPS(GS)=\sum_{G_k\in GS}EPG_{G_k}\left(t_{G_k},I_{G_k}\right) \quad (15)$$

$EPS(GS)$ represents the performance of grouping structure (GS). The grouping performance depends on grouping structure (GS), departure time of the maintenance team (t_{G_k}) and maintenance route (I_{G_k}) to maintain each group G_k of GS .

3. The proposed GaB algorithm

This section introduces an advanced hybrid algorithm, referred to as GaB, which integrates Genetic Algorithm (GA) and Branch and Bound (BAB) techniques to optimize both maintenance grouping and routing in the proposed framework. Genetic Algorithms are widely recognized for their effectiveness in solving complex optimization problems in maintenance planning, particularly in large-scale systems involving multiple components and dispersed sites. The traditional GA process begins with the random generation of potential grouping configurations. In each iteration, these configurations are evaluated based on their total economic profit (EPS), which serves as the fitness or objective function. High-performing groupings are then selected and refined through genetic operations such as crossover, mutation, and elitism. The algorithm proceeds until a predefined stopping condition, such as reaching a maximum number of generations, is satisfied. However, relying solely on GA poses challenges in effectively solving the routing aspect of the maintenance plan. To overcome this limitation, a Branch and Bound algorithm is embedded within the GA framework, enhancing its capability to address routing optimization. The structure of the proposed GaB algorithm is illustrated in Figure 2, and the detailed procedure is outlined below:

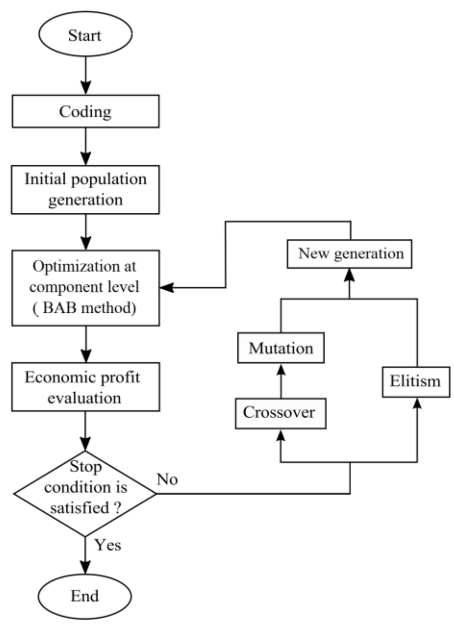

Figure 2. *The principle of GaB for grouping maintenance and routing optimization problem*

- *Step 1 (Coding)*: This step defines the way to generate a grouping structure in GaB. An array including elements, denoted GS, represents a grouping structure. For example, The coding of a grouping structure containing 3 groups $G_1=\{1,2\}$, $G_2=\{3,5\}$, $G_3=\{4\}$ is shown in Figure 3.

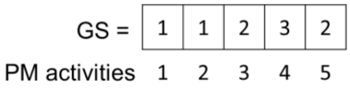

Figure 3. *Encoding for a grouping structure*

- *Step 2 (Generating a population of grouping structures)*: GaB randomly generates an initial population of grouping structures. Let length(GS) denote the maximum number of groups in a grouping structure. length(GS) equal to the length of the array GS. A grouping structure is created by choosing randomly the number of groups in [1; length(GS)]. Next, all PM activities are randomly put into the chosen groups.

- *Step 3 (Optimization at component level):* This step aims to find the best maintenance itinerary ($I^*_{G_k}$) and the best departure time ($t^*_{G_k}$) for a group G_k in terms of economic profit of grouping structure (*EPG*) presented as shown in Equation 9. Based on these results, the performance of a grouping structure can be evaluated at the next step.

- *Step 4 (Evaluating the performance of grouping structure*): Based on the grouping structure as well as all optimal itineraries and departure times of its groups, the performance of a grouping structure in the population is assessed by fitness functions, denoted in Equation 15.

- *Step 5 (Elitism)*: *This step aim*s to protect the best grouping structure from the high level of disruption. To do that, the two best grouping structures are directly copied to the next generation.

- *Step 6 (Crossover)*: Crossover is performed by combining a pair of grouping structure (parent) to create next grouping structure (new generation) that preserves their characteristics. Firstly, two PM activities are randomly chosen as the crossover points. Then, the elements between these points of the selected parent are exchanged (see Figure 4A). The probability that the crossover is done for a pair of grouping structures is around 80%.

