Swarm and Evolutionary Computation 43 (2018) 225–243

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

A novel strategy for the combinatorial production planning problem using integer variables and performance evaluation of recent optimization algorithms

Sandeep Singh Chauhan, Chinta Sivadurgaprasad, Rajasekhar Kadambur, Prakash Kotecha *

Department of Chemical Engineering, Indian Institute of Technology Guwahati, Guwahati, 781 039, Assam, India

ARTICLE INFO

Keywords:
Sanitized teaching-learning-based optimization
Genetic algorithm
Multi-population based ensemble differential evolution algorithm
Enhanced binary particle swarm optimization algorithm
Artificial bee colony algorithm
Production planning
Combinatorial optimization

ABSTRACT

In this article we propose a multi-unit production planning based optimization strategy that employs a set of integer and continuous variables to overcome many of the drawbacks of the formulation/strategies in literature [1,2] and helps in determining efficient production plans. Additionally, we have utilized an efficient strategy to handle the domain-hole constraints that does not rely on imposing penalties for violation. The benefits of the proposed strategy are demonstrated on eight cases, which have been previously discussed in literature to potentially guide the petrochemical industries, and provide an improvement of up to 34.66% in the profit. In addition to proposing an efficient strategy, this work also discusses the computational performance of five-optimization algorithms viz., the recently proposed sanitized-teaching-learning-based optimization algorithm, multi-population based ensemble differential evolution, enhanced binary particle swarm optimization algorithm, artificial bee colony algorithm and the popular real coded genetic algorithm on this combinatorial optimization problem.

1. Introduction

Petrochemical plants use a series of complex operations to convert naturally occurring raw materials into intermediate products, which can be subsequently converted into everyday consumer products. The complex nature of petrochemical network requires the systemic optimization of a large number of factors to be competitive in a globalized environment. Some of the previous works in this direction include the use of graph theory concepts in genetic algorithms for the scheduling of crude oil operations [3], use of linear programming and ArcGIS spatial analysis for the organization of the petrochemical industry in China [4], use of differential evolution to boost the hydrogen and octane number in naphtha reactor [5], use of particle swarm and non-linear programming techniques to optimize the operating parameters of refrigeration cycle [6], use of parallel differential evolution for the multi-objective operation optimization of naphtha pyrolysis process [7], use of NSGA – II and SPEA for the intelligent design of sensor networks [8], use of jumping gene adaptation of genetic algorithm for the multi-objective optimization of fuel oil blending [9] and the use of elitist TLBO for the production planning problem [2]. Swarm intelligence and gravitational search algorithm have also been used for the multi-objective optimization of synthesis gas production [10] and fuzzy optimization in production planning [11]. A review of recent soft computing techniques and its applications can be obtained from literature [12–15].

Some of the works that used mathematical programming models include design of multisite refinery and petrochemical network for optimal integration and coordination [16], identification of synergies in capacity expansion [17], the design of petrochemical networks based on an aggregated multi-objective function of profit, safety and environment impact [18], short-term scheduling of refinery operations using continuous [19,20] and discrete time optimization formulations [21]. Some of the works that have considered uncertainty include the optimal network design and storage management in petroleum distribution network [22], optimizing the supply chain using a two stage stochastic algorithm for uncertainty in operating and economic conditions [23], and the development of investment model for long-range capacity expansion under uncertain demand forecast scenarios [24].

In this article, we critically review the formulation/strategies

* Corresponding author.
E-mail address: pkotecha@iitg.ernet.in (P. Kotecha).

https://doi.org/10.1016/j.swevo.2018.04.004
Received 15 December 2017; Received in revised form 2 March 2018; Accepted 10 April 2018
Available online 14 June 2018
2210-6502/© 2018 Elsevier B.V. All rights reserved.

Nomenclature

i	product index, $i \in (1, \cdots, I)$, I denotes total number of products
j	process index, $j \in (1, \cdots, J)$, J denotes total number of processes
t	raw material index, $t \in (1, \cdots, T)$, T denotes total number of raw materials
l	production capacity level index, $l \in (1, \cdots, L)$, L denotes the number production levels of a process
b_{jt}	amount of raw material t required in process j for producing per ton of product
B	total available monetary resources ($)
c_{lj}	production capacity of level l for process j
C_{lj}	production cost of capacity level l for process j
C_j	total production cost of process j
E_j	selling price of product ($/ton) produced from process j
l_j, m_j, h_j	low, medium and high production capacity level of process j (tons of product/year)
L_j, M_j, H_j	represents the proportions of production using production capacity levels l_j, m_j, and h_j
n_i	number of processes employed to produce product i
N_{lj}	number of production unit of process j for production capacity level c_{lj}
Profit	profit obtained from the production plan
P	objective function used to obtain overall production from the production plan
P_t^{raw}	penalty incurred in fitness function due to insufficient raw material of type t
$P^{Investment}$	penalty incurred in fitness function due to insufficient investment cost
P_i^{unique}	penalty incurred in fitness function due to violation of unique process constraint for product i
R_t	available feedstock of raw material t (tons/year)
S_i	set of all processes that can produce product i
V_{lj}	investment cost of capacity level l for process j
v_{lj}	investment cost for producing x_{lj} and is based on the line connecting (c_{lj}, V_{lj}) and (c_{l+1j}, V_{l+1j})
x_{lj}	amount of product produced from capacity level between c_l and c_{l+1} of process j
X_j	total amount of product produced from process j (from all the levels)
Y_j	a binary variable with a value of 1 if $X_j \leq m_j$, else 0
Z_j	a binary variable with a value of 1 if $X_j > 0$, else 0

available in literature [1,2,25] for determining production planning and propose a multi-unit based strategy to determine the optimal production plan, which overcomes the limitations and helps in the determination of optimal production plans with an improved profit. We have also utilized an efficient strategy to handle decision variables that have domain-holes. The benefits of the proposed strategy are demonstrated on all the eight cases that have been discussed in literature. Additionally, this article also helps in comparing the performance of real coded Genetic Algorithm (GA), Sanitized Teaching Learning based Optimization (s-TLBO), Multi-Population based Ensemble Differential Evolution (MPEDE), enhanced Binary Particle Swarm Optimization (BPSO) and Artificial Bee Colony (ABC) on large problems, involving both integer (>150) and continuous variables (>100) with complex constraints. To the best of our knowledge, there is no study which has evaluated the performance of these five algorithms on combinatorial problems, on identical test conditions, of this size involving complex domain-hole and unique process constraints.

The article is structured as follows: In the following section, we provide the problem description and discuss the drawbacks of the formulation/strategies in literature. Subsequently, we propose the multi-unit strategy to determine efficient production plans. Subsequently, the benefits of the proposed strategy are demonstrated on eight cases from the literature and the performance of the five algorithms is evaluated. We conclude the article by summarizing the developments in this work and briefly discuss possible future work.

2. Problem description

In the production planning problem considered in this work and in literature [1,2,25], the objective is to determine the optimal production plan that yields the maximum profit. The information available for the design of production plan include (i) the set of potential processes along with their (a) production capacity levels, (b) production costs of the various levels, (c) investment costs of the various levels and (d) the products produced by these processes, (ii) the selling price and the demand of each product, (iii) the amount of raw materials available for the entire production plan (iv) the amount of raw material required in each process to produce a unit quantity of the product, and (v) the total amount of monetary resource that is available for investment. A schematic representation of the production-planning problem is shown in Fig. 1. Table A1-A4 provides details of a typical production planning problem that has been previously used in literature [1] to potentially guide the petrochemical industries in Saudi Arabia. It involves 54 different processes, which can operate at three different production levels (low, medium and high) to produce 24 unique products. In addition, the tables also provide investment costs (V_{lj}, V_{mj}, V_{hj}), production costs (C_{lj}, C_{mj}, C_{hj}), amount of raw materials (b_j) required by every process, and the selling price (E_j) of the various products. A production plan requires the specification of (i) the products that are to be produced, (ii) the processes that are to be used to produce the selected products and (iii) the production levels of the selected processes. The production plan should ensure that the investment budget required for the entire production plan is not more than the available monetary resources (B).

Similarly, the production plan should also ensure that the amount of different raw materials required by all the processes in the optimal production plan should be less than the quantity of raw material (R_t) that is available. In certain cases, a unique process constraint is required to be satisfied in which the production plan should not have more than a single process for producing a product. Fig. 2 shows the variation of production and investment cost with the production level for a typical process. From Fig. 2, it can be seen that the determination of production cost and investment is linear in between two successive production levels. It can also be observed that any production that is greater than zero but less than the lower level capacity (l_j) is not permissible. Similarly, any production greater than the higher capacity level (h_j) is not permissible. Thus it can be seen that the production planning problem is a combinatorial problem involving domain holes and piecewise continuous functions.

Limitations of the formulation/strategy in literature:

For the production planning described in the preceding discussion, the amount of production from a process should ideally be restricted either by the amount of resources available (investment as well as raw materials) to produce the product or by the demand of the product in the market. In literature, a MILP formulation [1] as well as a multi-level [2,25] has been proposed for determining the optimal production plan. The MILP formulation artificially restricts the maximum production from a process to the capacity of high-level production (h_j) whereas the multi-level based strategy artificially restricts the maximum production

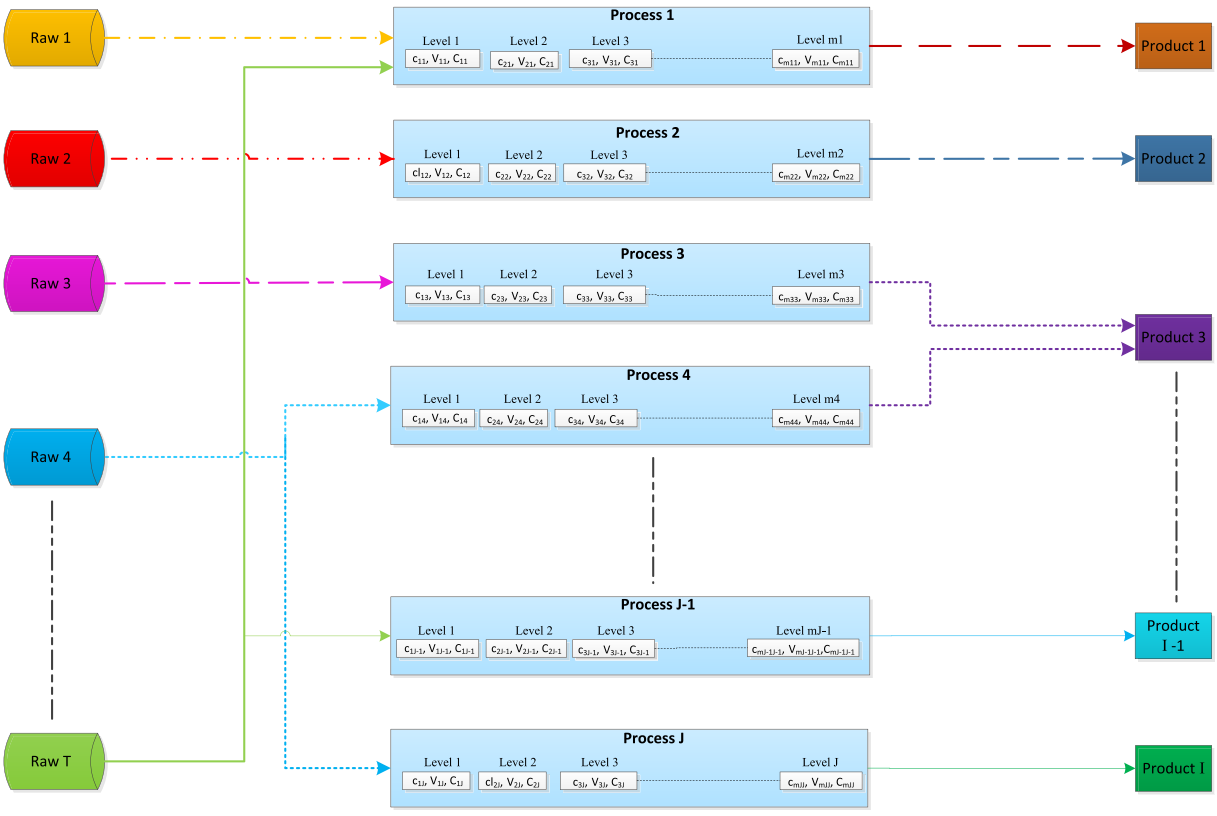

Fig. 1. Schematic representation of the production planning problem.

from a process to the sum of the capacity of the medium and high-level process $(m_j + h_j)$.

