






## Article

# A Hybrid Grey Wolf Optimizer for Process Planning Optimization with Precedence Constraints

Mijodrag Milosevic <sup>1</sup>, Robert Cep <sup>2</sup>, Lenka Cepova <sup>2</sup>, Dejan Lukic <sup>1</sup>, Aco Antic <sup>1</sup> and Mica Djurdjev <sup>3,\*</sup>

<sup>1</sup> Department of Production Engineering, Faculty of Technical Sciences, University of Novi Sad, 21000 Novi Sad, Serbia; mido@uns.ac.rs (M.M.); lukicd@uns.ac.rs (D.L.); antica@uns.ac.rs (A.A.)

<sup>2</sup> Department of Machining, Assembly and Engineering Metrology, Faculty of Mechanical Engineering, Technical University of Ostrava, 70800 Ostrava, Czech Republic; robert.cep@vsb.cz (R.C.); lenka.cepova@vsb.cz (L.C.)

<sup>3</sup> Department of Mechanical Engineering, Technical Faculty “Mihajlo Pupin”, University of Novi Sad, 23000 Zrenjanin, Serbia

\* Correspondence: mica.djurdjev@tfzr.rs; Tel.: +381-62-8019-715

**Abstract:** Process planning optimization is a well-known NP-hard combinatorial problem extensively studied in the scientific community. Its main components include operation sequencing, selection of manufacturing resources and determination of appropriate setup plans. These problems require metaheuristic-based approaches in order to be effectively and efficiently solved. Therefore, to optimize the complex process planning problem, a novel hybrid grey wolf optimizer (HGWO) is proposed. The traditional grey wolf optimizer (GWO) is improved by employing genetic strategies such as selection, crossover and mutation which enhance global search abilities and convergence of the traditional GWO. Precedence relationships among machining operations are taken into account and precedence constraints are modeled using operation precedence graphs and adjacency matrices. Constraint handling heuristic procedure is adopted to move infeasible solutions to a feasible domain. Minimization of the total weighted machining cost of a process plan is adopted as the objective and three experimental studies that consider three different prismatic parts are conducted. Comparative analysis of the obtained cost values, as well as the convergence analysis, are performed and the HGWO approach demonstrated effectiveness and flexibility in finding optimal and near-optimal process plans. On the other side, comparative analysis of computational times and execution times of certain MATLAB functions showed that the HGWO have good time efficiency but limited since it requires more time compared to considered hybrid and traditional algorithms. Potential directions to improving efficiency and performances of the proposed approach are given in conclusions.

**Keywords:** process planning optimization; grey wolf optimizer; precedence constraints; constraint handling; selection; crossover; mutation



**Citation:** Milosevic, M.; Cep, R.; Cepova, L.; Lukic, D.; Antic, A.; Djurdjev, M. A Hybrid Grey Wolf Optimizer for Process Planning Optimization with Precedence Constraints. *Materials* **2021**, *14*, 7360. <https://doi.org/10.3390/ma14237360>

Academic Editor: Kevin M. Van Geem

Received: 6 November 2021

Accepted: 27 November 2021

Published: 30 November 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Process planning optimization (PPO) problems consist of two subproblems: operations selection and operations sequencing [1]. Operations selection is a task of selecting necessary machining operations for each machining feature recognized on a part or a product. In order to perform these operations, adequate machining resources are required. In that sense, the PPO problem includes selection of machine candidate, cutting tool candidate and tool approach direction (TAD) candidate for each machining operation. On the other hand, operations sequencing represents a task of finding the order of selected machining operations with respect to the predetermined precedence constraints based on relationships between machining features. Tackling with such a problem requires dealing with a number of alternatives which makes the PPO a combinatorial optimization problem. A number of alternative process plans grows in accordance with problem dimensions. In other words, as a number of machining features increases, so does a number of machining operations. This

leads to exponential time growth required to find optimal or near-optimal process plans, which means that the PPO problem belongs to the class of NP-hard (non-deterministic polynomial) optimization problems.

As it is impossible to check every possible solution to an NP-hard problem, conventional non-heuristic methods have proved to be ineffective. The PPO problem requires the implementation of metaheuristic algorithms which have shown promising performance for solving difficult problems. This is especially due to their ability to achieve a trade-off between local and global search [2]. This paper proposes a novel hybrid grey wolf optimizer (HGWO) to deal with the PPO problem with precedence constraints. The manuscript is organized as follows. Section 2 provides recent studies related to the PPO. Section 3 describes the problem, focusing on manufacturing interactions and precedence constraints, representation of process plans and mathematical modelling of the PPO. Section 4 emphasizes the proposed HGWO approach starting from traditional GWO, then genetic strategies that are included, and constraint handling algorithm as part of the structure of the HGWO. Section 5 represents the experimental research that consider three experimental studies, or three prismatic parts. With detailed description of these cases, comparative analysis of cost values and convergence curves are performed to test the performances of the HGWO. Later, discussion in Section 6, emphasize the effectiveness of the proposed method and the computational efficiency. Conclusions with directions for future research are given in Section 7.

## 2. Related Research Studies

In the past few decades, many search algorithms have been employed to deal with the PPO problem. In the initial stages of research on the PPO, genetic algorithms have been widely reported among other metaheuristics. One of the earliest implementations of genetic algorithms (GAs) can be found in Zhang et al. [3]. Besides the GA, many other algorithms such as ACO, PSO, TS, SA are applied to this day.

GA it is the most popular evolutionary algorithm based on a Darwinian theory of natural selection and genetics. Traditional GAs, as well as many other algorithms, require additional modifications to improve their convergence as well as global and local search abilities. Study performed in Salehi and Bahreininejad [4] adopts hybrid GA and intelligent search for optimization of process planning in preliminary and detailed planning stages. The initial and feasible solutions were checked by order and clustering constraints. Li et al. [5] proposed a hybridization model of GA and SA where the initial process plans were generated using the GA approach while the optimal or near-optimal process plans were obtained using the SA algorithm. Kafashi [6] developed a GA approach in order to tackle integrated setup planning and operation sequencing problem in flexible manufacturing environment. With the emphasis on technological constraints, tolerance relation analysis and feature-based representation, satisfactory setup plans and operation sequences were generated. Cai et al. [7] used the GA approach based on optimization toolbox to address adaptive setup planning problem towards the process planning and scheduling integration. They considered machine availability based on tool accessibility analysis as a constraint, and selected certain scheduling criteria such as cost, makespan or machine utilization. Huang et al. [8] proposed a hybrid graph and GA approach. Combining graph theory as well as matrix theory, precisely operation precedence graphs and adjacency matrices, precedence constraints were formulated and a GA approach based on modified crossover and mutation strategies were utilized to solve the PPO problem. Two heuristic approaches are incorporated within this hybrid approach in order to adjust infeasible process plans to a feasible domain. Li et al. [9] employed a novel two phase genetic algorithm to optimize process parameters and machining sequence for two-tool parallel drilling operations with multiple blind holes distributed in a pair of parallel faces and in multiple pairs of parallel faces. Hybridization of GA and local search for the PPO on turning machine tool was proposed by Su et al. [10]. They formulated the PPO problem as a mixed 0–1 integer programming model and used a novel encoding strategy

to deal with complicated precedence constraints. In the study presented by Su et al. [11], precedence constrained operation sequencing problem was formulated and edge selection based strategy was adopted to assure feasibility of solutions and improve GA's converging efficiency. Dou et al. [12] proposed the improved GA by considering fragment crossover and mutation operators with adaptive operation probabilities as well as a new elitist-based crossover strategy. Luo et al. [13] dealt with a large-sized process sequencing problem with complex association constraints. The problem with huge and complex solution space was decomposed to smaller multi-neighborhood spaces and hybrid GA and variable neighborhood search (VNS) approach were used to obtain the best solutions through all neighborhood spaces.

Furthermore, the ant colony optimization algorithm also found its application to the PPO. Liu et al. [14] used ant colony optimization for process planning optimization. They based their research on mapping the PPO problem to a weighted graph and converting to a constraint-based travelling salesman problem. Constraint and state matrices were used to ensure the feasibility of process plans. Wang et al. [15] dealt with the PPO problem by using a weighted directed graph for process plan representation. Wang et al. [16] proposed a two-stage ACO algorithm to solve the PPO problem for prismatic parts. The PPO was formulated using a directed graph and then the proposed ACO approach was used for optimizing process plans for two prismatic parts from the literature. First stage included operation selection while the second was focused on operation sequencing. Hu et al. [17] considered precedence constraint and clustering constraint relationships to ensure feasible permutations and enhanced search abilities. They used the updating method and the local search mechanism to modify the ACO algorithm.

Particle swarm optimization (PSO) is also one of the most popular metaheuristics based on swarm intelligence. This method is inspired by social behavior of living organisms that live in groups (swarms), such as birds or fish [18]. One of the recent studies concerning the PSO in process planning can be found in [19]. Here, authors proposed a modified PSO algorithm improved by adopting efficient encoding, updating and random search methods in order to tackle seven different case studies that consider prismatic parts. Petrović et al. [20] developed a new chaotic PSO algorithm for flexible process planning. By emphasizing different types of flexibilities, authors used AND/OR networks to represent process plans and then tested performances of the cPSO on several cases including cylindrical as well as prismatic parts. Miljković and Petrović [21] developed a modified multi-objective PSO algorithm for flexible process planning. By emphasizing different types of flexibilities, authors used AND/OR networks to represent feasible process plans, and tested performances of the cPSO on several cases including cylindrical as well as prismatic parts. Dou et al. [22] proposed a novel feasible sequence-oriented discrete PSO algorithm to solve the operation sequencing problems in CAPP. It incorporates novel crossover and mutation operators with adaptive probabilities to evolve particles and improve exploration ability.

Lian et al. [23] proposed the imperialist competitive algorithm to address the PPO problem. This socio-politically motivated population-based metaheuristic was inspired by imperialist competition. The approach utilizes steps of assimilation, competition, revolution and elimination to obtain optimal process plans according to predefined networks and types of flexibilities.

Wen et al. [24] proposed a new method based on honey bees mating optimization (HBMO) algorithm to optimize the PPO. The solution encoding, crossover operator and local search strategies were developed, and three experiments were carried out to demonstrate improvement of the HBMO approach.

A new heuristic method, called cross entropy approach was proposed to optimize flexible process planning by Lv and Qiao [25]. Its adopted AND/OR networks to represent flexible process plans and established mathematical model for the minimization of total processing time and total cost of flexible process plans. The new sample representation

and probability distribution parameter were introduced and case studies were carried and discussed to indicate the performance and adaptability of the CE approach.

Wang et al. [26] adopted a hybrid bat algorithm for the PPO focusing on both crucial tasks, operations selection and operations sequencing. Encoding, decoding and initialization strategies were proposed and two local search strategies were incorporated into the standard bat algorithm (BA) to improve its local convergence. A classical simulation experiment was conducted to verify the validity of the hybrid BA.

Musharavati and Hamouda [27] investigated the possibility of implementing four different configurations of simulated annealing algorithm to solve the process planning problem in reconfigurable manufacturing systems. They used knowledge exploitation and quasi-parallelism as concepts to enhance the SA algorithms. Performances of the variant algorithms are compared and ANOVA methodology is used in computational analysis to test the means and indicate improvements towards better optimal solutions.

Mohammadi et al. [28] dealt with the multi-objective optimization of the integrated process planning and scheduling problem. They designed a slot-based mixed integer linear programming model that accounts for sequence-dependent preparation times. Minimization of manufacturing cost in process planning, and minimization of preparation times and total tardiness were considered as optimization objectives. To solve the problem, a hybrid simulated annealing approach was introduced. The SA was modified with tabu search algorithm as well as local search strategies in order to improve solution's quality and avoid premature convergence.

Xu et al. [29] introduced a novel NC process reuse-oriented flexible process planning optimization approach for prismatic parts with the objective of minimizing total manufacturing cost. A hybrid ant colony algorithm (ACA) and simulated annealing (SA) approach based on operation precedence graph (OPG) was presented to find the global optimal NC process scheme for the part.

Lian et al. [30] proposed a multi-dimensional tabu search algorithm to optimize four dimensions of a process plan, such as operation sequence, machine sequence, tool sequence, and tool approach direction sequence. Classical neighborhood strategies such as insertion and swap are used to improve the performance of MDTs.

Falih and Shammari [31] proposed a novel hybrid constrained permutation algorithm and genetic algorithm approach for process planning problem. A set of feasible operation sequences is generated using a CPA algorithm and the GA with mixed crossover operator is further employed to search for optimal or near optimal process plans.

Gao et al. [32] suggested the intelligent water drop algorithm for solving the process planning problem. Firstly, operation units were defined according to the processing characteristics, and later, the IWD algorithm was combined with the process planning problem.

Kizys et al. [33] developed an iterative local search metaheuristic and quadratic programming approach to deal with variants of the mean-variance portfolio optimization problem subject to cardinality and quantity constraints.

Sawik and Sawik [34] used stochastic programming approach to optimize cybersecurity investment in supply chains. A mixed binary optimization problem is transformed to an unconstrained binary program in order to maximize total cybersecurity value of control portfolio. It is also shown that the portfolio of security controls with maximum total cybersecurity value reduces the losses from security breaches and mitigates the impact of cyber risk.

In one of the recent studies, Milosević et al. [35] presented the intelligent process planning for the concepts of smart factory and smart manufacturing. Nature-inspired metaheuristic algorithms as smart services of artificial intelligence for intelligent process planning optimization within smart manufacturing were proposed. Three modern algorithms such as grey wolf optimizer (GWO), whale optimization algorithm (WOA) and crow search algorithm (CSA) were employed on three classical case studies from the literature.

Djurdjev et al. [36] proposed a novel genetic crow search approach (GCSA) for operation sequencing in process planning. The traditional CSA algorithm has been improved

by adopting genetic strategies such as tournament selection, three-string crossover, shift mutation and resource mutation. Adaptive probabilities were introduced to improve local and global capabilities of the GCSA. Besides, the nearest mechanism strategy was added to ensure feasibility of machine, tools and TADs vectors. Lastly, repairment heuristic strategy was used to handle precedence constraints among features and machining operations.

Aforementioned studies have shown that the improvements made by authors largely increased efficiency of metaheuristic algorithms. However, as stated in [20], main drawbacks reflect in time-consuming optimization and slow convergence of TS, SA and GA approaches particularly. The key and also the most challenging task in the implementation of metaheuristic algorithms is to find a proper balance between exploration and exploitation phases [37], and thereby, maintain its simplicity and flexibility. Many different concepts have been developed and tested so far, making them available for application in different scientific fields. In this research, we focus on implementing novel metaheuristic approach to solve the PPO problem.

### 3. Process Planning Optimization Problem

Computer-aided process planning represents the link between computer-aided design and computer-aided manufacturing within the computer integrated environment. The aim of process planning is to manufacture a part or a product starting from its initial design stages to its final stage (a finished part or product). The input to process planning includes valuable geometric information obtained from a CAD file. Accordingly, process planners have a goal to map product information to a technological domain. Parts/products are generally described by design features which are characterized by geometric forms or shapes with technological attributes such as tolerances and surface finishes. Design features are transformed to manufacturing features which represent manufacturing meaning of the geometry of a part, a product or an assembly associated with certain manufacturing activities [38]. An important subset of manufacturing features consists of machining features such as holes, steps, grooves, pockets, planes, etc. Recognized machining features are most often considered as an input information for the PPO problem.

Apart from machining features that require certain operations in order to be machined on a part or a product, process planning considers other activities such as finding the sequence of machining operations, determining machining resources—machine tools, cutting tools, fixtures, determining cutting conditions and calculating machining time and cost. In the PPO, the quality of a single process plan is determined by selecting machining resources such as machine tools, cutting tools, and tool approach directions (TADs) for each machining operation, as well as generating optimal sequence of machining operations for an observed part or a product. During the decision-making process of selecting alternatives of machining resources and TADs, in order to find feasible and optimal order of machining operations, the sequence have to satisfy precedence constraints that are based on relationships between machining features and machining operations.