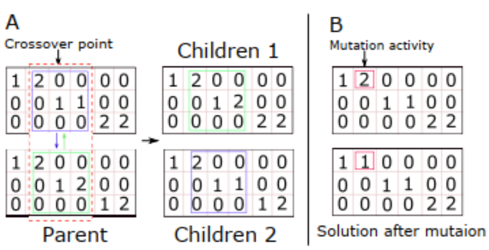

Figure 4. *Examples of crossover and mutation operators*

- *Step 7 (Mutation)*: Mutation helps to prevent falling GA into the local extreme, but it should not occur very often. Mutation probability is chosen in [2% - 5%]. For each selected solution, a maintenance action of a group is randomly selected and then moved to another group (see Figure 4B).

- *Step 8 (New generation)*: A set of new grouping structures is established at this step. It is noticed that new grouping structures are generated thanks to genetic operations such as elitism, crossover and mutation. The new generation is evaluated in the next iteration until stop criteria are triggered.

- *Step 9 (Stopping)*: GaB will be stopped when the maximum number of generations is reached.

4. Numerical studies

This section aims to demonstrate the effectiveness of the proposed GaB algorithm by comparing its performance with two benchmark algorithms: GAaES and GA-BAB, as described in [15]. While GA-BAB also integrates Genetic Algorithm and Branch and Bound techniques, it focuses solely on minimizing the length of maintenance routes and does not guarantee the lowest total maintenance cost. In contrast, GAaES combines a Genetic Algorithm with an Exhaustive Search approach to determine optimal routing. The main difference between GaB and GAaES lies in the routing strategy: GaB employs the Branch and Bound method, whereas GAaES utilizes Exhaustive Search, as referenced in the work of S. Vukmirović et al. [16]. All algorithms are implemented in MATLAB R2016a and executed on a system running Windows 10, equipped with an Intel Core i5-1035G1 processor and 12 GB of RAM. Computational performance is measured using computer time units (ctu). The simulation is conducted on a representative GDPS configuration, as illustrated in Figure 5.

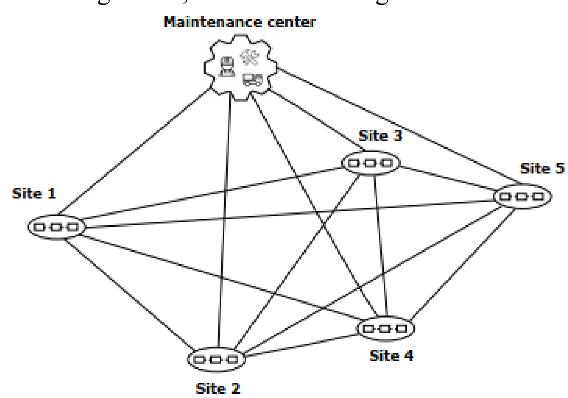

Figure 5. *A typical GDPS containing 5 production sites*

4.1. Algorithm execution

The numerical study is performed for a GDPS containing 5 sites and one maintenance center as Figure 5. The travel distances among sites and their maintenance center are given in Table 1.

Table 1: *Travel distances between the production sites and their maintenance center*

Sites	0	1	2	3	4	5
0	-	160	92	240	262	298
1	160	-	200	196	174	500
2	92	200	-	136	206	308
3	240	196	136	-	108	322
4	262	174	206	108	-	418
5	298	500	308	322	418	-

It should be noticed that, in this study, all parameters are given in arbitrary units, i.e., arbitrary time unit (atu), arbitrary distance unit (adu), or arbitrary cost unit (acu). The preventive maintenance date of 15 components based on Weibull distribution (refer to [17]) is shown in Table 2.

The labor cost rates R^{lb} are fixed at 100, 200, 300 for required levels (l_i) of maintenance team skills 1, 2, and 3 respectively. The average speed in moving of the maintenance team is AS=15 (adu)/(atu). The travel time is then can be easily determined by dividing the travel distance by the average speed, i.e., $d_{ij} = L_{ij}/AS$. The transportation cost rate is considered to be the same for all components $R_i^{tr} = R^{tr}$, and is equal to 15 (acu).