In the following discussion, we briefly review the relevant parts of these two works to discuss their limitations, which have been overcome by the proposed multi-unit strategy in this article. The MILP formulation in literature [1] has been proposed only for three production levels and uses two binary variables (Y_j, Z_j) for each process j. The binary variable (Y_j) is required to assume a value of 1 if the production (X_j) from process j is less than or equal to the capacity of the medium level (m_j) of process j

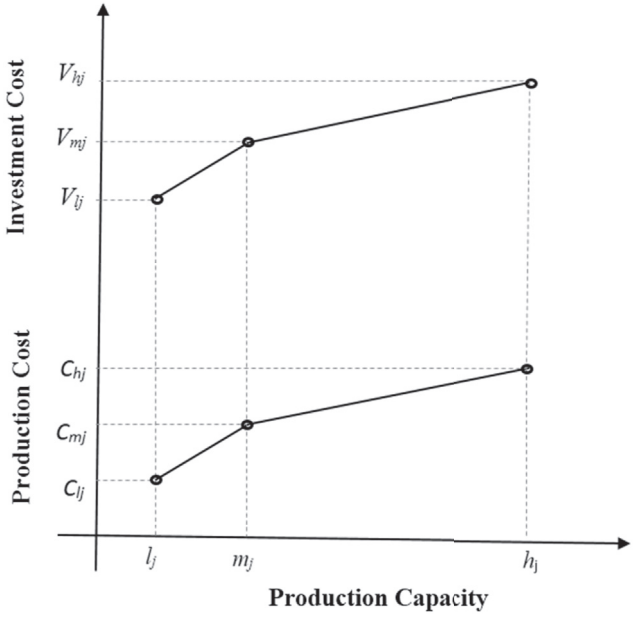

Fig. 2. Costs as a function of production level capacity for process j [1].

and a value of 0 if the production from process j is greater than m_j. The binary variable (Z_j) is required to assume a value of 1 if process j produces any product and should assume a value of zero otherwise. These requirements were ensured by the following set of equations

$$X_j = l_j L_j + m_j M_j + h_j H_j \tag{1}$$

$$L_j \le Y_j \tag{2}$$

$$H_j \le 1 - Y_j \tag{3}$$

$$L_j + M_j + H_j = Z_j \tag{4}$$

$$X_j \le NZ_j, N \text{ is a large number } (\text{as in the big} - M \text{ method}) \tag{5}$$

The incorporation of these binary variables enables the determination of the profit, despite the piecewise linear function of production cost and investment cost shown in Fig. 2, using the linear equation in Equation (6) and the investment cost shown in Equation (7)

$$\text{Maximize} \quad P = \sum_{j=1}^{J} E_j X_j - \left(C_{lj} L_j + C_{mj} M_j + C_{hj} H_j\right) \tag{6}$$

$$\text{Investment cost} = \sum_{j=1}^{J} \left(V_{lj} L_j + V_{mj} M_j + V_{hj} H_j\right) \tag{7}$$

A careful analysis of Equation (1) to Equation (5) can lead to the conclusions in Equation (8) if the process j produces a product.

$$\begin{aligned} Y_j = 1 &\Rightarrow H_j = 0 \Rightarrow X_j = l_j - M_j\left(l_j - m_j\right) \Rightarrow X_j \le m_j \\ Y_j = 0 &\Rightarrow L_j = 0 \Rightarrow X_j = h_j - M_j\left(m_j - h_j\right) \Rightarrow X_j \le h_j \end{aligned} \tag{8}$$

Thus, it can be seen that the constraints in the MILP formulation artificially restrict the maximum production of a product X_j to the high

production level (h_j). Thus even if the resources are available and if there is a demand in the market, the process j is restricted to produce utmost h_j amount of product from process j thereby determining suboptimal solutions. Additionally, the MILP formulation would require substantial modification for production planning involving more than three levels. For a production-planning problem with J processes and three production levels, it requires $2J$ binary variables and this would increase in the presence of additional levels potentially leading to computational issues. It should be noted that the MILP formulation contains several other constraints related to unique process requirement, satisfaction of the raw material and budget constraints which are not discussed here as these do not contribute to its limitations.

(i) **Decision Variables:** Without the loss of generality, let us assume that there are J processes with each of them having L different production capacities for which the production cost and investment cost is known. In this case, we will use LJ integer variables to represent the number of units of each production capacity that would be used by every process. Additionally $(L-1)J$ continuous variables are required wherein each continuous variable corresponds to the production capacity of the unit that is in between the two successive production levels for which the production cost and the investment cost is known. Thus according to this strategy, for J process with L levels for each of the process, the entire set of decision variables is as shown in Equation (9)

$$\left[\overbrace{N_{11},N_{12},\cdots,N_{1J}}^{Integer1},\underbrace{N_{21},N_{22},\cdots,N_{2J}}_{Integer\ 2},\cdots,\overbrace{N_{L1},N_{L2},\cdots,N_{LJ}}^{Integer\ L},\underbrace{X_{1-2,1},X_{2-3,1},\cdots,X_{(L-1)-L,1}}_{continuous1},\overbrace{X_{1-2,2},X_{2-3,2},\cdots,X_{(L-1)-L,2}}^{continuous2},\cdots,\underbrace{X_{1-2,J},X_{2-3,J},\cdots,X_{(L-1)-L,J}}_{continuous\ J}\right] \tag{9}$$

In subsequent works [2,25], a multi-level production strategy was employed in which for every process having three different production capacities (low (l_j), medium (m_j) and high (h_j), two continuous variables $(x_{l-m,j}, x_{m-h,j})$ were used to represent the production by a unit with capacity in between the low-medium $(x_{l-m,j})$ levels and another unit with capacity in between the medium-high levels (x_{m-hj}). This multi-level strategy showed an improvement over the results of the MILP formulation [1]. However, this strategy artificially restricted the maximum production from process j to $m_j + h_j$ thereby leading to production plans with lower profit. Moreover, the use of elitist TLBO [2] requires the incorporation of the duplicate removal step in the TLBO algorithm and leads to considerable computational burden. In this work, the domain hole constraints were handled using a penalty approach which can potentially delay the discovery of the optimal solution. In this current work, we have overcome the drawbacks of the single [1] and multi-level production planning [2,25] strategies based by using a strategy which incorporates the concept of multiple units along with multi-level.

3. Proposed multi-unit strategy for production planning

In this proposed strategy, we employ a set of integer and continuous decision variables to account for the production using multiple units and multiple levels from the process j as shown in Fig. 3. The upper bounds on the integer variables are based on the availability of resources, in terms of raw materials and budget, and the demand of the product. This ensures maximum production of the profitable products which would lead to the determination of better production plans than those suggested in literature [1,2,25] as it does not artificially restrict the production. For example, consider a process j which has three production levels of l_j, m_j and h_j. If profitable, one can employ N_{lj}, N_{mj}, and N_{hj} numbers of units of capacity l_j, m_j and h_j respectively along with a unit that is capable of producing quantity xlm_j that is in between l_j and m_j and another unit that is capable of producing quantity xmh_j in between m_j and h_j. Thus, the decision variables are N_{lj}, N_{mj}, and N_{hj} which are integer variables and the two continuous variables xlm_j and xmh_j whose ranges are as specified earlier. This leads to the maximum production of the most profitable product, which is restricted only by the availability of the resources and the demand in the market.

In Equation (9), *Integer*1 represents the collection of variables that denote the number of units for capacity level 1 for each of the J process with N_{lj} indicating the number of the units for level l of process j. For example N_{12} would indicate the number of units of capacity level 1 for process S2. *Integer*2 represents the collection of variables that denote the number of units of capacity level 2 for each of the J process while *IntegerL* represents the collection of variables that denote the number of units of capacity level L for each of the process J. In this, N_{LJ} would indicate the number of units of capacity level L for process J. Thus, it can be noted that each of the L parts have J integer variables and hence there are a total of LJ integer variables. Similarly, *continuous*1 denotes the collection of variables that denote the production from process S1 whose capacity is in between two successive levels. For example, $X_{2-3,1}$ would indicate the amount of production from a unit whose production capacity is in between level 2 and level 3 from process S1. Similarly, if L is the production capacity of the last level and $L-1$ is the production capacity of the penultimate level, then $X_{(L-1)-L,1}$ would indicate the production from a unit whose production capacity is in between these two levels for process S1. Similarly *continuous*2 indicates the collection of variables that denote the production from units whose production capacity is in between two successive levels for process S2 while *continuousJ* indicates the collection of variables that denote the production from units whose production capacity is in between two successive levels for process J. It can be noted that there are $L-1$ continuous variables in each of these terms and hence a total of $(L-1)J$ continuous variables are required. In the rest of article, the number of units used in the production from the process j from the production capacity l will be denoted as N_{lj} and the production from a unit whose production capacity is between c_l and c_{l+1} will be denoted as x_{lj}.

For example, consider a production-planning problem involving 54 processes (J) and that each process has three (L) production capacities (low, medium and high). In this case, we will have 162 $(=LJ)$ integer decision variables and 108

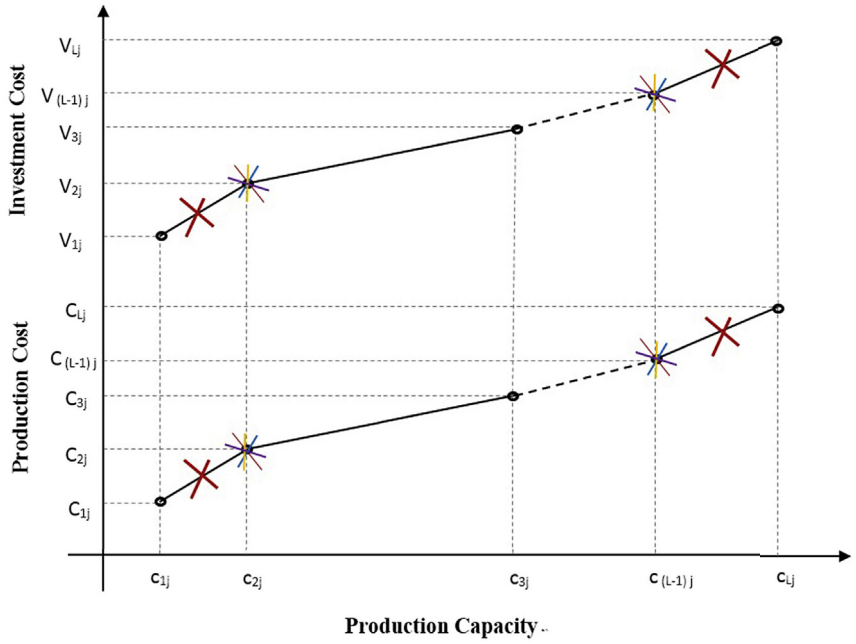

Fig. 3. Variation of costs as a function of production capacity for process j with multiple units.

continuous decision variables $(= J(L-1))$. In the 162 integer decision variables, each processes will have 3 integer variables corresponding to the number of production units with low level capacity (N_{1j}), medium level capacity (N_{2j}) and high level capacity (N_{3j}). In the 108 continuous decision variables, two continuous variables will correspond to every process. One continuous variable per process (x_{1j}) will represent the production between the low-level capacity and the medium level capacity whereas the other continuous variable (x_{2j}) per process will represent the production between the medium level capacity and the high-level capacity.

(ii) **Domains:** For the entire set of the decision variables defined in the preceding discussion, it is evident that the lower bound is zero to account for zero production if a product is not produced. The upper bound of the integer variables should represent the maximum number of units of process j that can be utilized and can be determined by the amount of various resources, i.e., the amount of investment budget and the amount of raw materials that are available in limited quantities. For example, if V_{lj} is the investment cost required for process j with capacity level l and if B denotes the amount of budget that is available for the production planning, then the upper bound on the integer variable N_{lj} based on the budget and investment cost is given by $\left\lfloor B/V_{lj} \right\rfloor$ where $\lfloor . \rfloor$ denotes the floor operator. Similarly if R_t denotes the maximum amount of raw material t that is available for the production

Table 1
Comparison of the five algorithms.

	GA	TLBO	MPEDE	BPSO	ABC
Origin	Holland, early 70s	Rao et al. [29]	Wu et al. 2015	Mirjalili et al. 2012	Karaboga 2005
Inspiration	Genetics	Teaching-learning process	Population partitioning in the conventional DE	Transfer functions to accommodate the discrete nature in the conventional PSO	Behaviour of honeybees
Major Phases/Operations	Selection, Crossover, Mutation	Teachers Phase, Students Phase	Three mutation strategies and selection of best mutation strategy	Updating the velocity and position	Employed bees, onlooker bee, and scout bee phase
Variants	Binary Coded, Real Coded [38] with multiple variants of operators [27].	Elitist TLBO [39], SPMGTLO [40], Improved TLBO [41]	Multi-search differential evolution algorithm [42]	Hybridized with gravitational search algorithm [43]	Modified ABC [36], Gbest-guided ABC [44], Chaotic bee colony [45]
Tuning Parameters	Among others, number of generations, population size, crossover and mutation probability.	Number of generations, population size.	Number of generations, population size, ratio between indicator population and whole population, generation gap.	Number of generations, population size, acceleration constants and inertia weight.	Number of maximum function evaluations, population size.
Function Evaluations	Among other parameters, depends on the type of crossover, tournament size, and size of mating pool.	Requires the evaluation of exactly two solutions per population member in every generation.	One functional evaluation per population member in each generation.	One functional evaluation per population member in each generation.	A maximum of two functional evaluations per population member in every generation for the employed bees phase and onlooker bees and a maximum of one functional evaluation in the scout bee phase

Table 2a
Statistical performance of the five algorithms (without guess).

