Optimization of a Kraft Pulping System: Using Particle Swarm Optimization and Differential Evolution

Millie Pant[1], Radha Thangaraj[1], Ajith Abraham[2]
[1]Department of Paper Technology, IIT Roorkee, Saharanpur – 247001, India.
[2]Center of Excellence for Quantifiable Quality of Service, Norwegian University of Science and Technology, Norway
millifpt@iitr.ernet.in, radhadpt@iitr.ernet.in, ajith.abraham@ieee.org

Abstract

Expectation of profit is the economic driving force motivating business activity in a free-enterprise economy. An increase in this profit for a given organization can be accomplished by discovering and following any of a number of courses of improved action. This paper presents an economic optimization of a hypothetical but realistic Kraft pulping system, which forms an integral part of a Pulp and Paper industry, using Particle Swarm Optimization (PSO) and Differential Evolution (DE). Simulation results obtained are comparable (and are slightly better) with the given results which show that these algorithms are quite competent for solving large scale industrial optimization problems in a very small time.

1. Introduction

Paper industry accounts for nearly 3.5% of world's industrial production and 2% of world trade. Current annual consumption of paper is of the order of 270 million tones. This industry is 10^{th} major section in India. Paper industry has some major sections like pulping and recovery cycle, stock preparation and machine operation etc. Kraft process is a dominant chemical pulping process which uses NaOH and Na_2S as puling chemicals. The Kraft pulping and recovery cycle operations comprise a reasonably isolated, yet complex, subsystem of an integrated papermaking process which requires sophisticated mathematical techniques for optimization.

For the past few decades, stochastic techniques have become very popular for solving complex optimization problems which are otherwise difficult to solve by the classical optimization techniques [1], [2]. The popularity of these algorithms is mainly because of two reasons; firstly they work with population of solutions rather than with a single point and secondly they do not depend on the mathematical properties of the objective function or the search space. Some well known stochastic techniques include Evolutionary and Genetic Algorithms [3] – [6], PSO [7], [8], DE [9], [10] etc. In this paper we have chosen PSO and DE for optimizing the Kraft pulping process for a Pulp and Paper industry.

The remaining paper is organized as follows: In section 2 and 3, we have briefly described the PSO and DE algorithms; section 4 explains the constrained handling approach for these algorithms. Section 5 describes the mathematical model of the given problem and simulation results, finally the paper concludes with section 6.

2. Particle Swarm Optimization

Particle swarm optimization technique is a population based stochastic search technique first suggested by Kennedy and Eberhart in 1995. The mechanism of PSO is inspired from the complex social behavior shown by the natural species. For a D-dimensional search space the position of the ith particle is represented as $X_i = (x_{i1},x_{i2},..x_{iD})$. Each particle maintains a memory of its previous best position $P_i = (p_{i1}, p_{i2}\ldots p_{iD})$ and a velocity $V_i = (v_{i1}, v_{i2},\ldots v_{iD})$ along each dimension . At each iteration, the P vector of the particle with best fitness in the local neighborhood, designated g, and the P vector of the current particle are combined to adjust the velocity along each dimension and a new position of the particle is determined using that velocity. The two basic equations which govern the working of PSO are that of velocity vector and position vector are given by:

$$v_{id} = \omega v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \quad (1)$$

$$x_{id} = x_{id} + v_{id} \quad (2)$$

The first part of equation (1) represents the inertia of the previous velocity, the second part is tells us about the personal thinking of the particle and the third part represents the cooperation among particles and is therefore named as the social component. Acceleration constants c_1, c_2 and inertia weight ω are predefined by the user and r_1, r_2 are the uniformly generated random numbers in the range of [0, 1] [16]14,15,16].

3. Differential Evolution

Differential Evolution is a simple powerful evolutionary algorithm for global optimization proposed by Storn and Price. It is a population based algorithm like genetic algorithms using the similar operator; crossover, mutation and selection. The main difference in constructing better solutions is that genetic algorithms rely on crossover while DE relies on mutation operator [11]. DE works as follows: First, all individuals are initialized with uniformly distributed random numbers and evaluated using the fitness function provided. Then the following will be executed until maximum number of generation has been reached or an optimum solution is found.