The main characteristics related to the components site, the downtime cost rate $\left(R_j^{dt}\right)$ and the preparation cost $\left(S_j^0\right)$ are given in Table 3.

Table 2: *Data and individual optimization of 15 components*

Unit	λ_i	β_i	C_i^{sp}	C_i^c	ω_i	t_i^e	l_i	S_i^{tr}	ϕ_i^*	t_i
1	2499	2.85	1757	561	30	1398	1	4800	4.1452	5526.5
2	3257	2.77	2546	669	48	2729	2	4800	4.5875	7955.5
3	2423	3.72	3651	628	63	781	3	4800	10.613	5071.3
4	2815	2.87	1874	665	27	2418	1	2760	3.5731	4428.6
5	3505	2.74	2676	722	45	2962	2	2760	4.1107	8415.7
6	2327	3.19	3597	619	66	1142	3	2760	11.067	6436.3
7	2645	2.82	1704	517	33	3281	1	7200	4.4215	5022.6
8	3057	2.76	2341	547	51	3799	2	7200	5.0947	7818.1
9	2264	3.54	3313	611	72	1035	3	7200	10.881	6244.7
10	2462	2.88	1946	635	39	2489	1	7860	4.6199	5469.1
11	2748	3.42	3369	714	69	2999	3	7860	9.6272	7295.8
12	2327	3.73	3529	691	75	3214	3	7860	11.361	7661.1
13	2968	2.64	1697	429	36	1740	1	8940	4.0547	4575.9
14	3359	3.16	2432	609	51	3589	2	8940	5.0827	5629.8
15	2483	3.65	3532	314	72	1819	3	8940	8.8367	5796.3

Table 3: *Given information related to the site level*

Sites	1	2	3	4	5
Components	1, 2, 3	4, 5, 6	7, 8, 9	10,11, 12	13,14, 15
R_j^{dt}	294	316	349	365	207
S_j^0	210	170	120	90	190

Based on this data, if components are maintained individually, each component is maintained only once within the planning horizon $PH = [0; t_5 + \omega_5] =$ [0;8460.7] The using of this individual maintenance plan will lead to an average maintenance cost of the system, which is equal to

$$\emptyset_{sys}^{individual} = \sum_{i=1}^{15} \emptyset_i^* = 99.852 \quad (16)$$

From the average maintenance cost above, the total maintenance cost is equal to 844817.8 (acu). Referring to Table 2, the transportation cost ($\sum S_i^{tr}$) is 94680 (acu) occupying up to 11.2%. According to Table 3, the total maintenance setup cost ($\sum S_j^0$) is 2340 (acu). The reduction of the costs is necessary. The joint optimization of dynamic grouping and maintenance routing is therefore a promising solution.

To find the optimal grouping structure, our proposed algorithm (GaB) is applied. The simulator was used to determine the GA parameters (crossover probability, mutation probability, and population size) by testing their different values. The appropriate values were finally chosen as follows: crossover probability = 0.80; mutation probability = 0.02; population size = 60 solutions. In addition, the evolution graph of the genetic algorithm, which represents the convergence of the total economic profit (TEP) over 400 iterations, is determined and shown in Figure 6. Therefore, the limited number of 250 iterations is chosen as the stopping condition for the GA, because the evolution graph indicates that there is no change in the total economic profit beyond this number.

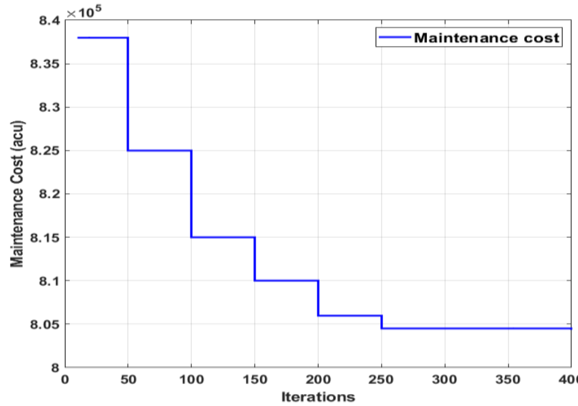

Figure 6. *The convergence curve of the GaB algorithm*

The best maintenance plan provided by GaB is a grouping structure including $G_1 = \{3,6,9,12,13,14,15\}$, $G_2 = \{1,4,7,10\}, G_3 = \{2,5,8,11\}$. The obtained results are shown in Table 4 and Table 5.