Case		GA	s-TLBO	MPEDE	BPSO	ABC
Case 1	Best	1E+84	**−709.95**	3.41E+20	−261.45	1E+93
	Worst	1E+99	−551.80	1E+33	8.8E+21	1E+102
	Mean	1.5E+98	−652.85	1E+31	1.37E+21	3.9E+101
	Median	1E+93	−655.23	6.14E+22	5.9E+20	1E+99
	St.dev	3.59E+98	31.27	1E+32	1.8E+21	4.9E+101
Case 2	Best	1E+84	**−839.25**	1.75E+21	−357.20	1E+93
	Worst	1E+99	−635.40	1E+33	7.24E+21	1E+102
	Mean	1.5E+98	−766.61	1E+31	1.35E+21	3.9E+101
	Median	1E+93	−769.32	5.39E+22	6.93E+20	1E+99
	St.dev	3.59E+98	40.95	1E+32	1.62E+21	4.9E+101
Case 3	Best	1E+96	**−1262.05**	2.24E+22	−538.55	1E+99
	Worst	1E+105	−952.60	1E+63	5.24E+21	1E+105
	Mean	1.1E+104	−1118.94	1E+61	4.17E+20	2.2E+104
	Median	1E+102	−1111.23	5.38E+23	1.97E+19	1E+102
	St.dev	3.1E+104	64.39	1E+62	9.36E+20	4.2E+104
Case 4	Best	1E+96	**−1509.15**	1.46E+22	−790.9	1E+99
	Worst	1E+105	−1250.55	1E+63	5.21E+21	1E+105
	Mean	1.1E+104	−1402.23	1E+61	3.54E+20	2.2E+104
	Median	1E+102	−1398.41	5.77E+23	1E+18	1E+102
	St.dev	3.1E+104	63.15	1E+62	7.92E+20	4.2E+104
Case 5	Best	9.26E+23	**−702.70**	−617.10	−451.62	3.18E+24
	Worst	1.98E+24	−575.47	4.81E+20	1.42E+20	4.8E+24
	Mean	1.33E+24	−640.38	5.96E+18	5.84E+18	4.12E+24
	Median	1.31E+24	−642.33	−498.02	−210.49	4.15E+24
	St.dev	2.27E+23	28.37	4.93E+19	2.23E+19	3.04E+23
Case 6	Best	8.04E+23	**−834.05**	−781.40	−579.25	3.3E+24
	Worst	1.8E+24	−680.30	−22.80	2.68E+20	4.69E+24
	Mean	1.3E+24	−774.99	−629.00	7.37E+18	4.07E+24
	Median	1.3E+24	−776.53	−645.62	−265.09	4.11E+24
	St.dev	2.07E+23	31.67	100.64	3.64E+19	3.06E+23
Case 7	Best	4.24E+24	**−1200.75**	−1045.61	−766.85	1.35E+25
	Worst	8.62E+24	−979.17	1.75E+19	1.19E+18	1.93E+25
	Mean	5.9E+24	−1095.99	1.75E+17	1.19E+16	1.65E+25
	Median	5.79E+24	−1099.82	−775.99	−512.51	1.67E+25
	St.dev	9.64E+23	48.39	1.75E+18	1.19E+17	1.16E+24
Case 8	Best	3.9E+24	**−1469.95**	−1258.7	−1008.23	1.36E+25
	Worst	9.51E+24	−1233.43	−140.90	−362.00	1.92E+25
	Mean	5.87E+24	−1353.37	−950.67	−741.32	1.64E+25
	Median	5.73E+24	−1359.07	−959.75	−743.49	1.64E+25
	St.dev	1.05E+24	47.59	149.12	103.12	1.17E+24

planning, if b_{jt} denotes the amount of raw material t that is required in process j for producing per ton of the product and c_{lj} denotes the capacity of the level l, then the maximum number of units that can be determined by $\left\lfloor \frac{R_t}{b_{jt}c_{lj}} \right\rfloor$. Thus the domain of the integer decision variables can be given by Equation (10)

$$0 \le N_{lj} \le \min\left(\left\lfloor \frac{B}{V_{lj}} \right\rfloor, \min_{\forall t=1,2,\cdots,T} \left\lfloor \frac{R_t}{b_{jt}c_{lj}} \right\rfloor \right) \forall l = 1,2,\cdots,L; \forall j = 1,2,\cdots,J \quad (10)$$

In the above expression, T denotes the total number of raw materials that are required in the process j. Thus for the example involving three production levels, the domain of the integer variables can be given as follows

$$0 \le N_{1j} \le \min\left(\left\lfloor \frac{B}{V_{1j}} \right\rfloor, \min_{\forall t=1,2,\cdots,T} \left\lfloor \frac{R_t}{b_{jt}c_{1j}} \right\rfloor \right) \quad \forall j = 1,2,\cdots,J$$

$$0 \le N_{2j} \le \min\left(\left\lfloor \frac{B}{V_{2j}} \right\rfloor, \min_{\forall t=1,2,\cdots,T} \left\lfloor \frac{R_t}{b_{jt}c_{2j}} \right\rfloor \right) \quad \forall j = 1,2,\cdots,J$$

$$0 \le N_{3j} \le \min\left(\left\lfloor \frac{B}{V_{3j}} \right\rfloor, \min_{\forall t=1,2,\cdots,T} \left\lfloor \frac{R_t}{b_{jt}c_{3j}} \right\rfloor \right) \quad \forall j = 1,2,\cdots,J$$

As the continuous variables denote the amount of product to be produced using a unit whose capacity is in between two successive specified levels, if produced, x_{lj} should be in the region between c_{lj} and $c_{l+1\,j}$. However, it can also remain unproduced thereby necessitating the inclusion of zero in its domain. Thus, the domain of continuous variables is given by

$$0 \le x_{lj} \le c_{l+1,j} \quad \forall l = 1,2,\cdots,L-1; \forall j = 1,2,\cdots,J \quad (11)$$

For the example with three different production capacities, the domain of the two continuous decision variables for process j is as shown in Equation (12)

$$\begin{aligned} 0 \le x_{1j} \le m_j \quad \forall j = 1,2,\cdots,J \\ 0 \le x_{2j} \le h_j \quad \forall j = 1,2,\cdots,J \end{aligned} \quad (12)$$

(iii) **Domain Hole:** From Fig. 3, it can be realized that if the production from a process is greater than zero, then it should not be less than the lower level capacity of the process. This is because of the discontinuity in the region 0 to c_{1j} where the production and investment cost cannot be estimated. Instead of penalizing the decision variables for assuming a value that is greater than 0 but less than c_{1j}, the following correction strategy is employed [27].

$$(x_{lj} > 0) \& (x_{lj} < c_{lj}) \Rightarrow x_{lj} = 0; \forall l = 1,2,\cdots,L-1; \forall j = 1,2,\cdots,J \quad (13)$$

For the case of above example with three production capacities, the above equation can be written into the following two constraints

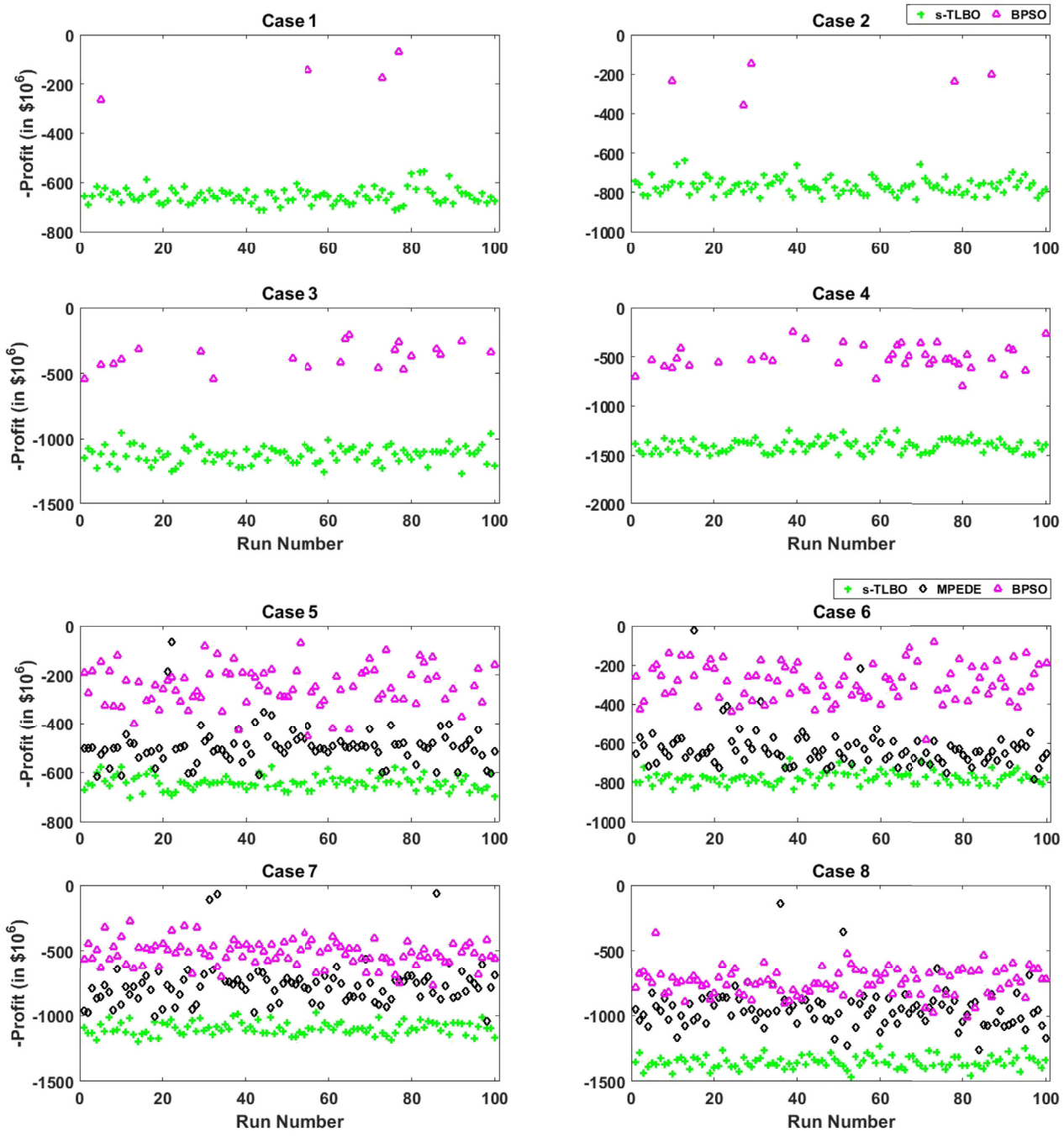

Fig. 4. Performance of the algorithms without initial guess.

$$\begin{aligned}&(x_{1j} > 0)\&(x_{1j} < l_j) \Rightarrow x_{1j} = 0 \quad \forall j = 1, 2, \cdots, J\\ &(x_{2j} > 0)\&(x_{2j} < m_j) \Rightarrow x_{2j} = 0 \quad \forall j = 1, 2, \cdots, J\end{aligned} \tag{14}$$

Due to the inherent nature of the computational intelligence techniques designed for continuous variables, the integer variables can assume continuous values and these are rounded off to the nearest integer thereby ensuring that all the LJ variables are in their respective bounds.

(iv) **Raw Material and Investment Constraint:** In order to ensure the satisfaction of the raw material and investment constraints, a penalty function approach is employed. The total amount of a product produced from all the multiple units and multiple levels of each process can be determined using Equation (15).

$$X_j = \sum_{l=1}^{L} N_{lj} c_{lj} + \sum_{l=1}^{L-1} x_{lj} \quad \forall j = 1, 2, \cdots, J \tag{15}$$

The total amount of raw material t that is consumed in the entire production plan is given by $\sum_{j=1}^{J} b_{jt} X_j$. If this exceeds the available quantity R_t, a penalty is determined as shown in Equation (16).

$$P_t^{raw} = \begin{cases} \left(R_t - \sum_{j=1}^{J} b_{jt} X_j\right)^2 & R_t < \sum_{j=1}^{J} b_{jt} X_j \\ 0 & R_t \geq \sum_{j=1}^{J} b_{jt} X_j \end{cases} \quad \forall t = 1, 2, \cdots, T \tag{16}$$

Table 2b
Statistical performance of the five algorithms (with guess).

