Applied Soft Computing 68 (2018) 458–477

Contents lists available at ScienceDirect

Applied Soft Computing

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

An efficient multi-unit production planning strategy based on continuous variables

Sandeep Singh Chauhan, Prakash Kotecha*

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

ARTICLE INFO

Article history:
Received 30 November 2016
Received in revised form 1 March 2018
Accepted 7 March 2018
Available online 28 March 2018

Keywords:
Production planning
Artificial bee colony
Sanitized teaching learning based optimization
Dynamic neighborhood learning particle swarm optimizer
Multi–population ensemble differential evolution

ABSTRACT

Production planning can significantly enhance the competitiveness and hence plays a crucial role in manufacturing industries. In the production planning problem discussed in this article, a set of processes are available which can be operated at various capacity levels with the production and investment costs being proportional to the production of the processes. The determination of the optimal production plan, in terms of the selection of product to be manufactured, the selection and operating capacity of the manufacturing processes in the presence of multiple resource constraints so as to obtain maximum profit is a combinatorial optimization problem. In this article, we state the limitations of the formulation/strategies employed in literature and propose a multi-unit strategy which utilizes only continuous variables to overcome them. The proposed strategy is generic and demonstrated in the context of production planning in a petrochemical industry that has been previously used to potentially guide the petrochemical industries. The proposed strategy is demonstrated with multiple computational intelligence algorithms viz., artificial bee colony, dynamic neighbourhood learning particle swarm optimizer, multi-population ensemble differential evolution, and sanitized–teaching–learning–based optimization. For the cases discussed in literature, the proposed strategy shows an improvement of up to 12.1% in the profit.

1. Introduction

The products from petrochemical industries are feedstock to large number of industries of the manufacturing sector and thus have a significant impact on the economy of a nation. In addition, these are also a major source of revenue for countries that have rich sources of natural gas and crude oil which are feedstock for the petrochemical industries. For example, a significant number of the Middle Eastern countries use their natural resources to produce and export petrochemical products to benefit their economy [1,2]. Globally, the petrochemical sector is estimated to be worth more than $600 billion [3] and is capital intensive [4] as it relies on large scale production. A petrochemical industry employs a set of complex processes to convert its raw materials into chemical products that are required to manufacture everyday consumer products. The complex nature of the petrochemical industries requires the optimization of a number of factors and researchers have focussed on optimizing various diverse aspects of petrochemical industries including supply chain management [5], determination of optimal production plans [1], efficient integration of refineries and petrochemical plants [6], capacity expansion [7], efficient job scheduling [8], strategies for mergers and acquisitions [9,10] and spatial organization in petrochemical plants [11].

Some of the objectives that have been used to address the complex challenges in petrochemical plant include improving the safety index [12], maximizing the annual profit [1,13], minimizing the total cost [14], minimizing the raw material requirement [15], minimizing the harmful environmental impact [16] and maximizing the thermodynamic availability [17]. Various optimization models have been built for specific countries to obtain maximum benefits including the development of a Mixed Integer Linear Programming (MILP) models for Mexican petrochemical industries [18], the development of Norwegian petrochemical industries [19,20], the development of MILP models to identify the synergy in mergers and acquisitions in the Korean petrochemical industry [9,10,21] and MILP models for guiding the petrochemical industry in Saudi Arabia to maximize the annual profit [1]. Some of the previous works that have used heuristic or evolutionary algorithms in petrochemical industries or refineries include the intelligent design of sensor networks in the petrochemical industry [22], crude oil scheduling [23], multi–objective optimization of a crude distillation unit [24] and optimizing the operating conditions of a fluid catalytic cracking unit [25].

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

https://doi.org/10.1016/j.asoc.2018.03.012
1568-4946/

Nomenclature

b_t^j	Amount of raw material t required in process j for producing per ton of product
c	Index for production capacity level
c_c^j	Production capacity of capacity level c for process j
f_p	Static penalty factor
i	Index for product, $i \in (1, .., I)$
j	Index for process, $j \in (1, .., J)$
l^j, m^j, h^j	Low, medium and high production capacity level of process j
n_i	Number of processes employed to produce product i
t	Raw material index, $t \in (1, .., T)$
x_{l-m}^j	Amount of product produced in between the production capacities l and m for process j
x_{m-h}^j	Amount of product produced in between production capacities m and h for process j
$x_{(c-(c+1)),u}^j$	Amount of product produced in between production capacity level c and $c+1$ of process j for the unit u
B	Total available monetary resource
C_c^j	Production cost of capacity level c for process j
C^j	Total production cost of process j
E^j	Selling price per ton of the product produced from process j
I	Total number of products
J	Total number of processes
L	Total number of production capacity level of a process
L^j, M^j, H^j	Represents the proportions of production using production capacity levels l^j, m^j and h^j
Q	Constant coefficient, represents a large value (as in the big–M method)
$N_{c-(c+1)}^j$	Number of units with capacity in between c and $c+1$ that can be established for process j
P_t^{raw}	Penalty incurred in fitness function due to insufficient raw material of type t
P^{Inv}	Penalty incurred in fitness function due to insufficient investment cost
P_i^{uni}	Penalty incurred in fitness function due to violation of unique process constraint for product i
R_t	Available feedstock of raw material t
T	Total number of raw materials
V_c^j	Investment cost of capacity level c for process j
V^j	Total investment cost of process j
X^j	Total amount of product produced from process j
Y^j	Binary variable to indicate the production level from process j; = 1 if $X^j \le m^j$, else 0
Z^j	Binary variable to indicate the production from process j; = 1 if $X^j > 0$, else 0
Ω^{rand}	Random member from the population
Ω^{best}	Best solution in the population
Ω^{mean}	Average value (decision variable) of the population
Ω_ψ^{new}	New solution generated either in the teacher or learner phase using the ψ^{th} member of the population
σ	Class or population size in s-TLBO
Γ	Maximum number of iterations in s-TLBO
ψ	Index used to refer the population in s-TLBO
ϕ	Index used to refer to the iteration in s-TLBO

In this article, we critically review the formulation/strategies available in literature for the production–planning problem and propose a multi–unit production strategy based on only continuous variables to determine the optimal production plan, which overcomes the limitations and helps in the determination of optimal production plans with a higher profit. The benefits of the proposed strategy are demonstrated on all the eight cases that have been discussed in literature [1,13,26]. Additionally, this article also helps in evaluating the performance of four algorithms, *viz.,* artificial bee colony (ABC), dynamic neighbourhood learning particle swarm optimizer (DNLPSO), multi–population ensemble differential evolution (MPEDE) and sanitized–teaching–learning–based optimization (s-TLBO) on large problems, involving large number of variables (> 2000) and complex constraints involving hard penalty as well as unique process requirement.

In the following section, we provide the problem description and follow it up with a critical review of the formulation/strategies and state its limitations. Subsequently, we provide a brief review of the optimization techniques employed in this work and propose the multi–unit production strategy to determine efficient production plan. Subsequently, the benefits of the proposed strategy is demonstrated on eight cases from the literature and the article is concluded by summarizing the developments in this work and discussion on possible future work.

2. Problem description

The production planning problem can be defined as follows. Given (i) the list of products and the list of all available processes that can be used to manufacture each of the product, (ii) the selling price of the product, (iii) the different capacity levels of each process along with its production and investment costs, (iv) the amount of raw materials required in each process to produce the particular product, (v) the total demand of each of the product in the market, (vi) the monetary resource that is available for investment, and (vii) the amount of each of the raw materials that is available for the production planning, the objective is to maximize the profit by determining the optimal production in terms of (i) the amount of product that needs to be produced, (ii) the processes to be used for producing each product, and (iii) the production capacity of the processes that have been selected for producing a particular product.

The production plan should ensure that (i) the investment and raw materials required for the entire production plan should not be more than the available budget and the available quantity of raw materials respectively, (ii) the production from every process should be in between the permissible levels, and (iii) no product should be produced from more than one process. The last constraint is optional and is known as "unique process requirement." An instance of this data in the context of production planning in petrochemical industries is shown in Tables A1–A4 involving 24 products (T1-T24) that can be produced from 54 processes (represented as S1-S54). Each of the process has the production cost $\left(C_l^j, C_m^j, C_h^j\right)$ and investment cost $\left(V_l^j, V_m^j, V_h^j\right)$ data available for low level $\left(l^j\right)$, medium level $\left(m^j\right)$ and high level capacity $\left(h^j\right)$. Any production that is greater than zero but less than the capacity of the low level $\left(l^j\right)$ is forbidden. As shown in Fig. 1, the production and investment costs vary linearly between two successive capacity levels thereby enabling the determination of production and investment cost for a unit with capacity X^j that lies in between l^j and h^j. It should be noted that a process j can remain without producing any product but cannot produce less than l^j and greater than h^j.

In literature, a single–level MILP formulation [1] as well as a multi–level MILP [26] and elitist TLBO based strategy [13] have been proposed for determining the optimal production plan in

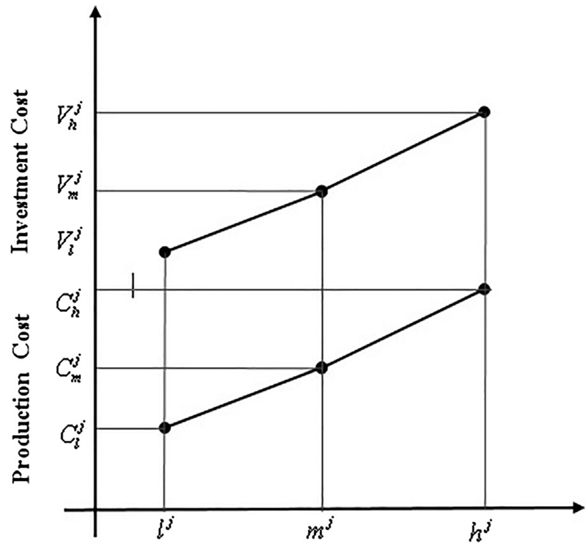

Fig. 1. Production cost and investment cost as a function of production level capacity [1].

petrochemical industries. We will briefly review the relevant parts of these two works to discuss their limitations, which have been overcome in this article.

