Journal of Optimization in Industrial Engineering
Vol.13, Issue 1, Winter & Spring 2020, 123-130
DOI:10.22094/JOIE.2019.387.0

Research Article

Operation Sequencing Optimization in CAPP Using Hybrid Teaching-Learning Based Optimization(HTLBO)

Hassan Haleh[a], Azam Sadati[b,*], Naser Hajisharifi[c]

[a]*Golpayegan University of Technology, Golpayegan, Iran*
[b]*Islamic Azad University. Khomein Branch, Khomein, Iran*
[c]*Department of Mathematics, Collage of Basic Sciences, Islamic azad University, Khomein Branch, Khomein, Iran*
Received 18 October 2014; Revised 09 July 2019; Accepted 16 July 2019

Abstract

Computer-aided process planning (CAPP) is an essential component in linking computer-aided design (CAD) and computer-aided manufacturing (CAM). Operation sequencing in CAPP is an essential activity. Each sequence of production operations which is produced in a process plan cannot be the best possible sequence every time in a changing production environment. As the complexity of the product increases, the number of feasible sequences increase exponentially, consequently the best sequence is to be chosen. This paper aims at presenting the application of a newly developed meta-heuristic called the hybrid teaching–learning-based optimization (HTLBO) as a global search technique for the quick identification of the optimal sequence of operations with consideration of various feasibility constraints. To do so, three case studies have been conducted to evaluate the performance of the proposed algorithm and a comparison between the proposed algorithm and the previous searches from the literature has been made. The results show that HTLBO performs well in operation sequencing problem.

***Keywords*:** Teaching–learning-based optimization; Computer-aided process planning (CAPP); Operation sequence; Hamilton path

1. Introduction

Computer aided process planning (CAPP) is applying computer in creation and development of technical plans needed to produce a part. This application is important in terms of linking of CAD and CAM. In general, CAPP has two approaches: variant and generative. The first approach uses a standard plan or similar part plan to prepare a process plan (PP). The PP is stored for the main components in the computer and used in the design of next components. The original part is a combination of all the shapes that may be present in the desired parts. This approach uses part classification techniques which are similar to group technology, to determine the shape of the parts and matches them with the corresponding shapes in the original part. In the second approach, a PP is generated based on the analysis of the information in the manufacturing databases and decision rules. The PP invoIes two main tasks: 1) selecting the appropriate operations for machining based on the form–feature geometry and its technological requirements, 2) operation sequencing (OS) that determines the order in which a set of selected operations are performed, taking into account the precedence constraints created by both parts and operations (Weil et al., 1982). Selection of the optimal operations sequence is an essential activity in CAPP.

Although, there is a lot of research on CAPP systems in the literature, one of the few issues that has been addressed is to determine the optimal sequence.

A variety of classical optimization techniques such as branch and bound method (Koulmas, 1993), linear programming (Lin and Wang, 1993) and dynamic programming (Halevi and Weill, 1988) have been studid. Since the OS problem includes various precedence constraints, formulating and solving it based on classical techniques alone is dificult. Recently, most studies considered meta- heuristics for solving OS problems. Turkay and Huseyin (1999), Bhaskara Reddy et al. (1999) and Li et al. (2005) used genetic algorithm to generate the optimal OS. Li et al. (2004) proposed a tabu search approach for optimization of PPs. Guo et al. (2006) considered particle swarm optimization (PSO) for OS problem. Krishna et al. (2006) studied ant colony optimization (ACO). Salehi and Tavakkoli-moghaddam (2009) proposed a genetic algorithm to OS problem which optimizes selection of the machine, cutting tools and tool access directions. Nallakumarasamy et al. (2011) used simulated annealing technique (SAT) to generate the feasible OS based on the precedence cost matrix and reward/penalty matrix. Hu et al. (2017) developed a novel modified ACO algorithm. They proposed the mathematical model that considered total weighted production cost or weighted resource transformation time. Dou et al. (2018) proposed a novel discrete particle swarm optimization (DPSO) to solve the OS problem in CAPP.