Case		GA	s-TLBO	MPEDE	BPSO	ABC
Case 1	Best	−659.87	**−706.25**	−692.22	−669.07	−525.80
	Worst	−222.40	−540.27	−497.77	−450.12	−70.50
	Mean	−503.28	−645.86	−600.71	−559.72	−240.21
	Median	−516.76	−651.01	−615.07	−561.52	−216.38
	St.dev	87.75	31.58	45.01	41.97	111.98
Case 2	Best	−784.30	**−841.76**	−818.15	−742.10	−668.42
	Worst	−214.50	−637.85	−564.52	−479.50	−60.33
	Mean	−589.90	−775.56	−725.44	−630.71	−265.03
	Median	−611.74	−782.65	−733.51	−630.98	−238.85
	St.dev	105.78	41.02	55.76	54.60	126.13
Case 3	Best	−1082.03	−1237.25	**−1251.80**	−894.70	−1007.63
	Worst	−472.45	−885.07	−881.72	−582.75	−95.45
	Mean	−770.12	−1120.58	−1073.99	−754.85	−386.72
	Median	−760.12	−1116.56	−1058.30	−753.15	−334.72
	St.dev	134.68	77.53	83.86	64.19	189.35
Case 4	Best	−1356.73	**−1514.55**	−1499.26	−1145.38	−1100.78
	Worst	−492.30	−1295.06	−1029.80	−821.92	−95.45
	Mean	−975.05	−1412.56	−1329.92	−975.02	−498.55
	Median	−979.10	−1405.96	−1354.71	−979.03	−477.98
	St.dev	158.72	52.09	101.50	72.06	225.22
Case 5	Best	−671.11	**−709.25**	−686.35	−676.16	−531.52
	Worst	−266.80	−582.47	−467.47	−455.01	−74.50
	Mean	−553.04	−649.61	−596.36	−599.43	−234.01
	Median	−575.30	−652.12	−610.06	−603.19	−212.46
	St.dev	85.17	22.01	50.67	42.77	113.77
Case 6	Best	−783.76	**−844.35**	−823.05	−771.35	−668.42
	Worst	−360.45	−715.75	−603.40	−604.33	−60.33
	Mean	−646.42	−790.92	−737.70	−687.93	−262.75
	Median	−665.10	−787.87	−744.97	−690.50	−241.783
	St.dev	92.04	24.56	42.37	42.29	131.50
Case 7	Best	−1188.55	**−1224.90**	−1221.86	−1061.50	−956.87
	Worst	−449.15	−986.12	−888.58	−691.82	−95.45
	Mean	−855.42	−1124.55	−1089.79	−927.26	−378.27
	Median	−859.68	−1128.63	−1081.33	−935.73	−332.48
	St.dev	142.06	55.92	72.71	64.73	172.69
Case 8	Best	−1419.10	−1439.48	**−1514.55**	−1308.20	−1139.47
	Worst	−657.20	−1235.88	−1089.48	−992.20	−95.45
	Mean	−1106.00	−1364.58	−1359.80	−1173.63	−494.16
	Median	−1129.37	−1372.39	−1376.60	−1168.02	−469.68
	St.dev	157.75	41.44	88.40	70.98	235.48

The investment cost for each process can be determined using Equation (17)

$$V_j = \sum_{l=1}^{L} N_{lj} V_{lj} + \sum_{l=1}^{L-1} x_{lj} v_{lj} \quad \forall j = 1, 2, \cdots, J; l = 1, 2, \cdots, L \tag{17}$$

In Equation (17), v_{lj} indicates the investment cost of x_{lj} and is based on the line connecting (c_{lj}, V_{lj}) and (c_{l+1j}, V_{l+1j})

$$v_{lj} = V_{lj} + \left(\frac{V_{l+1j} - V_{lj}}{c_{l+1j} - c_{lj}}\right)(x_{lj} - c_{lj}) \quad \forall j = 1, 2, \cdots, J; l = 1, 2, \cdots, L-1 \tag{18}$$

If the total investment cost of the entire production plan, V_j exceeds the available budget B, a penalty is determined as shown in Equation (19)

$$P^{Investment} = \begin{cases} \left(B - \sum_{j=1}^{J} V_j\right)^2 & if \sum_{j=1}^{J} V_j > B \\ 0 & if \sum_{j=1}^{J} V_j \leq B \end{cases} \tag{19}$$

(v) **Unique Process Constraint:** As explained earlier, a product can be produced from different processes. However at times, it is desired that a product be produced from a single type of process. In such circumstances, the unique process constraint has to be included. Given the values for the entire set of decision variables (as in evolutionary algorithms), it is simple to determine the number of processes employed for producing for each product. If n_i represents the number of processes employed for producing the product i, then the penalty (P_i^{unique}) for violating the unique process constraint can be determined as shown in Equation (20)

$$P_i^{unique} = \begin{cases} 1000^{n_i} & n_i > 1 \\ 0 & n_i \leq 1 \end{cases} \quad \forall i = 1, 2, \cdots I \tag{20}$$

The above equation ensures that the penalty increases with an increase in the number of processes used to manufacture the product i. In case a product is not produced from any of the processes, the unique process requirement is not applicable and hence no penalty would be incurred. Similarly, there would be no penalty incurred if it is produced from a single process.

(vi) **Objective Function:** In order to determine the profit from a production plan, the production cost from each process can be determined from Equation (21) in which the first term corresponds to the production cost from the multiple units whose production capacity is c_{lj} and production cost is C_{lj}. The second term corresponds to the production cost for the production of x_{lj} using the multiple levels.

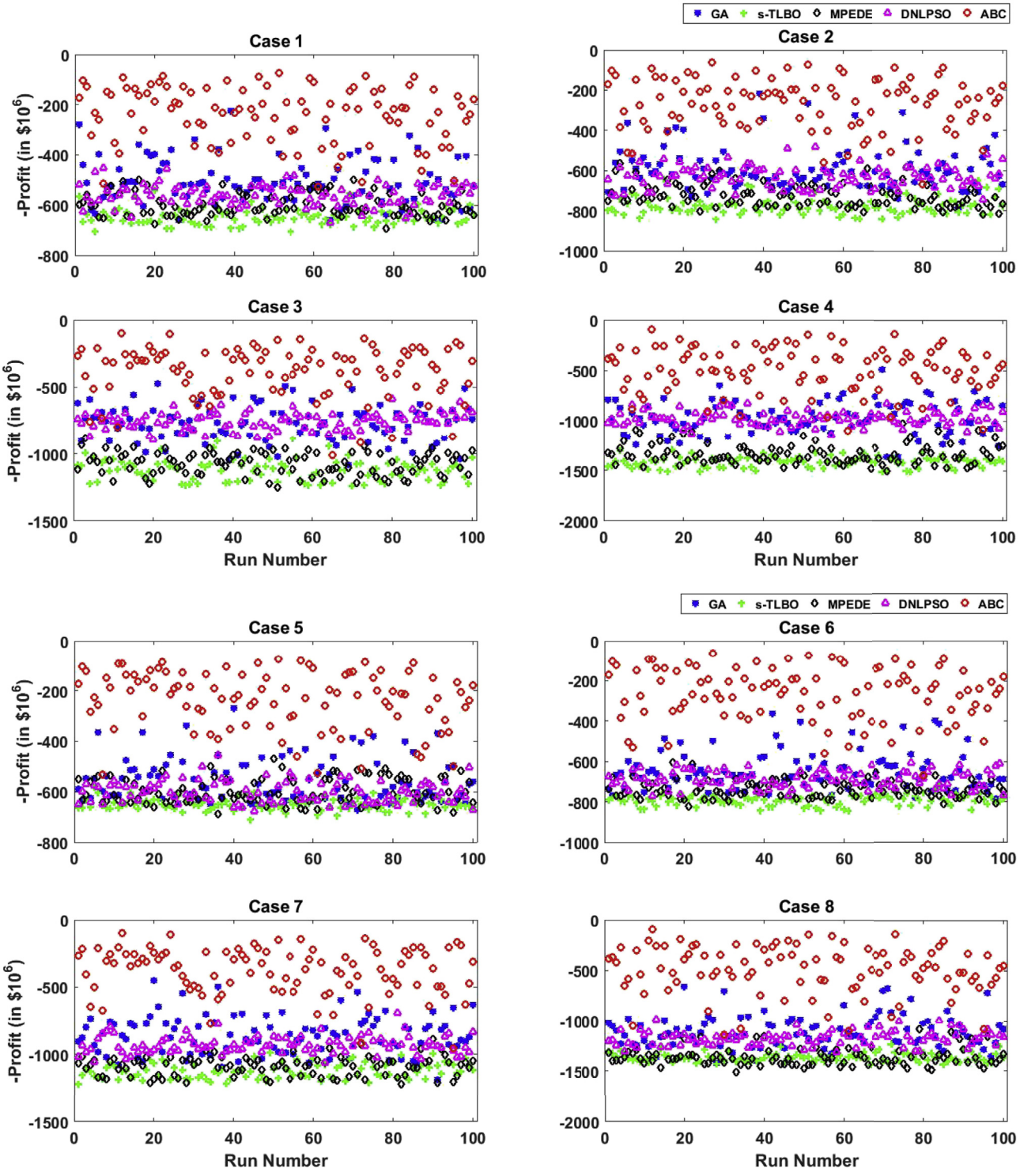

Fig. 5. Performance of the five algorithms with initial guess.

$$C_j = \begin{cases} \sum_{l=1}^{L} N_{lj} C_{lj} + \sum_{l=1}^{L-1} \left(C_{lj} + \left(\frac{C_{l+1j} - C_{lj}}{c_{l+1j} - c_{lj}} \right) (x_{lj} - c_{lj}) \right) & ; c_{lj} \le x_{lj} \le c_{l+1j} \\ \sum_{l=1}^{L} N_{lj} C_{lj} & ; x_{lj} = 0 \end{cases} \quad \forall j = 1, 2, \cdots, J \tag{21}$$

The profit from the production plan can be determined using Equation (22) in which E_j denotes the selling price of the product from process j,

$$Maximize \quad \text{Profit} = \left(\sum_{j=1}^{J} E_j X_j - \sum_{j=1}^{J} C_j \right) \tag{22}$$

In order to accommodate the constraints, the fitness function is designed as shown in Equation (23) which considers the objective function in Equation (22) and the penalty for violation shown in constraints through Equation (16), Equation (19) and Equation (20).

$$Max \quad = \text{Profit} - \left(f_p \left(\sum_{t=1}^{T} P_t^{raw} + P^{Investment} + \sum_{i=1}^{I} P_i^{unique} \right) \right) \tag{23}$$

In view of the piece-wise linear cost function, the proposed strategy leads to fewer variables rather than employing multiple continuous variables and also enables the discovery of better solutions [27]. In Equation (23), f_p is a penalty factor with a high value in order to ensure that feasibility is given more importance than the value of profit. The objective function in Equation (23) ensures that (i) a feasible solution will have a better fitness function value than an infeasible solution, (ii) between two feasible solutions, the solution with greater profit will have a better function value, and (iii) between two infeasible solutions, the solution with lower violation has a better fitness function value. We have used a static penalty as the primary aim of this article is to show the benefits of using multiple units and multiple levels and study the performance of the various algorithms on this problem. An interested reader can study the performance of other constraint handling techniques including adaptive penalties that have been proposed in the literature [26]. The p codes for evaluating the objective function are available at https://goo.gl/H49Crh.

Table 3
Performance of the proposed strategy with s-TLBO.

Case	Items	Amount of resource available	MILP Literature [1]	Multi-Level Strategy with TLBO [2]	Proposed Multi-Unit Strategy with s-TLBO	% Increase in Profit
Case 1	B ($ 10^6)	1000	1000	1000	995.8	4.32
	R_1 (10^3 tons/year)	500	500	500	366.7	
	R_2 (10^3 tons/year)	500	500	500	488.5	
	Profit ($ 10^6/year)	-	**692.8**	**692.8**	**722.7**	
Case 2	B ($ 10^6)	1000	1000	1000	997.1	7.05
	R_1 (10^3 tons/year)	1000	847.2	971.8	882.5	
	R_2 (10^3 tons/year)	1000	660.5	571.0	999.7	
	Profit ($ 10^6/year)	-	**759.7**	**796.5**	**852.6**	
Case 3	B ($ 10^6)	2000	1952	1999.8	1998.2	34.66
	R_1 (10^3 tons/year)	500	500	500	432.6	
	R_2 (10^3 tons/year)	500	500	500	499.8	
	Profit ($ 10^6/year)	-	**894.3**	**953.8**	**1284.3**	
Case 4	B ($ 10^6)	2000	1989.3	2000	1994.7	25.67
	R_1 (10^3 tons/year)	1000	1000	975.1	983.1	
	R_2 (10^3 tons/year)	1000	979.5	979.5	966.6	
	Profit ($ 10^6/year)	-	**1111.5**	**1202.2**	**1510.8**	
Case 5	B ($ 10^6)	1000	993	990.2	991.2	0.63
	R_1 (10^3 tons/year)	500	500	500	494.53	
	R_2 (10^3 tons/year)	500	500	500	499.54	
	Profit ($ 10^6/year)	-	**726.0**	**730.1**	**734.70**	
Case 6	B ($ 10^6)	1000	1000	1000	993	1.19
	R_1 (10^3 tons/year)	1000	1000	1000	882.57	
	R_2 (10^3 tons/year)	1000	571.0	571.0	950.20	
	Profit ($ 10^6/year)	-	**834.3**	**834.3**	**844.3**	
Case 7	B ($ 10^6)	2000	1991.7	1985.3	1989.7	6.78
	R_1 (10^3 tons/year)	500	500	500	495.43	
	R_2 (10^3 tons/year)	500	500	500	487.20	
	Profit ($ 10^6/year)	-	**1173.1**	**1177.3**	**1257.14**	
Case 8	B ($ 10^6)	2000	1998.1	1998.1	1946.6	1.96
	R_1 (10^3 tons/year)	1000	1000	1000	954.60	
	R_2 (10^3 tons/year)	1000	954.6	954.6	969.88	
	Profit ($ 10^6/year)	-	**1452.8**	**1452.8**	**1481.3**	

Table 4
Optimal production plans determined using the proposed strategy with s-TLBO.

