Single Level Production Planning in Petrochemical Industries using Moth-flame Optimization

Sandeep Singh Chauhan
Department of Chemical Engineering
Indian Institute of Technology Guwahati
Guwahati – 781 039, India

Prakash Kotecha
Department of Chemical Engineering
Indian Institute of Technology Guwahati
Guwahati – 781 039, India
pkotecha@iitg.ernet.in

***Abstract*— The determination of optimal production planning is a combinatorial optimization problem and requires specific strategies for evolutionary computation techniques to be successful. In this work, we provide such a strategy that enables the use of evolutionary algorithms to efficiently solve the single level production planning. The efficacy of the strategy is demonstrated with the recently proposed Moth-flame optimization which also enables in evaluating the performance of this technique. It was observed that the Moth-flame optimization is able to consistently solve the production planning problem with the suggested strategy.**

Keywords— Production Planning; Moth-flame Optimization; Combinatorial Optimization; Petrochemical industry

I. INTRODUCTION

The worldwide petrochemical industry is expected to be worth over $758 billion by 2022 [1]. The petrochemical industry produces products which are used as raw materials for manufacturing everyday consumer products. Production planning plays a crucial role in the petrochemical industry as it helps in the efficient use of the limited resources and also enables to satisfy various requirements in the presence of complex models and networks. With an increase in competition from renewable sources, production planning in petrochemical industry has become mandatory to remain competitive and profitable. However, production planning is a combinatorial optimization problem that can be difficult to solve with the mathematical programming techniques, especially for larger problems. A limiting feature of these techniques is that these require the optimization problem to be postulated in terms of the traditional inequalities represented as $g(x) \le 0$ and $h(x) = 0$. On the contrary, evolutionary optimization techniques do not require the optimization problem to be postulated in terms of such inequalities and provide a greater modelling flexibility as they are capable of solving the optimization problem even if it is a complete black-box [2-5]. However it should be noted that evolutionary optimization techniques can also have difficulty in solving combinatorial optimization problems, particularly while using the mathematical models developed for mathematical programming techniques. Thus it becomes necessary to develop an appropriate strategy to harness the benefits of evolutionary algorithms for solving combinatorial production planning problem. In this work, we provide an efficient framework for solving single level production planning optimization problems.

In literature, a Mixed Integer Linear Programming formulation [6] has been proposed for solving single level production planning but suffers from several drawbacks viz., (i) it cannot incorporate various non-linear cost functions without losing its MILP structure, (ii) it employs a significant number of artificial binary variables which can make the solution procedure prohibitively expensive for production planning involving a large number of processes and levels, (iii) it is limited to production planning problems in which the processes have only three capacity levels and its extension to multiple production levels is not trivial, (iv) in its naïve form, the solution techniques for MILP are not designed to determine value added solutions, and (v) it cannot be used to solve multi-objective optimization problems in a single run. In this work, we develop a strategy which can overcome all the above drawbacks of the MILP formulation. A unique feature of the developed strategy is that it does not employ any binary variable and the number of decision variables do not exponentially increase with an increase in the number of processes or capacity levels. In fact, the decision variables increase linearly with an increase in the number of processes and is independent of the number of capacity levels. Since the strategy has been specifically developed for solving with evolutionary algorithms, the determination of value added solutions, solution of multi-objective production planning problems and incorporation of various non-linear cost functions becomes possible. Additionally, to the best of our knowledge, Moth-flame optimization (MFO) technique [7] that has been very recently proposed has not been applied to any combinatorial optimization problem. Thus this work also serves as a test case to evaluate the performance of MFO on complex optimization problems. The developed strategy with its successful application using MFO has been demonstrated on a case study that has been previously used in literature [5, 6, 8] to guide the Saudi Arabia petrochemical industry.

The article is organized as follows: In Section II, we provide the problem description and provide a strategy to solve the production planning problem with an evolutionary algorithm. However, in view of scarcity of space, this entire section is provided as supplementary information [9]. In Section III, we provide a description of the recently developed Moth-flame optimization. This is followed by a discussion on the results of the developed strategy on an industrial case study from the literature. We conclude the article in Section V by summarizing the developments in this work. The article is organized as follows:

978-1-5090-2597-8/16/$31.00

III. Moth flame Optimization Algorithm

Moth-flame optimization is a stochastic population based technique that has been proposed in 2016 by Mirjalili to solve unconstrained single objective optimization problems. It is inspired from the transverse orientation mechanism employed by moths to travel in straight paths by flying at a fixed angle to celestial objects such as the moon. However, this mechanism is effective only when the source of light is far away and misleads the moths to spiral around and converge to the light source (such as artificial lights by humans) if it is closer to the moth. In MFO, two types of population are employed, viz., (i) the moth population and (ii) the flame population. The moths and flames in their respective population correspond to the solutions of the optimization problem whereas their positions (or directions) are considered to represent the decision variables. The size of the moth population (and flame population) is N x V where N is the number of moths and V is the number of decision variables. MFO requires only two user-defined parameters, viz, (i) the number of moths (or the search agents) and (ii) the maximum number of iterations to be used for terminating the algorithm. The extension of the technique to various other commonly used termination criteria is trivial. In a single iteration, the objective function is evaluated as many times the number of the moths in the population.