Table 4: *Optimal maintenance itineraries of the maintenance team provided by GaB*

Groups	Departure times	Maintenance itineraries	Distances
G_1	4992.1	0 → 2 → 1 → 4 → 3 → 5 → 0	1194
G_2	5073.2	0 → 2 → 3 → 4 → 1 → 0	670
G_3	7852.4	0 → 1 → 4 → 3 → 2 → 0	670

Table 5: *Economic profit provided by GaB*

Groups	$\Delta S_{G_k}^{tr}$	$\Delta S_{G_k}^0$		$\Delta S_{G_k}^{lb}$	ΔH_{G_k}	EPG	EPS
G_1	31530	380		12300	3938.5	15671.5	40462.8
G_2	12570	0		0	257.9	12312.1	
G_3	12570	0		0	90.8	12479.2	

By using GaB, the provided maintenance plan helps to reduce 59.85% transportation cost and save up to 16.23% setup cost in comparison with the individual maintenance. The average maintenance cost of the system when the components are performed joint maintenance

$$\emptyset_{sys}^{grouping} = \emptyset_{sys}^{individual} - \frac{EPS}{t_{end}} = 95.07 \quad (17)$$

From the average maintenance cost, the total maintenance cost is equal to 804355 (acu). Therefore, grouping maintenance helps to save up to 7.44% on total maintenance cost in comparison with individual maintenance. This result is similar to using exhaustive search algorithms like GAaES [15].

A similar process is performed to GA-BAB, except that the grouping penalties are not integrated into the route scheduling process. The optimal route is found in order to minimize the total travel distance only. By applying GA-BAB to the considered GDPS, the obtained results are shown in Tables 6 and 7.

***Table 6:** Optimal maintenance itineraries of the maintenance team provided by GA-BAB*

Groups	Departure times	Maintenance itineraries	Distances
G_1	4914.8	0 → 2 → 5 → 3 → 4 → 1 → 0	1164
G_2	5073.2	0 → 2 → 3 → 4 → 1 → 0	670
G_3	7852.4	0 → 1 → 4 → 3 → 2 → 0	670

***Table 7:** Economic profit provided by GA-BAB*

Groups	$\Delta S_{G_k}^{tr}$	$\Delta S_{G_k}^{0}$	$\Delta S_{G_k}^{lb}$	ΔH_{G_k}	EPG	EPS
G_1	31980	380	12300	5468.8	14591.2	39382.5
G_2	12570	0	0	257.9	12312.1	
G_3	12570	0	0	90.8	12479.2	

By comparing the obtained results in Tables 6 and 7, we can see that both GaB and GA-BAB provide the same grouping structure, however, the optimal itineraries and departure time of group G_1 given by the two algorithms are not the same. Although travel distance and travel cost provided by GA-BAB are less than GaB, maintenance itineraries given by GA-BAB are not an optimal result in terms of cost. The reason is that the GA-BAB tends to choose the shortest itineraries that ignore optimization of penalty cost. The travel cost is minimal but the penalty cost is too large. It leads to EPS provided by GA-BAB is less than GaB. Hence, the results provided by GaB are better than GA-BAB in terms of economic profit.

4.2. Performance study of GaB

This subsection aims to evaluate the performance of the GaB algorithm by comparing it with GAaES in generating optimal maintenance plans for a GDPS. It is important to note that while the GA-BAB algorithm (refer to [15]) focuses on maximizing economic profit primarily through minimizing transportation costs, both GaB and GAaES provide comprehensive solutions that consider the combined economic benefits from reduced transportation and penalty costs. As such, the comparison in this subsection is limited to GaB and GAaES.

A major challenge in this comparison arises from the stochastic nature of genetic operations such as crossover and mutation, which cause the subsequent generations in GaB and GAaES to diverge after each iteration. To ensure a fair and consistent comparison, an adjustment procedure is applied to synchronize the next generations produced by both algorithms. Assuming both methods start with an identical initial population, they are each executed for 10 iterations across different scenarios involving varying numbers of production sites. This ensures the resulting maintenance plans are consistent, allowing for a valid comparison of their computational performance. To compare the optimal solution search time between GaB and GAaES, the study used 6 datasets (each dataset has several sites varying from 3 to 8). Each algorithm was run 10 times on 1 dataset and the computational time (CT) is the average time of these 10 runs. The differences in computational time are presented in Table 8 and illustrated in Figure 7.