*Corresponding author Email address: industrial7781@yahoo.com

2. Process Planning Problem

CAPP in this paper consists of the following steps : 1) a part drawing , which consins details of geometric specifications and technological information such as tolerance and surface finish requirements. By interpreting the part drawing, one can identify the machining operations, machine tools, cutting tools and cutting parameters required to produce the part. 2) an operation precedence graph (OPG). 3) either a relative precedence cost matrix (PCM) or a reward/penalty matrix (RPM) 4) Operation sequencing , which is the sequence of operations represented as a Hamiltonian path that visits all of the operatins once and only once with the lest cost. (Nallakumarasamy et al. ,2011). steps 2 - 4 are discussed in detail in the subsequent sections.

2.1. Operation precedence graph

Considering manufacturing constraints to identify the feasible sequences , an OPG is obtained based on the precedence relationship among the machining operations. each directed edge between any two operations in OPG is labeled depending on the type of constraints between them (such as location constraint, non-destructive constraints, datum-holding or geometric tolerance constraint and accessibility constraint). an edge with no direction between any two operations shows that either operation can precede another operation.

2.2. Precedence cost matrix generation

A PCM is produced for any two operations based on the relative costs associated with the number of tasks that must be performed in each feature category, such as changing the machining parameter, changing the tool and changing the setting and also the type of constraints which one operation has with others. Numerical values are allocated on the basis of complexity to different costs.

2.3. Reward/penalty matrix

The RPM for each set-up is automatically prepared based on a reward (negative penalty) / penalty generator according to the selected rules that are defined in the system. The authorized rules in the preparation of RPM are as follows: if precedence of operations such as k and l complies with the rules of the sequence, it will be rewarded according to the degree of satisfaction of the rules. Otherwise, a positive penalty will be given.

2.4. Operation sequencing

OS determines the order in which a set of selected operations is performed, taking into account the precedence constraints created by both parts and operations.

An OS problem in CAPP is a combination of varient choices (choice in the operatins or machine selection) and constraints. For this reason, PP is a combinatorial problem (Korde et.al, 1992).

3. Mathematical Model

Considered graph G(V ,A) and arc cost c_{kl} for each arc $(k,l)\epsilon A$, a Hamiltonian path is a path that visits all the vertexes once only once. Matemathical model to find shortest Hamiltonian path from node b to node e is the following (Castro de Andrade, 2016):

Notations:

N Number of operations

k,l Operation indices $(k,l,=1,2,\dots,N)$

x_{kl} Equals 1, if the (k,l) belong to the path

; and 0, otherwise

c_{kl} Cost of arc (k,l)

U_k representing the sequence the node k is being visited.

According to the above-mentioned notations, the proposed problem is formulated as follows:

$$Min\ f=\sum_{k=1}^{N}\sum_{l=1}^{N}c_{kl}x_{kl} \quad (1)$$

$$\sum_{l=1}^{N}x_{kl}-\sum_{l=1}^{N}x_{lk}=\begin{cases}1 & if\ k=b\\ -1 & if\ k=e\\ 0 & else\end{cases} \quad \forall k \quad (2)$$

$$\sum_{l=1}^{N}x_{kl}\leq 1 \quad \forall k \quad (3)$$

$$U_k-U_l+Nx_{kl}\leq(N-1)x_{kl} \quad 2\leq k\neq l \leq N \quad (4)$$

$$x_{kl}=0\ or\ 1; \quad \forall k,l \quad (5)$$

$$U_k\geq 0 \quad \forall k \quad (6)$$

This model finds a minimum-cost path between node b and node e by objective function (1). Constraints (2) guarantee that one arc is entering and leaving each node $(k\neq b,e)$. Constraints (3) ensure that the outgoing degree of each node is at most one. Constraints (4) avoid subtours (subtour elimination constraints). Constraints (5) and (6) define the type of decision variable.