A detailed flowchart depicting the entire MFO algorithm is provided in Figure 1 (as supplementary information) [9] that has been prepared in accordance with the code given by its inventor. The flowchart accurately reflects the implementation and overcomes some of the minor discrepancies in the explanation of MFO in literature. In this section, we provide a brief description as additional information can be obtained from the literature. Similar to many other evolutionary techniques, MFO begins with the creation of N uniformly generated population members (or moths and denoted as M) in the bounds of the decision variables and evaluating its fitness. The moth members are sorted based on their fitness and are additionally saved as flames (denoted as F). It should be noted that the member i in the flame population need not be the same as the member i in the moth population as the flames are determined by sorting the moth population. However, every member of the moth will also be present in the flame population but at a different index depending on its fitness. Subsequent to this, an iterative procedure is employed which begins with the determination of the flame number (f) based on the following equation

$$f = round\left(N - i\left(\left(N-1\right)/I\right)\right) \tag{1}$$

In the above equation, i denotes the current iteration number, I denotes the maximum number of iteration whereas N denotes the size of the moth population. The above equation ensures that the flame number decreases with the progress in the number of iteration and is 1 in the last iteration, i.e., $i = I$. This adaptive nature on the number of flames has been reported to balance exploration and exploitation. This is followed by determining the convergence constant a using

$$a = -1 - \frac{i}{I} \tag{2}$$

It can be observed that the parameter a will always be lower than -1 but will never be lower than -2. For each of the moth i , its distance to the flame i is determined using the following equation

$$D_{i,j} = \left| M_{i,j} - F_{i,j} \right| \qquad \forall i = 1,2,..,N; \forall j = 1,2,..,V \tag{3}$$

In the above equation, $M_{i,j}$ denotes the direction j of member i (equivalent to the index i) of the moth population whereas $F_{i,j}$ indicates the direction j of member i of the flame population. As mentioned earlier, the flame population is sorted whereas moth population is not sorted and hence the member i in each of the two population need not be equal. The position of a moth i in direction j is varied as per the following equation

$$\alpha_{i,j} = D_{i,j} e^{bt_j} \cos\left(b\pi t_j\right) \qquad \forall i = 1,2,..,f; \forall j = 1,2,..,V \tag{4}$$

$$M_{i,j} = \alpha_{i,j} + F_{i,j} \qquad \forall i = 1,2,..,f; \forall j = 1,2,..,V \tag{5}$$

$$M_{i,j} = \alpha_{i,j} + F_{f,j} \qquad \forall i = f+1,..,N; \forall j = 1,2,..,V \tag{6}$$

In the above equations, b is a constant parameter whose value is 1 whereas the term t_j is determined using

$$t_j = \left(a-1\right) r_j + 1 \quad \forall j = 1,2,..,V \tag{7}$$

However, the new moth positions may not be necessarily in the bounds of the decision variables and they are bounded using the following equation where l_j and u_j indicate the lower and upper bounds of the variable j

$$\begin{aligned} M_{i,j} &= \max\left(M_{i,j}, l_j\right) \quad \forall i = 1,2,..,N; \forall j = 1,2,..,V \\ M_{i,j} &= \min\left(M_{i,j}, u_j\right) \quad \forall i = 1,2,..,N; \forall j = 1,2,..,V \end{aligned} \tag{8}$$

The fitness of the bounded moths are determined and thus there are $2N$ solutions (N flames in F and N newly generated moths in M). Among these $2N$ solutions, the best N are selected based on their fitness functions and are considered as the flame population of the next iteration. The moth population remains the same that was obtained from Equation (8). This completes one iteration of MFO and the next iteration begins with the determination of the flame number using Equation (1). The entire procedure is repeated until the termination criterion of maximum number of iterations is not satisfied. It can be observed that MFO is an easy to implement, simple algorithm with a very low computational complexity. It should be noted that MFO in its inherent form does not require the identification and removal of duplicate solutions. It should also be noted that maintaining the flame population also enables the algorithm to exhibit a monotonic convergence. Though proposed as a technique for unconstrained optimization, it can be easily extended to solve constrained problems by using any of the constraint handling methods.