***Table 8:** Comparison of computational time between GaB and GAaES*

CT \ Sites	3	4	5	6	7	8
GAaES	1.754	8.425	21.542	55.781	2359.549	21856.187
GaB	1.691	7.024	11.988	28.881	1112.456	8935.207

The results indicate that when the number of sites is relatively small, the computational times of GaB and GAaES are comparable. However, as the number of sites increases, GaB demonstrates a clear advantage, achieving a reduction in computational time of over 59% compared to GAaES.

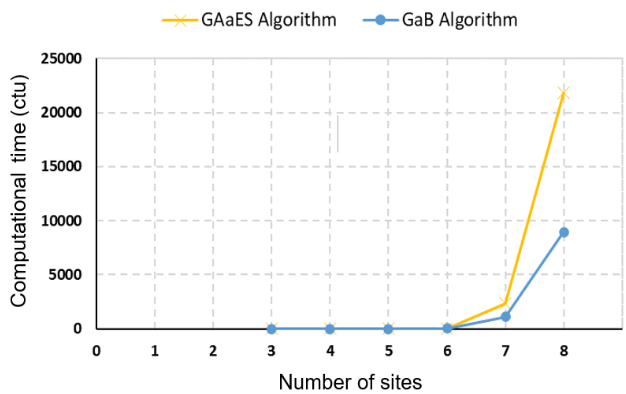

***Figure 7.** Comparison of computational time between GaB and GAaES*

5. Conclusions

In this study, an advanced Evolutionary Algorithm (GaB) was proposed to solve a complex optimization problem of maintenance planning of GDPS. This algorithm is a combination of GA and BAB. GA generates random grouping plans and then improves them by applying different operators including selection, crossover, elitism, and mutation. At each iteration of GA, BAB tries to find the best maintenance itinerary for each group by enumerating and ruling out, simultaneously, domains of the search space that cannot contain promising itineraries. The best grouping plan and their itineraries are found by minimizing not only the maintenance cost but also the transportation and preparation costs.

To underline the performance of the proposed algorithm (BAB and GaB), they have been compared against ES and GAaES algorithms respectively. The comparison was done by considering different scenarios where the number of sites has been varied. The obtained results show that our proposed methods (BAB and GaB) outperform the ES and GAaES in terms of computational time when the number of sites increases. GaB is then a very promising solution for maintenance planning and routing of complex systems with a large number of components such as GDPS. For future research, other metaheuristic methods such as ACO (Ant Colony Optimization), SA (Simulated Annealing), Tabu Search, or exact methods (MILP) such as branch-and-cut, cutting planes may be considered to further improve solution quality and efficiency.

REFERENCES

[1] R. Dekkert, A. Smit, and J. Losekoot, “Combining maintenance activities in an operational planning phase: a set-partitioning approach”, *IMA Journal of Management Mathematics*, vol. 3, no. 4, pp. 315-331, 1991.

[2] R. Dekker, R. E. Wildeman, and F. A. Van der Duyn Schouten, “A review of multi-component maintenance models with economic dependence”, *Mathematical Methods of Operations Research*, vol. 45, no. 3, pp. 411–435, 1997.

[3] Z. Hameed and K. Wang, “Development of optimal maintenance strategies for offshore wind turbine by using artificial neural network”, *Wind Engineering*, vol. 36, no. 3, pp. 353–364, 2012.

[4] R. Liu, Y. Tao, and X. Xie, “An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and synchronized visits”, *Computers & Operations Research*, vol. 101, pp. 250–262, 2019.

[5] A. Moura, “A model-based heuristic to the vehicle routing and loading problem”, *International Transactions in Operational Research*, vol. 26, no. 3, pp. 888–907, 2019.

[6] M. Soto, M. Sevaux, A. Rossi, and A. Reinholz, “Multiple neighborhood search, tabu search and ejection chains for the multi-depot open vehicle routing problem”, *Computers & Industrial Engineering*, vol. 107, pp. 211–222, 2017.