In this paper, nodes b and e are determined according to operation precedence graph and PCM (or RPM), and parameter c_{kl} is determined according to PCM (or RPM). This problem is solved with Lingo 8.0 and results are shown in Tables 6-9, in which the results of proposed algorithm for the same problems are presented as well.

Because Hamiltonian path is NP-hard (Keshavarz-Kohjerdi and Bagheri, 2017; Hosseini, 2018), a meta-heuristics, namely teaching–learning-based optimization (TLBO), is proposed to find the minimum-cost path.

4. TLBO Algorithm

TLBO, proposed by Roa et al. (2011) is based on learning a group of learners of a teacher in a class and this group is considered as the population. The teacher is the best learner in each population. The learning process includes the teacher stage, that the learners learn from the teacher, and the learner stage, that the learners increase their knowledge through interaction with each other, as follows:

4.1. Teacher phase

In this stage, the learners' level of knowledge in iteration *t* ($x_{i,t}$) is transferred by using DM_t (i.e., difference between the teacher ($x_{T,t}$) and mean result of learners (M_t)). Updated learner ($x_{i,t}^{new}$) is considered by:

$$x_{i,t}^{new} = x_{i,t} + DM_t \quad (7)$$

Where $DM_t = r_t(x_{T,t} - T_F M_t)$ (8)

where T_F is the teaching factor and its value can be either 1 or 2 and r_t is a random number in the range [0, 1]. The detailed implementation for our discrete problem is given below.

4.1.1. Selecting the teacher

In each iteration, the teacher is considered as the best learner. In this paper, teacher is the learner that has lowest objective function.

4.1.2. Mean Result of Learners, M_t

To obtain the mean result of learners, the mean of the population is calculated column-wise.

4.1.3. DM_t

The difference between the teacher and the mean result at iteration *t*, is computed using Eq. 8.

4.1.4. Update the learners

Each learner in population is updated by using Eq.7. If the updated number is negative convert it to a positive and if the obtained number is greater than 1, subtract it from 1 and put it instead.

4.1.5. Acceptance Updated Learner

If $f(x_{i,t}^{new}) < f(x_{i,t})$, then replyce $x_{i,t}$ by $x_{i,t}^{new}$ else replyce $x_{i,t}$ by 2-OPT ($x_{i,t}^{new}$).

4.2. Learner Stage

the following steps in iteration *t* are carried out:

Step 1. For learner $x_{i,t}$, randomly select another learner $x_{j,t}$ $(i \neq j)$.

Step 2. Update learner $x_{i,t}$ by using Eq.9 or Eq.10.

If $f(x_{i,t}) < f(x_{j,t})$

$$x_{i,t}^{new} = x_{i,t} + r_t(x_{i,t} - x_{j,t}) \quad r_t \epsilon (0,1) \quad (9)$$

If $f(x_{j,t}) < f(x_{i,t})$

$$x_{i,t}^{new} = x_{i,t} + r_t(x_{j,t} - x_{i,t}) \quad r_t \epsilon (0,1) \quad (10)$$

Step 3. If updated number is is negative convert it to a positive and if the obtained number is greater than 1, subtract it from 1 and put it instead.

Step 4. If $f(x_{i,t}^{new}) < f(x_{i,t})$, then replace $x_{i,t}$ by $x_{i,t}^{new}$ else replace $x_{i,t}$ by 2-OPT ($x_{i,t}^{new}$).

4.3. 2-OPT

the speed and performance of the proposed algorithm is improved, if TLBO is integrated with a local search such as 2-OPT (Chen et al., 2017). So, we'll have hybrid TLBO (HTLBO).

4.4. Solution representation

Our representation is an array consisting of *N* real values between (0, 1). An example with eight Operations is presented in Fig. 1.