2.1. Limitations of the formulation/strategy in literature

In the single level MILP formulation [1], the total amount of a product produced $\left(X^j\right)$ from a particular process j is given by Eq. (1) where l^j, m^j and h^j denote the production capacities of the low, medium and high level units.

$$X^j = l^jL^j + m^jM^j + h^jH^j \tag{1}$$

In this equation, L^j,M^j and H^j are continuous, positive decision variables that can at most assume a value of 1 and are constrained by Eqs. (2)–(4) in which $Y^j$and Z^j are binary variables.

$$L^j \le Y^j \tag{2}$$

$$H^j \le 1 - Y^j \tag{3}$$

$$L^j + M^j + H^j = Z^j \tag{4}$$

$$X^j \le QZ^j, \text{Q}\,\textit{is a large number}\,(\textit{as in the big}-M\ \textit{method}) \tag{5}$$

In Eqs. (1)–(5), Y^j and $Z^j$are binary variables whereas L^j, M^j, and H^j are continuous variables in the domain of 0–1. In particular, Y^j will assume a value of 1 if X^j is less than or equal to m^j and a value of 0 if X^j is greater than m^j. Similarly, Z^j will be 1 if X^j is greater than 0 and Z^j will be 0 if X^j is 0. If process j produces a product and if $Y^j = 1$, Eq. (3) would ensure that $H^j = 0$ which can be used in Eq. (4) thereby transforming Eq. (1) to $X^j = l^j - M^j\left(l^j - m^j\right)$. Since M^j is always positive and $\left(l^j - m^j\right)$ will always be negative, X^j will always lie in between l^j and m^j. On the contrary, if process j produces a product and if $Y^j = 0$, Eq. (2) would ensure that $L^j = 0$ and Eq. (4) can be used to transform Eq. (1) to $X^j = h^j - M^j\left(h^j - m^j\right)$. Since M^j and $\left(h^j - m^j\right)$ will always be positive, X^j will always lie in between m^j and h^j. This enabled the determination of profit by using Eq. (6) where C_l^j, C_m^j, and C_h^j are the production costs.

$$\textit{Maximize}\quad P = \sum_{j=1}^{J} E^jX^j - \left(C_l^jL^j + C_m^jM^j + C_h^jH^j\right) \tag{6}$$

It can be easily shown if $Y^j = 0$, the second term in Eq. (6), denoting the production cost of process j, would correspond to the production cost determined by interpolation from the line connecting the points $\left(l^j, C_l^j\right)$ and $\left(m^j, C_m^j\right)$ in Fig. 1. Similarly, if $Y^j = 1$, the second term would correspond to production cost denoting the production cost determined from the line connecting the points $\left(m^j, C_m^j\right)$and $\left(h^j, C_h^j\right)$ in Fig. 1. Thus the single level MILP formulation artificially restricts the production of a product X^j to the high production level since the production has to be either in between the level l^j and m^j or between m^j and 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 at most h^j thereby determining suboptimal solutions.

In a subsequent work, an elitist TLBO based strategy [13] and a MILP formulation [26] with multiple–levels was used for this production planning problem that overcame the drawbacks of the single–level MILP formulation [1] by allowing the deployment of more than unit. In particular, for a production planning problem involving J processes with three production levels $\left(l^j, m^j, h^j\right)$ for each process j, this work used a total of $2J$ continuous variables $\left(x_{l-m}^j, x_{m-h}^j\right)$ to denote the amount of product produced from the J processes. The effective domain of the variables were restricted to $l^j \le x_{l-m}^j \le m^j$ and $m^j \le x_{m-h}^j \le h^j$ representing the production from two units with production capacity between low-medium and medium-high capacity levels. Thus the maximum amount of product produced from a process j is restricted to $m^j + h^j$ which is better than the single level MILP formulation [1] which restricted the production to h^j. Though this work overcame the drawbacks of the single level MILP formulation, it nevertheless artificially restricts the production from process j to $m^j + h^j$ thereby possibly leading to production plans that are suboptimal. The amount of production from a process should ideally be restricted either by the amount of resources available to produce the product or by the demand of the product in the market. In this current work, we overcome the drawbacks of the formulation and strategies used in literature by accommodating multiple units across the various production levels. We also evaluate the performance of four computational intelligence techniques to solve the production planning problem using the proposed strategy.

3. Review of computational intelligence algorithms

For the sake of brevity, we briefly describe the four computational intelligence algorithms used in this work. Additional details can be obtained from literature.

3.1. Artificial bee colony

ABC is a population–based evolutionary algorithm that has been proposed in 2005 and is motivated by the intelligent behaviour of honey bees. In ABC algorithm, the employed bees independently explore the nectar source, the onlookers bees exploit the food source based on the information received from employed bees and scouts bees searches for new food sources. The employed and onlooker bees are reported to help in exploitation whereas the scout bees aid in the exploration of the search space. Several issues regarding functional evaluation and appropriate implementation of ABC has been described in the literature [27,28]. ABC requires only two user–defined parameters in terms of a number of food sources (population size) and the maximum number of function evaluations (termination criteria). ABC uses a parameter "limit" which determines the maximum number of trial for abandoning a source [28]. Conventionally, it has been set to dimension of the problem x employed bees. It has been recently reported [29] that “limit” has to be suitably tuned. Thus ABC has three tuning parameters.

3.2. Dynamic neighbourhood learning particle swarm optimizer

DNLPSO has been proposed [30] in 2012 and is a novel variant of the particle swarm optimization. In DNLPSO, the velocity of a particle is updated using a learning strategy that utilizes the historical best information of all other particles with the exemplar particle being selected from a neighbourhood. The neighbourhood is reformed after periodic intervals to aid in the diversity. The selection of exemplar from the neighbourhood is done randomly for updating the velocity. A ring topology is used for the implementation of the neighbourhood in which the entire population (swarm) is divided into equal subgroups and connected with a ring–shaped graph. A code review of DNLPSO shows seven user defined parameters (refreshing gap, regrouping period, parameter for determination of velocity bound, two acceleration parameters and two parameters to calculate the inertia factor) in addition to population (swarm) size and termination criteria. In this work, for out of bound variables, the variables are set to its nearest bound [31].

3.3. Multi–population ensemble differential evolution

MPEDE has been proposed [32] in 2015 and is a variant of the differential evolution algorithm. Its performance has been evaluated on benchmark optimization problems and it has been shown to outperform other DE variants. MPEDE employs a multi population–based approach and incorporates multiple mutation strategies. It consists of two types of population (viz., three indicator subpopulations, and one reward subpopulation). A separate strategy (“current–to–pbest/1”, “current–to–rand/1”, and “rand/1”) is assigned to each of the indicator subpopulation and the reward population is assigned randomly to one mutation strategy. After a specified number of generations, the reward population employs the mutation strategy that has shown the best performance. Binomial crossover is used with “rand/1” and “current–to–pbest/1” mutation strategies while the “current–to–rand/1” strategy is used without any crossover. Other than population size and termination criteria, MPEDE requires two user–defined parameters viz., (i) ratio between indicator population to the whole population and (ii) generation gap. Variables violating the bounds are set to their nearest bounds [31].

3.4. Sanitized teaching–learning–based optimization

TLBO has been proposed [33] multiple times since 2011 and it simulates the teaching–learning phenomenon in a classroom wherein each learner (or population member) learns from the teacher (in the Teacher Phase) as well as other learners (in the Learners Phase). In each generation, a new solution is generated in teacher phase using the best student and mean of the students in the class. In the student phase, a new solution is generated using a random student from the class. Whenever a new solution is generated, its fitness is compared with the learner undergoing the teacher or learners phase. If the new solution has a better fitness, it is used to replace the current learner. The TLBO variant, s-TLBO, used in this work does not employ any duplicate removal procedure and requires only two functional evaluations per member of the population in every iteration. The total number of fitness evaluation is (population size + 2 x population size x number of generation). In view of the various ambiguities [34,35] reported in the literature, we are providing the pseudo code of the variant of TLBO employed in this work. Recent variants of TLBO include SPMGTLO [36] and FTLBO [37].

Set $\phi = 0$ (iteration counter); Maximum allowed iterations, Γ
Initialize a random population (Ω) of size σ
Evaluate Ω for their fitness*, $f(\Omega)$
for $\phi = 1 : \Gamma$
 for $\Psi = 1 : \sigma$

TEACHER PHASE
  Choose Ω^{best} (solution with the best fitness)
  Set the scalar T_f randomly to either 1 or 2.
  Determine Ω^{mean} and $\Omega_\psi^{new} = \Omega_\psi + r(\Omega^{best} - T_f \Omega^{mean})$
  For out of bound values of Ω_ψ^{new}, set to nearest bound.
  Evaluate Ω_ψ^{new} for its fitness*
  Replace Ω_ψ with Ω_ψ^{new} if fitness of Ω_ψ^{new} is better than Ω_ψ

STUDENT PHASE
  Choose any solution randomly, Ω^{rand}
  Determine Ω_ψ^{new} as
   if $f(\Omega_\psi) < f(\Omega^{rand})$
    $\Omega_\psi^{new} = \Omega_\psi + r(\Omega_\psi - \Omega^{rand})$
   else
    $\Omega_\psi^{new} = \Omega_\psi + r(\Omega^{rand} - \Omega_\psi)$
   end
  For out of bound values of Ω_ψ^{new}, set to nearest bound.
  Evaluate Ω_ψ^{new} for its fitness*
  Replace Ω_ψ with Ω_ψ^{new} if fitness of Ω_ψ^{new} is better than Ω_ψ
 end
end

* *indicates the evaluation of the fitness function; r indicates a random number between 0 and 1. For every decision variable, a separate value of r is selected. For a problem with D decision variables, 2D random numbers would be required for every member in a generation; no duplicate removal is employed in this variant.*

4. Proposed multi–unit strategy for production planning

Many mathematical programming techniques guarantee global optimality for well–structured problems, are deterministic in nature and do not require multiple runs of a problem. However, these require the constraints to be in the form of explicit equalities and inequalities thereby rendering the modelling tedious in many instances. Sometimes these also require the addition of a large number of “artificial” variables so as to model the constraints in terms of conventional equalities and inequalities. The addition of these additional variables, especially discrete, can significantly increase the requirement of computational resources due to increased branching of the variables and have computational issues with scalability.

On the contrary, evolutionary techniques provide greater modelling flexibility as these do not require the constraints to be in the form of linear equalities and inequalities. In this article, we harness this benefit of evolutionary algorithms to overcome the limitations that either restricted the production from a process to the high level production capacity or to the sum of the medium and high level production capacity. In this work, the potential production from a process, if profitable, is not artificially restricted as in literature [1,13,26] but it is restricted by the availability of resources, in terms of raw materials and budget, and the demand of the product produced from the process in the market. The maximum production is achieved by incorporating the ability to include production from multiple units and using multiple levels. This aids in the maximum production of the most profitable product and leads to better solutions than those reported in literature. A unique feature of the proposed multi–unit strategy is that it does not involve any discrete variable and thereby helps in the direct application of the evolu-

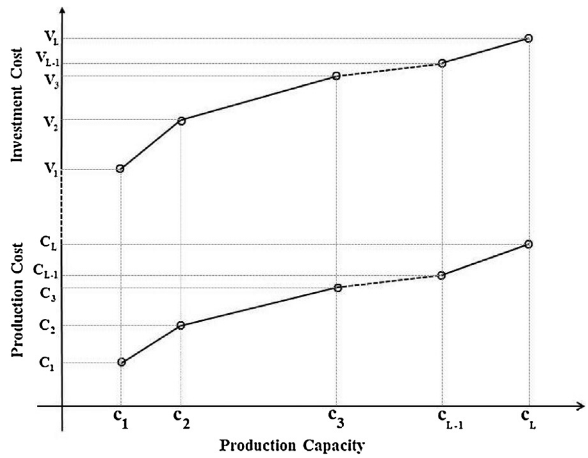

Fig. 2. Production and Investment Cost as a function of Production Capacity.

tionary algorithms, which are essentially developed for continuous variables. Moreover, the proposed strategy can easily accommodate non–linear cost functions.