IV. Results and Discussions

In this section, we demonstrate the strategy discussed in this article to solve a combinatorial production planning problem using MFO. We use the case study of determining the optimal production plans for a petrochemical industry that is available in literature [5] and used for guiding the petrochemical industry in Saudi Arabia. The data has not been reproduced in this article in view of the page limits and can be obtained from the literature. The case study consists of 54 chemical processes which can produce 24 different petrochemical products. All the process have three capacity levels and two of the raw materials used by various processes are available in limited quantities. Other data such as the production cost, investment cost and the raw material required for each process is also known along with the selling price of the products. Eight different cases of this problem have been defined in literature depending on the amount of resources available and the constraint on production through unique process. Case 1 – Case 4 require that no more than one process be employed to produce a particular product whereas Case 5 – Case 8 do not enforce this criterion. The investment budget available for Case 1 and Case 2 is \$1000 x 10^6 whereas Case 3 and Case 4 have an investment of \$2000 x 10^6. Case 1 and Case 2 differ in the quantities of the raw materials that is available for the production planning. The resource availability of all the eight cases is given in Table I under the column "Avail". The number of decision variables is 54 whereas the number of constraints is 81 with unique process constraints (57 in the absence of unique process constraints). The raw material value in Table 1 is given in terms of 10^3 tons/year whereas the budget is in terms of \$$10^6$.

TABLE I. Resource Availability and Utilization

Case	Raw Material 1		Raw Material 2		Budget	
	Avail	Utli	Avail	Utli	Avail	Utli
Case1	500	475.64	500	500.00	1000	968.08
Case2	1000	615.52	1000	723.82	1000	1000.00
Case3	500	500.00	500	474.93	2000	1965.18
Case4	1000	725.27	1000	847.16	2000	2000.00
Case5	500	434.07	500	500.00	1000	858.30
Case6	1000	882.24	1000	929.24	1000	986.00
Case7	500	500.00	500	471.99	2000	2000.00
Case8	1000	1000.00	1000	850.43	2000	1987.59

In view of the stochastic nature of MFO, all the eight cases are run for 51 runs by varying the seed of the random number generator ('*twister*') in MATLAB2015a from 1-51. The population size was 25 and the number of maximum iterations was 1000 thereby leading to 25000 functional evaluations. The statistical results of the 51 runs are consolidated for each case in Table II. Among the 51 x 8 instances, MFO was not able to determine a feasible solution in 14 instances (4 runs in Case, 6 runs in Case, 3 and 4 runs in Case 4). Since the infeasible solution has a very high fitness value, the calculation of worst, mean and standard deviation for these three cases have been reported based on the feasible runs. The convergence curve for the run with the best objective function value determined in all the 51 runs for the eight cases is shown in Figure 2.

TABLE II. Stastical results of 51 Runs

	Best	Worst	Mean	Std. dev
Case 1	**-557.37**	-108.20	-348.84	101.94
Case 2	**-619.21**	-149.40	-404.67	108.35
Case 3	**-791.78**	-331.80	-550.12	119.37
Case 4	**-908.51**	-475.05	-700.67	103.35
Case 5	**-531.43**	-159.21	-379.44	87.88
Case 6	**-636.65**	-227.60	-477.59	87.92
Case 7	**-876.20**	-319.92	-597.62	131.33
Case 8	**-1128.4**	-616.20	-837.43	125.90