1. Producing 8 random real numbers in (0,1):

1	2	3	4	5	6	7	8
0.9	0.127	0.913	0.964	0.097	0.278	*0.546*	0.957

2. Sorting real numbers in descending order:

4	8	3	1	7	6	*2*	5
0.9	0.957	0.913	0.905	0.546	0.278	*0.127*	0.097

3. Operation sequencing: 4,8,3,1,7,6,2,5

Fig. 1. Solution representation, encoding and decoding procedures

In following, to evaluate the proposed algorithm several case studies like Case study 1, Case study 2and Case study 3are considered from the literature.

Table 1
PCM for CS1

		1	2	3	4	5	6	7	8
(A1)	1	-	100	100	1	100	100	100	100
(B1)	2	11	-	0	100	1	100	100	100
(B2)	3	11	100	-	100	1	100	1	1
(C1)	4	100	100	100	-	100	100	100	100
(D1)	5	11	1	100	100	-	0	100	100
(D2)	6	11	1	100	100	100	-	100	100
(D3)	7	11	100	100	100	100	100	-	100
(E1)	8	11	1	100	100	100	100	1	-

5. Case Study 1

Case study 1 is a product that is taken from Bhaskara Reddy et al. (1999) according to Fig.2. The operations are labeled A1, B1, C1, D1, D2, D3 and E1. Fig.3 shows the operations precedence graph (OPG) and Table 1 indicates PCM. The symbol "-" states that there is no task

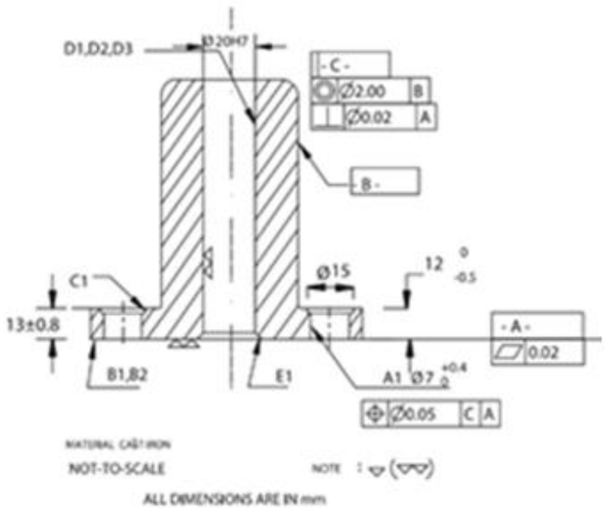

Fig. 2. Part drawing for CS1

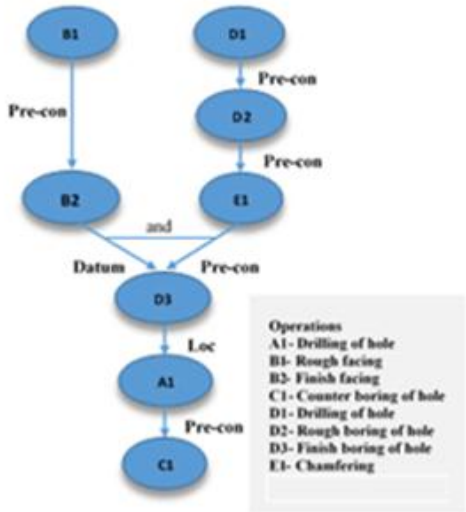

Fig. 3. OPG for CS1

Table 2
PCM for CS2

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A1	1	-	1	999	999	999	999	999	999	999	999	999	999	999	999	999	999
A2	2	999	-	10	999	999	999	999	999	999	999	999	999	999	999	999	999
B1	3	999	999	-	2	999	999	999	2	999	999	999	999	999	999	999	2
B2	4	999	999	999	-	2	999	999	2	999	999	999	999	999	999	999	2
B3	5	999	999	999	999	-	2	999	999	999	999	999	999	999	999	999	999
B4	6	999	999	999	999	999	-	1	999	999	999	999	999	999	999	999	999
B5	7	999	999	999	999	999	999	-	2	999	2	1	2	999	999	999	2
C1	8	999	999	2	2	999	999	999	-	2	2	999	999	999	999	999	2
C2	9	999	999	999	999	999	999	999	999	-	2	999	999	999	999	999	2
D1	10	999	999	999	999	999	999	999	2	999	-	2	2	999	999	999	2
D2	11	999	999	999	999	999	999	999	999	999	999	-	2	2	2	999	999
E1	12	999	999	999	999	999	999	999	999	999	2	999	-	2	2	999	999
E2	13	999	999	999	999	999	999	999	999	999	999	2	999	-	2	2	2
F1	14	999	999	999	999	999	999	999	999	999	999	2	2	2	-	2	2
F2	15	999	999	999	999	999	999	999	999	999	999	999	2	2	999	-	2
G	16	999	999	2	2	999	999	999	2	2	2	999	999	999	2	2	-