4.1. Decision variables

For the sake of simplicity, consider a process j which has three production levels of l^j, m^j and h^j. If beneficial, one can employ more than one unit whose capacities are in between l^j and m^j and also more than one unit whose capacities are in between m^j and h^j. However the maximum number of such units that can be employed with production capacity in between l^j and m^j is dependent on the budget that is available for the entire production planning and the investment cost of level l^j. Thus if the investment cost of level l of process j is V^j_l and if the total budget available for investment is B, the maximum number of units of level l for process j that can be established is $\left\lfloor B/V^j_l \right\rfloor$. Similarly the maximum number of units of level m for process j that can be established is $\left\lfloor B/V^j_m \right\rfloor$. Thus for process j, there are a total of $\left(\left\lfloor B/V^j_l \right\rfloor + \left\lfloor B/V^j_m \right\rfloor\right)$ decision variables in which $\left\lfloor B/V^j_l \right\rfloor$ have the domain in between l^j and m^j whereas the other $\left\lfloor B/V^j_m \right\rfloor$ decision variables have a domain in between m^j and h^j. In general, if there are J processes with each of them having L different production capacities (as shown in Fig. 2) for which the production cost and investment cost is available, then the total number of decision variables involved in the production planning is given by

$$N = \sum_{j=1}^{J}\sum_{c=1}^{L-1} N^j_{c-(c+1)} \tag{7}$$

$$N^j_{c-(c+1)} = \left\lfloor \frac{B}{\min\left(V^j_c, V^j_{c+1}\right)} \right\rfloor \qquad \forall c = 1, 2, .., L-1, j = 1, 2, .., J \tag{8}$$

The value of $N^j_{c-(c+1)}$ indicates the number of units with capacity in between c and $c+1$ that can be established for process j if the entire budget is to be used for establishing units of capacity c for process j. It is quite evident that these many numbers of units may not be established but however the exact number of units cannot be known a priori and the optimization algorithm has to determine the optimal number of units of appropriate capacity that lead to beneficial results. For the sake of better readability, we will refer $N^j_{c-(c+1)}$ as N^j_c

The set of decision variables given in Eq. (7) can be represented as shown in Eq. (9)

$$\left\{\overbrace{x^1_{1-2,1}, x^1_{1-2,2}, \ldots, x^1_{1-2,N^1_1}}^{\text{Process 1 (Level 1-Level 2)}}, \underbrace{x^2_{1-2,1}, x^2_{1-2,2}, \ldots, x^2_{1-2,N^2_1}}_{\text{Process 2 (Level 1-Level 2)}}, \ldots, \overbrace{x^J_{1-2,1}, x^J_{1-2,2}, \ldots, x^J_{1-2,N^J_1}}^{\text{Process J (Level 1-Level 2)}}\right\},$$
$$\left\{\overbrace{x^1_{2-3,1}, x^1_{2-3,2}, \ldots, x^1_{2-3,N^1_2}}^{\text{Process 1 (Level 2-Level 3)}}, \underbrace{x^2_{2-3,1}, x^2_{2-3,2}, \ldots, x^2_{2-3,N^2_1}}_{\text{Process 2 (Level 2-Level 3)}}, \ldots, \overbrace{x^J_{2-3,1}, x^J_{2-3,2}, \ldots, x^J_{2-3,N^J_2}}^{\text{Process J (Level 2-Level 3)}}\right\},$$
$$\ldots, \left\{\begin{array}{l} \overbrace{x^1_{(L-1)-L,1}, x^1_{(L-1)-L,2}, \ldots, x^1_{(L-1)-L,N^1_{(L-1)}}}^{\text{Process 1 (Level } (L-1)\text{-Level } L)}, \underbrace{x^2_{(L-1)-L,1}, x^2_{(L-1)-L,2}, \ldots, x^2_{(L-1)-L,N^2_{(L-1)}}}_{\text{Process 2 (Level } (L-1)\text{-Level } L)}, \ldots \\ \ldots, \overbrace{x^J_{(L-1)-L,1}, x^J_{(L-1)-L,2}, \ldots, x^J_{(L-1)-L,N^J_{(L-1)}}}^{\text{Process J (Level } (L-1)\text{-Level } L)} \end{array}\right\} \tag{9}$$

In Eq. (9), the term $x^J_{1-2,1}$ indicates the production for process J from a unit whose capacity is between level 1 (c_1) and level 2 (c_2). The number of such units is N^J_1 that is to be determined from Eq. (8). Similarly the term $x^J_{(L-1)-L,N^J_{(L-1)}}$ indicates the production from process J from a unit whose capacity is between level (L-1) and level L. If there are three levels, i.e. $L=3$, then this term would reduce to x^J_{2-3,N^J_2}. For the data given in Tables A1–A4 involving 54 process with three production capacity level $\left(l^j, m^j, h^j\right)$ and a budget of 1000, the maximum number of units for process S1 with a production capacity equal to the low level capacity l_1 $(= 70)$ is 18 $\left(= \left\lfloor \frac{1000}{55} \right\rfloor\right)$ whereas the maximum number of units with a production capacity equal to the medium level capacity m_1 $(= 135)$ is 12 $\left(= \left\lfloor \frac{1000}{81.1} \right\rfloor\right)$. Thus there will be a total of 30 (=18 + 12) variables corresponding to process S1. Similarly, for process S54, the maximum number of units with low production capacity and medium level production capacity is 5 and 3 respectively and a total of 8 decision variables will be required.

$$\left\{\overbrace{x^1_{1-2,1}, x^1_{1-2,2}, \ldots, x^1_{1-2,18}}^{\text{Process 1 (low-medium)}}, \underbrace{x^2_{1-2,1}, x^2_{1-2,2}, \ldots, x^2_{1-2,17}}_{\text{Process 2 (low-medium)}}, \ldots, \overbrace{x^{54}_{1-2,1}, x^{54}_{1-2,2}, x^{54}_{1-2,3}, x^{54}_{1-2,4}, x^{54}_{1-2,5}}^{\text{Process 54 (low-medium)}}\right\},$$
$$\left\{\overbrace{x^1_{2-3,1}, x^1_{2-3,2}, \ldots, x^1_{2-3,12}}^{\text{Process 1 (medium-high)}}, \underbrace{x^2_{2-3,1}, x^2_{2-3,2}, \ldots, x^2_{2-3,11}}_{\text{Process 2 (medium-high)}}, \ldots, \overbrace{x^{54}_{2-3,1}, x^{54}_{2-3,2}, x^{54}_{2-3,3}}^{\text{Process 54 (medium-high)}}\right\}$$

The total production from process j, denoted by X^j, can be determined using the following equation

$$X^j = \sum_{c=1}^{L-1} \sum_{u=1}^{N_c^j} x^j_{c-(c+1),u} \qquad \forall j = 1, 2, .., J \tag{10}$$

The above discussion is based on the assumption that the budget available for investment is the only limiting resource. In the event of additional limiting resources such as different types of raw materials, Eq. (8) can be modified as shown below which determines the maximum number of units depending on all limiting resources

$$N_c^j = \min\left(\left(\left\lfloor \frac{B}{\min\left(V_c^j, V_{c+1}^j\right)} \right\rfloor\right), \left(\min_{\forall t=1,2,\ldots,T} \left\lfloor \frac{R_t}{b_t^j c_c^j} \right\rfloor\right)\right)$$
$$\forall c = 1, 2, .., L-1, j = 1, 2, .., J \tag{11}$$

In Eq. (11), $\left\lfloor \frac{R_t}{b_t^j c_c^j} \right\rfloor$ denotes the maximum number of units of capacity c_c^j that can be established if b_t^j is the amount of raw material t that is required for producing unit quantity of product from process j and R_t is the total amount of raw material t that is available. The term $\left(\min_{\forall t=1,2,\ldots,T} \left\lfloor \frac{R_t}{b_t^j c_c^j} \right\rfloor\right)$ indicates the maximum number of units with a production capacity in between c and $c+1$ that can be established considering all the T limiting raw materials for process j. Since the value of N_c^j obtained from Eq. (11) cannot be greater than the value determined from Eq. (8), using Eq. (11) would lead to lower number of decision variables without compromising on the optimal solution.

4.2. Domains

The domains of the decision variable play a crucial role in the success of evolutionary algorithms as these are basically search techniques and a smaller domain may possibly require lower computational resources and potentially have a higher probability of discovering the globally optimal solution. However the domain of the variables should be such that the entire feasible region of the problem is accounted thereby preventing the loss of any optimal solutions.

As the variables denote the amount of product to be produced using a unit whose capacity is in between two successive specified levels, if produced, $x^j_{(c-(c+1)),u}$ should be in between c_c^j and c_{c+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

$$0 \le x^j_{(c-(c+1)),u} \le c^j_{c+1} \qquad \forall c = 1, 2, \ldots, L-1;$$
$$\forall j = 1, 2, \ldots, J; u = 1, 2, \ldots, N_c^j \tag{12}$$

For the example with three different production capacities, the domain of the decision variables is as shown below

$$0 \le x^j_{1-2,u} \le m^j \quad \forall j = 1, 2, \ldots, J; \forall u = 1, 2, \ldots, N_1^j$$
$$0 \le x^j_{2-3,u} \le h^j \quad \forall j = 1, 2, \ldots, J; \forall u = 1, 2, \ldots, N_1^j$$

As explained earlier, it should be noted that no production is possible in the region between 0 and c_l^j and in order to handle this hole in the domain of $x^j_{(c-(c+1)),u}$, we will employ a correction/repair strategy.

4.3. Constraint handling

In techniques that are primarily designed for unconstrained optimization problems, the constraints can be handled either explicitly or implicitly. By explicitly, we mean that an external penalty is added to the objective function if a population member violates the constraint. In implicit method, the strategy itself is modified so as to ensure that the population member satisfies the constraint. This is primarily done by suitably modifying the population member through some additional operations/operators and is known as "repairing" of the solution (as shown in Fig. 3). Hybrid versions also exist wherein some hard to satisfy constraints are handled implicitly whereas the other constraints are handled by an explicit method. In this work, we have used the implicit constraint handling to satisfy the domain hole constraint and have used an explicit approach to handle the other constraints to solve the complex optimization problem involving large number of variables, hard penalties, domain holes and complex constraints such as unique process constraint.