#### 3.1. Representation of Process Plans

When dealing with the PPO problem, the first step is to select an appropriate representation of a process plan. In recent studies [20,21,23,25] five types of flexibilities were considered, and AND/OR networks are used to represent the PPO problem. The main advantage of these networks is in the fact that a single AND/OR network visualizes the detailed representation of all flexibilities for a considered part or a product. By using AND/OR connectors and links, a network can easily be traversed and a feasible alternative process plan can be generated.

Knowledge-based representation is the second method used for representing process plans [8,39]. Here, a process plan is represented using vectors with  $n$  bits of data information. Each bit represents a machining operation and the order of bits form an operation sequence. As mentioned, each machining operation requires a set of a machine candidate, a cutting tool candidate and a TAD candidate which are used in order to perform a given

machining operation. Therefore, apart from an operation vector, this representation also consists of a machine vector, a tool vector and a TAD vector.

To form a process plan, we considered that each machining feature on a part contains a certain number of operation units [32]. Operation units represent single machining operations associated with its candidate resources and can be formulated using the following expression:

$$mo_i = [M_j, CT_k, TAD_l] \quad (1)$$

where  $mo_i$  represents a machining operation  $i$ , while  $M_j$ ,  $CT_k$  and  $TAD_l$  stand for a machine candidate, a cutting tool candidate and a TAD candidate for an operation  $i$ , respectively.

Operation units become building elements of an operation sequence that can be represented using modified knowledge-based approach similar to the one previously mentioned. Figure 1 shows a simple example of a representation of process planning problem.

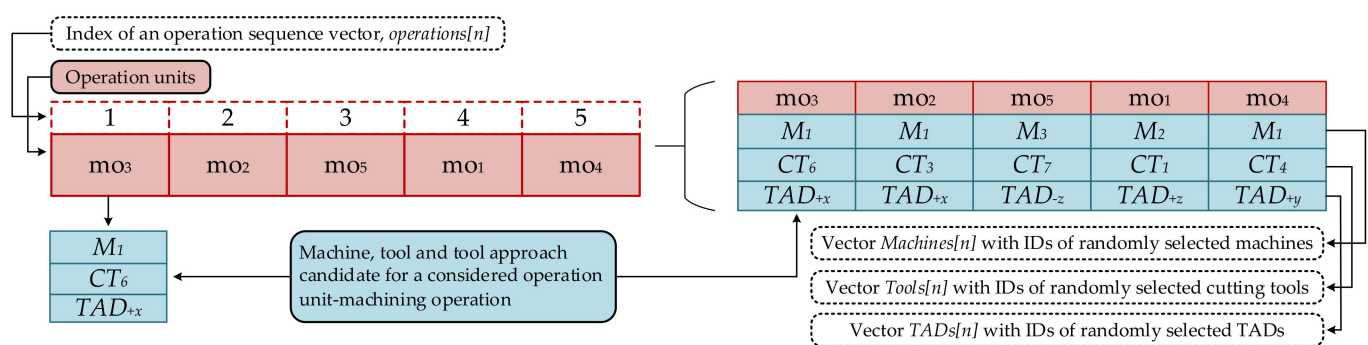


Figure 1. Process plan representation.

In Figure 1, we have the total of 5 operation units that match the permutation of 5 numbers. They are shown in the dashed boxes in the first row of the represented process plan (left picture). Position 1 in the vector contains the operation unit 3 which further include  $M_1$ —machine 1,  $CT_6$ —cutting tool 6 and  $TAD_{+x}$ —TAD “+x” as candidates for  $mo_3$ —machining operation 3. All these machining operations form an operation sequence vector called  $operations[n]$  ( $n$  is a total number of machining operations). Candidate solutions are assigned to each machining operation and presented in the right picture. Therefore, randomly selected candidates form machine vector, cutting tool vector and TADs vector, respectively.

### 3.2. Manufacturing Relationships and Precedence Constraints

Process planning is a precondition for executing a machining process. According to manufacturing and geometric specifications of machining features, manufacturing relationships occur followed by potential interferences between certain consecutive operations. Therefore, in order to machine a feature, process planners are required to pay thorough attention to manufacturing relationships which strongly affect the feasibility of process plans. Consequently, generating operation sequence without considering manufacturing relationships may result in a situation where executing one operation may make it difficult or impossible to execute others. These relationships form precedence constraints in the PPO problem which affect the sequence of all machining operations required to machine a certain product or a part. With the emphasis on machining, two distinctive types of precedence constraints based on relationships among machining operations are hard and soft constraints [40].

Since the PPO problem considered in this study is formulated as an optimization problem, precedence constraints need to be respected in order to find feasible and optimal process plans. Generally, as the name implies, hard constraints are more stringent compared to soft constraints. They directly affect the feasibility of process plans and process planners must ensure the consistency with hard constraints. On the other hand, soft constraints

have the influence solely on the quality, cost and efficiency of a feasible process plan and therefore can be violated in some cases.

According to [39,40], hard and soft constraints can be divided into several types. Figure 2 represents the schematic illustration of precedence constraints based on manufacturing relationships among features.

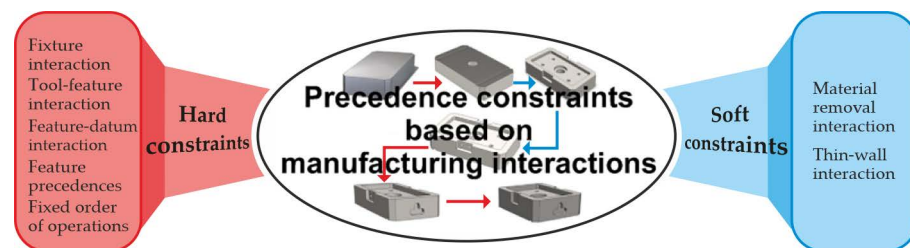


Figure 2. Schematic representation of precedence constraints.

### 3.3. Representation and Modelling of Precedence Constraints

Precedence constraints define precedence relationships between features/machining operations whose purpose is to help in finding a feasible operation sequence. With the emphasis on hard and soft constraints, we will consider two different approaches in this research. One approach will focus only on hard precedence constraints and the other will consider both types of constraints, hard and soft. Accordingly, two different prismatic parts adopted from the literature sources are adopted to identify precedence relationships and define types of constraints based on relationships or interactions between machining operations.

The simplest way to graphically illustrate precedence relationships is with operation precedence graphs (OPGs), which are effectively proposed in [8,41]. Assuming the fact that the emphasis will be placed on two different approaches, we will adopt two different OPGs. For hard constraints only, representation will be based on classical connected OPGs (cOPGs), while for both hard and soft constraints representation of precedence relationships will be based on disconnected graphs (dOPGs).

In order to represent aforementioned constraints and relationships, we adopted the first part from Guo et al. [42]. Its 3D solid model with assigned features and operations, as well as the cOPG that depicts only hard constraints, are shown in Figure 3. According to this figure, the total number of machining operations for prismatic part 1 is 20 and can be defined as the set of machining operations  $[mo_1, mo_2, mo_3, \dots, mo_{20}]$ . Each machining operation from the cOPG is represented as a node while connections among these nodes are directed edges that show precedence relationships among them. An operation sequence that satisfies these precedence relationships is considered a feasible operation sequence.

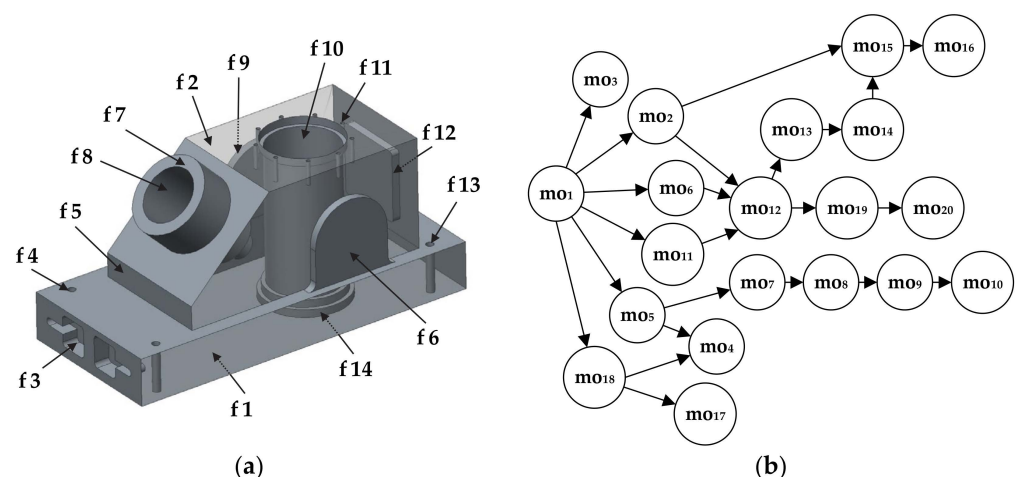


Figure 3. (a) Semi-transparent solid model and (b) the cOPG of the prismatic part 1.

The best way to manipulate such graphical data in a selected programming environment is by converting it to a matrix form. In that sense, we adopted the adjacency matrix [8]. The adjacency matrix is generally used to represent nodes of a graph. To convert graph information to an adjacency matrix, we used the following expression:

$$Adj\_matrix = (PR_{ij})n \times n \tag{2}$$

where  $PR_{ij}$  stands for a precedence relationship between machining operation  $i$  (regarding the row) and machining operation  $j$  (regarding the column). For hard constraints only, the  $PR_{ij}$  values are binary, ones and zeroes. The first rule for this case is when  $PR_{ij} = 1$  and  $PR_{ji} = 0$ . Here, a directed edge connects the nodes  $mo_i$  and  $mo_j$  pointing from  $mo_i$  towards  $mo_j$  and, therefore states that machining operation  $i$  has to be performed prior to machining operation  $j$ . Otherwise,  $PR_{ij} = 0$ . Another rule is where  $PR_{ii} = 0$  for all operations from the operation set (here  $[mo_1, \dots, mo_{20}]$ ). The expression  $n \times n$  means that the adjacency matrix is a square matrix where  $n$  stands for the total number of machining operations.

It can be noticed from Figure 4, that the first row for  $mo_1$  regarding the machining operation 1 has six precedence relationships that equal number 1, meaning that the machining operation  $mo_1$  has to be performed prior to  $mo_2, mo_3, mo_5, mo_6, mo_{11}$  and  $mo_{18}$ . The same rule stands for each row in the matrix.

	mo1	mo2	mo3	mo4	mo5	mo6	mo7	mo8	mo9	mo10	mo11	mo12	mo13	mo14	mo15	mo16	mo17	mo18	mo19	mo20
mo1	0	1	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0
mo2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
mo3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mo4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mo5	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
mo6	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
mo7	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
mo8	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
mo9	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
mo10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mo11	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
mo12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
mo13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
mo14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
mo15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
mo16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mo17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mo18	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
mo19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
mo20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4. Schematic representation of precedence constraints.

To put an emphasis on both hard and soft constraints we included another prismatic part proposed by [3]. Three-dimensional (3D) solid model of the prismatic part 2 with assigned features and machining operations along with the corresponding dOPG is shown in Figure 5.

Figure 5b presents the disconnected dOPG with 9 different precedence relationships among machining operations. Only node  $mo_5$  does not have precedence relationships with other nodes. There are 4 hard constraints and 5 soft constraints depicted with red and blue edges. Accordingly, two different numbers have to be used in order to form a reliable adjacency matrix. Herewith, number 1 stands for a hard constraint and number 2 in the matrix assumes that node  $mo_i$  and node  $mo_j$  form a soft constraint. The formulation and rules are the same as for the cOPG. The operation precedence matrix for the prismatic part 2 is given in Figure 6. Hard constraints are the ones whose violation must be avoided and therefore have total priority over soft constraints. This means that relationships among machining operations assigned with number 2 in the matrix can in some cases be neglected if they get into conflict with hard relationships assigned with number 1.



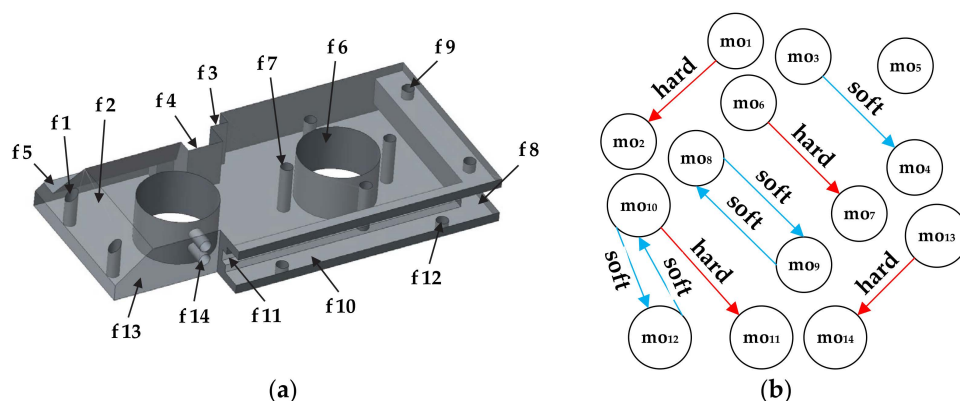


Figure 5. (a) Semi-transparent solid model and (b) the dOPG of the prismatic part 2.

	mo1	mo2	mo3	mo4	mo5	mo6	mo7	mo8	mo9	mo10	mo11	mo12	mo13	mo14
mo1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
mo2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mo3	0	0	0	2	0	0	0	0	0	0	0	0	0	0
mo4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mo5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mo6	0	0	0	0	0	0	1	0	0	0	0	0	0	0
mo7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mo8	0	0	0	0	0	0	0	0	2	0	0	0	0	0
mo9	0	0	0	0	0	0	0	2	0	0	0	0	0	0
mo10	0	0	0	0	0	0	0	0	0	0	1	2	0	0
mo11	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mo12	0	0	0	0	0	0	0	0	0	2	0	0	0	0
mo13	0	0	0	0	0	0	0	0	0	0	0	0	0	1
mo14	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 6. The adjacency matrix for the prismatic part 2.

### 3.4. Mathematical Model of Process Planning

So far, the most popular evaluation criteria for the PPO problem are the minimization of the total machining time and the minimization of the total machining cost. The criterion of the total machining time has been successfully used in [9,11,20,21,25]. The machining time is composed of five time factors: machine processing time, tool processing time, transportation time (between machines), total tool change time and total setup change time. On the other side, the minimization of the total machining cost has been more frequently considered in scientific studies [4,6,8,10,11,14–16,24,29,31,32]. This criterion is adopted here, and consists of five cost components that are described below.

1. Total machine cost (TMC) represents the total cost of all machines that are used in a process plan and is calculated using the following equation:

$$TMC = \sum_{i=1}^n MCI_j \tag{3}$$

where  $j$  stands for  $operations[i].Machines$  (i.e.,  $j = 1 \dots nMach$ , where  $nMach$  stands for total number of machines),  $n$  is the total number of machining operations while  $MCI_j$  represents the machine candidate cost index for using a machine  $j$ , a constant cost value for a specific machine.

2. Total cutting tool cost (TCTC) is the total sum of cutting tool costs used in a process plan and is calculated in the following way:

$$TCTC = \sum_{i=1}^n CTCT_j \tag{4}$$

where  $j$  stands for  $operations[i].Tools$  ( $j = 1 \dots nTools$ , where  $nTools$  stands for total number of cutting tools),  $CTCI_j$  stands for the cutting tool cost index for using a cutting tool  $j$ , a constant cost value for a specific cutting tool.

3. Total machine change cost ( $TMCC$ ) and the number of machine changes ( $NMC$ ) are important cost factors considered when two successive machining operations in  $operations[n]$  are performed on different machines. Number of machine changes is computed as:

$$NMC = \sum_{i=1}^{n-1} \Omega_1 (operations[i].Machines, operations[i+1].Machines) \quad (5)$$

$$TMCC = \sum_1^{NMC} MCC \quad (6)$$

$$\Omega_1(X, Y) = \begin{cases} 1, & \text{if } X \neq Y \\ 0, & \text{if } X = Y \end{cases} \quad (7)$$

where  $MCC$  represents the machine candidate change cost index while  $operations[i].Machines$  stands for the machine ID used for performing the machining operation  $i$  from  $operations[n]$ .