For a D-dimensional search space, each target vector $x_{i,g}$, a mutant vector is generated by

$$v_{i,g+1} = x_{r_1,g} + F * (x_{r_2,g} - x_{r_3,g}) \quad (3)$$

where $r_1, r_2, r_3 \in \{1,2,....,NP\}$ are randomly chosen integers, must be different from each other and also different from the running index i. F (>0) is a scaling factor which controls the amplification of the differential evolution $(x_{r_2,g} - x_{r_3,g})$. In order to increase the diversity of the perturbed parameter vectors, crossover is introduced [10]. The parent vector is mixed with the mutated vector to produce a trial vector $u_{ji,g+1}$,

$$u_{ji,g+1} = \begin{cases} v_{ji,g+1} & if\ (rand_j \le CR)\ \ or\ \ (j = j_{rand}) \\ x_{ji,g} & if\ (rand_j > CR)\ \ and\ \ (j \ne j_{rand}) \end{cases}$$

where j = 1, 2,……, D; $rand_j \in [0,1]$; CR is the crossover constant takes values in the range [0, 1] and $j_{rand} \in (1,2,....., D)$ is the randomly chosen index.

Selection is the step to choose the vector between the target vector and the trial vector with the aim of creating an individual for the next generation.

4. Penalty Method for Constrained Optimization Problems

Many real-world optimization problems are solved subject to sets of constraints. The search space in COPs consists of two kinds of solutions: feasible and infeasible. Feasible points satisfy all the constraints, while infeasible points violate atleast one of them. Therefore the final solution of an optimization problem must satisfy all constraints.

In this paper, the two algorithms PSO and DE handle the constraints using the concept of penalty functions. In the penalty function approach, the constrained problem is transformed into an unconstrained optimization algorithm by penalizing the constraints and building a single objective function, which is minimized using an unconstrained optimization algorithm. That is,

$$F(x) = f(x) + \lambda\ p(x) \quad (4)$$

Where $$p(x_i, t) = \sum_{m=1}^{n_g + n_h} \lambda_m(t) p_m(x_i) \quad (5)$$

$$p_m(x_i) = \max\{0, g_m(x_i)^\alpha\}$$

$$\text{if } m \in [1,......, n_g] \text{ (inequality)} \quad (6)$$

$$p_m(x_i) = |h_m(x_i)|^\alpha$$

$$\text{if } m \in [n_g + 1,......, n_g + n_h] \text{ (equality)} \quad (7)$$

with α a positive constant, representing the power of the penalty. The inequality constraints are considered as g(x) and h(x) represents the equality constraints. n_g and n_h denotes the number inequality and equality constraints respectively. λ is the constraint penalty coefficient.

5. Economic Optimization of a Kraft Pulping for Pulp and Paper Industry

5.1. Mathematical Model of the Hypothetical Kraft Pulping System

The Kraft pulping [12] and recovery cycle operations comprise a reasonably isolated, yet complex, subsystem of an integrated papermaking process. Figure 1 [12] shows the various subsystems involved in the process of paper making. It is evident from the figure that Pulping and recovery cycle operations form the center of integrated paper making process. The given hypothetical system consists of the interrelated digester and recovery cycle operations for a

Kraft mill producing, under certain realistic conditions, a fixed daily amount of unbleached spruce pulp. Also there are certain typical revenues and variable costs, and a number of realistic constraints. For more details please refer to [12].