[7] M. Gendreau, G. Laporte, and F. Semet, “A branch-and-cut algorithm for the undirected selective traveling salesman problem”, *Networks: An International Journal*, vol. 32, no. 4, pp. 263–273, 1998.

[8] G. Laporte, “The traveling salesman problem: An overview of exact and approximate algorithms”, *European Journal of Operational Research*, vol. 59, no. 2, pp. 231–247, 1992.

[9] E. Lopez-Santana, R. Akhavan-Tabatabaei, L. Dieulle, N. Labadie, and A. L. Medaglia, “On the combined maintenance and routing optimization problem”, *Reliability Engineering & System Safety*, vol. 145, pp. 199–214, 2016.

[10] M. Stålhane, L. M. Hvattum, and V. Skaar, “Optimization of routing and scheduling of vessels to perform maintenance at oshore wind farms”, *Energy Procedia*, vol. 80, pp. 92–99, 2015.

[11] L. Dai, M. Stålhane, and I. B. Utne, “Routing and scheduling of mainte-nance fleet for oshore wind farms”, *Wind Engineering*, vol. 39, no. 1, pp. 15–30, 2015.

[12] E. López-Santana, G. Méndez, and C. Franco, "On the multi-period combined maintenance and routing optimisation problem", *International Journal of Production Research*, vol. 61, no. 23, pp. 8265-8290, 2023.

[13] F. Delavernhe, B. Castanier, C. Guéret, and J. E. Mendoza, "The joint maintenance operation selection and technician routing problem", *Computers & Operations Research*, vol. 167, pp. 106667, 2024.

[14] D. Moros, N. Berrabah, I. Ashton, K. Searle, and I. Lazakis, "A Multi-timescale Framework for Maintenance Planning at Offshore Wind Farms", *SSRN*, vol. 5147352, 2025.

[15] H. S. H. Nguyen, P. Do, H.-C. Vu, T. B. Le, and K. A. Nguyen, “An efficient evolutionary algorithm for joint optimization of maintenance grouping and routing,” in *2020 5th International Conference on Green Technology and Sustainable Development (GTSD)*, IEEE, 2020, pp. 604–611.

[16] S. Vukmirović, Z. Čapko, and A. Babić, "Model of using the exhaustive search algorithm in solving of traveling salesman problem (tsp) on the example of the transport network optimization of primorje-gorski kotar county (PGC)", *Journal of Corporate Governance, Insurance, and Risk Management,* Vol. 6, no. 2, 2019.

[17] P. Do, A. Barros, C. Bérenguer, K. Bouvard and F. Brissaud, "Dynamic grouping maintenance with time limited opportunities", *Reliability Engineering & System Safety*, vol. 120, pp. 51-59, 2013.

[18] R. Reichwald, C. M. Stotko, and F. T. Piller, “Distributed mini-factory networks as a form of real-time enterprise: concept, flexibility potential and case studies”, *The Practical Real-Time Enterprise, Berlin, Heidelberg*, Springer Berlin Heidelberg, pp. 403–434, 2005.

APPENDIX A. IMPLEMENTATION OF BRANCH AND BOUND

procedure OPTIMAL ITINERARY WITH HIGHEST ECONOMIC PROFIT

L = {Ljm}; % Distance matrix

Route = {0}; %Maintenance itinerary is started at global maintenance center (0)

SoS = {0, 1, 2, ..., n}; %Set of all visited sites

CSS = {S0}; %Complete solution set; there is one element at beginning position

$S_t = 1$; %Declare condition to stop; algorithm stop when $S_t=2$

q = 0; %Root node

while $S_t < 2$ **do**

  for m ∈ SoS − (Route ∩ SoS) **do**

    q = q + 1;

    $VN_{0\to q} = \{j|j \in Route\}$; %Vertical set of visited nodes

    SJ = SoS − Route{1, ..., (end − 1)} − {m}; %The set of sites that the maintenance team could be in before visiting site

    $ETC = \{t'_{i_s} | i \epsilon j \& j \epsilon VN_{0\to q}\}$; %Execution time of components in $VN_{0\to q}$

    w = m; % Current site

    $TLS_w = TAS_w + sum(\omega_i)$; %Time that maintenance team leaves site w

    LBC = LBCF(LBTC, SJ, ETC, L, Route, TLS_w, q, w, t_i);

