Differential Evolution for the Optimal Design of Heat Exchangers

B.V.Babu* and S.A.Munawar

Department of Chemical Engineering, Birla Institue of Technology and Science, PILANI-333 031, Rajasthan, India.

Abstract

This paper presents the application of Differential Evolution (DE) for the optimal design of shell-and-tube heat exchangers. A primary objective in the heat exchanger (HE) design is the estimation of the minimum heat transfer area required for a given heat duty, as it governs the overall cost of the heat exchanger. However, many number of discrete combinations of the design variables are possible. Hence the design engineer needs an efficient strategy in searching for the global minimum heat exchanger cost. In the present study, for the first time DE, an improved version of Genetic Algorithms (GA), has been successfully applied with 1,61,280 design configurations obtained by varying the design variables: tube outer diameter, tube pitch, tube length, number of tube passes, baffle spacing and baffle cut. Bell's method is used to find the heat transfer area for a given design configuration. For a case study taken up, it is observed that DE, an exceptionally simple evolution strategy, is significantly faster compared to GA and is also much more likely to find a function's true global optimum.

Keywords: Heat exchanger design, Shell-and-tube heat exchanger, Genetic algorithms, Simulated annealing, Differential evolution, and optimization.

1.Introduction

The transfer of heat to and from process fluids is an essential part of most of the chemical processes. So the Heat Exchangers (HEs) are used extensively and regularly in the process and allied industries and are very important during design and operation. The most commonly used type of heat exchanger is the shell-and-tube heat exchanger, the optimal design of which is the main objective of this study. Computer software marketed by companies such as HTRI and HTFS are used extensively in the thermal design and rating of HEs. These packages incorporate various design options for the heat exchangers including the variations in the tube diameter, tube pitch, shell type, number of tube passes, baffle spacing, baffle cut, etc. A primary objective in the Heat Exchanger Design (HED) is the estimation of the minimum heat transfer area required for a given heat duty, as it governs the overall cost of the HE. But there is no concrete objective function that can be expressed explicitly as a function of the design variables and in fact many number of discrete combinations of the design variables are possible as is elaborated below. The tube diameter, length, shell types etc. are all standardized and are available only in certain sizes and geometry. And so the design of a shell-and-tube heat exchanger usually involves a trial and error procedure where for a certain combination of the design variables the heat transfer area is calculated and then another combination is tried to check if there is any possibility of reducing the heat transfer area. Since several discrete combinations of the design configurations are possible, the designer needs an efficient strategy to quickly locate the design configuration having the minimum heat exchanger cost. Thus the optimal problem of HED can be posed as a large scale, discrete, combinatorial optimization problem (Chaudhuri et al., 1997).

Most of the traditional optimization techniques based on gradient methods have the possibility of getting trapped at local optimum depending upon the degree of non-linearity and initial guess. Hence, these traditional optimization techniques do not ensure global optimum and also have limited applications. In the recent past, some expert systems based on natural phenomena such as Simulated Annealing and Genetic Algorithms have been developed to overcome this problem (Goldberg, 1989; Deb, 1996). Simulated Annealing (SA)

*Corresponding author. Fax: 0091 1596 44183. Ph: 0091 1596 45073 EXT. 205
Group Leader, Engineering Technology Group, Engineering Services Division
E-mail: {bvbabu,munawar}@bits-pilani.ac.in

is a probabilistic based nontraditional optimization technique, which resembles the thermodynamic process of cooling of molten metals to achieve the minimum energy state. Since, its introduction by Kirkpatrick et al.(1983), SA has diffused widely into many diverse applications. Rutenbar (1989) gives a detailed discussion of the working principle of SA and its applications along with the algorithm for coding. Genetic Algorithms (GA) are computerized search and optimization algorithms based on the mechanics of natural genetics and natural selection. They mimic the ‘ survival of the fittest ‘ principle of nature to make a search process (Goldberg, 1989). Since their inception three decades ago, they have evolved like the species they try to mimic and have been applied successfully in many diverse fields.

Chaudhuri et al. (1997) used Simulated Annealing for the optimal design of heat exchangers and developed a command procedure, to run the HTRI design program coupled to the annealing algorithm, iteratively. They have compared the results of the SA program with a base case design and concluded that significant savings in the heat transfer area and hence the HE cost can be obtained using SA. Manish et al. (1999) used a genetic algorithm framework to solve this optimal problem of HED along with SA and compared the performance of SA and GAs in solving this problem. They also presented GA strategies to improve the performance of the optimization framework. They concluded that these algorithms result in considerable savings in computational time compared to an exhaustive search and GAs have an advantage over other methods in obtaining multiple solutions of the same quality, thus providing more flexibility to the designer.

This paper demonstrates the first successful application of Differential Evolution, an improved version of GA, to the optimal heat exchanger design problem. Differential Evolution (DE), an improved version of GA, is an exceptionally simple evolution strategy that is significantly faster and robust at numerical optimization and is more likely to find a function’s true global optimum. Unlike simple GA that uses a binary coding for representing problem parameters, DE uses real coding of floating point numbers. The mutation operator here is addition instead of bit-wise flipping used in GA. And DE uses non-uniform crossover and tournament selection operators to create new solution strings. Among the DE’s advantages are its simple structure, ease of use, speed and robustness. It can be used for optimizing functions with real variables and many local optima. DE has been used to design several complex digital filters (Price and Storn, 1997) and to design fuzzy logic controllers (Sastry et al., 1998). DE can also be used for parameter estimations. Babu and Sastry (1999) used DE for the estimation of effective heat transfer parameters in trickle-bed reactors using radial temperature profile measurements. They concluded that DE takes less computational time to converge compared to the existing techniques without compromising with the accuracy of the parameter estimates.