Case	Product number	Process number	Decision variables					Production	Net production (ton/yr)
	N_T	N_J	N_{1j}	N_{2j}	N_{3j}	x_{1j}	x_{2j}	X_j	
Case 1	T1	S3	0	0	0	0	310	310	310
	T5	S9	0	0	0	0	115.3	115.3	115.3
	T21	S48	0	1	1	0	588.5	1718.59	1718.59
Case 2	T1	S3	0	0	2	0	310	930	930
	T15	S31	1	0	1	0	352.4	852.4	852.4
	T21	S48	0	0	0	0	680	680	680
Case 3	T1	S3	0	0	0	0	310	310	310
	T2	S4	0	1	0	0	0	145	145
	T17	S36	0	0	5	0	0	2700	2700
	T21	S48	0	1	1	0	628.2	1758.2	1758.2
Case 4	T1	S3	0	0	3	0	0	930	930
	T5	S9	0	0	0	0	160	160	160
	T19	S41	0	0	1	0	0	50	50
	T21	S48	0	0	5	0	0	3400	3400
Case 5	T1	S1	0	0	0	0	211.3	211.3	521.3
		S3	0	0	0	0	310	310	
	T19	S41	0	0	0	0	50	50	50
	T21	S48	0	0	1	397	680	1757.1	1757.1
Case 6	T1	S3	0	0	2	0	310	930	930
	T15	S31	0	0	1	0	400	800	800
	T19	S41	0	0	1	0	0	50	50
	T21	S48	0	0	0	0	680	680	680
Case 7	T1	S3	0	0	0	0	310	310	310
	T5	S9	0	0	1	0	160	320	320
	T10	S22	0	0	0	0	100	100	100
	T17	S36	0	0	2	0	540	1620	1620
	T21	S46	0	0	0	0	680	680	1702.2
		S48	0	0	0	450	572.2	1022.2	
Case 8	T1	S2	0	0	0	0	300	300	610
		S3	0	0	0	0	310	310	
	T2	S4	0	0	0	0	290	290	290
	T5	S9	0	0	0	0	160	160	160
	T21	S46	0	0	0	0	680	680	3400
		S48	0	0	3	0	680	2720	

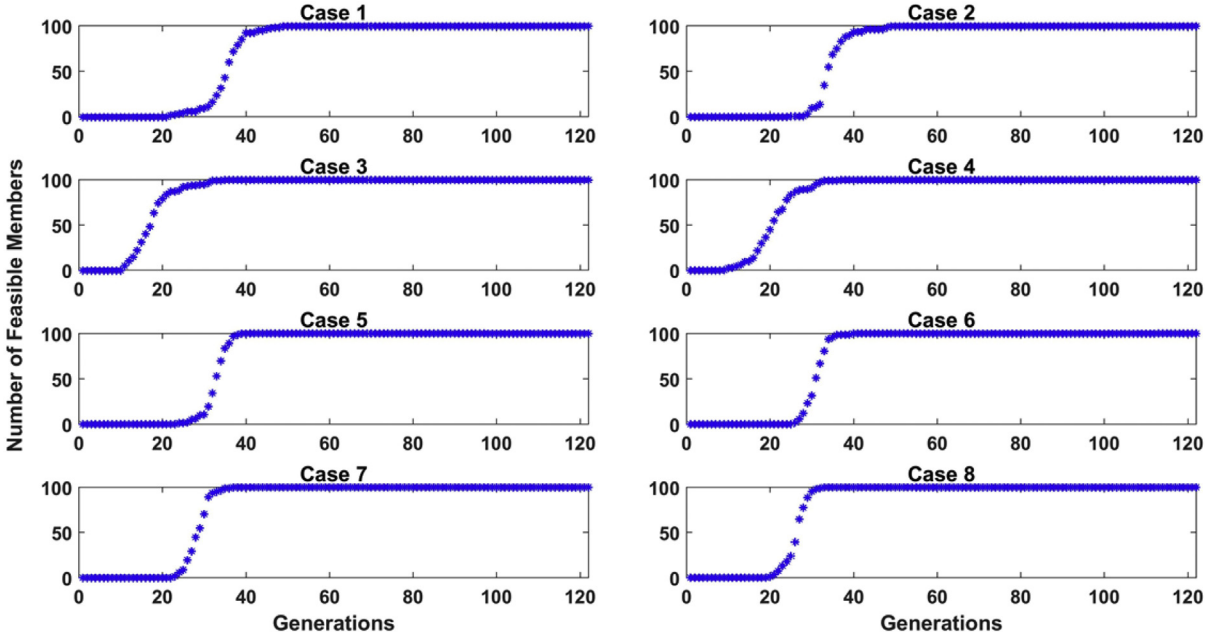

Fig. 6. Number of feasible solutions in each generation.

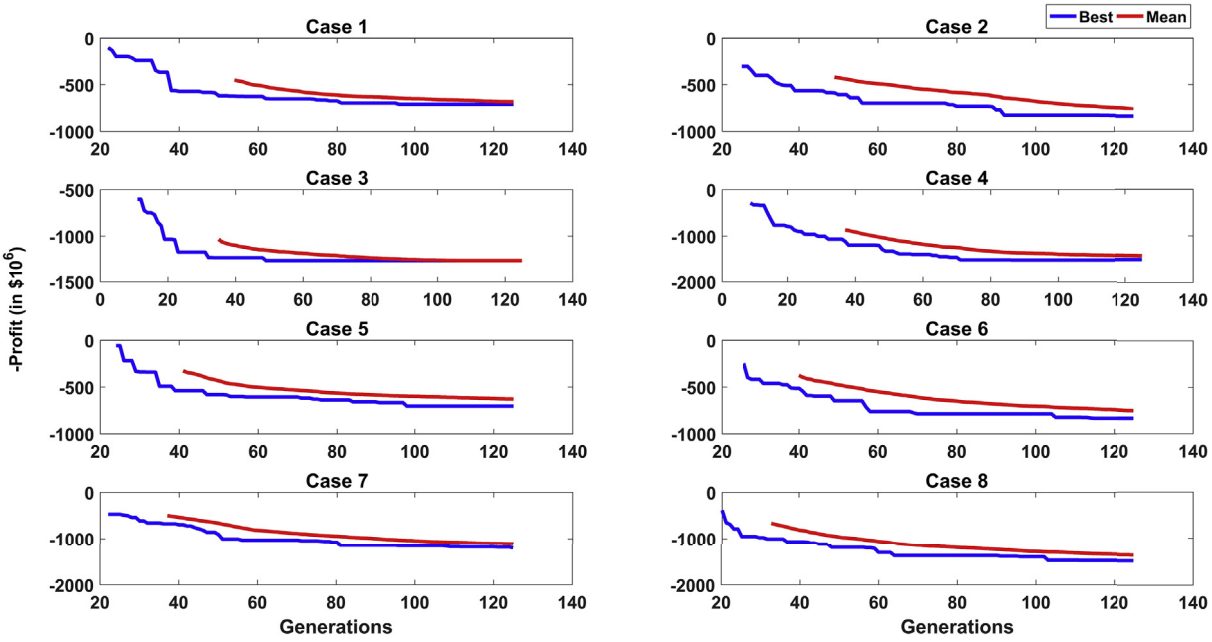

Fig. 7. Performance of the s-TLBO algorithm for the multi-unit based strategy without initial guess.

4. Brief description of five evolutionary techniques

Evolutionary algorithms exhibit several advantages over mathematical programming techniques as these do not require the optimization problem to be explicitly expressed in terms of the conventional equalities and inequalities. This enables the use of these techniques with comparatively low modelling effort whereas the population-based nature of these algorithms helps to possibly overcome issues with local optima. In most instances, these techniques do not require any artificial binary variable to transform the problem into a set of exact equalities and inequalities and thus do not exhibit substantial computational issues with scalability. However, due to their stochastic nature, all these five methods require multiple runs of the problem. For the sake of brevity, we have not provided the detailed description of the five algorithms but restrict with a brief discussion. An interested reader can obtain additional details from the cited literature.

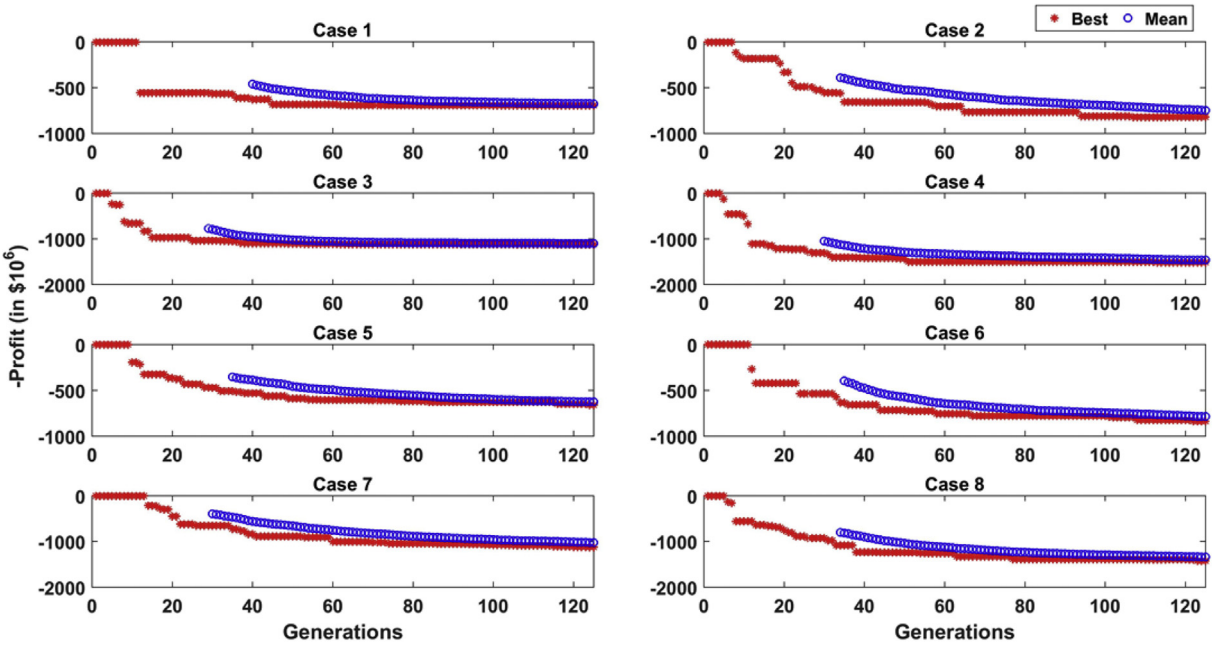

Fig. 8. Performance of the s-TLBO algorithm for the multi-unit based strategy with initial guess.

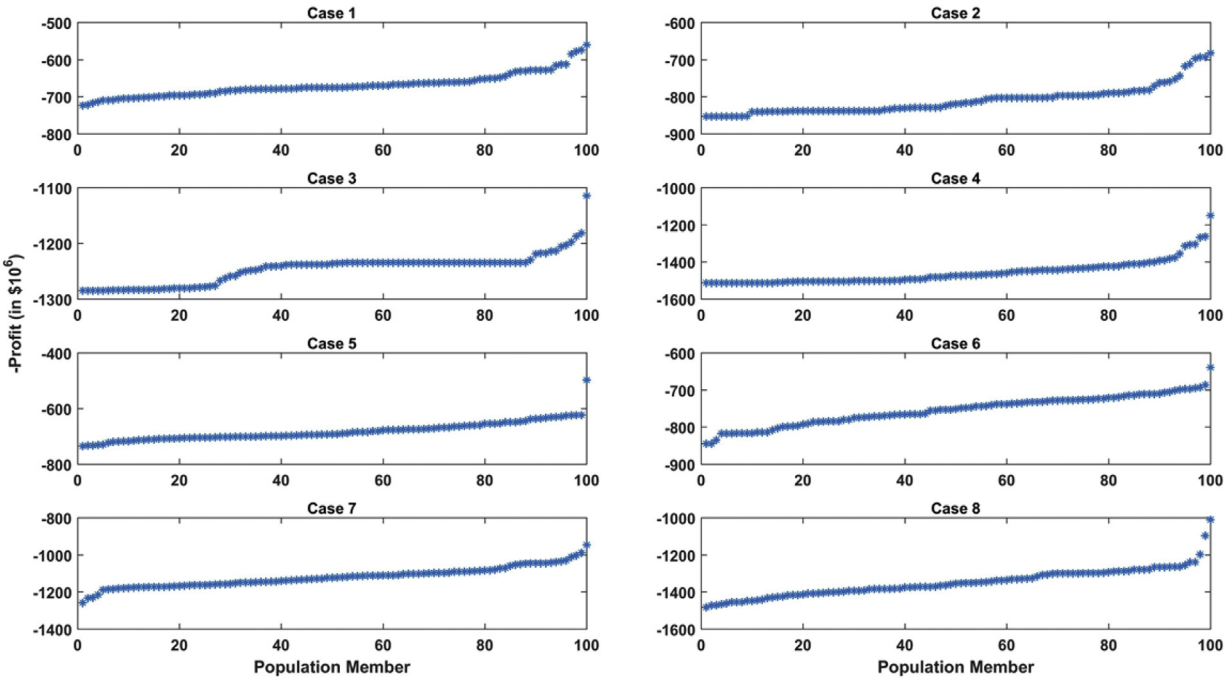

Fig. 9. Fitness of the population members in the last generation.

4.1. Genetic algorithm

Genetic Algorithm is a stochastic population-based optimization technique that has continuously evolved since the early 70s [28] and is based on the Darwin's theory of evolution. GA employs specialized selection operators to select population members (parents) that would be used to generate potential solutions. Subsequently, variation operators such as crossover are used to combine the parent solutions whereas mutation operators are employed to modify the individual solutions to generate the offspring solutions. Several strategies are available in literature to select the population members for a subsequent generation from the parent and offspring solutions. A detailed review of GA is available in literature [28]. TLBO suffers from various drawbacks including additional computational burden for removal of duplicates. In this work, sanitized TLBO has been used which does not incorporate duplicate removal and overcomes the drawbacks of TLBO [27]. The code for Sanitized TLBO is available at https://goo.gl/vk8gLb.