4. Total cutting tool change cost ( $TCTCC$ ) and the number of cutting tool changes ( $NCTC$ ) are similarly significant cost factors compared to those of machines. Cutting tool changes do not occur only in situations when same cutting tool and same machine are used for machining two successive machining operations. In other cases, tool changes are required. Accordingly, number of cutting tool changes is computed as follows:

$$NCTC = \sum_{i=1}^{n-1} \Omega_2 \left( \begin{array}{c} \Omega_1 \left( \begin{array}{c} operations[i].Machines, \\ operations[i+1].Machines \end{array} \right), \\ \Omega_1 \left( \begin{array}{c} operations[i].Tools, \\ operations[i+1].Tools \end{array} \right) \end{array} \right) \quad (8)$$

$$TCTCC = \sum_1^{NCTC} CTCC \quad (9)$$

$$\Omega_2(X, Y) = \begin{cases} 0, & X = Y = 0 \\ 1, & \text{otherwise} \end{cases} \quad (10)$$

where  $CTCC$  is the cutting tool change cost index and  $operations[i].Tools$  stands for the cutting tool ID used for performing the operation  $i$  from  $operations[n]$ .

5. Total setup cost ( $TSC$ ), number of setup changes ( $NSC$ ) and the number of setups ( $NS$ ) are the cost factors considered when two successive machining operations are not performed on the same machine and using the same tool approach direction. This stands for 3-axis machines which are considered in Section 4. Firstly, the  $NSC$  is calculated as follows:

$$SC = \sum_{i=1}^{n-1} \Omega_2 \left( \begin{array}{c} \Omega_1 \left( \begin{array}{c} operations[i].Machines, \\ operations[i+1].Machines \end{array} \right), \\ \Omega_1 \left( \begin{array}{c} operations[i].TADs, \\ operations[i+1].TADs \end{array} \right) \end{array} \right) \quad (11)$$

The corresponding  $NS$  is calculated as:

$$NS = NSC + 1 \quad (12)$$

Number one assumes the starting machine setup which is then added to the corresponding number of setup changes calculated by Equation (11). Lastly, the  $TSC$  costs are computed as:

$$TSC = \sum_1^{NS} SCC \quad (13)$$

where  $SCC$  represent the setup change cost index and  $operations[i].TADs$  stands for the TAD ID used for performing the operation  $i$  from  $operations[n]$ . Similar to tool changes, setup

changes do not appear only when the same TAD and the same machine are used when performing two successive machining operations. In other cases, the change is required.

6. Additional penalty costs (*APC*) and the number of violating constraints (*NVC*) are the cost factors included in the part of our study focused on soft and hard constraints. Even if soft constraints are allowed to be violated, number of violations is subject to minimization. Firstly, the number of violated soft constraints is calculated using the following equation:

$$NVC = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \Omega_3 \left( \begin{array}{l} operations[i], \\ operations[j] \end{array} \right) \quad (14)$$

where *operations[i]* and *operations[j]* stand for two consecutive machining operations in *operations[n]* vector. The *APC* is thereby calculated as:

$$APC = \sum_1^{NVC} penalty \quad (15)$$

$$\Omega_3(X, Y) = \begin{cases} 0, & X \rightarrow Y \text{ violates soft constr.} \\ 1, & X \rightarrow Y \text{ in accordance with soft constr.} \end{cases} \quad (16)$$

where *penalty* assumes the fixed penalty cost index which is applied for each violated soft constraint. In this study, the penalty cost applies to prismatic part 2 that involves both hard and soft constraints. As the conflicts between these two types of precedence constraints may occur, hard constraints become the priority. In that case, soft constraints that are numerically assigned with number 2 in Figure 6, have to be violated.

7. Total weighted machining cost (*TWMC*) sums up all the previous cost factors in the following equation:

$$TWMC = w_1 \cdot TMC + w_2 \cdot TCTC + w_3 \cdot TMCC + w_4 \cdot TCTCC + w_5 \cdot TSC + w_6 \cdot APC \quad (17)$$

where  $w_1 - w_6$  are the weight coefficients used for experimental studies. The *APC* costs are not included in Equation (17) in those studies which consider only hard precedence constraints.

8. Fitness function is maximized in order to verify the total weighted machining costs of a process plan and is defined as follows:

$$f_f = \frac{1}{TWMC} \quad (18)$$

## 4. A Hybrid Grey Wolf Optimizer (HGWO) for Process Planning Optimization

### 4.1. Traditional Grey Wolf Optimizer

Grey wolf optimizer (GWO) is proposed by Mirjalili et al. [37]. It is a modern swarm-based algorithm whose inspiration derives from the social intelligence of grey wolves.

Grey wolves (lat. *Canis lupus*) are predatory animals which brings them to the top of food chain in animal world. The advantages of social life of grey wolves mostly reflect in social hunting, group care about infants and territorial defense against dangerous forces outside the pack. A strict social hierarchy among members of the pack is what brings grey wolves to the fore. From the most dominant to the most submissive wolves there are alphas, betas, deltas and omegas. According to the social dominance hierarchy, a mathematical model of GWO was designed. Three main stages of grey wolf hunting process, such as tracking, encircling and attacking a prey, are included in this model.

According to the social dominance hierarchy previously described, a mathematical model of GWO was designed. Three main stages of grey wolf hunting process, such as tracking, encircling and attacking a prey, are included in this model.

When designing the social hierarchy in the GWO, the three best solutions in a population of individuals represent alpha ( $\alpha$ ), beta ( $\beta$ ) and delta ( $\delta$ ) wolves, respectively. All the other candidates are considered to be omegas ( $\omega$ ). The hunting (optimization) process is guided by the three fittest wolves.

The first mechanism in the GWO is encircling of prey which is defined according to the following equations:

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_p(it) - \vec{X}(it) \right| \quad (19)$$

$$\vec{X}(it+1) = \vec{X}_p(it) - \vec{A} \cdot \vec{D} \quad (20)$$

$$\vec{A} = 2\vec{a} \vec{r}_1 - \vec{a} \quad (21)$$

$$\vec{C} = 2\vec{r}_2 \quad (22)$$

where  $it$  stands for current iteration,  $\vec{A}$  and  $\vec{C}$  are coefficient vectors that are defined according to random numbers  $\vec{r}_1$  and  $\vec{r}_2$  in  $[0,1]$ .  $\vec{X}$  is the position vector of the prey while  $\vec{X}_p$  is the position vector of a grey wolf. Vector  $\vec{a}$  indicate a value that linearly decreases from 2 to 0.

When modelling hunting behaviour, the fittest wolves, the alpha ( $\alpha$ ), the beta ( $\beta$ ) and the delta ( $\delta$ ) have the main role. It is assumed that these wolves respectively have better knowledge about potential location of the prey. Therefore, omega wolves ( $\omega$ ) update their positions based on the positions of the alpha ( $\alpha$ ), the beta ( $\beta$ ) and the delta ( $\delta$ ). This process is mathematically formulated by the following equations:

$$\vec{D}_\alpha = \left| \vec{C}_1 \cdot \vec{X}_\alpha - \vec{X} \right| \quad \vec{D}_\beta = \left| \vec{C}_2 \cdot \vec{X}_\beta - \vec{X} \right| \quad \vec{D}_\delta = \left| \vec{C}_3 \cdot \vec{X}_\delta - \vec{X} \right| \quad (23)$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \left( \vec{D}_\alpha \right) \quad \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \left( \vec{D}_\beta \right) \quad \vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot \left( \vec{D}_\delta \right) \quad (24)$$

$$\vec{X}(it+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (25)$$

where  $\vec{X}$  represents a position of a grey wolf over a course of iterations  $it$  updated according to the distances from three fittest wolves, alpha,  $\vec{D}_\alpha$ , beta,  $\vec{D}_\beta$  and delta,  $\vec{D}_\delta$ .

Intensification and diversification of the GWO are maintained by the coefficient vector  $\vec{A}$  whose fluctuation range depends on linear decrease of the vector  $\vec{a}$ . During iterations, the vector  $\vec{a}$  tends to decrease the coefficient vector  $\vec{A}$  which has direct impact on exploitation and exploration. If the absolute value of  $\vec{A}$  is less than 1, the wolves perform an attack, i.e., converge towards the prey, and therefore local search abilities of the GWO are emphasized. On the other hand, if the absolute value of  $\vec{A}$  is greater than 1, the wolves diverge from the prey and global search abilities of the GWO are used.

Search for prey, or divergence from prey, is another component of the GWO that emphasizes exploration. It is defined by the coefficient vector  $\vec{C}$  which ranges from 0 to 2. Its purpose is to assign random weights to a potential prey and give wolves harder or easier way to reach it. In order to use global capacities of the GWO and avoid global optima, the value of the vector  $\vec{C}$  does not decrease linearly compared to the vector  $\vec{A}$ .

Grey wolf optimizer has been applied to various types of optimization problems to this day. One of its main advantages is less parameter tuning since all the vectors above mentioned are based on random values. However, main drawbacks of the GWO are low convergence rate and entrapment in local optima [20]. To overcome these issues, researchers have proposed different ways to improve or enhance the traditional GWO. Ahmed et al. [43] proposed niching GWO to deal with multi-modal optimization problems. To maintain balance between exploitation and exploration and avoid premature convergence, authors incorporated best features of the PSO algorithm and a local search technique. Yue et al. [44] developed a novel hybrid algorithm based on the GWO and fireworks algorithm (FWA)

in order to combine both global abilities of FWA and local abilities of GWO. Wang and Wang [45] used quantum computing principles and operations of differential evolution with grey wolf optimizer in order to improve performance when dealing with NP-complete combinatorial 0–1 knapsack problems. Martin et al. [46] proposed a novel discrete GWO algorithm which uses update rules to distinguish between exploitation and exploration stage. Jiang and Zhang [47] applied the GWO algorithm on job shop and flexible job shop scheduling problems. They embedded crossover operation and adaptive mutation method to maintain the search within discrete domain, and avoid premature convergence and local optima. Variable neighborhood search was also introduced to improve explorative capacities of the proposed GWO. Qin et al. [48] implemented hybrid discrete multi-objective GWO to solve the casting production scheduling problem where makespan, the total production cost and the total delivery delay time were used as objective functions. A strategy based on reducing job transportation time and processing time are adopted to improve initial solutions. Additionally, improved tabu search algorithm was incorporated into the GWO to improve performances of the proposed method. Premkumar et al. [49] developed a multi-objective GWO to solve the brushless direct current (BLDC) motor design problem. Firstly, the analytical model of the BLDC motor design problem that belongs to highly non-linear electromagnetic optimization problems is presented. MOGWO was verified on standard benchmark functions, and latter applied to mono and multi objective optimization of BLDC motor design problem.

In the next section, the hybrid grey wolf optimizer based on traditional GWO and GA operations is proposed for solving NP-hard process planning combinatorial problem.

#### 4.2. HGWO Methodology

When designing mathematical model of the HGWO method, certain adaptations of previous Equations (19), (20), (23) and (24) have to be considered. Firstly, the distances of gray wolves from the alpha ( $\alpha$ ), the beta ( $\beta$ ) and the delta ( $\delta$ ) wolf for the machine vector, the tool vector and the TAD vector respectively, are formulated as follows:

$$D_{\alpha,m} = \left| \vec{C}_1 \cdot \vec{X}_{\alpha,m} - \vec{X}_m \right|, D_{\beta,m} = \left| \vec{C}_2 \cdot \vec{X}_{\beta,m} - \vec{X}_m \right|, D_{\delta,m} = \left| \vec{C}_3 \cdot \vec{X}_{\delta,m} - \vec{X}_m \right| \quad (26)$$

$$D_{\alpha,t} = \left| \vec{C}_1 \cdot \vec{X}_{\alpha,t} - \vec{X}_t \right|, D_{\beta,t} = \left| \vec{C}_2 \cdot \vec{X}_{\beta,t} - \vec{X}_t \right|, D_{\delta,t} = \left| \vec{C}_3 \cdot \vec{X}_{\delta,t} - \vec{X}_t \right| \quad (27)$$

$$D_{\alpha,tad} = \left| \vec{C}_1 \cdot \vec{X}_{\alpha,tad} - \vec{X}_{tad} \right|, D_{\beta,tad} = \left| \vec{C}_2 \cdot \vec{X}_{\beta,tad} - \vec{X}_{tad} \right|, D_{\delta,tad} = \left| \vec{C}_3 \cdot \vec{X}_{\delta,tad} - \vec{X}_{tad} \right| \quad (28)$$

where  $\vec{X}_m$  stands for the machine position vector of a grey wolf,  $\vec{X}_{\alpha,m}$ ,  $\vec{X}_{\beta,m}$  and  $\vec{X}_{\delta,m}$  are the alpha ( $\alpha$ ), the beta ( $\beta$ ) and the delta ( $\delta$ ) wolf for the machine vector;  $\vec{X}_t$  is the tool position vector of a grey wolf,  $\vec{X}_{\alpha,t}$ ,  $\vec{X}_{\beta,t}$  and  $\vec{X}_{\delta,t}$  are the alpha ( $\alpha$ ), the beta ( $\beta$ ) and the delta ( $\delta$ ) wolf for the tool vector,  $\vec{X}_{tad}$  is the TAD position vector of a grey wolf and  $\vec{X}_{\alpha,tad}$ ,  $\vec{X}_{\beta,tad}$  and  $\vec{X}_{\delta,tad}$  are the alpha ( $\alpha$ ), the beta ( $\beta$ ) and the delta ( $\delta$ ) wolf for the TAD vector.

Furthermore, the position of a gray wolf in the iteration  $it + 1$  is formulated on the basis of the machine vector, the tool vector and the TAD vector relative to the position vectors of the alpha ( $\alpha$ ), the beta ( $\beta$ ) and the delta ( $\delta$ ) wolves. This can be expressed with the following expressions:

$$\vec{X}_m(it + 1) = \frac{\vec{X}_{1,m} + \vec{X}_{2,m} + \vec{X}_{3,m}}{3}, \vec{X}_t(it + 1) = \frac{\vec{X}_{1,t} + \vec{X}_{2,t} + \vec{X}_{3,t}}{3}, \vec{X}_{tad}(it + 1) = \frac{\vec{X}_{1,tad} + \vec{X}_{2,tad} + \vec{X}_{3,tad}}{3} \quad (29)$$

$$\vec{X}_{1,m} = \vec{X}_{\alpha,m} - \vec{A}_1 \cdot \left( \vec{D}_{\alpha,m} \right), \vec{X}_{2,m} = \vec{X}_{\beta,m} - \vec{A}_2 \cdot \left( \vec{D}_{\beta,m} \right), \vec{X}_{3,m} = \vec{X}_{\delta,m} - \vec{A}_3 \cdot \left( \vec{D}_{\delta,m} \right) \quad (30)$$

$$\vec{X}_{1,t} = \vec{X}_{\alpha,t} - \vec{A}_1 \cdot \left( \vec{D}_{\alpha,t} \right), \vec{X}_{2,t} = \vec{X}_{\beta,t} - \vec{A}_2 \cdot \left( \vec{D}_{\beta,t} \right), \vec{X}_{3,t} = \vec{X}_{\delta,t} - \vec{A}_3 \cdot \left( \vec{D}_{\delta,t} \right) \quad (31)$$

$$\vec{X}_{1,tad} = \vec{X}_{\alpha,tad} - \vec{A}_1 \cdot \left( \vec{D}_{\alpha,tad} \right), \vec{X}_{2,tad} = \vec{X}_{\beta,tad} - \vec{A}_2 \cdot \left( \vec{D}_{\beta,tad} \right), \vec{X}_{3,tad} = \vec{X}_{\delta,tad} - \vec{A}_3 \cdot \left( \vec{D}_{\delta,tad} \right) \quad (32)$$

where  $\vec{X}_m(it+1)$ ,  $\vec{X}_t(it+1)$  i  $\vec{X}_{tad}(it+1)$  represent the updated machine, tool and TAD position vectors of a grey wolf in iteration  $it+1$  respectively,  $\vec{X}_{1,m}$ ,  $\vec{X}_{2,m}$  and  $\vec{X}_{3,m}$  are the updated machine vectors according to the distance from the alpha ( $\alpha$ ), the beta ( $\beta$ ) and the delta ( $\delta$ ) wolf respectively,  $\vec{X}_{1,t}$ ,  $\vec{X}_{2,t}$  and  $\vec{X}_{3,t}$  are the updated tool vectors according to the distance from the alpha ( $\alpha$ ), the beta ( $\beta$ ) and the delta ( $\delta$ ) wolf respectively, and  $\vec{X}_{1,tad}$ ,  $\vec{X}_{2,tad}$  and  $\vec{X}_{3,tad}$  are the updated TAD vectors according to the distance from the alpha ( $\alpha$ ), the beta ( $\beta$ ) and the delta ( $\delta$ ) wolf respectively.