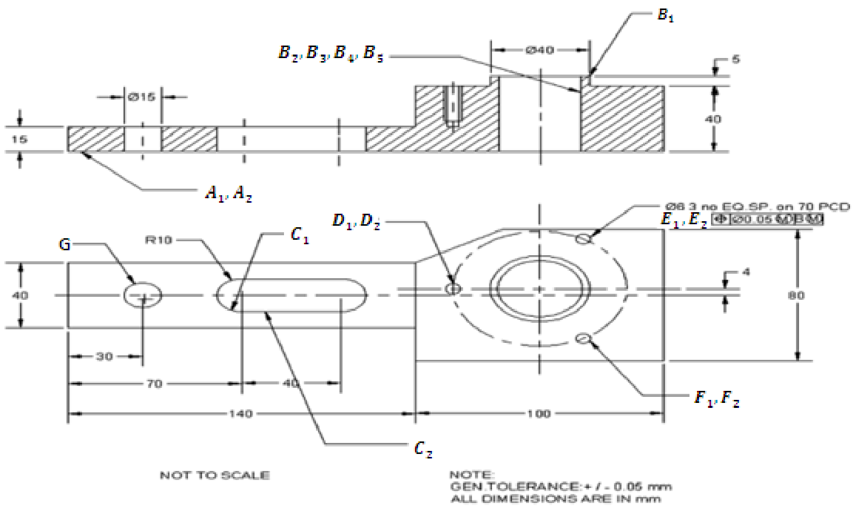

Fig. 4. Part drawing for CS2

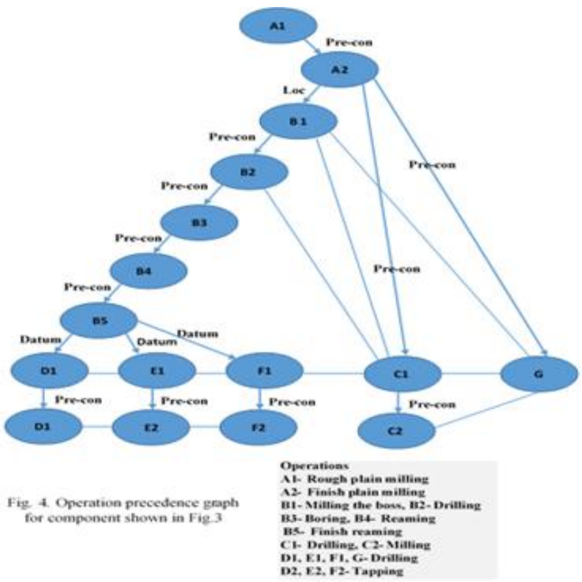

Fig. 5. OPG for Case study 2

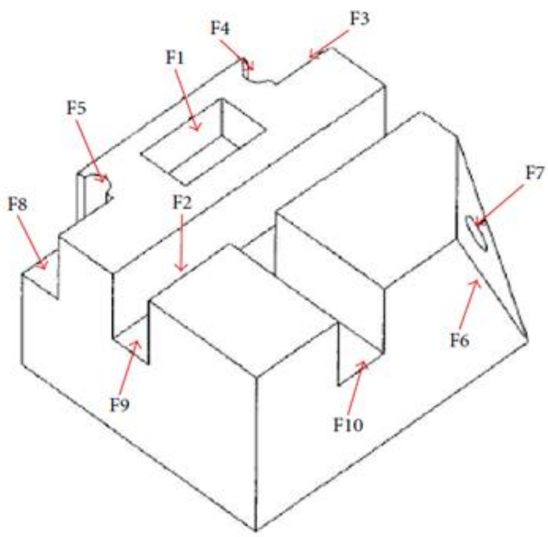

Fig. 6. Part drawing for Case study 3

Table 3
RPM of a highly difficult part for Case study 3

		1	2	3	4	5	6	7	8	9	10
F1	1	∞	-30	-20	-10	-10	-10	-10	-10	-10	-10
F2	2	50	∞	-10	-5	-5	-5	-5	-5	-10	-20
F3	3	40	10	∞	-30	-50	-10	5	10	15	20
F4	4	45	-45	-5	∞	4	40	-25	60	-30	-10
F5	5	100	40	60	55	∞	5	20	-10	-20	5
F6	6	100	70	-50	-15	40		40	20	-5	10
F7	7	60	5	30	-25	5	5	∞	-15	30	20
F8	8	90	30	-5	10	-5	-35	80	∞	15	-25
F9	9	-75	10	-5	5	-40	40	20	-90	∞	-5
F10	10	80	30	-20	45	75	25	-30	30	35	∞

Table 4

RPM of the sample part for Case study 3

		1	2	3	4	5	6	7	8	9	10
F1	1	∞	5	35	25	25	5	5	35	45	5
F2	2	5	∞	5	5	5	5	5	5	-45	5
F3	3	-25	5	∞	85	5	5	5	5	5	5
F4	4	-15	5	-75	∞	5	5	5	5	5	5
F5	5	-15	5	5	5	∞	5	5	-75	5	5
F6	6	5	5	5	5	5	∞	105	5	5	5
F7	7	5	5	5	5	5	-95	∞	5	5	5
F8	8	-25	5	5	5	85	5	5	∞	5	5
F9	9	-20	55	5	5	5	5	5	5	∞	-15
F10	10	5	5	5	5	5	5	5	5	25	∞

6. Case Study 2

Case study 2 is taken from Bhaskara Reddy et al. (1999), as shown in Fig. 4. The operations are labeled as A1, A2, B1, B2, B3, B4, B5, C1, C2, D1, D2, E1, E2, F1, F2, and G. Fig.5 shows OPG and Table 2 indicates PCM.

7. Case Study 3

Case study 3 is taken from Türkay and Hüseyin (1999), as shown in Fig. 6. and includs two examples, one highly difficult part with RPM in Table.3 and the other, a sample part with RPM in Table.4.

Table 5
Different value for the parameters

	Parameters	
	S_{pop}	T_F
Case study 1	100	1
Case study 2	600	2
Case study 3 (Sample part)	200	2
Case study 3 (Highly difficult part)	400	1

8. Computational Results

The quality of the solutions obtained from the proposed algorithm is affected by the values of their parameters. To set the value for the two key parameters, i.e., the population size S_{pop} and the teacher learning factor T_F, three types of case studies with different values are implented as shown in Table 5.

propsed HTLBO algorithm in this work has been used on Case study 1, Case study 2 and Case study 3. The results are compared with the previous works and the efficiency of HTLBO is shown in the following tables. The proposed HTLBO are coded in MATLAB R2016a software.