4.2. Sanitized-teaching-learning-based optimization

TLBO is a stochastic population-based method that has been proposed multiple times since 2011 [29]. TLBO is inspired from the classroom

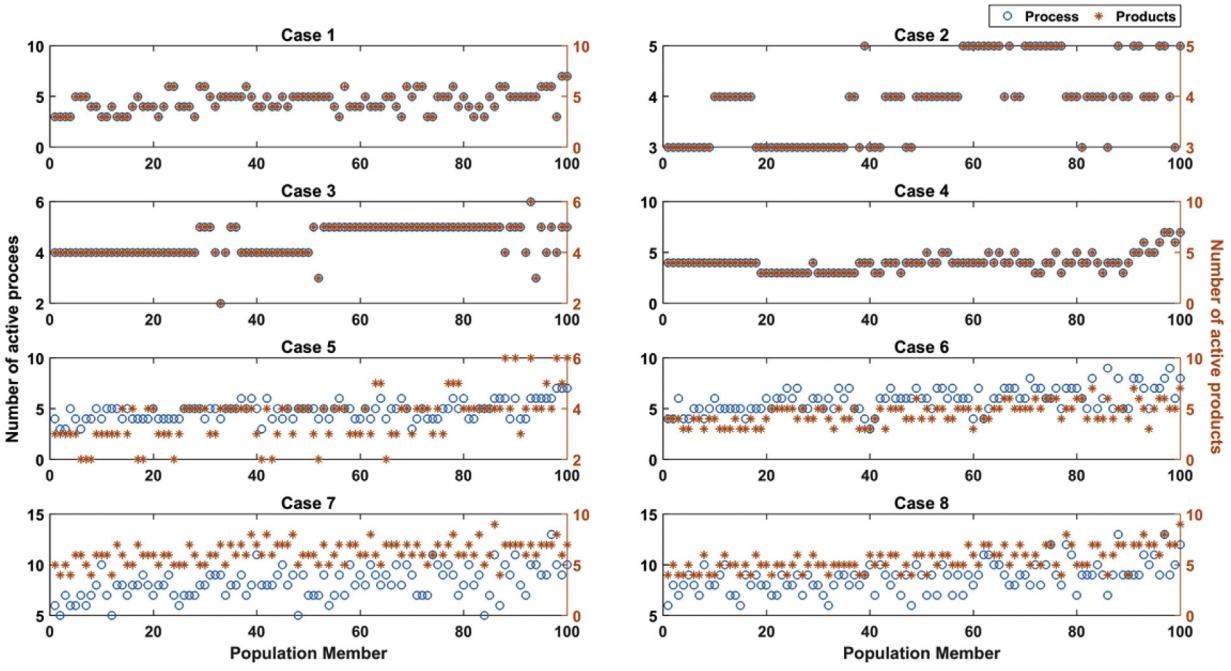

Fig. 10. Number of active process and active product corresponding to the population members in last generation.

teaching-learning process where students learn from the teacher (modelled as Teachers Phase) as well from interacting with other classmates (modelled as Learners Phase). TLBO does not require any algorithm specific controlling parameters except for the size of the initial population and the termination criterion. For every member of the class, in each generation, a potential solution is generated in each of the two phases. The new solution replaces the old solution if its fitness is better than the solutions used to generate it. The solution in the teacher phase is generated using the best student as well as the mean of the students whereas the solution in the learner phase is generated using a randomly selected member from the class. The lack of user defined parameters (except population size and the termination criteria) has been claimed [29,30] as one of the major benefits and has potentially contributed to its recent widespread use [31,32].

4.3. Multi-population based ensemble differential evolution algorithm

MPEDE is a novel differential evolution variant proposed in 2015 [33] and employs an ensemble approach to overcome issues related to the selection of appropriate mutation strategy in the conventional Differential Evolution technique. MPEDE simultaneously accommodates three mutation strategies, "current-to-pbest/1," "current-to-rand/1" and "rand/1" by assigning an indicator population to each of the mutation strategy. The performances of the mutation strategies are periodically evaluated and the reward population, relatively larger number of members than the individual indicator population, is assigned to the best performing mutation strategy. The mutation strategies "current-to-pbest/1" and "rand/1" use a binomial crossover whereas "current-to-rand/1" does not use any crossover. MPEDE also employs the adaptive strategy proposed as part of adaptive differential evolution to tune the crossover probability and the mutation factor [34]. MPEDE has been reported to outperform several other variants such as JADE, jDE, SaDE, EPSDE, CoDE and SHADE.

4.4. Enhanced binary particle swarm optimization algorithm

BPSO is a variant of PSO that has been proposed to solve problems combinatorial optimization problems and employs transfer functions to map the continuous search space to the discrete search space. The transfer function should be such that a large absolute value of the velocity should provide a high probability of changing the position whereas a small absolute value of the velocity should provide a small probability of changing the position. Similarly, the return value of a transfer function should increase with an increase in velocity and vice versa. A study of eight different transfer functions comprising of two families of s-shaped and v-shaped, was reported in 2011 [35]. For the production-planning problem, it was observed that the v-shaped transfer (referred as V1 and BPSO5 in Ref. [35]) was able to perform relatively better in determining feasible solutions.

4.5. Artificial bee colony algorithm

ABC is a population-based algorithm that has been proposed in 2005 [36] and is motivated by the foraging behaviour of honeybees. ABC consists of two essential components (food sources and foragers), and three main phases *viz.*, employed bee phase, onlooker bee phase and scout phase. Employed bees independently explore the nectar sources and provide the information to the onlooker bees about the quality of food source sites. Scout bees search for new food sources randomly with the help of internal motivations or external clues. The scout phase helps in the exploration whereas the other two phases help in the exploitation of the search space. Several issues in the implementation of ABC have been described in the literature [37]. A brief comparison of these five algorithms is given in Table 1. It should be noted that all the five algorithms selected for this work exhibit monotonic convergence, i.e., the best solution obtained in a particular generation is replaced if and only if a better solution than the current best solution is obtained.

5. Case study

In this section, the performance of the proposed multi-unit production strategy and the strategy to handle domain hole constraints is demonstrated on a production planning problem case study from literature [1]. In addition, the proposed strategy will be employed with five optimization algorithms to provide an opportunity to compare the

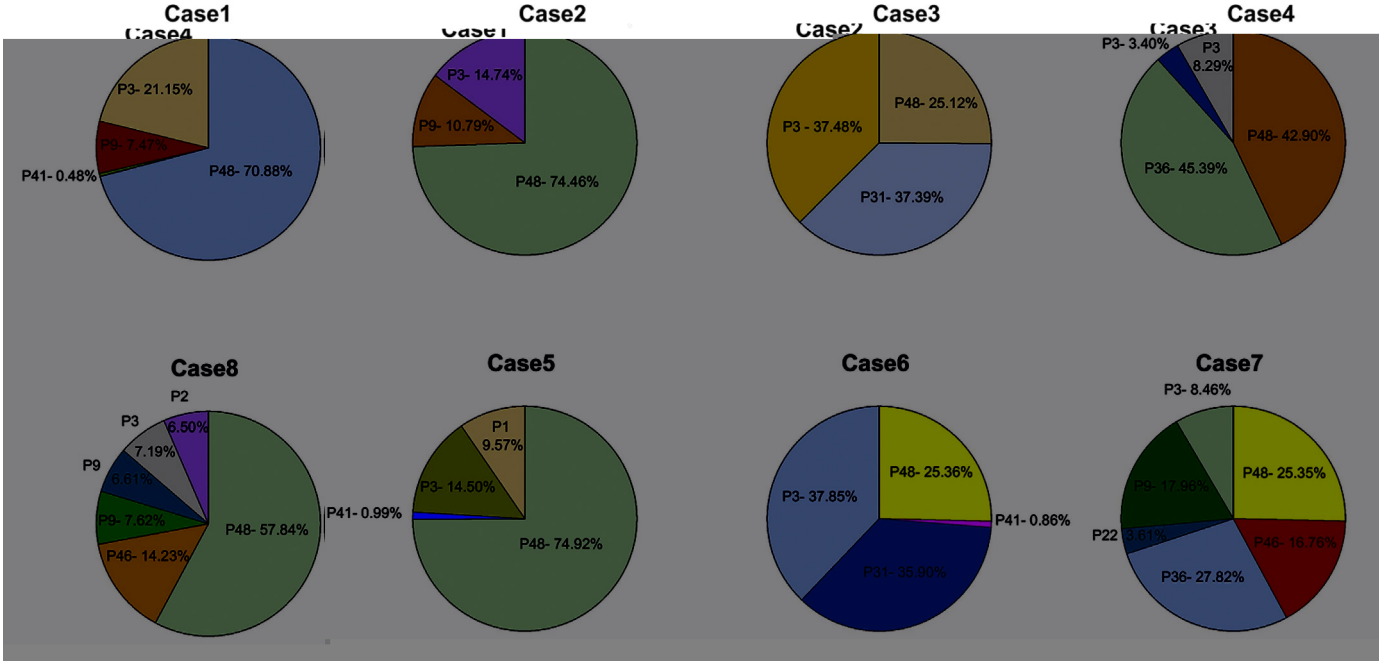

Fig. 11. Contribution of various processes to the profit.

performance of these algorithms. In this case study, the objective is to design a production plan to maximize profit subject to unique process constraints, raw material constraints and the constraint on the investment cost in addition to the domain hole of the decision variables and the piecewise functions of production and investment cost. The production-planning problem involves the possible production of 24 products (T1 to T24) from a set of 54 processes (S1 to S54) with three different production capacities. Thus, this problem will have 162 integer variables and 108 continuous variables. Table A1–Table A4 provide the necessary data such as selling price of the products, raw material consumption, and the production capacities of the three levels along with their production and investment cost. It can be observed from the tables that there are seven products (T4, T5, T8, T12, T13, T16, and T23) which can be produced by a single process whereas there are eight other products (T2, T3, T7, T9, T10, T19, T22, and T24) which can be produced by two processes. Three different processes are available to produce another seven products (T1, T11, T14, T15, T17, T18, and T20) whereas the product T21 and T6 can be produced from four and six processes respectively.

In accordance with literature [1], we have demonstrated the proposed strategy on eight case studies. Case 1 – Case 4 require that the production of a product be only from a single process whereas Case 5 – Case 8 do not enforce this constraint. In Case 1 – Case 4 (also in Case 5 – Case 8), the amount of raw materials that are available and the amount of the budget vary. In Case 1 and Case 2, the investment budget is restricted to $1 billion whereas in Case 3 and Case 4, it is restricted to $2 billion. In Case 1 and Case 3, the raw material availability is 0.5 million tons/year whereas in Case 2 and Case 4, it is 1 million tons/year. These cases help in understanding the impact of various resources on the profitability of the production plan.

The inbuilt GA function in MATLAB is used for demonstrating the performance of GA but it permits only the transfer of objective function value(s) from the objective function file to the genetic algorithm and does not seem to permit the transfer of a corrected population member to the genetic algorithm. This creates an impediment in implementing the correction approach for handling the domain hole constraint using the correction approach shown in equation (13). In order to overcome this drawback, a simple tweak (as suggested by Mathworks) is employed as shown in the following algorithm, which exploits the fact that a user can provide the initial population to the GA function in MATLAB.

Step 1: Create the initial population between the bounds of the decision variables.
Step 2: Run the inbuilt GA function for a single generation (with the correction approach implemented in the objective function file).
Step 3: Correct the population members obtained from Step 2 to satisfy the domain hole constraint. Use this population as an initial population set for the next generation.
Step 4: Go to Step 2 till the termination criteria of the specified number of generations (determined from maximum permissible functional evaluations) is satisfied.

For the rest of the discussion, the results of GA are those obtained from the inbuilt GA function in MATLAB with elite count is 0.05*(population size) i.e. 5, crossover fraction is 0.8, migration interval is 20, with a population size of 100 and the stopping criteria is set to 25100 functional evaluations. We have used the default setting values of the tuning parameters (except the population size and termination criteria) for MATLAB GA [46]. The default settings of MATLAB GA are as follows: the value of migration fraction is 0.20, the initial value of the penalty parameter is 10, penalty factor is 100, selection operator is stochastic universal sampling, crossover operator is position independent crossover and mutation operator is the adaptive mutation. Similarly, for s-TLBO, MPEDE, BPSO and ABC, the number of population members is set to 100. The stopping criterion for all the runs in each of these algorithms is set to 25100 functional evaluations so that 100 functional evaluations are employed for the initial population and the remaining 25000 functional evaluations are employed for the algorithm. The simulations are run on PC using MATLAB R2015a with Intel i7-4790 CPU@ 3.60 GHz processor and 12 GB of RAM. Due to the stochastic nature of evolutionary algorithms, we run each case 100 times with a different seed. The 100 independent runs are realized by varying the seed of the *twister* algorithm in MATLAB R2015a from 1 to 100.