So far, no recent discoveries have shown the proposal of the GWO for solving process planning optimization problem. In this paper, we present a novel hybrid grey wolf optimizer (HGWO) approach to address the PPO problem with various precedence constraints focusing on prismatic parts. The pseudocode of the proposed approach can be seen in Figure 7. It includes the following components: (1) knowledge-based representation of process plans (positions of grey wolves), (2) initializing population (pack of wolves), (3) applying constraint handling heuristic, (4) fitness evaluation of positions of wolves, (5) traditional GWO steps for updating positions of wolves, (6) tournament selection, (7) crossover, (8) shift mutation, (9) resource mutation, (10) fitness re-evaluation and (11) termination criteria of the HGWO.

Hybrid grey wolf optimizer (HGWO)

**Start**

**Initialize parameters** maximal number of iterations, pack size, tournament size, crossover probability  $p_c$  and mutation probabilities  $p_{m1}$  and  $p_{m2}$ , maximal and minimal inertia  $w_{max}$  and  $w_{min}$

**Initialize the pack of wolves** where each grey wolf is defined by its position in the search space and according to the representation scheme described in Section 3.1

**Adjust** infeasible positions of grey wolves to the feasible domain using constraint handling heuristic

**Evaluate** the position of all wolves using fitness function described in Section 3.4

**Generate** positions of the three fittest wolves in the pack—alpha ( $\alpha$ ), beta ( $\beta$ ) and delta ( $\delta$ )

**Iterate** for iteration = 1: maximal number of iterations

**Repeat** for each crow in the flock

**Initialize** GWO parameters—vector coefficients  $\vec{a}$ ,  $\vec{A}$  and  $\vec{C}$

**Generate** new positions of grey wolves according to the Eq. (29)

**End Repeat**

**Select** random wolves to participate in the tournament selection as described in Section 4.2.2

**Perform** crossover of the selected wolves according to the predefined crossover probability  $p_c$

**Select** random wolves for shift mutation according to the predefined mutation probability  $p_{m1}$

**Adjust** infeasible positions of grey wolves to feasible domain using constraint handling heuristic

**Select** random wolves for resource mutation according to the predefined mutation probability  $p_{m2}$

**Evaluate** newly obtained positions of all wolves in the pack

**Update** positions of the three fittest wolves in the pack—alpha ( $\alpha$ ), beta ( $\beta$ ) and delta ( $\delta$ )

**End** after maximal number of iterations is reached

**Output** best results

Figure 7. Pseudocode of the proposed HGWO.

#### 4.2.1. Initialization of Population

The first step after defining the representation of solutions is to initialize a starting population of individuals, here described as a pack of wolves. As mentioned earlier,

the modified knowledge-based representation is adopted in this study. For the adopted representation of individuals, a population can be initialized. To ensure the feasibility of individuals, the search methodology has to include an appropriate approach for converting infeasible individuals to a feasible domain. Therefore, we propose two heuristic algorithms for constraint handling that will be incorporated into the proposed HGWO in order to guarantee feasible solutions. The first heuristic is adopted from [8] and utilized for dealing with hard constraints only. The other heuristic was developed in [39] to address both hard and soft constraints.

Using the information about precedence constraints and operations precedence graph let us assume that a randomly generated sequence of operations for a single individual from a population would have the following form: [mo<sub>2</sub>, mo<sub>5</sub>, mo<sub>6</sub>, mo<sub>8</sub>, mo<sub>1</sub>, mo<sub>9</sub>, mo<sub>4</sub>, mo<sub>3</sub>, mo<sub>7</sub>]. To complete the initialization of a process plan, randomly selected machine, cutting tool and TAD candidate are assigned to each of these machining operations to form the operation units. Here, we will focus on the operations sequence which is crucial for elimination of infeasible solutions. Using the adjacency matrix as an input information, the constraint handling heuristic for the hard constraints is developed, Figure 8. The step-by-step procedure of the constraint handling heuristic algorithm is the following:

1. A vector *operation*[*n*] is used for representing a single solution; its adjacency matrix is given in Equation (2);
2. A variable *pt*<sub>1</sub> (*pt*<sub>1</sub> = *n*) representing an index pointer of the vector *operations*[*n*] is generated;
3. Repeat the procedure within the for loop until *pt*<sub>1</sub> = 0;
4. Identify the machining operation at the point *pt*<sub>1</sub> = *n*;
5. For the identified machining operation calculate the sum of all number values in a corresponding row of the adjacency matrix  $\sum_{i=1}^n (PR_{ij})$ ; Then, check the conditions:
  - I. If  $\sum_{i=1}^n (PR_{ij}) = 0$ 
    - (a) Leave the machining operation at the position *pt*<sub>1</sub>;
    - (b) Exclude the row and the column of the *Adj\_matrix* that match the value of the machining operation at the position *pt*<sub>1</sub>; instead of removing the row and the column, assign “not a number”;
    - (c) Initialize *n* = *n* − 1 and move pointer *pt*<sub>1</sub> one place left;
  - II. If  $\sum_{i=1}^n (PR_{ij}) \neq 0$ 
    - (a) Initialize *pt*<sub>2</sub> = *pt*<sub>1</sub> − 1;
    - (b) Identify the machining operation at the position *pt*<sub>2</sub>;
    - (c) Calculate the sum  $\sum_{i=1}^n (PR_{ij})$  of all number values in the corresponding row that matches the identified machining operation at the position *pt*<sub>2</sub>;
    - (d) Repeat the following steps while *pt*<sub>2</sub> > 0;
    - (e) If  $\sum_{i=1}^n (PR_{ij}) = 0$ 
      - Swap the machining operation at the position *pt*<sub>1</sub> with the machining operation at the position *pt*<sub>2</sub>;
      - Remove the row and the column of the corresponding *Adj\_matrix* which match the value of the machining operation at the position *pt*<sub>2</sub>; assign “not a number”;
      - Break the loop and move to the step (3);
    - (f) If  $\sum_{i=1}^n (PR_{ij}) \neq 0$ 
      - Initialize *pt*<sub>2</sub> = *pt*<sub>2</sub> − 1;
      - Move to the step (d);
6. Return to the step (3) and reduce the pointer *pt*<sub>1</sub> = *pt*<sub>1</sub> − 1;
7. Reset the *Adj\_matrix* and apply all the steps for the next individual.



**Figure 8.** The step-by-step procedure of the constraint handling heuristic: (a) Select the pointer identifying the last machining operation in the sequence, mo<sub>7</sub>; (b) Swap mo<sub>7</sub> and mo<sub>3</sub> according to the condition in Step 5.II; (c) Exclude row and column of mo<sub>3</sub>, and select the next one, mo<sub>4</sub>; (d) Swap mo<sub>7</sub> and mo<sub>4</sub> according to the Step 5.II, exclude row and column of mo<sub>4</sub>, and select the next operation, mo<sub>9</sub>; (e) Swap positions of mo<sub>7</sub> and mo<sub>9</sub>, exclude row and column of mo<sub>9</sub>, and select the next operation, mo<sub>8</sub>; mo<sub>1</sub> is skipped according to the Step 5.II.f; (f) Swap positions of mo<sub>7</sub> and mo<sub>8</sub>, exclude row and column of mo<sub>8</sub>, and select the next operation, mo<sub>1</sub>; (g) Swap positions of mo<sub>7</sub> and mo<sub>1</sub>, exclude row and column of mo<sub>7</sub>, and select the next operations, mo<sub>1</sub> and mo<sub>6</sub>; (h) Leave mo<sub>6</sub> and mo<sub>1</sub> as positioned, exclude row and column of mo<sub>6</sub>, and select the next operation, mo<sub>5</sub>; (i) mo<sub>1</sub> goes before mo<sub>5</sub> and before the last operation mo<sub>2</sub>; the final sequence is generated.

The step-by step constraint handling procedure begins with the adjacency matrix given in Figure 8a. The procedure is based on identifying index pointers  $pt_1$  and modifying the sequence within the for loop. The first selected pointer  $pt_1 = n = 9$  denotes the last machining operation in the matrix, mo<sub>7</sub>. The sum of all values in the row 9 meets the second condition in the Step 5 meaning that  $\sum_{i=1}^n (PR_{ij}) \neq 0$ . Then, the Step 5.II.a starts with initializing the second pointer  $pt_2 = pt_1 - 1 = 8$  that denotes mo<sub>3</sub>, Figure 8b. The same condition is checked and since the sum  $\sum_{i=1}^n (PR_{ij})$  equals zero, the operations mo<sub>7</sub> and mo<sub>3</sub> are swapped and the row and the column of the mo<sub>3</sub> are assigned with the non-numbering value. Operation mo<sub>3</sub> then takes the final position 8 in the sequence, Figure 8c. The procedure continues with the Step 6 that reduces the pointer  $pt_1$  to 8 and repeats from the Step 5. Pointers  $pt_1 = 8$  and  $pt_2 = 7$  denote the operations mo<sub>7</sub> and mo<sub>4</sub>, Figure 8c. After checking conditions in Step 5.II, the row and the column of mo<sub>4</sub> are assigned with the non-numbering value and removed from the following steps, Figure 8d. From Figure 8d to Figure 8i, the handling procedure continues to remove the rows and the columns of machining operations for which the sum  $\sum_{i=1}^n (PR_{ij})$  equals zero. The sequence of operations is modified accordingly. Figure 8i shows that the operation mo<sub>5</sub>



has to swap place with  $mo_1$  in order to complete the procedure of generating a feasible machining sequence.

The second constraint handling heuristic is adopted from [3]. Here, we will shortly focus on the most significant steps of this heuristic. In order to handle both hard and soft constraints, firstly, so called “linked list” is created. This list includes machining operations that form hard constraints, therefore leaving the positions of other operations unchanged. Herewith, a heuristic process is imposed on the linked list to ensure its consistency with the hard constraints. Such an approach is adopted for the part 2 shown in Figure 5. Afterwards, the next step of this constraint handling algorithm is to include the additional penalty costs for violated soft constraints in the objective function model, Equations (15) and (16). Considering the fact that soft constraints are not manipulated using the linked list, certain soft constraints may be subject to violation. In this step, soft constraints can be compromised and violated to achieve the minimal total weighted machining cost, Equation (17).

#### 4.2.2. Genetic Components of the HGWO Approach

Genetic algorithms are techniques based on natural evolution of species that mimic the viewpoint of modern genetics, so-called “survival of the fittest”. Apart from being fitness-oriented and belonging to the group of population-based optimization algorithms, their unique characterization is varying operations that mimic genetic gene changes and enable population individuals to evolve. These strategies are selection, crossover and mutation.

The adopted selection strategy for the HGWO algorithm is the tournament selection which belongs to the group of proportionate-based selection schemes. This strategy is based on selecting a number of individuals (tournament size) from the population that will participate in the so called “tournament”. The individual with the highest fitness value is considered as the fittest individual and therefore the winner of the tournament. The process of selecting individuals lasts until a new generation of individuals is created.

Crossover or recombination is the next strategy employed to recombine two selected parent individuals in order to obtain better offsprings. In this paper, a modified one-point crossover strategy is adopted to improve exploration capabilities of the proposed approach. Using the predefined crossover probability  $p_c$ , the steps of this recombination procedure are the following:

- Two wolf candidates selected from the tournaments are randomly chosen to be parent wolves;
- By generating a random crossover point, two parent wolves are divided into two sections to create two child wolves;
- Left section of the child wolf 1 is formed by copying left section of the parent wolf 1. Then, the right section of the child wolf 1 is formed in two parts. The first part is to find the remaining machining operations in the parent wolf 2 and copy them in the current order to the right section of the child wolf 1. Afterwards, the machine, cutting tool and TAD candidate are copied from these machining operations in the parent wolf 2 to the remaining machining operations in right section of the child wolf 1;
- The identical, but inverse procedure using left section of the parent wolf 2 and remaining machining operations from the parent wolf 1 is performed to produce the child wolf 2.

To improve exploration capabilities of the HGWO and avoid premature convergence, two appropriate mutation strategies are introduced. Here, the shift mutation and resource mutation strategy are used. The first strategy, takes two random genes (operation units) of a randomly selected wolf from the population (pack) and exchanges them. This procedure impacts the feasibility of the mutated position of the wolf.

The second mutation strategy called “resource mutation” does minor changes to a machine, cutting tool and TAD candidate, respectively. The steps of the resource mutation strategy (example for a machine vector) are the following:

- Randomly select a crow candidate for resource mutation;

- Randomly select the mutation point, i.e., machining operation that matches certain position in the vector;
- Check the availability of machines in the machine candidate list for the selected machining operation;
- Randomly select one of the available machines as the current machine;
- Identify all other machining operations that have the current machine in their machine candidate list;
- Assign the same alternative machine candidate as the current machine for remaining machining operations;
- Repeat the same steps for the CT vector and TAD vector using CT candidate list and TAD candidate list respectively.

Besides the above described genetic strategies within the HGWO approach, we also included inertia weight coefficient in order to achieve additional control of exploration and exploitation. The inertia is decreased linearly thereby emphasizing global exploration ability in the initial stages of the search process and moving towards local exploitation abilities in the latter stages. Convergence curves in Chapter 5 graphically show diversification of the search (primarily HGWO and HybGA curves) in the initial stages meaning that the exploration is at hand, whereas the latter stages clearly show less diversified search where local abilities of metaheuristics are more used. In that sense, the mathematical Equations (25)–(27) defined in the Section 4.2 will have the following form:

$$\vec{D}_{\alpha,m} = \left| \vec{C}_1 \cdot \vec{X}_{\alpha,m} - w \cdot \vec{X}_m \right|, \vec{D}_{\beta,m} = \left| \vec{C}_2 \cdot \vec{X}_{\beta,m} - w \cdot \vec{X}_m \right|, \vec{D}_{\delta,m} = \left| \vec{C}_3 \cdot \vec{X}_{\delta,m} - w \cdot \vec{X}_m \right| \quad (33)$$

$$\vec{D}_{\alpha,t} = \left| \vec{C}_1 \cdot \vec{X}_{\alpha,t} - w \cdot \vec{X}_t \right|, \vec{D}_{\beta,t} = \left| \vec{C}_2 \cdot \vec{X}_{\beta,t} - w \cdot \vec{X}_t \right|, \vec{D}_{\delta,t} = \left| \vec{C}_3 \cdot \vec{X}_{\delta,t} - w \cdot \vec{X}_t \right| \quad (34)$$

$$\vec{D}_{\alpha,tad} = \left| \vec{C}_1 \cdot \vec{X}_{\alpha,tad} - w \cdot \vec{X}_{tad} \right|, \vec{D}_{\beta,tad} = \left| \vec{C}_2 \cdot \vec{X}_{\beta,tad} - w \cdot \vec{X}_{tad} \right|, \vec{D}_{\delta,tad} = \left| \vec{C}_3 \cdot \vec{X}_{\delta,tad} - w \cdot \vec{X}_{tad} \right| \quad (35)$$

where  $w$  represents inertia weight that is decreased linearly using the next equation:

$$w = ((w_{max} - w_{min}) / MaxIt) \cdot it \quad (36)$$

where  $w_{max}$  and  $w_{min}$  are maximal and minimal weight,  $MaxIt$  is the maximal number of iterations and  $it$  is the current iteration.