4.4. Domain hole constraint

As explained earlier, if a product is manufactured, the production quantity should be greater than or equal to the lower limit of the production level or should not be manufactured. If the product manufactured $x^j_{(l-(l+1)),u}$ is greater than 0 and is less than c_l^j, the variable is set to zero.

$$\left(x^j_{(c-(c+1)),u} > 0\right) \& \left(x^j_{(c-(c+1)),u} < c_1^j\right) \Rightarrow x^j_{(c-(c+1)),u} = 0;$$
$$\forall c = 1, 2, .., L-1; j = 1, 2, \ldots, J; u = 1, 2, .., N_c^j \tag{13}$$

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

$$\left(x^j_{1-2,u} > 0\right) \& \left(x^j_{1-2,u} < c_1^j\right) \Rightarrow x^j_{1-2,u} = 0 \qquad \forall j = 1, 2, \ldots, J; u = 1, 2, .., N_1^j$$
$$\left(x^j_{2-3,u} > 0\right) \& \left(x^j_{2-3,u} < c_2^j\right) \Rightarrow x^j_{2-3,u} = 0 \qquad \forall j = 1, 2, \ldots, J; u = 1, 2, .., N_2^j$$

From the above equations, it can be seen that the domain hole constraint would always be satisfied by the repairing mechanism without incurring any penalty.

4.5. Raw material constraint and investment constraint

These constraints are handled using the conventional penalty method. If there are a total of T raw materials which are available in limited quantities, a penalty is incurred if the consumption for a production plan is more than the available quantity. The penalty incurred if the t^{th} raw material is consumed in a quantity that is greater than its availability is given by

$$P_t^{raw} = \begin{cases} \left(R_t - \sum_{j=1}^{J} b_t^j X^j\right)^2 & R_t < \sum_{j=1}^{J} b_t^j X^j \\ 0 & R_t \ge \sum_{j=1}^{J} b_t^j X^j \end{cases} \qquad \forall t = 1, 2, \ldots, T \tag{14}$$

In the above equation, X^j denotes the total production from process j and is given by Eq. (10) whereas b_t^j is the amount of raw material t that is required for a unit production in process j. Thus

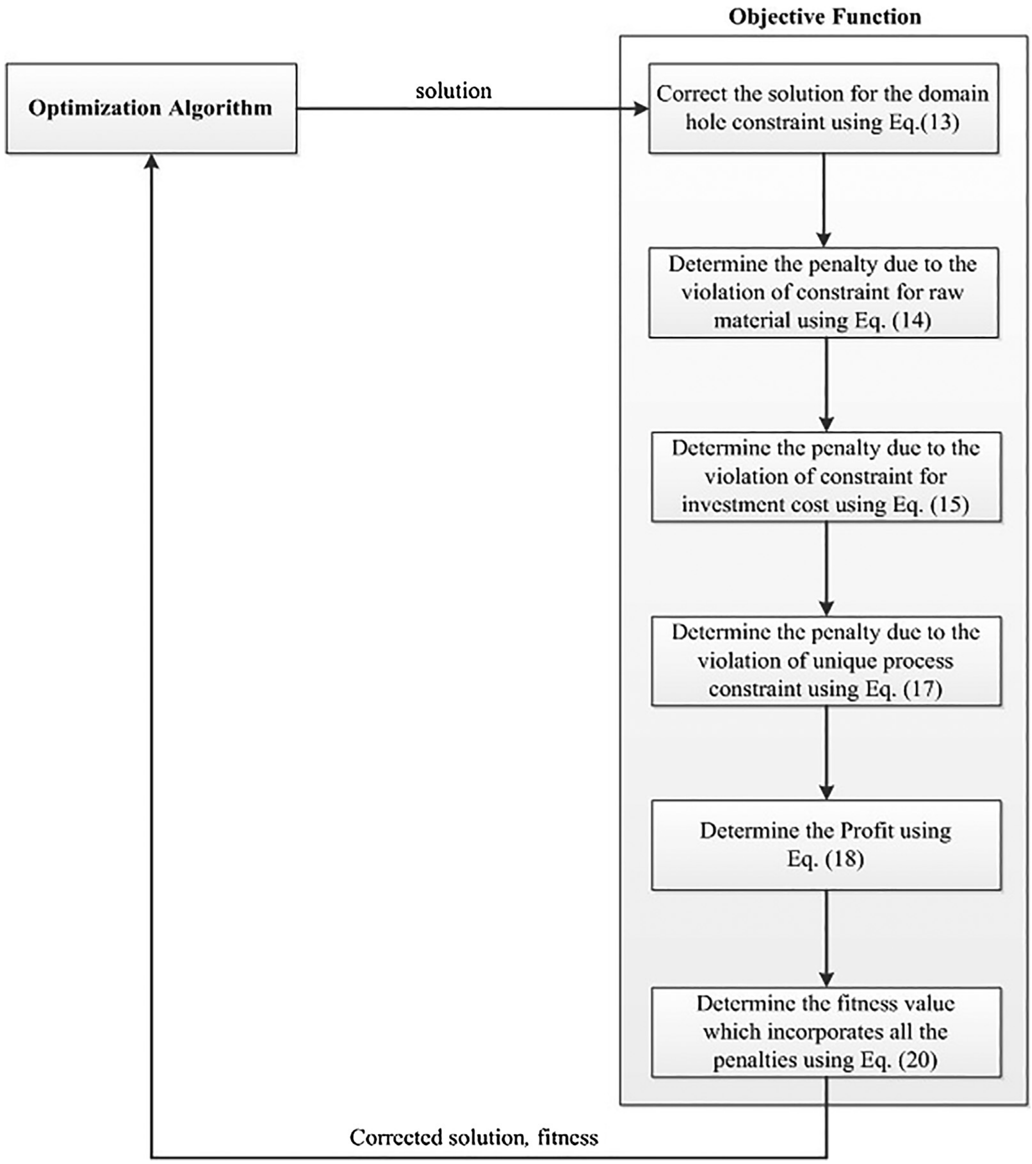

Fig. 3. Depiction of the evaluation of the fitness function.

the penalty $\left(P_t^{raw}\right)$ incurred for violating each of the raw materials can be determined. Similarly, the penalty P^{Inv} for exceeding the available investment budget (B) is given by

$$P^{Inv} = \begin{cases} \left(B - \sum_{j=1}^{J} V^j\right)^2 & B < \sum_{j=1}^{J} V^j \\ 0 & B \geq \sum_{j=1}^{J} V^j \end{cases} \quad (15)$$

In the above equation, V^j denotes the investment cost incurred for the production of X^j and is determined by

$$V^j = \begin{cases} \sum_{c=1}^{L-1} \sum_{u=1}^{N_c^j} \left(V_c^j + \left(\frac{V_{c+1}^j - V_c^j}{V_{c+1}^j - V_c^j} \right) \left(x_{c-(c+1),u}^j - V_c^j \right) \right) & ; c_c^j \leq x_{c-(c+1),u}^j \leq c_{c+1}^j \\ 0 & ; x_{c-(c+1),u}^j < c_c^j \end{cases} \quad \forall j = 1, 2, \ldots, J \quad (16)$$

It is easy to note that the investment cost determined in Eq. (16) can be determined by interpolating the line joining the points $\left(c_c^j, V_c^j\right)$ and $\left(c_{c+1}^j, V_{c+1}^j\right)$ shown in Fig. 2. It should be noted that the repairing mechanism ensures that any value of $x_{c-(c+1),u}^j$ which is less than c_c^j is converted to zero, the above equation will also ensure that the investment cost corresponds to a value of zero since there is no production from $x_{c-(c+1),u}^j$.

4.6. Unique process constraint

In certain circumstances, a product is to be produced using only one type of process and the production of the product from more than one process is considered as an invalid solution. This requirement is commonly known as "unique process constraint".

In the proposed strategy, the values of the decision variables (each population member in every generation) are known and hence it is straight forward to determine the number of processes used for producing each product. If there are I products and if n_i denotes the number of processes used for producing the product i, then the penalty P_i^u for using more than one process for producing the product i can be determined as shown in Eq. (17)

$$P_i^{uni} = \begin{cases} 1000^{n_i} & n_i > 1 \\ 0 & otherwise \end{cases} \quad \forall i = 1, 2, ..I \tag{17}$$

It is obvious from this equation that there is no penalty if the product is not produced ($n_i = 0$) or if a product is produced only from a single process. Otherwise a penalty is incurred which is in proportion to the number of processes used to manufacture the product i.

4.7. Objective function

The objective function in the production planning problem is the maximization of the profit which depends on the selling price of the products and the production cost incurred to manufacture the products. This can be determined as shown in Eq. (18) in which E^j denotes the selling price of the product from process j, X^j denotes the amount of product produced from process j and C^j denotes the production cost incurred in the process j.

$$Maximize \quad \left(\sum_{j=1}^{J} E^j X^j - \sum_{j=1}^{J} C^j \right) \tag{18}$$

The total amount of product produced can be determined by using Eq. (10) whereas the production cost C^j incurred in process j can be determined as shown in Eq. (19)

$$C^j = \begin{cases} \sum_{c=1}^{L-1} \sum_{u=1}^{N_c^j} \left(C_c^j + \left(\frac{C_{c+1}^j - C_c^j}{c_{c+1}^j - c_c^j} \right) \left(x_{c-(c+1),u}^j - c_c^j \right) \right) & c_c^j \le x_{c-(c+1),u}^j \le c_{c+1}^j \\ 0 & x_{c-(c+1),u}^j < c_c^j \end{cases} \quad \forall j = 1, 2, \ldots, J \tag{19}$$

In the above equation, C_c^j and C_{c+1}^j denote the production cost whereas c_c^j and c_{c+1}^j denote the production capacity of level c and $c+1$ of process j. The production cost determined in Eq. (19) is from the line connecting the points $\left(c_c^j, C_c^j\right)$ and $\left(c_{c+1}^j, C_{c+1}^j\right)$ as shown in Fig. 2. It should be noted that for a feasible solution, $x_{c-(c+1),u}^j$ is always greater than or equal to c_c^j as the effective domain of $x_{c-(c+1),u}^j$ is $c_c^j \le x_{c-(c+1),u}^j \le c_{c+1}^j$. Similar to the investment cost, the production cost is set to zero if $x_{c-(c+1),u}^j$ is less than c_c^j as the repairing mechanism will ensure that $x_{c-(c+1),u}^j$ is also set to zero indicating no production from this unit. As the evolutionary algorithms are inherently optimization techniques developed for unconstrained optimization problems, the constraints are incorporated into the objective function using a static penalty function as shown in Eq. (20)

$$Maximize \quad \left(\sum_{j=1}^{J} E^j X^j - \sum_{j=1}^{J} C^j \right) - \left(f_p \left(\sum_{t=1}^{T} P_t^{raw} + P^{Inv} + \sum_{i=1}^{I} P_i^{uni} \right) \right) \tag{20}$$

In Eq. (20), f_p denotes the static penalty factor with a large value so as to ensure that satisfaction of the constraint is given higher priority than improvement in the profit.