5.1. Comparative analysis of GA, s-TLBO, MPEDE, BPSO and ABC

In order to evaluate the performance of these five algorithms, each algorithm was executed for 100 different random seeds for all the eight cases. Table 2a shows (i) the best value of the fitness function, (ii) the worst of the best fitness function values, (iii) the mean of the best fitness function values, (iv) the median of the best fitness function values and (v) the standard deviation of the best fitness function values obtained in all the runs thereby helping us to evaluate the performance of all the algorithms. In the absence of any initial guess, from Table 2a, it can be observed that among 4000 scenarios (8 Cases x 100 runs x 5 algorithms), s-TLBO and BPSO are able to identify optimal solutions for all the eight cases whereas GA and ABC are not able to determine even a single feasible solution for any of the eight cases. It can also be observed from Table 2a that MPEDE is able to determine the feasible solution for all the cases (Case 5 - Case 8) which do not have the unique process requirement. It can be observed that for all the eight cases, the best solution discovered by s-TLBO is better than all other algorithms. Moreover, the worst, mean, median, and standard deviation values determined by s-TLBO are better than all other algorithms for all the eight cases. This consistent performance establishes the supremacy of s-TLBO for solving the production planning problem discussed in this article. The fitness function values obtained by all the algorithms for each run of the cases are shown in Fig. 4. In order to provide better visualization, only the runs in which a feasible solution is obtained are plotted. It can be seen that s-TLBO is able to determine feasible solutions in all the runs whereas BPSO fails to determine a feasible solution for 96 instances in Case 1, (95, 79, 59, 18, 10, 1 in Case 2 to Case 7 respectively), but is able to determine feasible solutions for all the instances in Case 8. However, GA and ABC were not able to determine even a single feasible solution for any of the cases. MPEDE, was able to obtain feasible solutions for 396 instances (without unique process constraints cases) out of 800 instances.

In the case of some real-life problems, it is easy to determine a feasible solution. A trivial feasible solution for this case study is not to produce any product (leading to a zero profit). This solution (assigning a value of zero for all 270 variables) can be used as an initial guess by providing it as a member of the initial population. This will ensure that GA and ABC, which are not able to determine even a feasible solution, would start with a feasible initial solution. On the other hand, MPEDE, BPSO, and s-TLBO are also able to determine solutions with better fitness values as compared to those obtained without using an initial guess. All the 4000 scenarios discussed were executed once again but with zeroes as initial guess (i.e., one population member is a vector of zeros). The statistical results of all the five algorithms are provided in Table 2b. With the initial guess, it can be observed that for all the eight cases, all the five algorithms are able to determine improved solutions. It can be seen that s-TLBO is able to obtain better best solutions in six cases (Case 1, Case 2, Case 4, Case 5, Case 6 and Case 7) than other four algorithms whereas MPEDE is able to obtain better best function values than the other four algorithms for Case 3 and Case 8. However, the worst values reported by s-TLBO in some cases are better than the best values obtained by ABC (Case 1, Case 4, Case 5, Case 6, Case 7 and Case 8) and BPSO (Case 4). From all the 16 unique instances for each algorithms (8 cases without initial guess, 8 cases with initial guess), s-TLBO is able to obtain better statistical results for 14 instances out of 16 instances than other four algorithms. Thus even in the presence of a trivial

initial guess, GA, ABC, BPSO and MPEDE are not able to outperform s-TLBO for this production planning problem. No instability was experienced in any of the 8000 scenarios (8 Cases x 100 runs x 5 algorithms) that have been executed for this work.

The best fitness value obtained in all the 100 runs for each of the case is shown in Fig. 5 for all the five algorithms. Due to the high penalty in the fitness functions for constraint violation, only the values that are obtained in the feasible runs are depicted in Fig. 5. It can be observed that in the majority of the individual instances, s-TLBO is able to outperform the other four algorithms except in Case 7 and Case 8. However for Case 7 there are 3 instances wherein MPEDE is able to determine a solution better than s-TLBO, and for Case 8, there are 22 instances where s-TLBO does not obtain better results than MPEDE algorithm. However on an overall basis, s-TLBO is able to outperform all the four algorithms, both in the presence and in absence of any initial guess. Thus the performance evaluation of five algorithms are studied for the production planning problem using 8000 (= 8 Cases x 100 runs x 2 set of guess x 5 algorithms) unique instances.

5.2. Performance of the proposed strategy

In the above discussion, it was demonstrated that s-TLBO is able to determine better solutions than GA, MPEDE, BPSO, and ABC algorithm for the production-planning problem. In this section, we use s-TLBO to demonstrate the benefits of the proposed multi-unit strategy. For this, we have used the solution in literature [2] as an initial guess as the objective is to demonstrate the utility of the proposed multi-unit formulation. We have used the same experimental settings (population size 100 and 25100 functional evaluations). Table 3 compares the resource utilization and the profit for all the cases from multi-unit strategy with the previous works in literature [1,2,27]. Table 4 provides the detailed production plan obtained for all the eight cases.

Fig. 6 shows the number of feasible solutions obtained in each generation of s-TLBO for the respective best run in each case. It can be observed that initially there are no feasible solutions and there is a phase wherein the number of feasible solutions increases significantly. Figs. 7 and 8 shows the performance of the s-TLBO for the multi-unit strategy in terms of the average fitness value of the entire population and also the best fitness function value in every generation. For the sake of better visualization, the best fitness function value is plotted after at least a single member becomes feasible and the average fitness value is plotted when the entire population becomes feasible. For Case 1 – Case 8, it can be seen that s-TLBO is able to determine a feasible solution in less than 26 generations and the entire population becomes feasible in less than 50 generations. Also in most instances, the average fitness function continues to improve even after the final solution is obtained. This shows that the rest of the population members keep improving and are trying to reach the optimal solution.

Fig. 9 shows the value of the objective function corresponding to each member of the population in the final generation. Fig. 10 shows the number of processes and the number of products corresponding to each member of the population member in the final generation. Figs. 9 and 10 are also indicative of the flexibility provided by the stochastic algorithms. From Fig. 9, it can be observed that some of the other solutions have an objective function value closer to the optimal solution and it can also be inferred (from Fig. 10) that many of these solutions have different number of processes and products thereby providing flexibility to the user to select production plans as per their needs. In Case 1 – Case 4, the unique process requirement is to be satisfied and hence the number of products and processes are identical in Fig. 10. However, this is not true for Case 5 – Case 8 as the unique process constraint is not enforced in these cases.

From Table 4 it can be seen that for all the eight cases, the proposed multi-unit strategy is able to determine better solutions than those currently reported in literature [1,2]. The maximum increase in profit can be seen in Case 3 whereas the minimum improvement in profit is in Case 5. It can be seen from Table 4 that the proposed multi-unit strategy using s-TLBO is able to determine a better solution for Case 1 (also in Case 3) despite using lower monetary and raw material resources. From Table 4, it can be observed that in Case 1 – Case 4 which requires the satisfaction of the unique process constraint, the products are produced by a single process showing the efficacy of the s-TLBO algorithm in handling the unique process constraints. It can also be seen from Table 4 that all the variables satisfy the domain hole constraint as all the non-zero continuous decision variables are between the low and high-level production capacities.

The improvement in the profit for the multi-unit strategy can be understood by analysing the production plans in Table 4. It can be seen that for Case 1, the amount of product T21 that is produced is greater than the sum of the medium and high-level capacities. This is because the proposed multi-unit strategy permits the use of multiple units. For example, T21 is produced by two units of high-level capacity which would not be possible for the formulation/strategies in the literature [1,2] as these artificially restrict the production and do not accommodate the use of multiple units. Similar phenomenon can be observed for Case 2 (T1 and T15), Case 3 (T17 and T21), Case 4 (T1 and T21), Case 5 (T21), Case 6 (T1 and T15), Case 7 (T17 and T21) and Case 8 (T2 and T21). It can be observed for Case 3 from Table 4 that five units are employed for the production of Product T17 from Process S36 and hence leads to a substantially higher improvement in the profit. Similarly for Case 4 it can be observed that five units are employed for the production of Product T21 from Process S48. The discovery of such solutions is possible because the proposed strategy is designed to accommodate production from multiple units.

Fig. 11 shows the contribution of each process to the profit and it can be seen that for most of the cases, the product T21 (produced by process S46, S47,S48) contributes significantly towards the profit and at times it is even greater than 70%. Thus in this section, we have demonstrated the benefits of the multi-unit strategy for production planning using s-TLBO and have also demonstrated the suitability of s-TLBO over GA, MPEDE, BPSO and ABC using 4000 unique instances.

6. Conclusion

In this article, we review the MILP formulation and the elitist TLBO based multi-level strategy modelled for guiding petrochemical industries in Saudi Arabia to determine the optimal production plan and discuss its various limitations. We have proposed an alternate strategy that accommodates multiple units and multiple levels of production to overcome the drawbacks and strategies in literature [1,2,25,27]. We have also utilized an efficient way to handle the domain hole constraints that corrects a potential solution for violation instead of assigning penalty to the objective function. The proposed strategy is able to determine efficient production plans with improved utilization of resources. The benefits of proposed strategy are validated with the eight cases from the literature [2] and is able to determine production plans that have increase profits up to 34.66%. The proposed strategy is generic and can be easily implemented in various industries for determining production plan. In addition, we have compared the performance of the s-TLBO to four other algorithms viz., MPEDE, GA, BPSO, and ABC. For this production planning problem involving more than 150 variables and complex unique process constraints, it is observed that s-TLBO is able to outperform the other four algorithms. Due to the inherent stochastic nature of these algorithms, one needs to exercise caution in generalizing the conclusion that s-TLBO is better than the other algorithms for all kinds of optimization problems. Future work in this direction can include the development of a mathematical programming model and inclusion of multiple objectives such as safety and environmental impacts to provide a comprehensive solution for the production planning problem.

Appendix A

Table A1
Data for propylene products and processes [1].

Product	Product no. i	Sale price	Process no. j	Process name	Propylene used/ton b_{j2}	Capacity (10^3 ton/yr)			Production cost ($10^6/yr)			Investment ($10^6)		
						l_j	m_j	h_j	C_{lj}	C_{mj}	C_{hj}	V_{lj}	V_{mj}	V_{hj}
Polypropylene copolymer	T1	975	S1	Amoco/Chisso	0.9480	70	135	270	50.7	90.1	170.7	55	81.1	131.6
			S2	BASF	0.9432	75	150	300	56.8	103.8	196.2	58	85.1	132.4
			S3	Himont	0.9490	77.5	155	310	56.9	103.7	195.7	60.2	86.8	134.1
Polypropylene block copolymer	T2	975	S4	Sumitomo (gas phase)	0.9546	70	145	290	51.7	97.6	184.8	55.1	83.1	132
			S5	UCC/Shell	0.9550	47.5	95	190	38.2	69.8	130.4	43.3	66.8	104.3
Polypropylene homopolymer	T3	780	S6	Borealis	1.0450	40	80	160	38.5	65.2	120.7	66.2	92.8	153.2
			S7	UCC/Shell	1.0500	40	80	160	31.8	57.1	105.5	40	61.4	95.1
Phenol	T4	735	S8	From C_6H_6/C_3H_8 via cumene	0.5103	45	90	180	37.8	57.7	94.9	106.6	151.7	231.5
Acrylic acid (ester grade)	T5	1450	S9	Two-stage oxidation	0.6289	40	80	160	38.5	65.6	119.1	82.8	125.4	207
Propylene oxide	T6	1130	S10	Arco process (styrene product)	0.8648	90	180	360	92.2	159.2	290.9	233.5	390.7	698.7
			S11	Texaco (T butanolbyproduct)	0.9546	90	180	360	86.7	154.1	287.7	185.8	304.5	537.1
			S12	Chlorohydrin	0.8265	90	180	360	95.8	175	330.9	119	179.4	289.2
			S13	Acro process (T butanolbyproduct)	0.7875	90	180	360	87.5	157.2	294.9	212.3	362.7	657.7
			S14	Cell liquor neutralization	0.8101	90	180	360	105.9	196.6	375.2	109.8	164.3	263.1
			S15	Shell process (styrene byproduct)	0.8782	90	180	360	93.1	131.1	239.4	221.7	376.1	672.7
N-butanol	T7	830	S16	Via Cobalt hydrocarbonyl catalyst	0.8150	50	100	200	41.4	68.7	117.2	115.5	180.4	287.4
			S17	Via N butryaldehyde Rh catalyst	0.6994	50	100	200	34.9	62	111.6	63.7	100.2	156.3
Cumene	T8	450	S18	From C_6H_6 and Propylene	0.3784	60	120	240	36.6	62.1	120.8	23.1	33.2	50.7

Table A2
Data for ethylene products and processes [1].

Product	Product no. i	Sale price	Process no. j	Process name	Propylene used/ton b_{j1}	Capacity (10^3 ton/yr)			Production cost ($10^6/yr)			Investment ($10^6)		
						l_j	m_j	h_j	C_{lj}	C_{mj}	C_{hj}	V_{lj}	V_{mj}	V_{hj}
Poly vinyl Chloride	T9	740	S19	Suspension polymerization		100	200	400	67.6	125.2	237.2	117.6	186	307.5
			S20	Bulk polymerization		50	100	300	33	63.1	163.8	62.5	114	209.6
Poly vinyl Chloride (dispersion)	T10	1250	S21	Batch emulsion polymerization		25	50	100	28.7	48.3	86	73.1	101.1	148
			S22	Continuous emulsion polymerization		25	50	100	24	43.1	79.5	46.5	70.7	110.1
Vinyl chloride	T11	430	S23	TOSOH technology		125	250	500	63.8	123.5	241	49.2	74.4	112.8
			S24	Pyrolysis		125	250	500	68.5	134.5	264	79.1	144.2	258.1
			S25	Chlorination/Oxychlorination	0.4678	250	500	1000	101.5	195	377	134	229.9	392.2
Ethylene glycol	T12	600	S26	Hydration of EO all EO for EG	0.7267	90	180	360	50.3	90	165.6	142.6	234.8	397.5
Vinyl acetate	T13	690	S27	From ethylene and acetic acid	0.3930	67.5	135	200	53.9	101.2	146.4	82.7	133.6	181.3
High density polyethylene	T14	860	S28	UCC process	1.0200	70	135	270	42.1	75.1	141.8	56.9	84.5	131.5
			S29	Du Pont process	1.0200	70	135	270	44.6	77.5	147.7	63.4	84.5	136.9
			S30	Philips process	1.0200	70	135	270	44.6	78.8	148	66.5	96.2	147.7
Linear low density polyethylene	T15	900	S31	Dry mode gas phase univation process	0.9461	100	200	400	55.7	106.8	208.4	51.4	83.0	144.5
			S32	Bimodal grade by mixed mettallocene/Ziegler catalyst	0.9387	75	150	300	48.3	90.2	172.8	46.9	66	98.6
			S33	Bimodal grade by unipol process	0.9430	122.5	245	490	92	174	336.2	82.4	116.6	175.6
Low density polyethylene	T16	870	S34	High pressure tubular reactor	1.0600	50	100	200	34.9	63.9	120.4	72	117.7	199.7

Table A3
Data for synthesis gas products and processes [1].