## 5. Experimental Studies and Results

The proposed HGWO algorithm was programmed in MATLAB environment and executed on a Windows 10 operating system by a 1.99 GHz Intel i7 processor and 8 GB RAM computer. Its performance was verified on three experimental studies that consider three different prismatic parts. The first experiment employed the prismatic part 1 taken from [42]. The 3D solid model with associated features and its cOPG are represented in Figure 3. The second experiment for testing the HECSA algorithm is the prismatic part 2 taken from [3]. Its model representation and dOPG are presented in Figure 5. The prismatic part 3 in the third experiment is adopted from [6]. Figures 9 and 10 depict the semi-transparent 3D model with its dOPG and adjacency matrix, respectively. Compared to the first and the second experimental study which considered prismatic parts with hard, and both hard and soft constraints respectively, precedence relationships in this model require certain operations to be performed in the same setup. In that sense, number 2 is added to the dOPG (e.g., mo<sub>10</sub> precedes mo<sub>3</sub> and both should be performed in the same setup). The prismatic part 3 is modeled for one condition in which all resources are available.

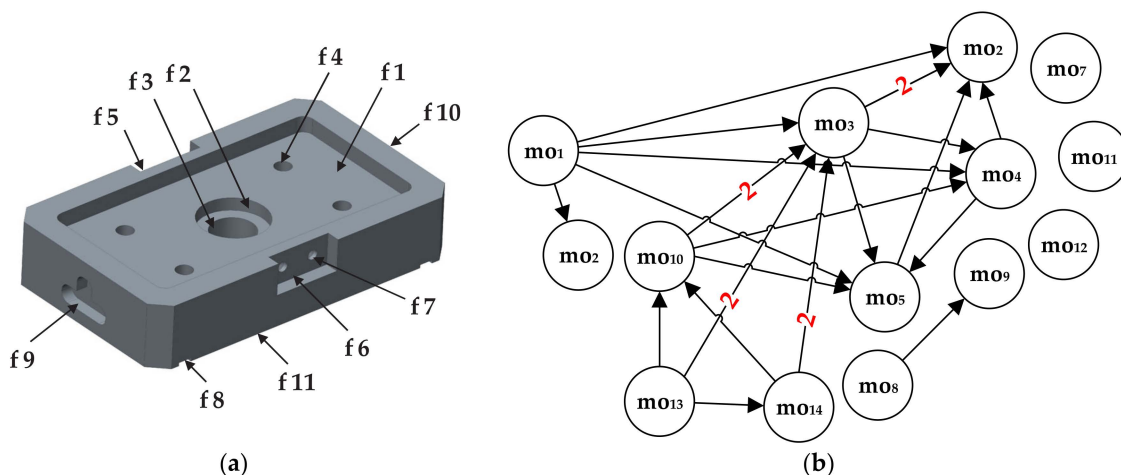


Figure 9. (a) Semi-transparent solid model and (b) dOPG of prismatic part 3.

	mo1	mo2	mo3	mo4	mo5	mo6	mo7	mo8	mo9	mo10	mo11	mo12	mo13	mo14
mo1	0	1	1	1	1	1	0	0	0	0	0	0	0	0
mo2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mo3	0	2	0	1	1	0	0	0	0	0	0	0	0	0
mo4	0	1	0	0	0	0	0	0	0	0	0	0	0	0
mo5	0	1	0	1	0	0	0	0	0	0	0	0	0	0
mo6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mo7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mo8	0	0	0	0	0	0	0	0	1	0	0	0	0	0
mo9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mo10	0	0	2	1	1	0	0	0	0	0	0	0	0	0
mo11	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mo12	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mo13	0	0	2	0	0	0	0	0	0	1	0	0	0	1
mo14	0	0	2	0	0	0	0	0	0	1	0	0	0	0

Figure 10. The adjacency matrix for prismatic part 3.

The detailed information about features and candidates for machines, cutting tools and TADs for prismatic part 1 is given in Table 1. Precedence relationships among machining features for prismatic part 1 are presented in Table 2. Similarly, information about features and resource candidates, as well as information about precedence relationships for prismatic part 2 are given in Tables 3 and 4, respectively. Lastly, information about features and resource candidates for prismatic part 3 are shown in Table 5. Available machines and cutting tools as well as the machining cost information about three prismatic parts are given in Tables 6 and 7, respectively.

Parameter tuning may be one of the most time-consuming challenges when testing the performances of metaheuristic algorithms. In this paper, we performed several manual tests of input parameters of the HGWO and finally we adopted the following: pack size:  $N = 100$ , maximal number of iterations:  $MaxIter = 700$ , tournament size:  $TourSize = 5$ , crossover probability:  $p_c = 0.8$ , shift mutation probability  $p_{m1} = 0.4$ , resource mutation probability:  $p_{m2} = 0.6$  and maximal and minimal inertia weights are  $w_{max} = 1.2$  and  $w_{min} = 0.2$ . The adopted optimization objective is to minimize the total weighted machining cost defined in Section 3.4. The output results included minimum and maximum results achieved in 20 runs, and mean values of the obtained results.

**Table 1.** Information about features and resource candidates for prismatic part 1.

Feat.	Feature Definition	Machining Operations	Machines	Cutting Tools	TADs
f1	Planar surface	Milling (mo <sub>1</sub> )	M <sub>2</sub> ,M <sub>3</sub>	CT <sub>6</sub> ,CT <sub>7</sub> ,CT <sub>8</sub>	TAD <sub>+z</sub>
f2	Planar surface	Milling (mo <sub>2</sub> )	M <sub>2</sub> ,M <sub>3</sub>	CT <sub>6</sub> ,CT <sub>7</sub> ,CT <sub>8</sub>	TAD <sub>-z</sub>
f3	Two pockets arranged as a replicated feature	Milling (mo <sub>3</sub> )	M <sub>2</sub> ,M <sub>3</sub>	CT <sub>6</sub> ,CT <sub>7</sub> ,CT <sub>8</sub>	TAD <sub>+x</sub>
f4	Four holes arranged as a replicated feature	Drilling (mo <sub>4</sub> )	M <sub>1</sub> ,M <sub>2</sub> ,M <sub>3</sub>	CT <sub>2</sub>	TAD <sub>+z</sub> , TAD <sub>-z</sub>
f5	A step	Milling (mo <sub>5</sub> )	M <sub>2</sub> ,M <sub>3</sub>	CT <sub>6</sub> ,CT <sub>7</sub>	TAD <sub>+x</sub> , TAD <sub>-z</sub>
f6	A rib	Milling (mo <sub>6</sub> )	M <sub>2</sub> ,M <sub>3</sub>	CT <sub>7</sub> ,CT <sub>8</sub>	TAD <sub>+y</sub> , TAD <sub>-z</sub>
f7	A boss	Milling (mo <sub>7</sub> )	M <sub>2</sub> ,M <sub>3</sub>	CT <sub>7</sub> ,CT <sub>8</sub>	TAD <sub>-a</sub>
f8	A compound hole	Drilling (mo <sub>8</sub> ) Reaming (mo <sub>9</sub> ) Boring (mo <sub>10</sub> )	M <sub>1</sub> ,M <sub>2</sub> ,M <sub>3</sub> M <sub>1</sub> ,M <sub>2</sub> ,M <sub>3</sub> M <sub>2</sub> ,M <sub>3</sub>	CT <sub>2</sub> ,CT <sub>3</sub> ,CT <sub>4</sub> CT <sub>9</sub> CT <sub>10</sub>	TAD <sub>-a</sub>
f9	A rib	Milling (mo <sub>11</sub> )	M <sub>2</sub> ,M <sub>3</sub>	CT <sub>7</sub> ,CT <sub>8</sub>	TAD <sub>-y</sub> , TAD <sub>-z</sub>
f10	A compound hole	Drilling (mo <sub>12</sub> ) Reaming (mo <sub>13</sub> ) Boring (mo <sub>14</sub> )	M <sub>1</sub> ,M <sub>2</sub> ,M <sub>3</sub> M <sub>1</sub> ,M <sub>2</sub> ,M <sub>3</sub> M <sub>3</sub> ,M <sub>4</sub>	CT <sub>2</sub> ,CT <sub>3</sub> ,CT <sub>4</sub> CT <sub>9</sub> CT <sub>10</sub>	TAD <sub>-z</sub>
f11	Nine holes arranged as a replicated feature	Drilling (mo <sub>15</sub> ) Tapping (mo <sub>16</sub> )	M <sub>1</sub> ,M <sub>2</sub> ,M <sub>3</sub> M <sub>1</sub> ,M <sub>2</sub> ,M <sub>3</sub>	CT <sub>1</sub> CT <sub>5</sub>	TAD <sub>-z</sub>
f12	A pocket	Milling (mo <sub>17</sub> )	M <sub>2</sub> ,M <sub>3</sub>	CT <sub>7</sub> ,CT <sub>8</sub>	TAD <sub>-x</sub>
f13	A step	Milling (mo <sub>18</sub> )	M <sub>2</sub> ,M <sub>3</sub>	CT <sub>6</sub> , CT <sub>7</sub>	TAD <sub>-x</sub> , TAD <sub>-z</sub>
f14	A compound hole	Reaming (mo <sub>19</sub> ) Boring (mo <sub>20</sub> )	M <sub>1</sub> ,M <sub>2</sub> ,M <sub>3</sub> M <sub>3</sub> ,M <sub>4</sub>	CT <sub>9</sub> CT <sub>10</sub>	TAD <sub>+z</sub>

The optimal process plans for minimal machining cost and three conditions concerning prismatic part 1 are shown in Table 8. In the first condition, all resources are available. The second condition excludes costs for cutting tools and cutting tool changes. The third condition is the same as the second, with unavailability of machine 2 and cutting tool 8.

To perform detailed comparative analysis, results were compared with different approaches in the literature, HGASA by Li et al. [5], PSO by Guo et al. [42], HybGA by Huang et al. [8], HBMO by Wen et al. [24], ACO by Wang et al. [15], TSACO by Wang et al. [16], cPSO by Petrović et al. [20], mACO by Hu et al. [17], ESGA by Su et al. [11], IWD by Gao et al. [32], TS, SA, GA and ACO by Liu et al. [14] and GA by Kafashi [6]. Numerical information about the comparative studies for three prismatic parts is given in Table 9.

Firstly, we will discuss the results for the prismatic part 1. For the first condition where all resources are available, the HGWO obtained the minimal TWMC of 2470 cost units in 20 runs. The total of six results have values below 2500 cost units. According to Table 9, the minimal TWMC of 2470 is one of the best results compared to the results obtained by other algorithms. Only the approach reported in [15] achieved better minimum. The average cost obtained by the HGWO is better than the average cost achieved by cPSO [20], PSO [42], and mACO [17]. However, the maximal value of 2805 shows a lack of consistency in achieved results compared to other metaheuristics. Considering the second condition where CT cost and CT change cost are not included, the TWMC of 1990 cost units appears to be the best minimal result. As shown in Table 9, HGWO outperforms all other metaheuristic approaches in terms of the minimal TWMC and the mean TWMC. On the other hand, the achieved maximal value was better compared to ACO [15], TSACO [16] and IWD [32] approaches. Lastly, the HGWO showed best performances for the third condition in which machine M<sub>2</sub> and cutting tool CT<sub>8</sub> are not available. Best minimal TWMC of 2490 cost units is found. Additionally, regarding the mean and the maximal values of the TWMC,

the HGWO demonstrated much better consistency and effectiveness than the considered modified and hybrid approaches.

**Table 2.** Precedence relationships for prismatic part 1.

Features	Machining Operations	Precedence Constraint Description
f1	Milling (mo <sub>1</sub> )	f1 (mo <sub>1</sub> ) is the datum and supporting face for the part; therefore it is machined before all features and operations
f2	Milling (mo <sub>2</sub> )	f2 (mo <sub>2</sub> ) is before f10 (mo <sub>12</sub> , mo <sub>13</sub> , mo <sub>14</sub> ) and f11 (mo <sub>15</sub> , mo <sub>16</sub> ) for the material removal interactions
f3	Milling (mo <sub>3</sub> )	
f4	Drilling (mo <sub>4</sub> )	
f5	Milling (mo <sub>5</sub> )	f5 (mo <sub>5</sub> ) is before f4 (mo <sub>4</sub> ) and f7 (mo <sub>7</sub> ) for the datum interactions
f6	Milling (mo <sub>6</sub> )	f6 (mo <sub>6</sub> ) is before f10 (mo <sub>12</sub> , mo <sub>13</sub> , mo <sub>14</sub> ) for the datum interaction
f7	Milling (mo <sub>7</sub> )	f7 (mo <sub>7</sub> ) is before f8 (mo <sub>8</sub> , mo <sub>9</sub> , mo <sub>10</sub> ) for the datum interactions
f8	Drilling (mo <sub>8</sub> ) Reaming (mo <sub>9</sub> ) Boring (mo <sub>10</sub> )	mo <sub>8</sub> is before (mo <sub>9</sub> and mo <sub>10</sub> ); mo <sub>9</sub> is before mo <sub>10</sub> for the fixed order of machining operations
f9	Milling (mo <sub>11</sub> )	f9 (mo <sub>11</sub> ) is before f10 (mo <sub>12</sub> , mo <sub>13</sub> , mo <sub>14</sub> ) for the datum interaction
f10	Drilling (mo <sub>12</sub> ) Reaming (mo <sub>13</sub> ) Boring (mo <sub>14</sub> )	mo <sub>12</sub> is before (mo <sub>13</sub> and mo <sub>14</sub> ); mo <sub>13</sub> is before mo <sub>14</sub> for the fixed order of machining operations; f10 (mo <sub>12</sub> , mo <sub>13</sub> , mo <sub>14</sub> ) is before f11 (mo <sub>15</sub> , mo <sub>16</sub> ) and mo <sub>12</sub> of f10 is before f14 (mo <sub>19</sub> , mo <sub>20</sub> ) for the datum interaction
f11	Drilling (mo <sub>15</sub> ) Tapping (mo <sub>16</sub> )	mo <sub>15</sub> is before mo <sub>16</sub> for the fixed order of operations
f12	Milling (mo <sub>17</sub> )	
f13	Milling (mo <sub>18</sub> )	f13 (mo <sub>18</sub> ) is before f4 (mo <sub>4</sub> ) and f12 (mo <sub>17</sub> ) for the material removal interaction
f14	Reaming (mo <sub>19</sub> ) Boring (mo <sub>20</sub> )	mo <sub>19</sub> is before mo <sub>20</sub> for the fixed order of machining operations