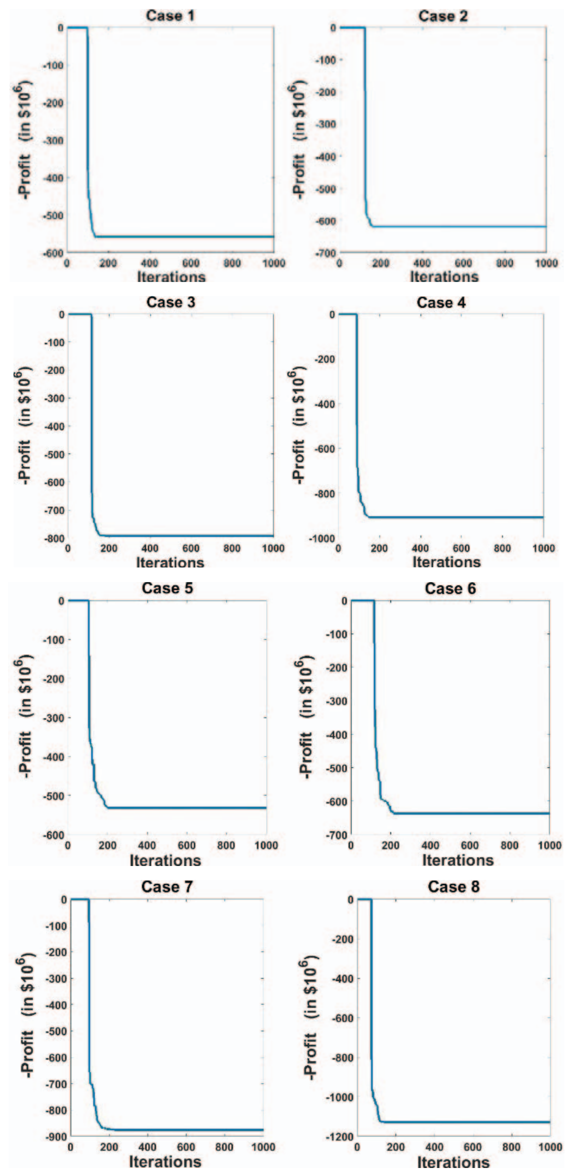

Fig. 2. Convergence curves for the best cases

It can be observed that MFO is able to quickly determine the best solution in very few iterations. Despite this, the number of iterations were set to the large value of 1000 to check if all the runs are able to determine optimal solutions. For the sake of brevity, the production plans corresponding to the best run are provided as supplementary information (Table III and Table IV) [9]. The amount produced is given in 10^3 ton/year. It can be observed from Table III, in all the four cases, the unique process constraint is satisfied i.e., a single product is not produced from more than one process. For example, all the seven processes selected (from 54 processes) in Case 1 produce seven unique products. This observation is valid for all the three other cases which employ the unique process constraint. It can also be observed that the products T1, T2, T15, and T21 are produced in all the four cases. However, these products are not produced by the same process in all the four cases. For example, the product T1 in Case 4 is produced by P1 whereas it is produced by process P3 in the three other cases. In many cases, the product is produced till the permissible level of maximum capacity level.

In Table IV, it can be observed that an identical product is produced from more than one process. For example, it can be observed in Case 6 that the product T2 is being produced from process P4 and P5. Similarly, T17 in Case 7 is produced by two different processes whereas T1 is produced in Case 8 by all the three process. This is because Case 5 – Case 8 did not have the unique process constraint. In all the eight cases, it can be observed that a maximum of 12 processes (in Case 5 and Case 6) are selected among the 54 processes whereas the decision variables of the other processes which are not selected takes a value of exactly zero. From the Table IV, it can be observed that in Case 2 and Case 4, the profit is limited by the amount of budget that is available for investment, as both the raw materials have not been fully utilized. Only in the Case 7 and Case 8, the raw material 1 has been completely exhausted. In few cases, the budget as well as the raw materials are not fully utilized as it may not be possible to produce any other product in their permissible limits with the unused budget or raw materials.

V. Conclusions

In this work, we have developed a strategy for the single level production planning problem that can be used with evolutionary algorithms to determine the optimal production plan of a production planning problem. The developed strategy has been used with Moth-flame optimization and the performance was evaluated on 408 unique instances (8 cases x 51 runs) of the production planning problem involving various complex constraints. It was observed that the performance was consistent and satisfactory. Though demonstrated for production planning in a chemical industry, this strategy can be used to determine the optimal production portfolio in many other industries. The proposed strategy can easily accommodate non-linear relations to determine the production and investment costs which would not be possible in a MILP framework. Future work can include evaluating the performance of various optimization algorithms with the developed strategy. Moreover, other efficient strategies to solve the single level production planning can also be explored.

References

1. *Grand View Research, I. Petrochemical Market Size To Reach $758.3 Billion By 2022. 2016 [cited 2016 June]; Available from* http://www.grandviewresearch.com/press-release/global-petrochemical-market.
2. Li, B. and Z. Shao, *Precise trajectory optimization for articulated wheeled vehicles in cluttered environments.* Advances in Engineering Software, 2016. **92**: p. 40-47.
3. Punnathanam, V. and P. Kotecha, *Yin-Yang-pair Optimization: A novel lightweight optimization algorithm.* Engineering Applications of Artificial Intelligence, 2016. **54**: p. 62-79.
4. Wong, J.Y.Q., S. Sharma, and G.P. Rangaiah, *Design of shell-and-tube heat exchangers for multiple objectives using elitist non-dominated sorting genetic algorithm with termination criteria.* Applied Thermal Engineering, 2016. **93**: p. 888-899.
5. Kadambur, R. and P. Kotecha, *Multi-level production planning in a petrochemical industry using elitist Teaching–Learning-Based-Optimization.* Expert Systems with Applications, 2015. **42**(1): p. 628-641.
6. Alfares, H. and A. Al-Amer, *An optimization model for guiding the petrochemical industry development in Saudi Arabia.* Engineering Optimization, 2002. **34**(6): p. 671-687.
7. Mirjalili, S., *Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm.* Knowledge-Based Systems, 2015. **89**: p. 228-249.
8. Kadambur, R. and P. Kotecha, *Optimal production planning in a petrochemical industry using multiple levels.* Computers & Industrial Engineering, 2016. **100**: p. 133-143.
9. S*upplementary information https://goo.gl/3RAC4G*, Last accessed september 29, 2016.