Product	Product no. i	Sale price	Process no. j	Process name	Propylene used/ton b_{j3}	Capacity (10^3 ton/yr)			Production cost ($\$10^6$/yr)			Investment ($\$10^6$)		
						l_j	m_j	h_j	C_{lj}	C_{mj}	C_{hj}	V_{lj}	V_{mj}	V_{hj}
Acetic acid	T17	480	S35	Low pressure carbonylation (Rh. Catalyst solution)		182.5	365	540	63.2	111.4	156.6	125.6	195.9	259.6
			S36	Low pressure carbonylation supported Rh. catalyst		182.5	365	540	60.3	103	142.6	116.4	168.2	213.5
			S37	Low pressure carbonylation Rh. Halide catalyst		180	360	550	64.7	110.2	154.6	133.2	196.3	248.9
Ammonia	T18	160	S38	ICI AMV process	6.3500	300	430	590	48.3	65	85	210.9	278.2	356
			S39	MW Kellog process	5.9280	300	430	590	52.8	71.4	92.7	243.5	322.4	412.6
			S40	ICI LCA process	6.6780	105	170	340	19.4	27.4	47.3	87	119.5	196.7
Formaldehyde 1	T19	500	S41	From methanol using silver catalyst	6.6780	15	25	50	6.6	9.7	17.7	15.3	20.2	32.7
			S42	From methanol using Fe Mo catalyst	6.6780	15	25	50	6.9	10.6	19.4	17.9	26.2	44.9
Methanol	T20	150	S43	Lurgi process	7.8670	415	830	1660	55.2	96.3	184.3	224.6	365.5	682.1
			S44	ICI process copper catalyst	7.7780	415	830	1660	56.5	100.5	194.3	228.5	384.6	727.6
			S45	ICI LCM process	7.6610	415	830	1660	51.9	98	187.6	199.1	371.5	702.9

Table A4
Data for aromatics (BTX) products and processes [1].

Product	Product no. i	Sale price	Process no. j	Process name	Ethylene used/ton b_{j1}	Capacity (10^3 ton/yr)			Production cost ($\$10^6$/yr)			Investment ($\$10^6$)		
						l_j	m_j	h_j	C_{lj}	C_{mj}	C_{hj}	V_{lj}	V_{mj}	V_{hj}
Styrene	T21	760	S46	Liquid phase alkyl/adiabatic dehydrogenation	0.2891	225	450	680	105.8	204.8	306	116.9	190	265.1
			S47	Liquid phase alkyl oxidative reheating	0.2878	225	450	680	108	209.7	313.5	115.6	191.8	266.8
			S48	Vapor phase alkyl/adiabatic dehydrogenation	0.2843	225	450	680	105.6	202.5	302.6	125.2	192.7	269
			S49	Vapor phase alkyl/isothermal dehydrogenation	0.2874	225	450	680	106.7	206.1	308.1	125.2	202	285.5
Pthalic anhydride	T22	700	S50	Attochem/Nippon		12.5	25	50	9.4	16.4	28.4	26	40.8	63.9
			S51	From O-Xylene by Alsuisse Italia process		12.5	25	50	9	15.4	27.3	27.7	39.6	56.9
Phenol	T23	735	S52	Liquid phase oxidation of toluene		45	90	180	36.8	64	118.7	108.8	157.2	251.6
PTA	T24	680	S53	Hydrolysis of dimethyl terephthalate		125	250	500	81.4	145.8	275.5	208.1	308.6	515.5
			S54	From P-xylene by bromine promoted air oxidation		125	250	500	78.4	145	277	170.5	267.3	452.7

References

[1] H.K. Alfares, A.M. Al-Amer, An optimization model for guiding the petrochemical industry development in Saudi Arabia, Eng. Optim. 34 (2002) 671–687.

[2] R. Kadambur, P. Kotecha, Multi-level production planning in a petrochemical industry using elitist Teaching–Learning-Based-Optimization, Expert Syst. Appl. 42 (2015) 628–641.

[3] M. Ramteke, R. Srinivasan, Large-scale refinery crude oil scheduling by integrating graph representation and genetic algorithm, Ind. Eng. Chem. Res. 51 (2012) 5256–5272.

[4] H. Liu, F. Jin, Y. Liu, J. Ding, X. Xu, Evaluation and optimization of the spatial organization of the petrochemical industry in China, J. Geogr. Sci. 23 (2013) 163–178.

[5] D. Iranshahi, E. Pourazadi, K. Paymooni, M.R. Rahimpour, Utilizing DE optimization approach to boost hydrogen and octane number in a novel radial-flow assisted membrane naphtha reactor, Chem. Eng. Sci. 68 (2012) 236–249.

[6] B. Ghorbani, M. Mafi, R. Shirmohammadi, M.-H. Hamedi, M. Amidpour, Optimization of operation parameters of refrigeration cycle using particle swarm and NLP techniques, J. Nat. Gas Sci. Eng. 21 (2014) 779–790.

[7] X. Wang, L. Tang, Multiobjective operation optimization of naphtha pyrolysis process using parallel differential evolution, Ind. Eng. Chem. Res. 52 (2013) 14415–14428.

[8] R.L. Cecchini, I. Ponzoni, J.A. Carballido, Multi-objective evolutionary approaches for intelligent design of sensor networks in the petrochemical industry, Expert Syst. Appl. 39 (2012) 2643–2649.

[9] D.K. Khosla, S.K. Gupta, D.N. Saraf, Multi-objective optimization of fuel oil blending using the jumping gene adaptation of genetic algorithm, Fuel Process. Technol. 88 (2007) 51–63.

[10] T. Ganesan, I. Elamvazuthi, K.Z.K. Shaari, P. Vasant, Swarm intelligence and gravitational search algorithm for multi-objective optimization of synthesis gas production, Appl. Energy 103 (2013) 368–374.

[11] F. Jiménez, G. Sánchez, P. Vasant, A multi-objective evolutionary approach for fuzzy optimization in production planning, J. Intell. Fuzzy Syst. 25 (2013) 441–455.

[12] P. Vasant, Hybrid optimization techniques for industrial production planning, in: Formal Methods in Manufacturing Systems IGI Global, 2013, pp. 84–111.

[13] P. Vasant, Handbook of Research on Novel Soft Computing Intelligent Algorithms: Theory and Practical Applications, vol. 2, IGI Global, 2014, pp. 1–1018.

[14] P. Vasant, Handbook of Research on Artificial Intelligence Techniques and Algorithms, vol. 2, IGI Global, 2015, pp. 1–796.

[15] P. Vasant, G.-W. Weber, V.N. Dieu, Handbook of Research on Modern Optimization Algorithms and Applications in Engineering and Economics, IGI Global, 2016, pp. 1–960.

[16] K. Al-Qahtani, A. Elkamel, Multisite refinery and petrochemical network design: optimal integration and coordination, Ind. Eng. Chem. Res. 48 (2009) 814–826.

[17] S.-G. Yoon, S. Park, J. Lee, P.M. Verderame, C.A. Floudas, Selecting the optimal target company based on synergy calculation for the vertical merger in a petrochemical complex, Ind. Eng. Chem. Res. 48 (2009) 1511–1521.

[18] G. Al-Sharrah, A. Elkamel, A. Almanssoor, Sustainability indicators for decision-making and optimisation in the process industry: the case of the petrochemical industry, Chem. Eng. Sci. 65 (2010) 1452–1461.

[19] S. Yadav, M.A. Shaik, Short-term scheduling of refinery crude oil operations, Ind. Eng. Chem. Res. 51 (2012) 9287–9299.

[20] Z. Jia, M. Ierapetritou, J.D. Kelly, Refinery short-term scheduling using continuous time formulation: crude-oil operations, Ind. Eng. Chem. Res. 42 (2003) 3085–3097.

[21] H. Lee, J.M. Pinto, I.E. Grossmann, S. Park, Mixed-integer linear programming model for refinery short-term scheduling of crude oil unloading with inventory management, Ind. Eng. Chem. Res. 35 (1996) 1630–1641.

[22] M. Ghatee, S.M. Hashemi, Optimal network design and storage management in petroleum distribution network under uncertainty, Eng. Appl. Artif. Intell. 22 (2009) 796–807.

[23] H.M.S. Lababidi, M.A. Ahmed, I.M. Alatiqi, A.F. Al-Enzi, Optimizing the supply chain of a petrochemical company under uncertain operating and economic conditions, Ind. Eng. Chem. Res. 43 (2004) 63–73.

[24] J.-K. Bok, H. Lee, S. Park, Robust investment model for long-range capacity expansion of chemical processing networks under uncertain demand forecast scenarios, Comput. Chem. Eng. 22 (1996) 1037–1049.

[25] R. Kadambur, P. Kotecha, Optimal production planning in a petrochemical industry using multiple levels, Comput. Ind. Eng. 100 (2016) 133–143.

[26] C.A.C. Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Comput. Meth. Appl. Mech. Eng. 191 (2010) 1245–1287.

[27] S. Chauhan, P. Kotecha, Production Planning Using Multiple-units, Technical Report - TR02, 2017.

[28] K. Deb, Multi-objective Optimization Using Evolutionary Algorithms, vol. 518, John Wiley & Sons, Inc., 2001.

[29] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching–learning–Based Optimization: a novel method for constrained mechanical design optimization problems, Comput. Aided Des. 43 (2011) 303–315.

[30] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching–Learning–Based Optimization: an optimization method for continuous non-linear large scale problems, Inf. Sci. 183 (2012) 1–15.

[31] S.C. Satapathy, A. Naik, Data clustering based on teaching-learning-based optimization, in: Swarm, Evolutionary, and Memetic Computing: Second International Conference, SEMCCO 2011, Springer Berlin Heidelberg, 2011, pp. 148–156.

[32] E. Uzlu, M. Kankal, A. Akpinar, T. Dede, Estimates of energy consumption in Turkey using neural networks with the teaching–learning-based optimization algorithm, Energy 75 (2014) 295–303.

[33] G. Wu, R. Mallipeddi, P.N. Suganthan, R. Wang, H. Chen, Differential evolution with multi-population based ensemble of mutation strategies, Inf. Sci. 329 (2016) 329–345.

[34] J. Zhang, A.C. Sanderson, JADE: adaptive differential evolution with optional external archive,", IEEE Trans. Evol. Comput. 13 (2009) 945–958.

[35] S. Mirjalili, A. Lewis, S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization, Swarm Evol. Comput. 9 (2013) 1–14.

[36] B. Akay, D. Karaboga, A modified artificial bee colony algorithm for real-parameter optimization, Inf. Sci. 192 (2012) 120–142.

[37] M. Mernik, S.-H. Liu, D. Karaboga, M. Črepinšek, On clarifying misconceptions when comparing variants of the Artificial Bee Colony Algorithm by offering a new implementation, Inf. Sci. 291 (2015) 115–127.

[38] K. Deb, A. Kumar, Real-coded genetic algorithms with simulated binary crossover: studies on multimodal and multiobjective problems, Complex Syst. 9 (1995) 431–454.

[39] R.V. Rao, V. Patel, An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems, Int. J. Ind. Eng. Comput. 3 (2012) 535–560.

[40] R. Kommadath, C. Sivadurgaprasad, P. Kotecha, Single phase multi-group teaching learning algorithm for computationally expensive numerical optimization (CEC 2016), in: 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, 2016, pp. 2989–2995.

[41] D. Chen, F. Zou, Z. Li, J. Wang, S. Li, An improved teaching–learning–based optimization algorithm for solving global optimization problem, Inf. Sci. 297 (2015) 171–190.

[42] X. Li, S. Ma, J. Hu, Multi-search differential evolution algorithm, Appl. Intell. 47 (2017) 231–256.

[43] S. Mirjalili, G.-G. Wang, L.D.S. Coelho, Binary optimization using hybrid particle swarm optimization and gravitational search algorithm, Neural Comput. Appl. 25 (2014) 1423–1435.

[44] G. Zhu, S. Kwong, Gbest-guided artificial bee colony algorithm for numerical function optimization, Appl. Math. Comput. 217 (2010) 3166–3173.

[45] B. Alatas, Chaotic bee colony algorithms for global numerical optimization, Expert Syst. Appl. 37 (2010) 5682–5687.

[46] Mathworks, Genetic Algorithm Options, Available from: http://in.mathworks.com/help/gads/genetic-.html?requestedDomain=in.mathworks.com (Accessed 1 March 2018).