**Table 3.** Information about features and resource candidates for prismatic part 2.

Feat.	Feature Definition	Machining Operations	Machines	Cutting Tools	TADs
f1	Two holes as a replicated feature	Drilling (mo <sub>1</sub> )	M <sub>1</sub> ,M <sub>2</sub> ,M <sub>3</sub>	CT <sub>1</sub>	TAD <sub>+z</sub> , TAD <sub>-z</sub>
f2	A chamfer	Drilling (mo <sub>2</sub> )	M <sub>2</sub> ,M <sub>3</sub>	CT <sub>8</sub>	TAD <sub>-x</sub> , TAD <sub>+y</sub> , TAD <sub>-y</sub> , TAD <sub>-z</sub>
f3	A slot	Milling (mo <sub>3</sub> )	M <sub>2</sub> ,M <sub>3</sub>	CT <sub>5</sub> ,CT <sub>6</sub>	TAD <sub>+y</sub>
f4	A slot	Milling (mo <sub>4</sub> )	M <sub>2</sub>	CT <sub>5</sub> ,CT <sub>6</sub>	TAD <sub>+y</sub>
f5	A step	Milling (mo <sub>5</sub> )	M <sub>2</sub> ,M <sub>3</sub>	CT <sub>5</sub> ,CT <sub>6</sub>	TAD <sub>+y</sub> , TAD <sub>-z</sub>
f6	Two holes as a replicated feature	Drilling (mo <sub>6</sub> )	M <sub>1</sub> ,M <sub>2</sub> ,M <sub>3</sub>	CT <sub>2</sub>	TAD <sub>+z</sub> , TAD <sub>-z</sub>
f7	Four holes as a replicated feature	Drilling (mo <sub>7</sub> )	M <sub>1</sub> ,M <sub>2</sub> ,M <sub>3</sub>	CT <sub>1</sub>	TAD <sub>+z</sub> , TAD <sub>-z</sub>
f8	A slot	Milling (mo <sub>8</sub> )	M <sub>2</sub> ,M <sub>3</sub>	CT <sub>5</sub> ,CT <sub>6</sub>	TAD <sub>+x</sub>
f9	Two holes as a replicated feature	Drilling (mo <sub>9</sub> )	M <sub>1</sub> ,M <sub>2</sub> ,M <sub>3</sub>	CT <sub>1</sub>	TAD <sub>-z</sub>
f10	A slot	Milling (mo <sub>10</sub> )	M <sub>2</sub> ,M <sub>3</sub>	CT <sub>5</sub> ,CT <sub>6</sub>	TAD <sub>-y</sub>
f11	A slot	Milling (mo <sub>11</sub> )	M <sub>2</sub> ,M <sub>3</sub>	CT <sub>5</sub> ,CT <sub>7</sub>	TAD <sub>-y</sub>
f12	Two holes as a replicated feature	Drilling (mo <sub>12</sub> )	M <sub>1</sub> ,M <sub>2</sub> ,M <sub>3</sub>	CT <sub>1</sub>	TAD <sub>+z</sub> , TAD <sub>-z</sub>
f13	A step	Milling (mo <sub>13</sub> )	M <sub>2</sub> ,M <sub>3</sub>	CT <sub>5</sub> ,CT <sub>6</sub>	TAD <sub>-x</sub> , TAD <sub>-y</sub>
f14	Two holes a replicated feature	Drilling (mo <sub>14</sub> )	M <sub>1</sub> ,M <sub>2</sub> ,M <sub>3</sub>	CT <sub>1</sub>	TAD <sub>-y</sub>

**Table 4.** Precedence relationships for prismatic part 2.

Manufacturing Interactions	Precedence Relationships	Type of Precedence Constraints
Tool-feature interaction	mo <sub>1</sub> must be executed before mo <sub>2</sub>	Hard constraints
Feature-datum interactions	mo <sub>6</sub> must be executed before mo <sub>7</sub> mo <sub>10</sub> must be executed before mo <sub>11</sub> mo <sub>13</sub> must be executed before mo <sub>14</sub>	
Thin-wall interactions	mo <sub>9</sub> must be executed before mo <sub>8</sub> mo <sub>12</sub> must be executed before mo <sub>10</sub>	
Material removal interactions	mo <sub>8</sub> must be executed before mo <sub>9</sub> mo <sub>10</sub> must be executed before mo <sub>12</sub> mo <sub>13</sub> must be executed before mo <sub>14</sub> mo <sub>3</sub> must be executed before mo <sub>4</sub>	Soft Constraints

**Table 5.** Information about features and resource candidates for prismatic part 3.

Feat.	Feature Definition	Machining Operations	Machines	Cutting Tools	TADs
f1	Pocket	Milling (mo <sub>1</sub> )	M <sub>2</sub> ,M <sub>3</sub>	CT <sub>5</sub> ,CT <sub>6</sub> ,CT <sub>7</sub>	TAD <sub>-z</sub>
f2	Blind hole	Drilling (mo <sub>2</sub> )	M <sub>1</sub> ,M <sub>2</sub> ,M <sub>3</sub>	CT <sub>2</sub> ,CT <sub>3</sub> ,CT <sub>4</sub>	TAD <sub>-z</sub>
f3	Through hole	Drilling (mo <sub>3</sub> )	M <sub>1</sub> ,M <sub>2</sub> ,M <sub>3</sub>	CT <sub>2</sub> ,CT <sub>3</sub> ,CT <sub>4</sub>	TAD <sub>+z</sub> , TAD <sub>-z</sub>
		Reaming (mo <sub>4</sub> )	M <sub>1</sub> ,M <sub>2</sub> ,M <sub>3</sub>	CT <sub>9</sub>	TAD <sub>+z</sub> , TAD <sub>-z</sub>
		Boring (mo <sub>5</sub> )	M <sub>3</sub> ,M <sub>4</sub>	CT <sub>8</sub>	TAD <sub>+z</sub> , TAD <sub>-z</sub>
f4	Four Through holes	Drilling (mo <sub>6</sub> )	M <sub>1</sub> ,M <sub>2</sub> ,M <sub>3</sub>	CT <sub>2</sub>	TAD <sub>+z</sub> , TAD <sub>-z</sub>
f5	Slot	Milling (mo <sub>7</sub> )	M <sub>2</sub> ,M <sub>3</sub>	CT <sub>5</sub> ,CT <sub>6</sub>	TAD <sub>-y</sub> , TAD <sub>-z</sub>
f6	Slot	Milling (mo <sub>8</sub> )	M <sub>2</sub> ,M <sub>3</sub>	CT <sub>5</sub> ,CT <sub>6</sub>	TAD <sub>+y</sub> , TAD <sub>-z</sub>
f7	Two Blind holes	Drilling (mo <sub>9</sub> )	M <sub>1</sub> ,M <sub>2</sub> ,M <sub>3</sub>	CT <sub>1</sub>	TAD <sub>+y</sub>
f8	Two slots	Milling (mo <sub>10</sub> )	M <sub>2</sub> ,M <sub>3</sub>	CT <sub>6</sub> ,CT <sub>7</sub> ,CT <sub>10</sub>	TAD <sub>+y</sub> , TAD <sub>-y</sub> , TAD <sub>+z</sub>
f9	Pocket	Milling (mo <sub>11</sub> )	M <sub>2</sub> ,M <sub>3</sub>	CT <sub>5</sub> ,CT <sub>6</sub> ,CT <sub>7</sub>	TAD <sub>-x</sub>
f10	Pocket	Milling (mo <sub>12</sub> )	M <sub>2</sub> ,M <sub>3</sub>	CT <sub>5</sub> ,CT <sub>6</sub> ,CT <sub>7</sub>	TAD <sub>+x</sub>
f11	Plane	Rough milling (mo <sub>13</sub> )	M <sub>2</sub> ,M <sub>3</sub>	CT <sub>5</sub> ,CT <sub>6</sub> ,CT <sub>7</sub>	TAD <sub>+x</sub> , TAD <sub>-x</sub> , TAD <sub>+y</sub> ,
		Finish milling (mo <sub>14</sub> )	M <sub>2</sub> ,M <sub>3</sub>	CT <sub>5</sub> ,CT <sub>6</sub> ,CT <sub>7</sub>	TAD <sub>-y</sub> , TAD <sub>+z</sub>

**Table 6.** Available machining resources for three prismatic parts.

Prismatic Part 1		Prismatic Part 2		Prismatic Part 3	
Machines					
M <sub>1</sub>	Drilling press	Drilling press	Drilling press	Drilling press	Drilling press
M <sub>2</sub>	3-axis vertical milling machine	Milling machine	3-axis vertical milling machine	3-axis vertical milling machine	3-axis vertical milling machine
M <sub>3</sub>	CNC 3-axis vertical milling machine	3-axis vertical milling machine	CNC 3-axis vertical milling machine	CNC 3-axis vertical milling machine	CNC 3-axis vertical milling machine
M <sub>4</sub>	Boring machine	-	Boring machine	Boring machine	Boring machine
Tools					
CT <sub>1</sub>	Drill	Drill 1	Drill 1	Drill 1	Drill 1
CT <sub>2</sub>	Drill	Drill 2	Drill 2	Drill 2	Drill 2
CT <sub>3</sub>	Drill	Reamer	Reamer	Reamer	Drill 3
CT <sub>4</sub>	Drill	Boring tool	Boring tool	Boring tool	Drill 4
CT <sub>5</sub>	Tapping tool	Milling cutter 1	Milling cutter 1	Milling cutter 1	Milling cutter 1
CT <sub>6</sub>	Mill	Milling cutter 2	Milling cutter 2	Milling cutter 2	Milling cutter 2
CT <sub>7</sub>	Mill	Slot cutter	Slot cutter	Slot cutter	Milling cutter 3
CT <sub>8</sub>	Mill	Chamfer tool	Chamfer tool	Chamfer tool	Boring tool
CT <sub>9</sub>	Reamer	-	-	-	Reamer
CT <sub>10</sub>	Boring tool	-	-	-	Slot cutter

**Table 7.** Cost information for the considered prismatic parts.

Resources	Prismatic part 1									
Machines	$M_1$	$M_2$	$M_3$	$M_4$						
Machine cost index (MCI)	10	40	100	60						
Cutting tools	$CT_1$	$CT_2$	$CT_3$	$CT_4$	$CT_5$	$CT_6$	$CT_7$	$CT_8$	$CT_9$	$CT_{10}$
Cutting tool cost index (CTCI)	7	5	3	8	7	10	15	30	15	20
Change indices	MCC	SCC	CTCC							
Cost	160	100	20							
Resources	Prismatic part 2									
Machines	$M_1$	$M_2$	$M_3$							
Machine cost index (MCI)	10	35	60							
Cutting tools	$CT_1$	$CT_2$	$CT_3$	$CT_4$	$CT_5$	$CT_6$	$CT_7$	$CT_8$		
Cutting tool cost index (CTCI)	3	3	8	15	10	15	10	10		
Change indices	MCC	SCC	CTCC							
Cost	300	120	15							
Resources	Prismatic part 3									
Machines	$M_1$	$M_2$	$M_3$	$M_4$						
Machine cost index (MCI)	10	40	100	60						
Cutting tools	$CT_1$	$CT_2$	$CT_3$	$CT_4$	$CT_5$	$CT_6$	$CT_7$	$CT_8$	$CT_9$	$CT_{10}$
Cutting tool cost index (CTCI)	7	5	3	8	7	10	15	30	20	15
Change indices	MCC	SCC	CTCC							
Cost	160	120	20							

**Table 8.** Optimal process plans for prismatic part 1 and three optimization conditions.

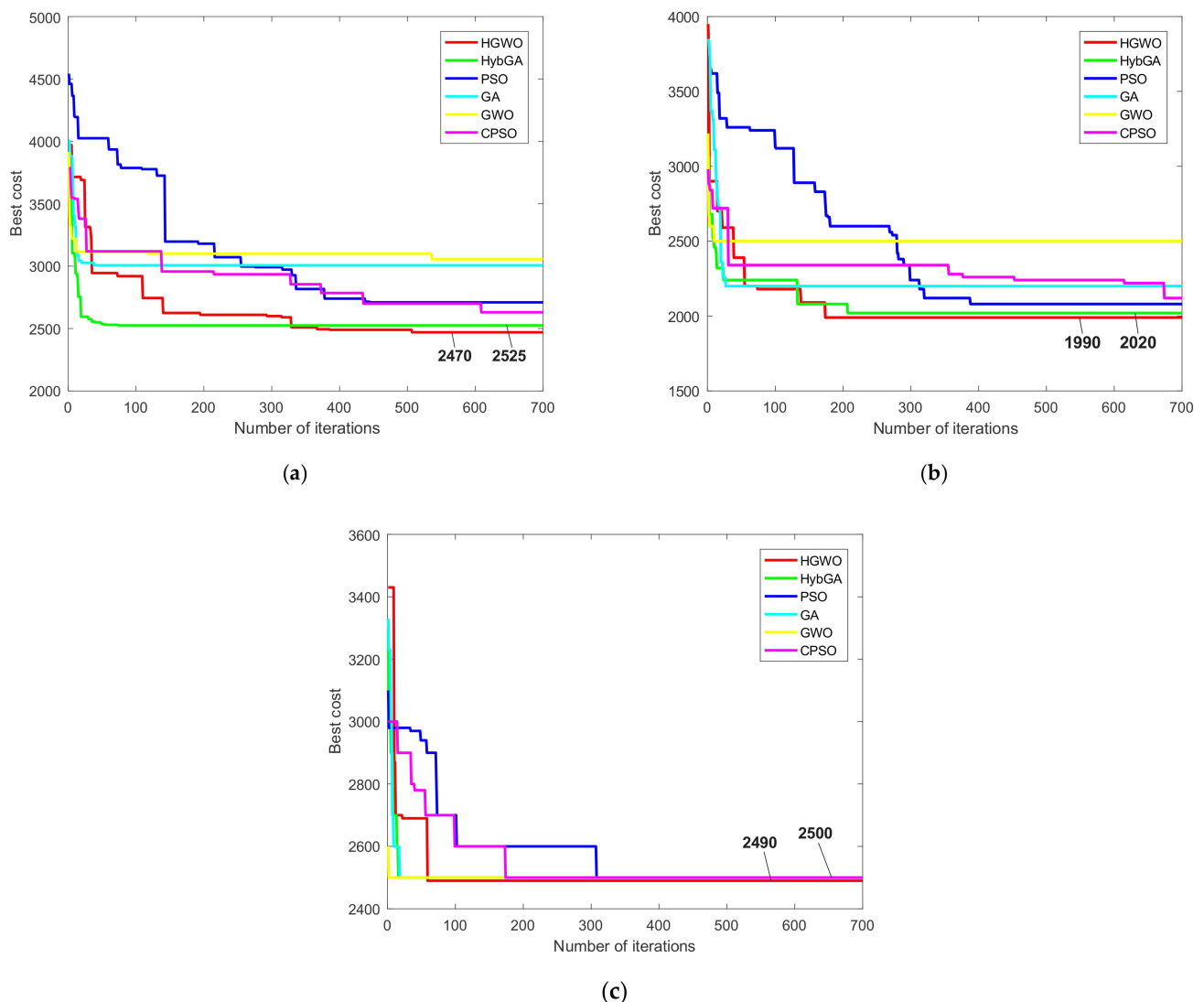
	Condition 1																			
Machining operation	1	5	3	2	18	4	17	11	6	12	13	7	8	9	19	20	14	10	15	16
Machine	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	4	4	4	1	1
Cutting tool	6	6	6	6	6	2	7	7	7	4	9	7	4	9	9	10	10	10	1	5
TAD	+z	+x	+x	-z	-z	-z	-z	-z	-z	-z	-z	-a	-a	-a	+z	+z	-z	-a	-z	-z
TMC = 800, TMCC = 320, NMC = 2, TCTC = 250, TCTCC = 200, NCTC = 10, TSC = 900, NSC = 9, TWMC = <b>2470</b>																				
	Condition 2																			
Machining operation	1	18	11	2	6	12	13	17	5	3	7	8	9	19	10	20	14	4	15	16
Machine	2	2	2	2	2	2	2	2	2	2	2	2	2	2	4	4	4	1	1	1
Cutting tool	8	7	8	8	8	3	9	8	7	8	8	3	9	9	10	10	10	2	1	5
TAD	+z	-z	-z	-z	-z	-z	-z	-z	-z	+x	-a	-a	-a	+z	-a	+z	-z	-z	-z	-z
TMC = 770, TMCC = 320, NMC = 2, TSC = 900, NSC = 9, TWMC = <b>1990</b>																				
	Condition 3																			
Machining operation	1	3	18	17	5	11	2	6	12	13	14	19	20	7	8	9	10	4	15	16
Machine	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	1	1	1
Cutting tool	7	7	7	7	7	7	6	7	4	9	10	9	10	7	4	9	10	2	1	5
TAD	+z	+x	-z	-z	-z	-z	-z	-z	-z	-z	-z	+z	+z	-a	-a	-a	-a	-z	-z	-z
TMC = 1730, TMCC = 160, NMC = 1, TSC = 600, NSC = 6, TWMC = <b>2490</b>																				