The objective function in Eq. (20) ensures that (i) a feasible solution is selected over an infeasible solution, (ii) between two feasible solutions, the solution with a higher profit is selected, and (iii) between two infeasible solutions, the solution with lower violation is selected. In this article, 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. An interested reader can study the performance of the various other constraint handling techniques including adaptive penalties that have been proposed in the literature. The p codes for evaluating the objective function are available at https://goo.gl/LQ8WrD.

5. Case studies

In this section, the performance of the proposed multi-unit based strategy is demonstrated on the production planning of a petrochemical industry involving 24 products using a set of 54 processes. For each process, the production cost and investment cost data is available for three (low, medium and high) different production capacities. The set of processes available to manufacture a product, selling price of the products and the production capacities of the three levels along with production and investment cost of each level is given in Tables A1–A4. The requirement of raw material in each process is also given in these tables. It can be noted that the product T1 can be produced from three different processes (S1, S2, S3) whereas the product T4 can be produced from only a single processes (S8) only. The products T4, T5, T8, T12, T13, T16, and T23 can be produced by only one process whereas the products T2, T3, T7, T9, T10, T19, T22, and T24 can be produced by two processes while the products T1, T11, T14, T15, T17, T18, T20 can be produced by three different processes. Product T21 can be produced by four processes whereas product T6 can be produced by six processes. For this case study, the objective is to maximize the annual profit and to determine the production plan associated with it. The constraints such as availability of two key raw materials (propylene and ethylene) and the investment budget should be satisfied in determining the profit. In literature [1,13,26], eight cases are given in which Case 1–Case 4 should satisfy the unique process requirement whereas Case 5–Case 8 do not require the satisfaction of unique process requirement. The quantity of key raw materials and the investment budget available for eight cases is as shown below

- For Case 1 and Case 5, the investment budget is limited to $1 billion whereas the availability of key raw materials is limited to 0.5 million tons/year.
- For Case 2 and Case 6, the investment budget is limited to $1 billion whereas the availability of key raw materials is limited to 1 million tons/year.
- For Case 3 and Case 7, the investment budget is limited to $2 billion whereas the availability of key raw materials is limited to 0.5 million tons/year.
- For Case 4 and Case 8, investment budget is limited to $2 billion whereas availability of key raw materials is limited to 1 million tons/year.

Based on the investment budget that is available and the investment cost of various processes in Tables A1–A4, the number of maximum of units can be determined using Eq. (8) and is shown in Fig. 4. The total number of decision variables in Case 1, Case 2, Case 5 and Case 6 (with an investment budget of $1billion) is determined to be 1287 using Eq. (7). Similarly, the number of maximum units

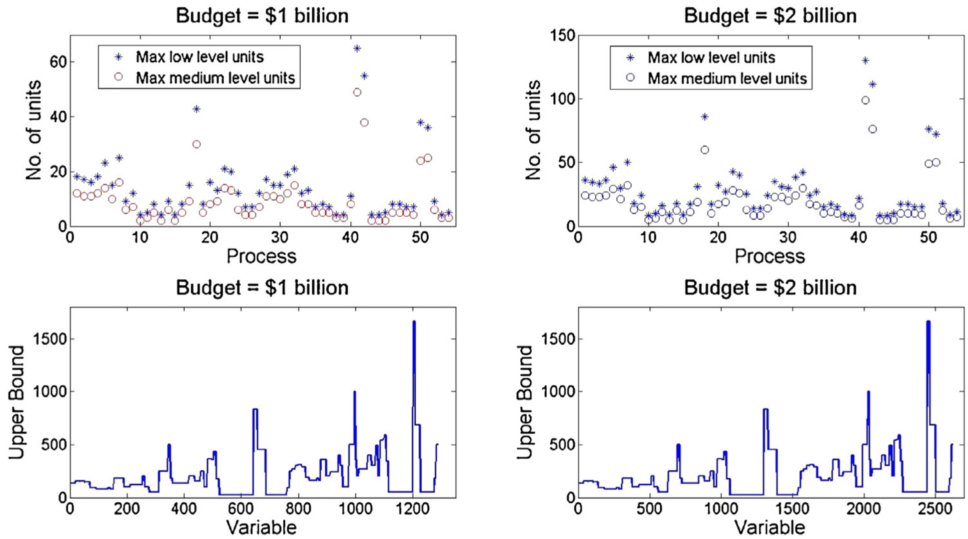

Fig. 4. Maximum number of units and the upper bounds of the variables.

and the total number of decision variables can be determined for the other four cases. In these four cases, the total number of decision variables is 2624. The lower bound for all the variables is set to zero to incorporate the possibility of not establishing the unit for production. The upper bound for the variables is as shown in Fig. 4.

The population size in ABC has been set to 5 after experimenting with other population sizes. Conventionally, the parameter "limit" in ABC has been recommended to be set to dimension of the problem x employed bees. However, it has been recently reported [29] that "limit" is a sensitive parameter and has to be suitably tuned. We have tried ten different values of limit (0, 100, 250, 500, 750, 1000, 1250, 1500, infinity, and dimension of the problem x employed bees) with five different population size (5, 10, 20, 50 and 100). From these 50 combinations (executed for each of the eight cases for the specified number of runs), it was concluded that only in two cases, the effect of "limit" was significant in the discovery of feasible solutions. The best results were obtained for "limit" being equal to 500, 750, 1000, 1250, 1500, infinity, and dimension of the problem x employed bees.

The number of population in s-TLBO, MPEDE and DNLPSO has been set to 100. The ratio between indicator and whole population is set to 0.2 whereas a value of 20 is taken for the generation gap. In DNLPSO, the refreshing gap and regrouping period are set to 7 and 10 respectively whereas the parameter for determination of velocity bound is set to 2. Both the acceleration parameters are set to 1.49445 * rand (as specified by the authors of the DNLPSO) and the two parameters to calculate the inertia factor are set to 0.9 and 0.1 respectively. It is ensured that all the algorithms employ a maximum of 60,100 functional evaluations.

Due to the stochastic nature of evolutionary algorithms [38], each algorithm is run for 26 different random seeds leading to 832 instances (8 Cases x 26 runs x 4 algorithms). In the following discussion, we demonstrate the performance of the proposed strategy on these eight cases [1,13,26]. The codes of DNLPSO and MPEDE are available online and have been used for this work. The codes for s-TLBO was built in–house as per the pseudo code explained earlier whereas the code for ABC was built in–house as per the pseudo code given in Algorithm 1 of literature [28]. The profit for each of the eight cases for all the 26 different random runs of the four algorithms are shown in Fig. 5 whereas Table 1 contains the statistical analysis of the 26 runs.

Since the problem of maximization of profit is solved as the minimization of the negative of the profit, any value of objective function which is greater than zero indicates an infeasible value.

Table 1
Statistical Analysis of the 26 runs for the eight cases by the four algorithms.

Case		s-TLBO	MPEDE	DNLPSO	ABC
Case 1	Best	−683.03	−639.97	−408.57	1E + 93
	Worst	−518.62	−446.53	−260.44	1E + 102
	Mean	−624.53	−541.12	−341.22	3.9E + 100
	Median	−631.25	−541.42	−341.69	1E + 96
	St.dev	39.25	49.56	39.03	2E + 101
Case 2	Best	−820.49	−762.98	−482.61	1E + 93
	Worst	−673.58	−538.46	−272.03	1E + 102
	Mean	−761.81	−649.19	−352.22	3.9E + 100
	Median	−758.16	−646.11	−355.72	1E + 96
	St.dev	34.17	55.98	48.67	2E + 101
Case 3	Best	−1024.56	−947.3	−715.5	1E + 102
	Worst	−780.12	−731.5	−285.23	1E + 105
	Mean	−927.4	−817.05	−541.14	6.5E + 104
	Median	−934.31	−801.74	−523.94	1E + 105
	St.dev	66.95	57.93	115.34	4.8E + 104
Case 4	Best	−1292.25	−1074.97	−780.48	1E + 102
	Worst	−1056.49	−744.93	−305.46	1E + 105
	Mean	−1186.42	−946.8	−587.13	6.5E + 104
	Median	−1189.17	−960.69	−610.8	1E + 105
	St.dev	55.44	85.34	110.08	4.8E + 104
Case 5	Best	−714.29	−643.65	−447.78	−329.64
	Worst	−573.71	−479.18	−277.6	1.04E + 21
	Mean	−661.1	−588.21	−358.88	1.08E + 20
	Median	−664.33	−582.44	−360.4	−158.81
	St.dev	36.42	36.97	46.13	2.91E + 20
Case 6	Best	−823.65	−803.78	−565.21	−387.13
	Worst	−758.91	−587.32	−307.1	9.8E + 20
	Mean	−793.17	−719.85	−378.32	4.46E + 19
	Median	−798.74	−727.1	−375.94	−189.64
	St.dev	18.16	55.66	60.63	1.92E + 20
Case 7	Best	−1118.28	−1070.54	−711.95	4.12E + 23
	Worst	−957.23	−808.81	−399.63	1.4E + 24
	Mean	−1042.06	−927.24	−575.14	8.11E + 23
	Median	−1043.53	−914.9	−590.2	7.89E + 23
	St.dev	41.83	69.27	82.33	2.65E + 23
Case 8	Best	−1420.48	−1362.59	−785.26	3.75E + 23
	Worst	−1255.66	−1026.01	−518.95	1.36E + 24
	Mean	−1343.22	−1164.83	−616.74	7.69E + 23
	Median	−1346.32	−1175.48	−617.41	7.65E + 23
	St.dev	42.08	87.81	68.9	2.4E + 23