Optimal solution for Case study 1 are shown in Table 6. HTLBO produces optimal solution 5-6-2-3-8-7-1-4 with optimal cost (OC) 15 similar to MIP. The optimal cost is the same as the one reported by Weill et al. (1982) , Bhaskara Reddy et al. (1999), Li et al. (2004) and Nallakumarasamy et al. (2011); however, the computational time (CT) is reduced. The various outputs for Case study 2 are shown in Table 7. HTLBO produces optimal solutions 1-2-3-16-4-5-6-7-8-9-10-11-12-13-14-15, 1-2-3-4-5-6-7-8-9-10-12-14-11-13-16-15 and 1-2-3-4-5-6-7-8-9-10-12-13-11-14-16-15 with optimal cost 36 similar to MIP. From Table 7, it is observed that HTLBO gives a lesser computational time and the same optimal cost compared to Bhaskara Reddy et al. (1999), Li et al. (2004) and Nallakumarasamy et al. (2011). Results for CS3 in sample part, optimal solutions 5-8-1-2-9-10-7-6-4-3, 4-3-1-2-9-10-5-8-7-6 and 4-3-5-8-1-2-9-10-7-6 with optimal cost -315, shows same results compared to MIP and Turkey and Huseyin (1999) in Table 8. Optimal solution for Case study 3 in highly difficult part are shown in Table 9. HTLBO produces optimal solution 1-2-10-7-4-9-8-6-3-5 with optimal cost -360 similar to MIP. The optimal cost is the same as the one reported by Nallakumarasamy et al. (2011); however, Turkey and Huseyin (1999) obtained optimal cost -345. The results showed that proposed HTLBO produces optimal sequences with a lesser computational time. Furthermore, it is effective in solving the NP-hard problems such as OS problems.

Table 6
Results for Case study 1

MIP			
Node *b*	Node *e*	OS	OC
5	4	5-6-2-3-8-7-1-4	15
2	4	2-3-5-6-8-7-1-4	114

Proposed HTLBO		
OS	OC	CT
5-6-2-3-8-7-1-4	15	<0.2

Other researches		
	OC	CT
Weill et al. (1982)	15	-
Bhaskara Reddy et al. (1999)	15	30 s
Li et al. (2004)	15	11 s
Nallakumarasamy et al. (2011)	15	1 s

Table 7
Results for Case study 2

MIP			
Node *b*	Node *e*	OS	OC
1	15	1-2-3-4-5-6-7-8-9-10-12-13-11-14-16-15	36

Proposed HTLBO		
OS	OC	CT
1-2-3-16-4-5-6-7-8-9-10-11-12-13-14-15	36	2-3s
1-2-3-4-5-6-7-8-9-10-12-14-11-13-16-15		
1-2-3-4-5-6-7-8-9-10-12-13-11-14-16-15		

Other researches		
	OC	CT
Bhaskara Reddy et al. (1999)	36	20–40 s
Li et al. (2004)	36	18 s
Nallakumarasamy et al. (2011)	36	7 s

Table 8
Results for Case study 3 (Sample part) sample part

MIP			
Node *b*	Node *e*	OS	OC
4	6	4-3-5-8-1-2-9-10-7-6	-315

Proposed HTLBO		
OS	OC	CT
5-8-1-2-9-10-7-6-4-3	-315	0.4-0.5 s
4-3-1-2-9-10-5-8-7-6		
4-3-5-8-1-2-9-10-7-6		

Other researches		
	OC	CT
Turkey and Huseyin (1999)	−315	-

Table 9
Results for Case study 3 (Highly difficult part)

MIP			
Node *b*	Node *e*	OS	OC
1	5	1-2-10-7-4-9-8-6-3-5	-360

Proposed HTLBO		
OS	OC	CT
1-2-10-7-4-9-8-6-3-5	-360	0.7-0.8 s

Other researches		
	OC	CT
Turkey and Huseyin (1999)	−345	4 min
Nallakumarasamy et al. (2011)	−360	13 s

Refrences

Bhaskara Reddy, S.V., Shunmugam, M.S., & Narendran, T.T.(1999). Operation sequencing in CAPP using genetic algorithms. *International Journal of Production Research*, 37(5), 1063–1074.

Castro de Andrade, R. (2016). New formulations for the elementary shortest-path problem visiting a given set of nodes. *European Journal of Operational Research*, 254, 755–768.

Chen, X., Zhou,Y., Tang ,Z., and Luo,Q. (2017). A hybrid algorithm combining glowworm swarm optimization and complete 2-opt algorithm for spherical travelling salesman problems. *Applied Soft Computing*, 58, 104–114.