    %Call function of lower bound of transportation cost

    $RouteS_q$ = {Route, m};

  end

  [Index, min Cost] = min (LBTC); %Position and value get minimum cost

  CSS = {CSS, S_{Index}}; %Update the complete solution set

  Route = $RouteS_q$; %Update the route

  if length(Route) = n + 1 **then**

    $Route_{final}$ = {Route, 0};

    $LBTC_{final} = LBTC_{Index}$;

    $S_t = S_t + 1$;

  end

end procedure

function LBC = LBCF(LBTC, $VN_{0\to q}$, SJ, ETC, L, Route, q, w, tentativeDates)

%Where LBTC - lower bound of travel cost, $VN_{0\to q}$ - vertical set of visited nodes, SJ - the set of sites that the maintenance team could be in before visiting site, ETC - execution time of components in $VN_{0\to q}$, L - distance matrix, TLS_w - time that maintenance team leaves site w, q - current node, w - current site, tentativeDates(i) - tentative dates of components.

if i ∈ $VN_{0\to q}$ then

  $H_{0\to q} = sum(f(ETC_i))$; %Total penalty cost of all components contained in itinerary from root node (0) to node q

else

  $minTime(i_s)$ = travelTimeLeaves(w) + minTravelTime(w, m);

  % The shortest time to maintain component *i*

  $maxTime(i_s)$ = travelTimeLeaves(w) + maxTravelTime(w, m)

  %The longest time to maintain component *i*

  if tentativeDates(i) < $minTime(i_s)$ **then**

    penaltyCost(i) = calculatePenalty($minTime(i_s)$);

  else

    if tentativeDates(i) > $maxTime(i_s)$ **then**

      penaltyCost(i) = calculatePenalty($maxTime(i_s)$);

    else

      penaltyCost(i) = 0;

    end

end

$\widehat{H}_{q\to 0} = \mathrm{sum}(\mathrm{penaltyCost}(i));$

end

$LBPC_q = H_{0\to q} + \widehat{H}_{q\to 0}$; %Lower bound of penalty cost

$LBTC_q = LBTCFunction(L, VN_{0\to q}, Route, SJ)$; %Call function of lower bound of transportation cost

$LBC = LBPC_q + LBTC_q$; %Lower bound cost at node q corresponding to w

end function

APPENDIX B. GAB ALGORITHM

procedure GaB ALGORITHM
sp = *Size_of_Population*
for g=1 → sp **do**
$GS_g = \{EGS(i^z) | EGS(i^z) = k\}$; %Coding
end
P = $\{GS_g\}$; iter = 1; Stop = *Number_of_Iteration;*
L = $\{L_{jm}\}$; %Distance matrix
while iter < Stop **do**
iter = iter + 1;
for g = 1 → sp **do**
for k = 1 → max(GS_g) **do**
$G_k = \{GS(i_z) = k\}$;
$\left(t^*_{G_k}, I^*_{G_k}\right) = \underset{\substack{t_{G_k}=0\to\infty \\ I_{G_k}\epsilon SI}}{\arg\max} EPG_{G_k}(t_{G_k}, I_{G_k})$;
$EPG_k^{max} = EPG_k(t^*_{G_k}, I^*_{G_k})$;
end
$EPS_g = \sum_{k=1}^{\max\{GS_g\}} EPG_k^{max}$
end
g* = arg max$\{EPS_g \mid g = 0 \to sp\}$;
$EPS_g^{max} = EPS_{g^*}$; %Economic profit evaluation
if iter = Stop **then**
Display($GS_{g^*}, I^*_{G_k}, t^*_{G_k}$);
else
$GS_1 = GS_{g^*}$;
for g = 2 → sp **do**
Parent1 = random$\{GS_q | q = 2 \to sp\}$;
Parent2 = random$\{GS_q | q = 2 \to sp\}$;
ChildrenGS = crossover(Parent1, Parent2);
GS_g = ChildrenGS;
end
for mg = 1 → round(sp/20) **do**
g = random(3 → length(P));
p = random(1 → length(GS_g));
$EGS_p(i^z)$ = random(1 → max(GS_g));
end
NGS = $\{GS_g | g = 1 \to s^p\}$; %New generation
end
end
end procedure