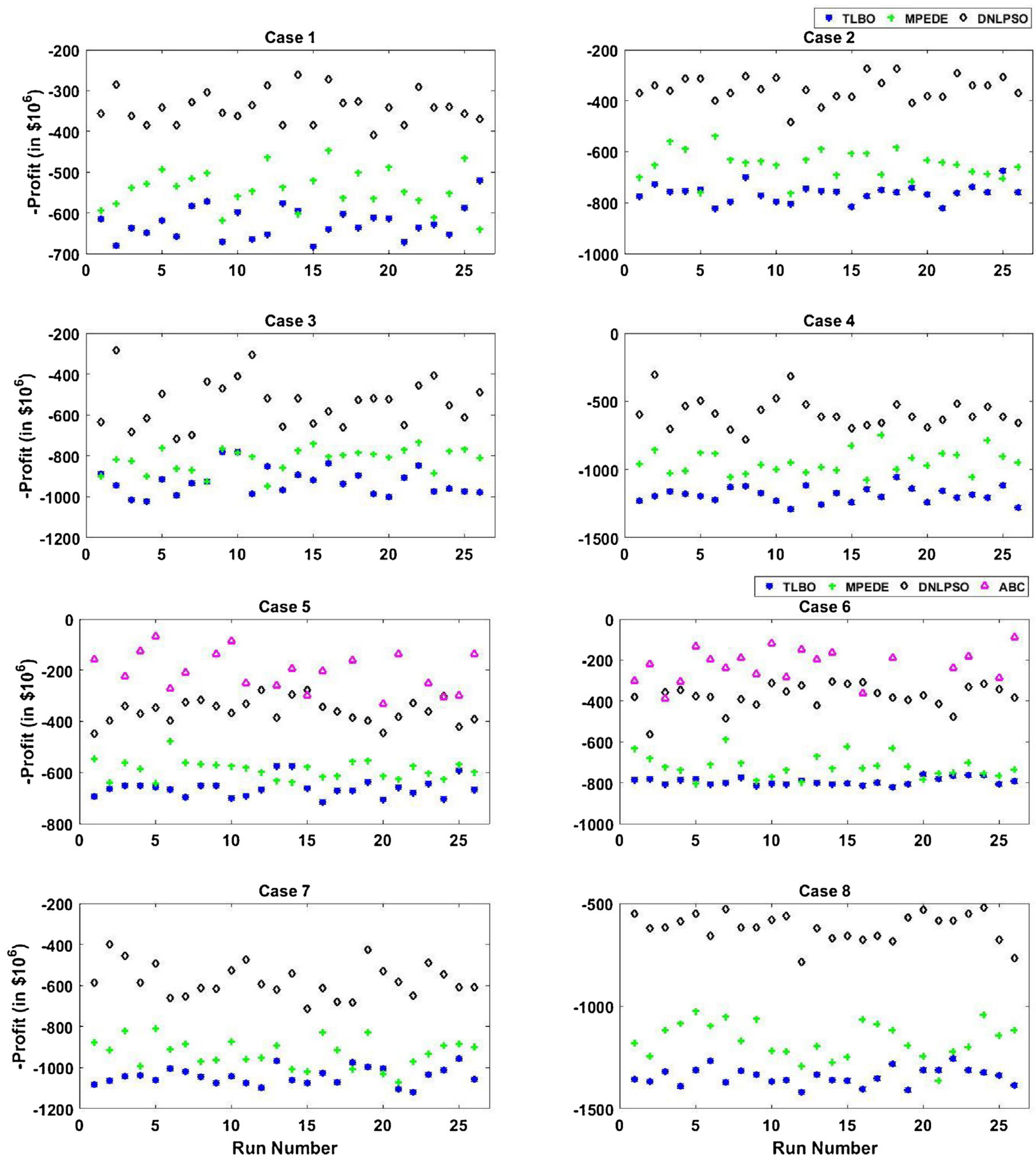

Fig. 5. Objective function determined in each of the 26 runs for the eight cases by the four algorithms.

In Fig. 5, only the feasible runs are plotted for better representation, as the fitness value in runs that are not able to discover even a single feasible solution is very poor (corresponds to a high positive value). It can be observed from Fig. 5 that ABC is able to find a feasible solution only in Case 5 and Case 6 whereas all the three other algorithms are able to discover a feasible solution in all the 26 runs. Even in these two cases, ABC is unable to determine feasible solutions in all the 26 runs. This is the reason for there being no point corresponding to ABC in plots corresponding to Case 1–Case 4 and Case 7–Case 8. In Case 5 and Case 6, ABC was able to determine feasible solutions in 20 of the 26 runs which are shown in Fig. 5.

From Table 1, it can be observed that the best value in all the eight cases is determined by s-TLBO followed by MPEDE and DNLPSO respectively. This trend is also observed in the worst, mean, median of the 26 runs across all the eight cases. The standard deviation values of the best solution in among the 26 runs is the least for s-TLBO (except Case 3). The performance of the proposed multi–unit strategy for the best run of s-TLBO for all the eight cases is shown in Fig. 6. It shows the best fitness value along with the average fitness value of the entire population with respect to the 300 generations (60,100 functional evaluations). In the initial phase of the algorithm, the solutions are not feasible and due to the high value of penalty, we have chosen to plot the fitness of the best solution only after a feasible is determined. Similarly the average fitness of the population has been plotted after the entire population becomes feasible. This explains the discontinuity in the initial phase of all the plots.

As expected, in all the eight cases, the best fitness value and the average fitness value increases monotonically and settles to a final value. This is because a new solution is included in s-TLBO only if it is better than the solution that is used to generate the new solution. It can be observed that the proposed strategy is able to determine feasible solutions in less than six generations. Similarly, the entire population becomes feasible in less than twelve genera-

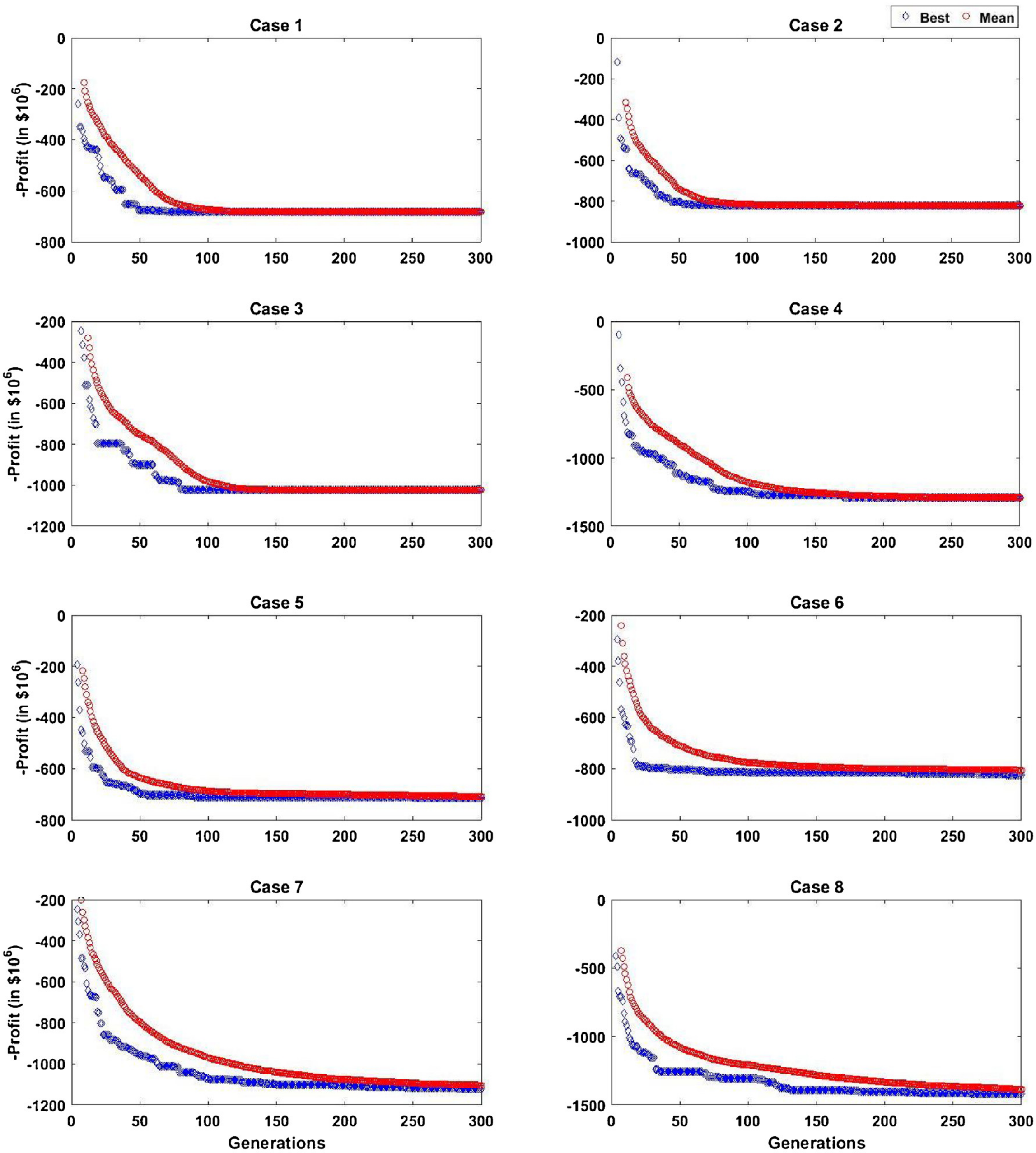

Fig. 6. Convergence Curve of s-TLBO for the best run.

tions which shows the efficacy of the proposed strategy and s-TLBO to solve this production problem despite the presence of complex unique process constraint. It should be noted that the proposed s-TLBO based strategy is able to determine all these solutions without being provided with any kind of initial guess from the previously available solutions in literature. However, caution should be exercised in extending this inference of s-TLBO being better than other three algorithms for other types of problems as s-TLBO has been reported to have an origin bias [39]. An inherent benefit of the evolutionary algorithms is that these provide a set of solutions which are closer to the best solution. The fitness value of all the 100 population members in the final generation is given in Fig. 7. In Fig. 7, the population members have been sorted based on the fitness value. It can be seen that in Case 1–Case 4, the fitness function value of a majority of the population members is approximately equal thereby providing additional solutions which are closer to the best solution determined by s-TLBO. This can enable the user to select solutions which are closer to the best determined solution and are able to satisfy additional soft constraints or objectives. Unlike in Case 1–Case 4, all the members of the last generation have not converged to the fitness of the best population member in Case 5–Case 8. This should not be interpreted to consider that the algorithm has not converged. The convergence of the algorithm can be observed in Fig. 6 in which there is no significant improvement in the value of the objective function in the last few generations. Increasing the functional evaluations may or may not lead to convergence of all the population members to the best solution.