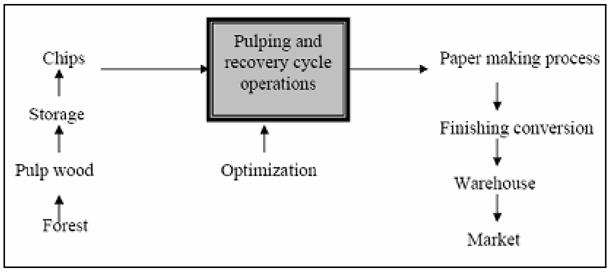

Figure 1 Integrated Paper Making Process

The mathematical model is given as [13]:

Minimize

$$f(x) = 0.00011y_{14} + 0.1365 + 0.00002358y_{13} + 0.000001502y_{16} + 0.0321y_{12} + 0.004324y_5 + 0.0001\frac{c_{15}}{c_{16}} + 37.48\frac{y_2}{c_{12}} - 0.0000005843y_{17}$$

Subject to:

$g_1 = \frac{0.28}{0.72}y_5 - y_4 \le 0$, $g_2 = x_3 - 1.5x_2 \le 0$,

$g_3 = 3496\frac{y_2}{c_{12}} - 21 \le 0$,

$g_4 = 110.6 + y_1 - \frac{62{,}212}{c_{17}} \le 0$,

$g_5 = y_1 - 405.23 \le 0$, $g_6 = 213.1 - y_1 \le 0$

$g_7 = y_2 - 1053.6667 \le 0$, $g_8 = 17.505 - y_2 \le 0$

$g_9 = y_3 - 35.03 \le 0$, $g_{10} = 11.275 - y_3 \le 0$,

$g_{11} = y_4 - 665.585 \le 0$, $g_{12} = 214.228 - y_4 \le 0$,

$g_{13} = y_5 - 584.463 \le 0$, $g_{14} = 7.458 - y_5 \le 0$,

$g_{15} = y_6 - 265.916 \le 0$, $g_{16} = 0.961 - y_6 \le 0$,

$g_{17} = y_7 - 7.046 \le 0$, $g_{18} = 1.612 - y_7 \le 0$,

$g_{19} = y_8 - 0.222 \le 0$, $g_{20} = 0.146 - y_8 \le 0$,

$g_{21} = y_9 - 273.366 \le 0$, $g_{22} = 107.99 - y_9 \le 0$,

$g_{23} = y_{10} - 1286.105 \le 0$,

$g_{24} = 922.693 - y_{10} \le 0$, $g_{25} = y_{11} - 1444.046 \le 0$,

$g_{26} = 926.832 - y_{11} \le 0$, $g_{27} = y_{12} - 537.141 \le 0$,

$g_{28} = 18.766 - y_{12} \le 0$, $g_{29} = y_{13} - 3247.039 \le 0$,

$g_{30} = 1072.163 - y_{13} \le 0$,

$g_{31} = y_{14} - 26844.086 \le 0$,

$g_{32} = 8961.448 - y_{14} \le 0$, $g_{33} = y_{15} - 0.386 \le 0$,

$g_{34} = 0.063 - y_{15} \le 0$, $g_{35} = y_{16} - 140{,}000 \le 0$,

$g_{36} = 71{,}084.33 - y_{16} \le 0$,

$g_{37} = y_{17} - 12{,}146{,}108 \le 0$,

$g_{38} = 2{,}802{,}713 - y_{17} \le 0$,

$704.4148 \le x_1 \le 906.3855$, $68.6 \le x_2 \le 288.88$,

$0 \le x_3 \le 134.75$, $193 \le x_4 \le 287.0966$,

$25 \le x_5 \le 84.1988$.

<u>Calculations:</u>

$y_1 = x_2 + x_3 + 41.6$, $c_1 = 0.024x_4 - 4.62$,

$y_2 = \frac{12.5}{c_1} + 12$,

$c_2 = 0.0003535x_1^2 + 0.5311x_1 + 0.08705y_2x_1$,

$c_3 = 0.052x_1 + 78 + 0.002377y_2x_1$,

$y_3 = \frac{c_2}{c_3}$, $y_4 = 19y_3$,

$$c_4 = 0.04782(x_1 - y_3) + \frac{0.1956(x_1 - y_3)^2}{x_2} + 0.6376y_4 + 1.594y_3,$$