**Table 9.** Comparative results of HECSA with other algorithms for three prismatic parts.

Algorithm	Condition 1			Condition 2			Condition 3		
	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min
Prismatic part 1									
<b>HGWO</b>	<b>2608.9</b>	<b>2805</b>	<b>2470</b>	<b>2092</b>	<b>2240</b>	<b>1990</b>	<b>2499.5</b>	<b>2500</b>	<b>2490</b>
HGASA (Li et al. [5])	2546	2585	2527	2120	2120	2120	-	2600	2590
PSO (Guo et al. [42])	2680.5	-	2535	-	-	-	-	-	-
HybGA (Huang et al. [8])	-	-	2527	-	-	2120	-	-	2590
HBMO (Wen et al. [24])	2543.5	2557	2525	2098	2120	2090	2592.4	2600	2590
ACO (Wang et al. [15])	2456.1	2527	2435	2115.4	2380	2090	2600	2740	2580
TSACO (Wang et al. [16])	2552.4	2557	2525	2120.5	2380	2090	2600.8	2740	2590
cPSO (Petrović et al. [20])	2629	2687	2520	2100	2220	2020	2515	2600	2500
mACO (Hu et al. [17])	2666	-	2530	2115	-	2090	-	-	-
ESGA (Su et al. [11])	2539.1	2562	2530	-	-	2090	-	-	2590
IWD (Gao et al. [32])	2553.5	2554	2527	2123	2380	2090	2615.3	2740	2590
Prismatic part 2									
<b>HGWO</b>	<b>1344.3</b>	<b>1363</b>	<b>1328</b>	<b>1193</b>	<b>1290</b>	<b>1170</b>			
HybGA (Huang et al. [8])	1370	1583	1328	1224	1410	1170			
TS (Liu et al. [14])	1342	1378	1328	1194	1298	1170			
SA (Liu et al. [14])	1373.5	1518	1328	1217	1345	1170			
GA (Liu et al. [14])	1611	1778	1478	1482	1650	1410			
ACO (Liu et al. [14])	1329.5	1343	1328	1170	1170	1170			
cPSO (Petrović et al. [20])	1654	1748	1493	1428	1530	1270			
Prismatic part 3									
<b>HGWO</b>	<b>1918.8</b>	<b>2003</b>	<b>1815</b>						
HGASA (Li et al. [5])	1927.1	2079	1823						
PSO (Guo et al. [42])	2408.5	2280	2593						
GA (Kafashi [6])	-	-	2057						
HybGA (Huang et al. [8])	1828	1895	1749						
cPSO (Petrović et al. [20])	1871.5	1934	1775						

Furthermore, in Figure 11 we presented convergence curves for prismatic part 1. As can be seen, convergence of six different metaheuristic approaches is compared considering three optimization conditions. Convergence of the proposed HGWO is compared with the convergence of three traditional algorithms, PSO [42], GA [14] and GWO, as well as the hybrid and the modified metaheuristics, HybGA [8] and cPSO [20]. From the information in Figure 11a, considering the first condition, GWO, GA and HybGA showed much faster convergence compared to other algorithms. However, the GWO and GA converged towards local optima, whereas the HybGA succeeded in finding much better solution. Although the HGWO showed much slower convergence than the HybGA for example, it converged towards the minimal value of 2470. This value may not be considered the global optimum, since the slightly better result in the literature has been reported so far. It could be safely argued that the HGWO converged towards the near-optimal solution. From Figure 11b and the second condition of the first study, it can be noticed that the convergence of the HGWO is similar to the one of the HybGA. After approximately 200 iterations, these approaches converged towards the optimal and near-optimal result. PSO, like CPSO, converged towards the good solutions, but with much slower convergence. GA and GWO got trapped in the local optima in the early stages. Lastly, the convergence curves for the third condition are given in Figure 11c. It can be seen that all algorithms in this analysis converged towards best solutions. The reason lies in the fact that the third condition excludes certain solution candidates as well as cost assigned with them. HGWO showed slightly slower convergence than GA, GWO and HybGA, but, on the other side, succeeded in finding the new global optimum of 2490 cost units for TWMC. This result showed the effectiveness of the HGWO for the third condition.





**Figure 11.** Convergence curves for the prismatic part 1: (a) 1st condition; (b) 2nd condition; (c) 3rd condition.

Best process plans for two conditions for prismatic part 2 are given in Table 10. After adopting similar parameters as in the previous study, we compared the results with six different approaches from the literature [8,14,20]. The comparative results are shown in Table 9. The proposed HGWO performed well by achieving optimal TWMC of 1328 cost units eight times under the first condition. The mean value was slightly greater compared to the same result achieved by TS and ACO, but much better compared to other algorithms. Similar conclusion can be made for the maximal TWMC where the HGWO showed better result compared to the HybGA, TS, SA, GA and CPSO. Under the second condition, the HGWO found the minimal TWMC of 1170 cost units which appeared 19 times in 20 runs. Besides the ACO approach, the HGWO achieved the best maximal and the best mean results compared to all other algorithms.

Figure 12 presents the convergence curves for two optimization conditions of the second experimental study concerning prismatic part 2. The same six metaheuristics are considered in the convergence analysis. As shown in Figure 12a for the first condition, the HGWO and the HybGA show similar performances, with the HGWO showing slower convergence compared to the hybrid metaheuristic algorithm. Both approaches converged towards the global TWMC of 1328 cost units. GA and GWO got trapped in the local optima in early stages, whereas the CPSO and the PSO showed progress in converging towards good solutions but not as good as the ones found by the HGWO and the HybGA. Similar

performances can be seen for the second condition shown in Figure 12b. Since the second condition excludes certain alternatives and costs, the HGWO showed faster convergence than for the previous condition. Like the HybGA, HGWO converged towards the global TWMC of 1170 cost units. It took HGWO approximately 150th iterations to converge towards the global optimum compared to the HybGA which was faster to some extent. However, compared to the traditional GA, GWO, PSO, as well as the CPSO, the HGWO showed more superior performance for the second condition.

Table 10. Optimal process plans for prismatic part 2 for two conditions.

Condition 1														
Machining operation	8	5	3	4	13	10	11	14	12	6	1	7	9	2
Machine	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Cutting tool	5	5	5	5	5	5	5	1	1	2	1	1	1	8
TAD	+x	+y	+y	+y	-y	-y	-y	-y	-z	-z	-z	-z	-z	-z
TMC = 490, TMCC = 0, NMC = 0, TCTC = 98, TCTCC = 60, NCTC = 4, TSC = 480, NSC = 4, TWMC = 1328														
Condition 2														
Machining operation	8	1	6	12	7	9	5	3	4	13	14	2	10	11
Machine	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Cutting tool	5	1	2	1	1	1	6	6	6	6	1	8	6	5
TAD	+x	-z	-z	-z	-z	-z	+y	+y	+y	-y	-y	-y	-y	-y
TMC = 490, TMCC = 0, NMC = 0, TSC = 480, NSC = 4, TWMC = 1170														

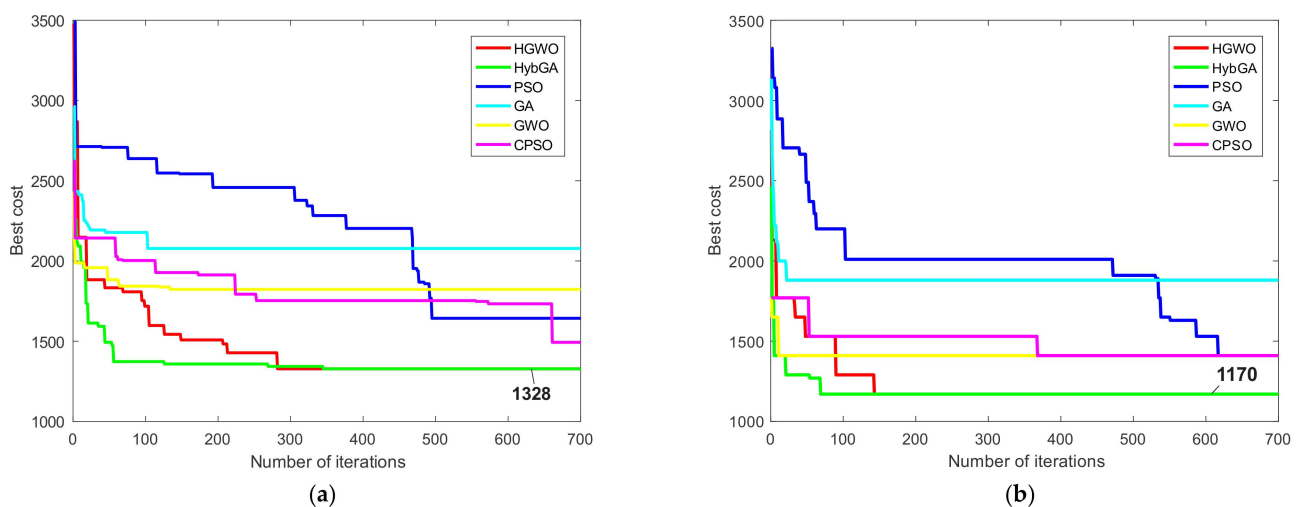


Figure 12. Convergence curves for the prismatic part 2: (a) 1st condition; (b) 2nd condition.

The optimal process plans for prismatic part 3 are obtained and presented in Table 11. The same input parameters are used for this experimental study. Table 9 shows the comparative results for the minimal, the maximal and the mean TWMC. In 20 runs, the HGWO obtained the TWMC below 1900 cost units eight times in total, and below 2000 cost units eleven times. The minimal TWMC of 1815 cost units is better than the minimal values obtained by the GA, PSO, as well as the HGASA. The HGWO outperforms these algorithms in terms of the best maximal and the best mean results.

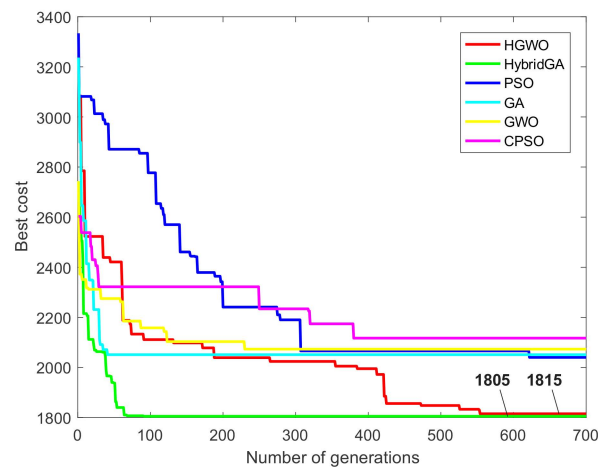
Figure 13 shows the convergence curves for the third experimental study. It can be noticed that the HybGA again shows the best convergence rate and by far outperforms other algorithms in this regard. As far as the HGWO is concerned, it shows much slower convergence compared to the first and the second experimental study. The HGWO obtained the TWMC of 1815 cost units after 500 iterations. Despite the slow convergence rate, the proposed HGWO algorithm achieved near-optimal solution for this prismatic part. Traditional GWO and especially GA also showed good convergence rates but they got trapped

in the local optima. CPSO and PSO performances could not match the performances of the previous algorithms.

**Table 11.** Optimal process plans for prismatic part 3 for two conditions.

Machining operation	11	13	12	14	8	9	10	7	1	3	5	4	2	6
Machine	2	2	2	2	2	2	2	2	2	2	3	3	3	3
Cutting tool	5	5	5	5	5	1	6	6	6	3	8	9	2	2
TAD	−x	+x	+x	+x	+y	+y	+y	−z	−z	−z	−z	−z	−z	−z

TMC = 800, TMCC = 160, NMC = 1, TCTC = 135, TCTCC = 120, NCTC = 6, TSC = 600, NSC = 5, TWMC = **1815**



**Figure 13.** Convergence curve for the prismatic part 3.

## 6. Time Efficiency of the HGWO

According to the obtained experimental results, the HGWO approach showed very good effectiveness and flexibility by finding optimal and near-optimal solutions of the process planning optimization problem. From the perspective of the solutions quality, the comparative results in Table 9 show that the HGWO obtained the near-optimal TWMC value for the first condition, and the new optimal values for the second and the third condition of the first experimental study. In terms of the consistency, the HGWO showed good mean values for all three conditions surpassing the mean results of several algorithms reported in the literature. The optimal TWMC cost values are obtained for both conditions of the second experimental study with much better consistency that is reflected in very good mean results. In the third experimental study, the HGWO approach obtained near-optimal results of TWMC with good consistency where mean results surpass those results obtained by traditional algorithms.

Although the proposed HGWO approach demonstrated flexibility and effectiveness in terms of the solution qualities, there are certain limitations regarding its computational time efficiency. Due to the fact a small number of researchers included computational times in their studies, a detailed comparative analysis could not be performed in this regard. In Table 12 we presented the mean computational times of six metaheuristic algorithms whose convergence curves were shown in Figures 11–13 for three experimental studies, respectively. They are expressed in seconds required to complete a single run, and the results in Table 12 are the mean computational times of 20 runs. From the perspective of the HGWO efficiency, better mean computational times are achieved against traditional PSO and modified cPSO approach. On the other side, they exceed the mean computational times of three algorithms, GWO, GA and HybGA. Similar values for the HGWO and the PSO are obtained for the prismatic part 2. Although some authors suggest less amount of time to be reasonable for large-sized NP-hard problems [50], we may argue that approximately 1 to 1.5 min of the running time is a reasonably well amount of time needed to solve different

instances of an NP-hard problem such as the PPO. GA and GWO as traditional algorithms obviously show the lowest computational times in all studies, since the modified and hybrid algorithms require more time to perform additional operations. One of the reasons for lower efficiency of the HGWO compared to the HybGA is the difference in input parameters, where HGWO considered larger probabilities for crossover and mutation strategies, as well as larger tournament size, compared to HybGA. According to the convergence curves in Figures 11–13, the convergence rate of the HGWO is slower in the first stages of the search process, especially for the first conditions of prismatic parts 1 and 2, as well as prismatic part 3. For example, curve in Figure 11a shows slower rate of the HGWO in the first 300 iterations, meaning that the probabilities of crossover and mutation are much higher than in the latter iterations, which leads to the increase in computational time of the HGWO.

**Table 12.** Mean computational times (s) of five different algorithms for three experimental studies.

Algorithm	Prismatic Part 1			Prismatic Part 2		Prismatic Part 3
	Cond. 1	Cond. 2	Cond. 3	Cond. 1	Cond. 2	Cond. 1
HGWO	67.9351	67.3275	89.2144	84.5813	86.2835	62.8624
HybGA	42.1551	42.9072	47.7353	50.0639	50.6163	38.1999
PSO	70.3245	70.0847	90.9951	83.9863	84.4812	64.7519
GA	38.9688	38.2773	44.7899	43.3343	44.1306	34.0308
GWO	43.6161	42.2103	60.7819	48.9564	49.5188	38.3231
cPSO	75.7522	78.1318	97.5841	93.0888	93.8023	72.2037

Time efficiency of the HGWO can also be evaluated by the function execution times that are called in the programming environment. Since the MATLAB was used to implement the HGWO, we provide the execution times of several functions coded in MATLAB which are presented in Table 13. Apart from the functions that consider calling six different algorithms considered in experimental studies, fitness evaluation and constraint handling functions are also presented. Two time values, total and self-time are shown. Total time represents the total time spent in a function including all child functions, while the self-time means the total time spent in a function without any time spent in child functions. All time values in Table 13 are obtained after 20 runs and consider only the first experimental study. It can be noticed that calling the HGWO algorithm requires more time compared to almost all other algorithms except the CPSO. The big difference in total and self-time means that the HGWO requires calling many child functions, including fitness evaluation and constraint handling algorithm.

**Table 13.** MATLAB profile summary report about the execution times of certain MATLAB functions.

Function Name (MATLAB)	Execution Times for the Several Functions Regarding Prismatic Part					
	Condition 1		Condition 2		Condition 3	
	Total Time	Self-Time	Total Time	Self-Time	Total Time	Self-Time
fitness_cost	3229.754	3229.754	3645.48	3645.48	3527.900	3527.900
cpsso_algorithm	2005.366	722.229	2185.712	791.54	2061.431	771.305
hgwo_algorithm	1823.554	583.585	2011.849	651.219	1895.894	638.906
pso_algorithm	1791.113	701.895	2006.323	798.087	1928.071	766.405
constr_handling	1283.064	1283.064	1387.015	1387.015	1353.541	1353.541
gwo_algorithm	1231.354	535.619	1332.102	579.912	1303.338	569.971
hybga_algorithm	971.847	42.707	1085.238	47.323	1015.884	44.017
ga_algorithm	889.644	27.057	989.088	30.224	964.780	29.588

According to the computational times and function execution times, we can argue that the HGWO demonstrated good time efficiency. However, this can also be considered as the main disadvantage of the proposed approach since the comparative results clearly

show that the HGWO falls behind other algorithms. The fact that the constraint handling heuristic, the fitness evaluation, the tournament selection, the crossover, the shift mutation, and the resource mutation represent a separate MATLAB functions that the HGWO has to call in order to perform a single iteration, the time efficiency should be treated with enough attention. With effectiveness and flexibility as the main advantages, and time efficiency as the main disadvantage of the HGWO approach, potential improvements in terms of time efficiency and convergence rate are beyond the scope of this paper and may be a subject for future research studies.