The profit obtained in each case along with the resource utilization is given in Table 2 while the detailed optimal production plan is given in Table 3. The profit obtained from each of the product for all the four cases is shown in Fig. 8. These results are generated by executing all the algorithms with the incorporation of the solution reported in literature [26] as one member of the initial population and a termination criterion of 60,100 functional evaluations. For

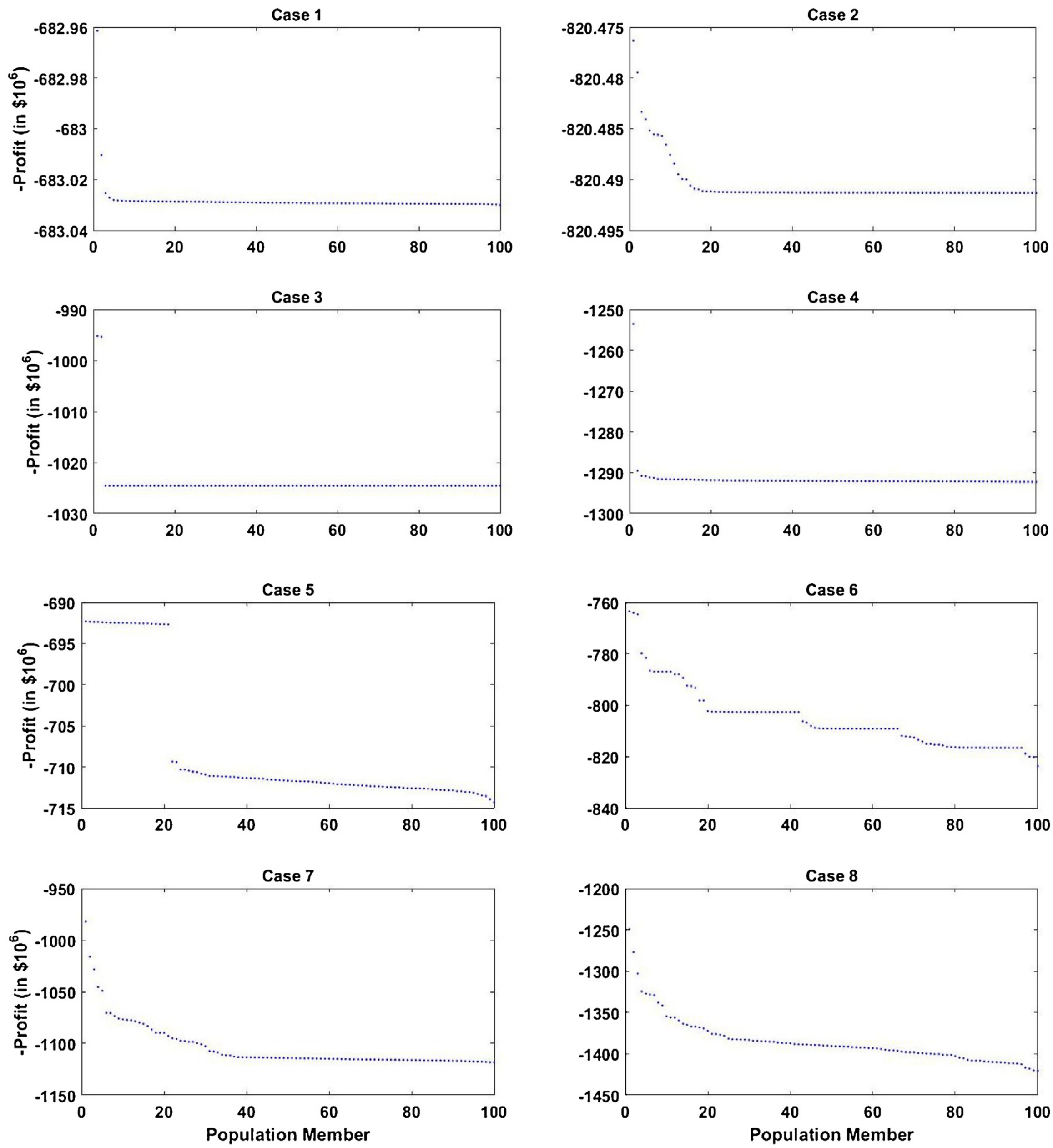

Fig. 7. Objective function value of the population members in the last generation.

all the cases with except Case 1 and Case 8, it was observed that s-TLBO was able to discover the best solution among the 26 runs. In these two cases, MPEDE was able to determine a better solution than s-TLBO. In Table 2, R_1 and R_2 denote the quantity of the raw materials available whereas B denotes the monetary resource that is available for the production planning. From Table 2, it can be observed that the proposed multi–unit strategy based on continuous variables leads to a maximum increase in profit of 12.1% in Case 3. In many cases, it can be observed that the proposed strategy leads to an increase in profit despite using lower resources. This is possible because the proposed multi–unit strategy does not artificially limit the production to $\left(m_j + h_j\right)$ but enables the production of higher profitable products. This can be verified from the production plans provided in Table 3. For example, the low, medium and high capacity values of process S3 (as shown in Table A1) are 77.5, 155 and 310 respectively. Thus the multiple–level strategy in literature would lead to a maximum production of 465 (=155 + 310) whereas the proposed multi–unit strategy is able to employ two units (in Case 1) in between the medium and high level capacity to produce an amount of 526.86 from process S3. This benefit of multiple–units can be observed across multiple processes such as S31 in Case 2 and Case 6, S36 in Case 3, S48 in Case 4 and Case 8. For Case 1–Case 4, it can be observed that a product is not produced from more than one process, though it can be produced from more than one unit of the same process. This is in view of the presence of unique process requirement and the solution satisfies this requirement. However, in Case 5–Case 8 there is no unique process requirement and hence a product can be produced from multiple process. For example, product T21 is produced from a single unit of process S46 and multiple units of process S48. From Fig. 8, it can be observed that product T21 contributes the maximum to the profit in Case1 to Case 8.

Table 2
Comparison of profit and resource utilization for Case 1–Case 8.

Case	Items	Amount of resource available	Single Level MILP Literature [1]	Multi-Level Strategy with TLBO [13]	Multi-Level Strategy with MILP [26]	Proposed Strategy	% Increase in Profit
Case 1	B ($ 10^6)	1000	1000	1000	995.7	994.5	0.12
	R_1 (10^3 tons/year)	500	500	500	500	500	
	R_2 (10^3 tons/year)	500	500	500	500	500	
	Profit ($ 10^6/year)	–	**692.8**	**692.8**	**715.9**	**716.8**	
Case 2	B ($ 10^6)	1000	1000	1000	1000	991.4	4.1
	R_1 (10^3 tons/year)	1000	847.2	971.8	971.8	1000	
	R_2 (10^3 tons/year)	1000	660.5	571.0	571.0	571.0	
	Profit ($ 10^6/year)	–	**759.7**	**796.5**	**796.5**	**829.0**	
Case 3	B ($ 10^6)	2000	1952	1999.8	2000	1982.4	12.1
	R_1 (10^3 tons/year)	500	500	500	500	449.1	
	R_2 (10^3 tons/year)	500	500	500	500	500	
	Profit ($ 10^6/year)	–	**894.3**	**953.8**	**1040.2**	**1165.5**	
Case 4	B ($ 10^6)	2000	1989.3	2000	2000	2000	8.6
	R_1 (10^3 tons/year)	1000	1000	975.1	1000	1000	
	R_2 (10^3 tons/year)	1000	979.5	979.5	1000	957.2	
	Profit ($ 10^6/year)	–	**1111.5**	**1202.2**	**1287.7**	**1399.1**	
Case 5	B ($ 10^6)	1000	993	990.2	1000	1000	0
	R_1 (10^3 tons/year)	500	500	500	500	500	
	R_2 (10^3 tons/year)	500	500	500	500	500	
	Profit ($ 10^6/year)	–	**726.0**	**730.1**	**731.9**	**731.9**	
Case 6	B ($ 10^6)	1000	1000	1000	1000	995.2	1.2
	R_1 (10^3 tons/year)	1000	1000	1000	1000	1000	
	R_2 (10^3 tons/year)	1000	571.0	571.0	571.0	865.2	
	Profit ($ 10^6/year)	–	**834.3**	**834.3**	**834.3**	**843.9**	
Case 7	B ($ 10^6)	2000	1991.7	1985.3	2000	1995.4	2.4
	R_1 (10^3 tons/year)	500	500	500	500	500	
	R_2 (10^3 tons/year)	500	500	500	500	495.4	
	Profit ($ 10^6/year)	–	**1173.1**	**1177.3**	**1191.9**	**1220.8**	
Case 8	B ($ 10^6)	2000	1998.1	1998.1	2000	2000	1.1
	R_1 (10^3 tons/year)	1000	1000	1000	1000	1000	
	R_2 (10^3 tons/year)	1000	954.6	954.6	939.0	944.6	
	Profit ($ 10^6/year)	–	**1452.8**	**1452.8**	**1465.0**	**1480.8**	

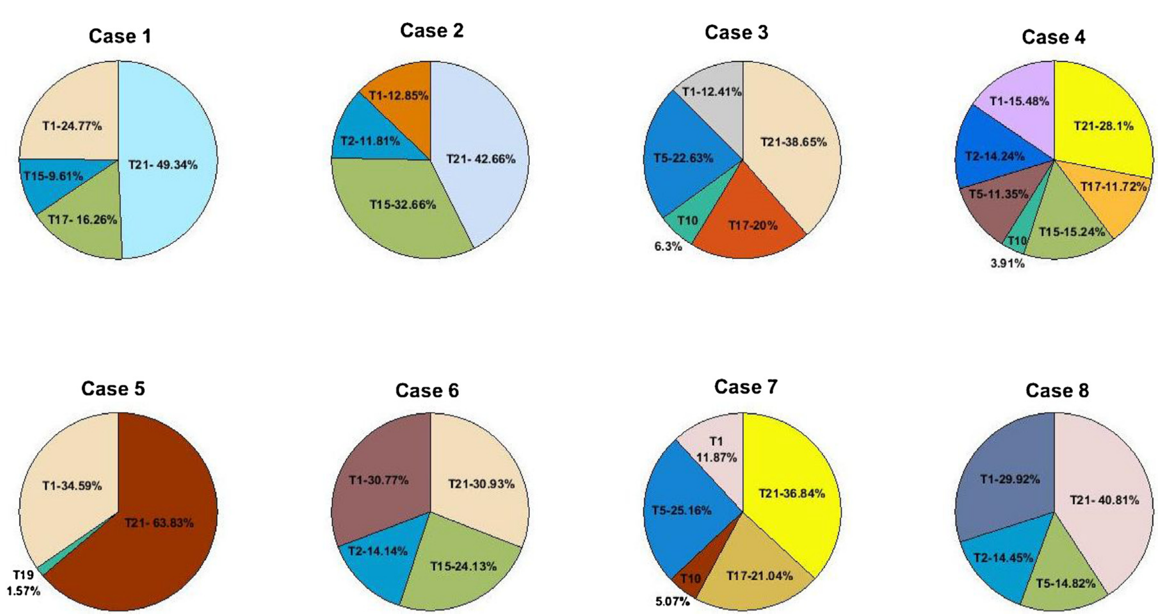

Fig. 8. Contribution of the individual products to the profit.

Table 3
Optimal production plans for Case 1–Case 8.