$c_5 = 100x_2$, $c_6 = x_1 - y_3 - y_4$, $c_7 = 0.95 - \frac{c_4}{c_5}$,

$y_5 = c_6c_7$, $y_6 = x_1 - y_5 - y_4 - y_3$

$c_8 = (y_5 + y_4)0.995$, $y_7 = \frac{c_8}{y_1}$, $y_8 = \frac{c_8}{3798}$,

$c_9 = y_7 - \frac{0.0663y_7}{y_8} - 0.3153$

$y_9 = \frac{96.82}{c_9} + 0.321y_1$,

$y_{10} = 1.29y_5 + 1.258y_4 + 2.29y_3 + 1.71y_6$,

$y_{11} = 1.71x_1 - 0.452y_4 + 0.58y_3$

$c_{10} = \frac{12.3}{752.3}$, $c_{11} = (1.75y_2)(0.995x_1)$,

$c_{12} = 0.995y_{10} + 1998$, $y_{12} = c_{10}x_1 + \frac{c_{11}}{c_{12}}$

$y_{13} = c_{12} - 1.75y_2$,

$y_{14} = 3623 + 64.4x_2 + 58.4x_3 + \frac{146312}{y_9 + x_5}$

$$c_{13} = 0.995y_{10} + 60.8x_2 + 48x_4 - 0.1121y_{14} - 5095$$

$$, \quad y_{15} = \frac{y_{13}}{c_{13}}$$

$$y_{16} = 148000 - 331000y_{15} + 40y_{13} - 61y_{15}y_{13} \quad ,$$

$$c_{14} = 2324y_{10} - 28740000y_2$$

$$y_{17} = 14130{,}000 - 1328y_{10} - 531y_{11} + \frac{c_{14}}{c_{12}} \quad ,$$

$$c_{15} = \frac{y_{13}}{y_{15}} - \frac{y_{13}}{0.52}$$

$$c_{16} = 1.104 - 0.72y_{15} \quad , \quad c_{17} = y_9 + x_5 \quad .$$

There are five independently adjustable variables, and they are defined as follows:

x_1 = total load of inorganic chemical as Na_2O in white liquor before losses

x_2 = volume of white liquor to digesters

x_3 = volume of black liquor to digesters

x_4 = rate of fresh wash water to washers

x_5 = time between cooks at each digester

5.2. Simulation Results

The initial population for both the algorithms is taken as number of particles in the swarm (swarm size) is taken as 30. A total of 30 runs were performed and best result throughout the run was recorded.

Table 1 Simulation Results

Item	PSO	DE	Results in [13]
x1	705.170955	705.180325	705.06
X2	68.6	68.6	68.6
x3	102.899995	102.899995	102.900
x4	282.324854	282.324033	282.341
x5	37.583506	37.571403	35.627
f (x)	-1.905168	-1.905134	-1.90500
Run time (sec)	0.63	1.23	---
Genera-tion	504	1763	---

Table 1 shows the experimental results. Figure 2 gives the convergence curves of PSO and DE. For the present study, the algorithms are coded in Turbo C++ and executed on a P IV.

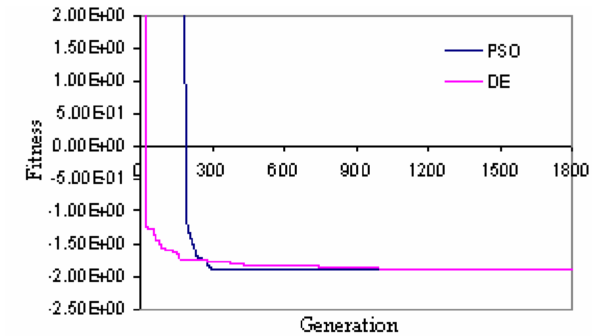

Figure 2 Convergence curves of PSO and DE

6. Conclusion

This paper presents an industrial application of two popular population based search algorithms namely PSO and DE, by taking a classical example of optimization of Kraft pulping system. The mathematical model of the problem consists of five unknown variables and thirty eight constraints.