## 7. Conclusions

A hybrid grey wolf optimizer is proposed to solve the NP-hard process planning optimization problem based on precedence constraints. The problem considers operation sequencing task along with the optimal selection of appropriate machines, cutting tools and tool approach directions that define a process plan. Modern knowledge-based representation scheme is used to represent process plans in a population of alternatives and the operation precedence graph approach with adjacency matrices is adopted to deal with precedence relationships among machining features. According to these relationships, precedence constraints are defined. To deal with the precedence constraint in an appropriate manner, we adopted constraint handling heuristic algorithm to deal with hard constraints, as well as both hard and soft constraints. In that sense, the feasibility of process plans was ensured.

To improve the performance of traditional grey wolf optimizer, the strategies of genetic algorithm were adopted. Tournament selection is utilized for selecting fittest individuals and classical two-point crossover strategy is applied on selected individuals to generate wolf offspring. Shift and resource mutation strategies perform minor changes to vector candidates and complete the evolutionary steps of the HGWO approach. To additionally achieve balance between exploration and exploitation inertia weight coefficients from the PSO are added to the mathematical model of the HGWO approach.

For the evaluation of process plans, traditional total weighted machining cost was adopted as an optimization criterion. The detailed mathematical model is presented and the criterion was formed for different conditions covered in three experimental studies which consider three prismatic parts proposed in the literature.

After manually determining input parameters of the proposed HGWO approach, popular results obtained by traditional and modern approaches were used to perform the detailed comparative analysis. Three experimental studies are conducted to test the effectiveness and efficiency of the HGWO approach. The detailed representation of optimal process plans and comparative results including the minimal, the maximal and the mean TWMC for 20 algorithm runs were given. The HGWO approach demonstrated flexibility and effectiveness by solving the process planning optimization problem and finding the optimal and the near-optimal results.

Although effectiveness and flexibility showed as the main advantages of the HGWO, time efficiency of the proposed algorithm showed certain limitations. The HGWO succeeded in finding optimal and near-optimal process plans in a reasonable time period, but the comparative analysis of computational times, as well as function execution times in MATLAB programming environment, confirm good efficiency of the HGWO. However, the HGWO falls behind certain hybrid and traditional algorithms in this regard.

One of the directions for future research may be focused on improving efficiency and convergence of the HGWO, primarily through consideration of different genetic strategies. Additionally, optimization of input parameters has already been reported in some studies and should be taken into account in order to further improve performances of the HGWO. Possible contribution to the area of process planning optimization can be directed towards the optimization of integrated process planning and scheduling. Likewise, the recent advances in the field of Industry 4.0 may consider the implementation of the HGWO as a cloud service within the Smart factory.

**Author Contributions:** Conceptualization, M.M. and M.D.; methodology, M.D., M.M. and R.C.; software D.L., A.A. and L.C.; validation, M.D., M.M. and L.C.; writing—original draft preparation, M.D. and M.M.; writing—review and editing, M.M. and D.L.; supervision R.C. and A.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This paper is part of a research on the projects: “Application of Edge computing and Artificial Intelligence methods in Smart products”, No. 142-451-2312/2021, supported by Provincial Secretariat for Higher Education and Scientific Research of the Autonomous Province of Vojvodina and “Innovative scientific and artistic research from the FTS domain”, No.451-03-68/2020-14/200156, supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data sharing is not applicable to this article.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Xu, X.; Wang, L.; Newman, S.T. Computer-aided process planning—A critical review of recent developments and future trends. *Int. J. Comput. Integr. Manuf.* **2011**, *24*, 518632. [\[CrossRef\]](#)
- Blum, C.; Roli, A. Metaheuristics in combinatorial optimization. *ACM Comput. Surv.* **2003**, *35*, 268–308. [\[CrossRef\]](#)
- Zhang, F.; Zhang, Y.; Nee, A. Using genetic algorithms in process planning for job shop machining. *IEEE Trans. Evol. Comput.* **1997**, *1*, 278–289. [\[CrossRef\]](#)
- Salehi, M.; Bahreininejad, A. Optimization process planning using hybrid genetic algorithm and intelligent search for job shop machining. *J. Intell. Manuf.* **2011**, *22*, 643–652. [\[CrossRef\]](#)
- Li, W.; Ong, S.K.; Nee, A.Y.C. Hybrid genetic algorithm and simulated annealing approach for the optimization of process plans for prismatic parts. *Int. J. Prod. Res.* **2002**, *40*, 1899–1922. [\[CrossRef\]](#)
- Kafashi, S. Integrated setup planning and operation sequencing (ISOS) using genetic algorithm. *Int. J. Adv. Manuf. Technol.* **2011**, *56*, 589–600. [\[CrossRef\]](#)
- Cai, N.; Wang, L.; Feng, H.-Y. GA-based adaptive setup planning toward process planning and scheduling integration. *Int. J. Prod. Res.* **2009**, *47*, 2745–2766. [\[CrossRef\]](#)
- Huang, W.; Hu, Y.; Cai, L. An effective hybrid graph and genetic algorithm approach to process planning optimization for prismatic parts. *Int. J. Adv. Manuf. Technol.* **2012**, *62*, 1219–1232. [\[CrossRef\]](#)
- Li, S.; Liu, Y.; Li, Y.; Landers, R.G.; Tang, L. Process planning optimization for parallel drilling of blind holes using a two phase genetic algorithm. *J. Intell. Manuf.* **2013**, *24*, 791–804. [\[CrossRef\]](#)
- Su, Y.; Chu, X.; Zhang, Z.; Chen, D. Process planning optimization on turning machine tool using a hybrid genetic algorithm with local search approach. *Adv. Mech. Eng.* **2015**, *7*, 1687814015581241. [\[CrossRef\]](#)
- Su, Y.; Chu, X.; Chen, D.; Sun, X. A genetic algorithm for operation sequencing in CAPP using edge selection based encoding strategy. *J. Intell. Manuf.* **2018**, *29*, 313–332. [\[CrossRef\]](#)
- Candan, G.; Yazgan, H.R. Genetic algorithm parameter optimisation using Taguchi method for a flexible manufacturing system scheduling problem. *Int. J. Prod. Res.* **2014**, *53*, 897–915. [\[CrossRef\]](#)
- Luo, Y.; Pan, Y.; Li, C.; Tang, H. A hybrid algorithm combining genetic algorithm and variable neighborhood search for process sequencing optimization of large-size problem. *Int. J. Comput. Integr. Manuf.* **2020**, *33*, 962–981. [\[CrossRef\]](#)
- Liu, X.-J.; Yi, H.; Ni, Z.-H. Application of ant colony optimization algorithm in process planning optimization. *J. Intell. Manuf.* **2013**, *24*, 1–13. [\[CrossRef\]](#)
- Wang, J.; Fan, X.; Wan, S. A Graph-Based Ant Colony Optimization Approach for Process Planning. *Sci. World J.* **2014**, *2014*, 1–11. [\[CrossRef\]](#) [\[PubMed\]](#)
- Wang, J.; Wu, X.; Fan, X. A two-stage ant colony optimization approach based on a directed graph for process planning. *Int. J. Adv. Manuf. Technol.* **2015**, *80*, 839–850. [\[CrossRef\]](#)
- Hu, Q.; Qiao, L.; Peng, G. An ant colony approach to operation sequencing optimization in process planning. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2017**, *231*, 470–489. [\[CrossRef\]](#)
- Li, W.; Wang, L.; Li, X.; Gao, L. Intelligent Optimisation for Integrated Process Planning and Scheduling. In *Multi-Objective Evolutionary Optimisation for Product Design and Manufacturing*; Springer: Singapore, 2011; pp. 305–324.
- Li, X.; Gao, L.; Wen, X. Application of an efficient modified particle swarm optimization algorithm for process planning. *Int. J. Adv. Manuf. Technol.* **2013**, *67*, 1355–1369. [\[CrossRef\]](#)
- Petrović, M.; Mitić, M.; Vuković, N.; Miljković, Z. Chaotic particle swarm optimization algorithm for flexible process planning. *Int. J. Adv. Manuf. Technol.* **2016**, *85*, 2535–2555. [\[CrossRef\]](#)
- Miljković, Z.; Petrović, M. Application of modified multi-objective particle swarm optimisation algorithm for flexible process planning problem. *Int. J. Comput. Integr. Manuf.* **2017**, *30*, 271–291. [\[CrossRef\]](#)

22. Dou, J.; Li, J.; Su, C. A discrete particle swarm optimisation for operation sequencing in CAPP. *Int. J. Prod. Res.* **2018**, *56*, 3795–3814. [[CrossRef](#)]
23. Lian, K.; Zhang, C.; Shao, X.; Gao, L. Optimization of process planning with various flexibilities using an imperialist competitive algorithm. *Int. J. Adv. Manuf. Technol.* **2012**, *59*, 815–828. [[CrossRef](#)]
24. Wen, X.-Y.; Li, X.-Y.; Gao, L.; Sang, H.-Y. Honey bees mating optimization algorithm for process planning problem. *J. Intell. Manuf.* **2014**, *25*, 459–472. [[CrossRef](#)]
25. Lv, S.; Qiao, L. A cross-entropy-based approach for the optimization of flexible process planning. *Int. J. Adv. Manuf. Technol.* **2013**, *68*, 2099–2110. [[CrossRef](#)]
26. Wang, J.; Fan, X.; Zhao, A.; Yang, M. A Hybrid Bat Algorithm for Process Planning Problem. *IFAC-PapersOnLine* **2015**, *48*, 1708–1713. [[CrossRef](#)]
27. Musharavati, F.; Hamouda, A.S.M. Enhanced simulated-annealing-based algorithms and their applications to process planning in reconfigurable manufacturing systems. *Adv. Eng. Softw.* **2012**, *45*, 80–90. [[CrossRef](#)]
28. Mohammadi, G.; Karampourhaghghi, A.; Samaei, F. A multi-objective optimisation model to integrating flexible process planning and scheduling based on hybrid multi-objective simulated annealing. *Int. J. Prod. Res.* **2012**, *50*, 5063–5076. [[CrossRef](#)]
29. Xu, C.; Zhang, S.; Huang, R.; Huang, B.; Li, X. NC process reuse-oriented flexible process planning optimization approach for prismatic parts. *Int. J. Adv. Manuf. Technol.* **2016**, *87*, 329–351. [[CrossRef](#)]
30. Lian, K.; Zhang, C.; Shao, X.; Zeng, Y. A multi-dimensional tabu search algorithm for the optimization of process planning. *Sci. China Ser. E Technol. Sci.* **2011**, *54*, 3211–3219. [[CrossRef](#)]
31. Falih, A.; Shammari, A.Z.M. Hybrid constrained permutation algorithm and genetic algorithm for process planning problem. *J. Intell. Manuf.* **2020**, *31*, 1079–1099. [[CrossRef](#)]
32. Gao, B.; Hu, X.; Peng, Z.; Song, Y. Application of intelligent water drop algorithm in process planning optimization. *Int. J. Adv. Manuf. Technol.* **2020**, *106*, 5199–5211. [[CrossRef](#)]
33. Kizys, R.; Juan, A.A.; Sawik, B.; Calvet, L. A Biased-Randomized Iterated Local Search Algorithm for Rich Portfolio Optimization. *Appl. Sci.* **2019**, *9*, 3509. [[CrossRef](#)]
34. Sawik, T.; Sawik, B. A rough cut cybersecurity investment using portfolio of security controls with maximum cybersecurity value. *Int. J. Prod. Res.* **2021**, 1–17. [[CrossRef](#)]
35. Milosevic, M.; Đurđev, M.; Lukić, D.; Antić, A.; Ungureanu, N. Intelligent Process Planning for Smart Factory and Smart Manufacturing. In Proceedings of the 5th International Conference on the Industry 4.0 Model for Advanced Manufacturing, Belgrade, Serbia, 1–4 June 2020; Springer: Cham, Switzerland, 2020; pp. 205–214. [[CrossRef](#)]
36. Djurdjev, M.; Cep, R.; Lukic, D.; Antic, A.; Popovic, B.; Milosevic, M. A Genetic Crow Search Algorithm for Optimization of Operation Sequencing in Process Planning. *Appl. Sci.* **2021**, *11*, 1981. [[CrossRef](#)]
37. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
38. Zhang, Y.; Luo, X.; Zhang, B.; Zhang, S. Semantic approach to the automatic recognition of machining features. *Int. J. Adv. Manuf. Technol.* **2017**, *89*, 417–437. [[CrossRef](#)]
39. Li, W.D.; Ong, S.K.; Nee, A.Y.C. *Integrated and Collaborative Product Development Environment—Technologies and Implementations*; Series on Manufacturing Systems and Technology; 2006; Volume 2. [[CrossRef](#)]
40. Faheem, W.; Hayes, C.; Castano, J.; Gaines, D. In What is manufacturing interaction? In Proceedings of the DETC'98, ASME Design Engineering Technical Conferences, Atlanta, GA, USA, 13–16 September 1998.
41. Dou, J.; Zhao, X.; Su, C. An Improved Genetic Algorithm for Optimization of Operation Sequencing. In Proceedings of the 2018 IEEE International Conference on Mechatronics and Automation (ICMA), Changchun, China, 5–8 August 2018; pp. 695–700.
42. Guo, Y.W.; Mileham, A.R.; Owen, G.W.; Li, W.D. Operation sequencing optimization using a particle swarm optimization approach. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2006**, *220*, 1945–1958. [[CrossRef](#)]
43. Ahmed, R.; Nazir, A.; Mahadzir, S.; Shorfuzzaman, M.; Islam, J. Niching Grey Wolf Optimizer for Multimodal Optimization Problems. *Appl. Sci.* **2021**, *11*, 4795. [[CrossRef](#)]
44. Yue, Z.; Zhang, S.; Xiao, W. A Novel Hybrid Algorithm Based on Grey Wolf Optimizer and Fireworks Algorithm. *Sensors* **2020**, *20*, 2147. [[CrossRef](#)]
45. Wang, Y.; Wang, W. Quantum-Inspired Differential Evolution with Grey Wolf Optimizer for 0-1 Knapsack Problem. *Mathematics* **2021**, *9*, 1233. [[CrossRef](#)]
46. Martin, B.; Marot, J.; Bourennane, S. Improved Discrete Grey Wolf Optimizer. In Proceedings of the 2018 26th European Signal Processing Conference (EUSIPCO), Rome, Italy, 3–7 September 2018; pp. 494–498.
47. Jiang, T.; Zhang, C. Application of Grey Wolf Optimization for Solving Combinatorial Problems: Job Shop and Flexible Job Shop Scheduling Cases. *IEEE Access* **2018**, *6*, 26231–26240. [[CrossRef](#)]
48. Qin, H.; Fan, P.; Tang, H.; Huang, P.; Fang, B.; Pan, S. An effective hybrid discrete grey wolf optimizer for the casting production scheduling problem with multi-objective and multi-constraint. *Comput. Ind. Eng.* **2019**, *128*, 458–476. [[CrossRef](#)]
49. Premkumar, M.; Jangir, P.; Kumar, B.S.; Alqudah, M.A.; Nisar, K.S. Multi-Objective Grey Wolf Optimization Algorithm for Solving Real-World BLDC Motor Design Problem. *Comput. Mater. Contin.* **2022**, *70*, 2435–2452. [[CrossRef](#)]
50. Zhang, H.; Buchmeister, B.; Li, X.; Ojstersek, R. Advanced Metaheuristic Method for Decision-Making in a Dynamic Job Shop Scheduling Environment. *Mathematics* **2021**, *9*, 909. [[CrossRef](#)]