	Product number	Process number	Decision variables	X^jNet production (103 ton/yr)
Case 1	T1	S3	254.56	526.86
			272.30	
	T15	S31	188.92	188.92
	T17	S36	540	540
	T21	S48	450	1130
			680	
Case 2	T1	S3	310	310
	T2	S4	290	290
	T15	S31	317.40	717.40
			400	
	T21	S48	450	1130
			680	
Case 3	T1	S1	262.06	262.06
	T5	S9	80	400
			160	
			160	
	T10	S22	100	100
	T17	S36	365	1270
			365	
			540	
	T21	S48	450	1580
			450	
			680	
Case 4	T1	S3	155	464.99
			309.99	
	T2	S4	145	435
			290	
	T5	S9	159.99	159.99
	T10	S22	87.25	87.25
	T15	S31	400	400
	T17	S36	539.99	539.99
	T21	S48	449.99	2186.26
			450	
			606.28	
			679.99	
Case 5	T1	S1	217.09	527.09
		S3	310	
	T19	S41	25	64.09
			39.09	
	T21	S46	618.26	618.26
		S48	450	1130
			680	
Case 6	T1	S3	310	620
			310	
	T2	S4	290	290
	T15	S31	130.90	852.37
			321.47	
			400	
	T21	S48	680	680
Case 7	T1	S3	310	310
	T5	S9	160	320
			160	
	T10	S22	100	100
	T17	S36	365	905
			540	
		S37	550	550
	T21	S46	618.26	618.26
		S48	450	1130
			680	
Case 8	T1	S2	299.99	299.99
		S3	310	310
	T2	S4	289.99	289.99
	T5	S9	144.10	144.10
	T21	S46	449.99	1129.98
			679.99	
		S47	551.55	551.55
		S48	449.99	1809.97
			679.99	
			679.99	

6. Conclusion

In this article, we have demonstrated the shortcomings of the single–level MILP formulation and the multi–level elitist TLBO based strategy and multi–level MILP formulation proposed in the literature for determining the optimal production plan that lead to the determination of suboptimal solutions. In this article, we have proposed a multi–unit production strategy that does not artificially restrict the production and leads to better utilization of the limiting resources and results in improved profit. An important feature of this strategy is that it can also be used for non–linear cost functions and does not utilize any discrete decision variables which enables the direct application of computational intelligence based optimization techniques that are usually developed for continuous variables. Another contribution of this work is that the proposed strategy has been demonstrated in the context of a petrochemical plant that has been previously addressed in literature. In this work, a preliminary comparison of four different optimization techniques, viz., artificial bee colony, dynamic neighbourhood learning particle swarm optimizer, multi–population ensemble differential evolution and a refined teaching–learning–based optimization also has been done. Of the four algorithms, it was observed that s-TLBO is most suited for this production planning problem. The proposed multi–unit production strategy is able to determine better solutions than those reported in literature with a maximum improvement of over 12%. In addition to providing the optimal solutions, the use of computational intelligence optimization algorithm provides near best solutions which can provide flexibility to the user to incorporate other soft objectives. This work also helps to establish the performance of the s-TLBO algorithm on problems involving very large number of variables in the presence of complex domain hole and unique process constraint without using any discrete variables. The proposed strategy is generic and can be used in other production planning problems that may have more than three levels.

Future work in this direction can utilize the proposed multi–unit strategy to rigorously evaluate the performance of other computational intelligence techniques. Moreover, due to the probabilistic nature of the computational intelligence techniques used in this work, there is no theoretical guarantee of optimality on the solutions that are determined. Hence, exact methods that rely on mathematical programming and provide guaranteed globally optimal solutions can be explored for the solution of this multi–unit production planning problem. As the primary aim of this work is to overcome the drawbacks of the formulations/strategies in literature, we have chosen to not modify the problem statement. However, efficient production plans involving multiple objectives such as safety, environmental impact, and ease of retrofitting along with uncertainty in the various parameters can be explored as future work.

Appendix A.

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

Product	Product no. i	Sale price ($/ton)	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_l^j	C_m^j	C_h^j	V_l^j	V_m^j	V_h^j
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 butanol byproduct)	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 butanol byproduct)	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 ($/ton)	Process no. *j*	Process name	Ethylene used/ton b_{j1}	Capacity (10^3 ton/yr)			Production cost ($\$10^6$/yr)			Investment ($\$10^6$)		
						l^j	m^j	h^j	C_l^j	C_m^j	C_h^j	V_l^j	V_m^j	V_h^j
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 ($/ton)	Process no. j	Process name	Methane used/ton b_{j3}	Capacity (10^3 ton/yr)			Production cost ($\$10^6$/yr)			Investment ($\$10^6$)		
						l^j	m^j	h^j	C_l^j	C_m^j	C_h^j	V_l^j	V_m^j	V_h^j
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 ($/ton)	Process no. j	Process name	Ethylene used/ton b_{j1}	Capacity (10^3 ton/yr)			Production cost ($\$10^6$/yr)			Investment ($\$10^6$)		
						l^j	m^j	h^j	C_l^j	C_m^j	C_h^j	V_l^j	V_m^j	V_h^j
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. Alfares, A. Al-Amer, An optimization model for guiding the petrochemical industry development in Saudi Arabia, Eng. Optim. 34 (2002) 671–687.
[2] G.K. Al-Sharrah, I. Alatiqi, A. Elkamel, Modeling and identification of economic disturbances in the planning of the petrochemical industry, Ind. Eng. Chem. Res. 42 (2003) 4678–4688.
[3] Al–Faresi Global petrochemical industry worth over USD 600 billion, Last accessed June 1, 2018 (Available from https://goo.gl/sQoNsX).
[4] S.–G. Yoon, S. Park, J. Lee, P.M. Verderame, C.A. Floudas, Evaluation of synergy effect in the merger of companies in a petrochemical complex, Comput. Aided Chem. Eng. 27 (2009) 2061–2066.
[5] 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.
[6] K. Al-Qahtani, A. Elkamel, Multisite refinery and petrochemical network design: optimal integration and coordination, Ind. Eng. Chem. Res. 48 (2008) 814–826.
[7] 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 (1998) 1037–1049.
[8] T. Lee, J.-H. Ryu, I.-B. Lee, H.-K. Lee, A synchronized feed scheduling of petrochemical industries simultaneously considering vessel scheduling and storage tank management, Ind. Eng. Chem. Res. 48 (2009) 2721–2727.
[9] S.-G. Yoon, S.B. Park, S. Park, J. Lee, P.M. Verderame, C.A. Floudas, Synergy in Mergers of petrochemical companies within a complex considering purchasing and selling advantage with process integration, Ind. Eng. Chem. Res. 47 (2008) 5556–5567.
[10] 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.
[11] L. He, J. Fengjun, L. Yi, D. Jinxue, X. Xu, Evaluation and optimization of the spatial organization of the petrochemical industry in China, J. Geog. Sci. 23 (2013) 163–178.
[12] G.K. Al-Sharrah, D. Edwards, G. Hankinson, A new safety risk index for use in petrochemical planning, Process Safety Environ. Protection 85 (2007) 533–540.
[13] 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.
[14] D.F. Rudd, Modelling the development of the intermediate chemicals industry, Chem. Eng. J. 9 (1975) 1–20.
[15] M.A. Stadtherr, D.F. Rudd, Systems study of the petrochemical industry, Chem. Eng. Sci. 31 (1976) 1019–1028.
[16] G.K. Al-Sharrah, I. Alatiqi, A. Elkamel, E. Alper, Planning an integrated petrochemical industry with an environmental objective, Ind. Eng. Chem. Res. 40 (2001) 2103–2111.
[17] A. Sophos, E. Rotstein, G. Stephanopoulos, Thermodynamic bounds and selection of technologies in the petrochemical industry, Chem. Eng. Sci. 35 (1980) 1049–1065.
[18] A.G. Jiménez, D.F. Rudd, Use of a recursive mixed–integer programming model to detect an optimal integration sequence for the Mexican petrochemical industry, Comput. Chem. Eng. 11 (1987) 291–301.
[19] J.K. Mikkelsen, D.F. Rudd, Development of a Norwegian petrochemical industry, Eng. Costs Prod. Econ. 5 (1981) 163–177.
[20] P.R. Stokke, W.K. Ralston, T.A. Boyce, I.H. Wilson, Scenario planning for Norwegian oil and gas, Long Range Plann. 23 (1990) 17–26.
[21] S.-G. Yoon, S. Park, J. Lee, P.M. Verderame, C.A. Floudas, Evaluation of synergy effect in the horizontal merger of companies in a petrochemical complex, Ind. Eng. Chem. Res. 48 (2009) 11017–11033.
[22] 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.
[23] 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.
[24] S.V. Inamdar, S.K. Gupta, D.N. Saraf, Multi–objective optimization of an industrial crude distillation unit using the elitist non–dominated sorting genetic algorithm, Chem. Eng. Res. Des. 82 (2004) 611–623.
[25] W. Zhao, D. Chen, S. Hu, Optimizing operating conditions based on ANN and modified GAs, Comput. Chem. Eng. 24 (2000) 61–65.
[26] R. Kadambur, P. Kotecha, Optimal production planning in a petrochemical industry using multiple levels, Comput. Ind. Eng. 100 (2016) 133–143.
[27] D. Karaboga, An idea based on honey bee swarm for numerical optimization, in: Technical Report –TR06, 2005.
[28] 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.
[29] N. Veček, S.-H. Liu, M. Črepinšek, M. Mernik, On the importance of the artificial bee colony control parameter 'Limit', Inf.Technol. Control 46 (2017) 566–604.
[30] M. Nasir, S. Das, D. Maity, S. Sengupta, U. Halder, P.N. Suganthan, A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization, Inf. Sci. 209 (2012) 16–36.
[31] S. Chauhan, P. Kotecha, Impact of bounding procedure in computational intelligence algorithms for production planning problems, in: Technical Report –TR01, 2017.
[32] M.Z. Ali, N.H. Awad, P.N. Suganthan, Multi–population differential evolution with balanced ensemble of mutation strategies for large–scale global optimization, Appl. Soft Comput. 33 (2015) 304–327.
[33] 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.
[34] M. Črepinšek, S.-H. Liu, L. Mernik, A note on teaching–learning–based optimization algorithm, Inf. Sci. 212 (2012) 79–93.
[35] M. Črepinšek, S.-H. Liu, L. Mernik, M. Mernik, Is a comparison of results meaningful from the inexact replications of computational experiments? Soft Comput. 20 (2016) 223–235.
[36] M. Črepinšek, S.-H. Liu, M. Mernik, Replication and comparison of computational experiments in applied evolutionary computing: common pitfalls and guidelines to avoid them, Appl. Soft Comput. 19 (2014) 161–170.
[37] J.K. Pickard, J.A. Carretero, V.C. Bhavsar, On the convergence and origin bias of the teaching–learning–based–optimization algorithm, Appl. Soft Comput. 46 (2016) 115–127.
[38] R. Kommadath, C. Sivadurgaprasad, P. Kotecha, Single phase multi-group teaching learning algorithm for computationally expensive numerical optimization (CEC 2016), IEEE Congress on Evolutionary Computation (CEC) (2016).
[39] R. Kommadath, P. Kotecha, Teaching Learning Based Optimization with focused learning and its performance on CEC2017 functions, IEEE Congress on Evolutionary Computation (CEC) (2017).