The solutions obtained by PSO and DE are more or less similar to the given solution (up to the third place of decimal), but since this is a real life optimization problem, the difference of even a small fraction may make a big difference in the industry. Moreover the main advantage of these algorithms is that they were able to solve the algorithms in very small time. PSO took only 0.63 seconds and 504 generations to solve the problem, where as DE took 1.23 seconds and 1763 generations to do the same. Thus it may be concluded that PSO and DE can be used for solving large scale industrial optimization problems. However, the PSO and DE models taken for this paper are relatively simple models and more advanced versions of both the algorithms are now available. In future we shall be using more sophisticated versions of PSO and DE for solving the large scale optimization problems related to industries. We shall also be using other stochastic optimization techniques for solving the problem analyzed in this paper.

7. References

[1] Pinter J.D., "Global Optimization in Action. Continuous and Lipschitz Optimization: Algorithms, Implementation and Applications", Kluwer, Dordrecht, 1996.

[2] A. Torn and Zilinskas, "Global Optimization", Springer Verlag, Berlin, 1989.

[3] I. Rechenberg, "Evolution Strategy: Optimization of Technical systems by means of biological evolution", Fromman-Holzboog, 1973.

[4] L. J. Fogel, A. J. Owens, and M. J. Walsh, "Artificial intelligence through a simulation of evolution", In M. Maxfield, A. Callahan and L. J. Fogel, editors, Biophysics and Cybernetic systems. Proc. of the 2nd Cybernetic Sciences Symposium, pp. 131 – 155, Spartan Books, 1965.

[5] Holland J. H., "Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence," Ann Arbor, MI: University of Michigan Press.

[6] Goldberg D., "Genetic Algorithms in Search Optimization and Machine Learning", Addison Wesley Publishing Company, Reading, Massachutes.

[7] Kennedy J. and Eberhart R., "Particle Swarm Optimization," IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, 1995, pg. IV: 1942-1948.

[8] Kennedy J., "The Particle Swarm: Social Adaptation of Knowledge," IEEE International Conference on Evolutionary Computation (Indianapolis, Indiana), IEEE Service Center, Piscataway, NJ, pg.303-308, 1997.

[9] R. Storn and K. Price, "Differential Evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces", Technical Report, International Computer Science Institute, Berkley, 1995.

[10] R. Storn and K. Price, "Differential Evolution – a simple and efficient Heuristic for global optimization over continuous spaces", Journal Global Optimization. 11, 1997, pp. 341 – 359.

[11] D. Karaboga and S. Okdem, "A simple and Global Optimization Algorithm for Engineering Problems: Differential Evolution Algorithm", Turk J. Elec. Engin. 12(1), 2004, pp. 53 – 60.

[12] C. W. Carroll, "An Operation Research Approach to the Economic Optimization of a Kraft Pulping Process", Ph. D. dissertation, The Institute of Paper Chemistry, Appleton, Wisconsin, 1959.

[13] D. M. Himmelblau, "Applied Non-Linear programming", New York: McGraw-Hill, 1972.

[14] H. Liu, A. Abraham and W. Zhang, A Fuzzy Adaptive Turbulent Particle Swarm Optimization, International Journal of Innovative Computing and Applications, Volume 1, Issue 1, 2007, pp. 39-47.

[15] C. Grosan, A. Abraham and M. Nicoara, Search Optimization Using Hybrid Particle Sub-Swarms and Evolutionary Algorithms, International Journal of Simulation Systems, Science & Technology, UK, Volume 6, Nos. 10 and 11, 2005, pp. 60-79.

[16] A. Abraham, H. Guo and H. Liu, Swarm Intelligence: Foundations, Perspectives and Applications, Swarm Intelligent Systems, Nadia Nedjah and Luiza Mourelle (Eds.), Studies in Computational Intelligence, Springer Verlag, Germany, pp. 3-25, 2006.