Doua, J., Li, J. & Su, C. (2018). A discrete particle swarm optimisation for operation sequencing in CAPP. *International Journal of Production Research*, 56(11), 3795- 3814.

Guo, Y.W., Mileham, A.R., Owen, G.W., & Li, W.D. (2006). Operation sequencing optimization using a particle swarm optimization approach. Proceeding of the Institution of Mechanical Engineering, Part B: *Journal of Engineering Manufacture*, 220(B12), 1945– 1958.

Halevi, G., & Weill, R. (1988). Development of flexible optimum process planning procedures. CIRP, 29(1), 313–317. Hosseini, S.M.H. (2018). Modeling and Solving the Job Shop Scheduling Problem Followed by an Assembly Stage Considering Maintenance Operations and Access Restrictions to Machines *Journal of Optimization in Industrial Engineering*, 12(1), 63-78.

Hu, Q., Qiao, L., & Peng, G. (2017). An Ant Colony Approach to Operation Sequencing Optimization in Process Planning. Proceedings of the Institution of MechanicalvEngineers, Part B: *Journal of Engineering Manufacture* 231(3), 470–489.

Keshavarz-Kohjerdi, F., & Bagheri, A. (2017). A linear-time algorithm for finding Hamiltonian (s,t)-paths in even- sized rectangular grid graphs with a rectangular hole. *Theoretical Computer Science*, 690, 26-58.

Krishna, A.G., and Rao, K.M. (2006). Optimisation of operations sequence in CAPP using an ant colony algorithm.*The International Journal of Advanced Manufacturing Technology*, 29, 159–164.

Korde U.P., Bora B.C., Stelson K.A., & Riley D.R.(1992) Computeraided process planning for turned parts using fundamental and heuristic principles. *Journal of Engineering for Industry*, 114, 31–40.

Koulamas, C. (1993). Operation sequencing and machining economics. *International Journal of Production Research*, 31(4), 957–975.

Li, L., Fuh, J.Y.H., Zhang, Y.F., & Nee, A.Y.C. (2005). Application of genetic algorithm to computer-aided process planning in distributed manufacturing environments. *Robotics and Computer-Integrated Manufacturing*, 21, 568–578.

Li, W.D., Ong, S.K., & Nee, A.Y.C. (2004). Optimization of process plans using a constraint-based tabu search approach. *International Journal of Production Research*, 42(10), 1955–1985.

Lin, C.J., & Wang, H.P. (1993). Optimal operation planning and sequencing; minimization of tool changeovers. *International Journal of Production Research*, 31(2), 311–324.

Nallakumarasamy, G., Srinivasan, P.S.S., Venkatesh Raja, K., & Malayalamurthi, R. (2011). Optimization of operation sequencing in CAPP using simulated annealing technique (SAT). *International Journal of Advanced Manufacturing Technology*, 54, 721–728.

Rao, R.V., Savsani, V.J., and Vakharia, D.P. (2011). Teaching– learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, 43, 303–315.

Salehi, M. & Tavakkoli-Moghaddam, R. (2009). Application of genetic algorithm to computer-aidedprocess planning in prelimirary and detailed planning. *Engineering Applications of Artificial Intelligence*, 22, 1179-1187.

Türkay, D.I., & Hüseyin, F. (1999). Optimisation of process planning functions by genetic algorithms. *International Journal of Computer and Industrial Engineering*, 36(2), 281–308.

Weill, R., Spur, G., & Eversheim, W. (1982). Survey of computer-aided process planning systems. CIRP Annals, 31(2), 539–551.

This article can be cited: Haleh, H., Sadati, A. & Hajisharifi, N. (2020). Operation Sequencing Optimization in CAPP Using Hybrid Teaching-Learning Based Optimization(HTLBO).
Journal of Optimization in Industrial Engineering. 13 (1), 123-130.

http://www.qjie.ir/article_666491.html
DOI: 10.22094/JOIE.2019.387.0