In the next section the general procedure of shell-and-tube heat exchanger design is discussed followed by the optimal problem formulation. In this study Bell’s method is used to find the heat transfer area for a given design configuration along with the pressure drop constraint. The design variables considered are: tube outer diameter, tube pitch, shell type, number of tube passes, tube length, baffle spacing and baffle cut. An exhaustive search will require 1,61,280 function evaluations to locate the global minimum heat exchanger cost. For a case study taken up the performance of GA and DE are compared and it is observed that DE is significantly faster compared to GA and also GA finally could not converge to the global optimum value obtained by DE. Thus DE, a simple evolution strategy is significantly faster, more likely to find a function’s true global optimum and proves to be a potential source for accurate and faster optimization.

2. The Optimal HED problem

The proper use of basic heat transfer knowledge in the design of practical heat transfer equipment is an art. The designer must be constantly aware of the differences between the idealized conditions for which the basic knowledge was obtained versus the real conditions of the mechanical expression of his design and its environment. The result must satisfy process and operational requirements (such as availability, flexibility, and maintainability) and do so economically. Heat exchanger design is not a highly accurate art under the best of conditions (Perry and Green, 1993).

2.1. Generalized Design Procedure for Heat Exchangers

The *design of a process heat exchanger* usually proceeds through the following steps: (Perry and Green, 1993)

- Process conditions (stream compositions, flow rates, temperatures, pressures) must be specified.
- Required physical properties over the temperature and pressure ranges of interest must be obtained.
- The type of heat exchanger to be employed is chosen.
- A preliminary estimate of the size of the exchanger is made, using a heat transfer coefficient appropriate to the fluids, the process, and the equipment.
- A first design is chosen, complete in all details necessary to carryout the design calculations.
- The design chosen is now evaluated or rated, as to its ability to meet the process specifications with respect to both heat transfer and pressure drop.
- Based on this result a new configuration is chosen if necessary and the above step is repeated. If the first design was inadequate to meet the required heat load, it is usually necessary to increase the size of the exchanger, while still remaining within specified or feasible limits of pressure drop, tube length, shell diameter, etc. This will sometimes mean going to multiple exchanger configurations. If the first design more than meets heat load requirements or does not use all the allowable pressure drop, a less expensive exchanger can usually be designed to fulfil process requirements.
- The final design should meet process requirements (within reasonable expectations of error) at lowest cost. The lowest cost should include operation and maintenance costs and credit for ability to meet long-term process changes as well as installed (capital) cost. Exchangers should not be selected entirely on a lowest first cost basis, which frequently results in future penalties.

The flow chart given in Fig. 1 briefly gives the sequence of steps involved in the optimal design of a shell-and-tube heat exchanger. (Sinnott, 1993).

2.2. The optimal problem formulation

The objective function and the optimal problem of shell-and-tube HED can be represented as shown in Table-1 (Manish et al., 1999). The objective function can be minimization of HE cost C(X) or heat transfer area A(X) and X is a solution string representing a design configuration. The design variable x_1 takes 12 values for tube outer diameter in the range of 1/4" to 2.5". x_2 represents the tube pitch either square or triangular taking two values represented by 1 and 2. x_3 takes the shell head types : floating head, fixed tube sheet, U tube, and pull through floating head represented by the numbers 1,2,3 and 4 respectively. x_4 takes number of tube passes 1-1, 1-2, 1-4, 1-6, 1-8 represented by numbers from 1 to 5. The variable x_5 takes eight values of the various tube lengths in the range 6ft - 24 ft. represented by numbers 1 to 8. x_6 takes six values for the variable baffle spacing, in the range 0.2 to 0.45 times the shell diameter. x_7 takes seven values for the baffle cut in the range 15 to 45 percent. The total number of design combinations with these variables are 12 x 2 x 4 x 5 x 8 x 6 x 7 = 1,61,280. It means that if an exhaustive search is to be performed it will take at the maximum 1,61,280 function evaluations before arriving at the global minimum heat exchanger cost. So the strategy which takes few function evaluations is the best one.

Table-1. Optimal Heat Exchanger Design Problem

min C(X) or A(X)

$X \in \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$

where

x_1	=	{1,2,...,12}
x_2	=	{1,2}
x_3	=	{1,2,3,4}
x_4	=	{1,2,...,5}
x_5	=	{1,2,..,8}
x_6	=	{1,2,...,6}
x_7	=	{1,2,...,7}

subject to

feasibility constraints [pressure_drop]

Considering HE cost as the objective function, differential evolution technique is applied to find the optimum design configuration. The performance of DE is compared with GA and the results are discussed in the fourth section.

3. Differential evolution: at a glance

We all must accept that 'Nature knows the best!' and we, the human beings try to mimic nature and its natural phenomenon in our efforts for the development of technology and resources. Few of the outcomes of such efforts are the evolution of nontraditional optimization techniques such as Simulated Annealing (SA), Genetic Algorithms (GA), and Differential Evolution (DE). In the recent past, these algorithms have been successfully applied for solving complex engineering optimization problems and are widely known to diffuse very close to the global optimum solution.

A genetic algorithm that is well adapted to solving a combinatorial task like the traveling salesman problem may fail miserably when used to minimize functions with real variables and many local optima. The concept of binary coding used by simple GA limits the resolution with which an optimum can be located to the precision set by the number of bits in the integer. So, the simple GA was later modified to work with the real variables as well. Just as floating point numbers are more appropriate than integers for representing points in continuous space, addition is more appropriate than random bit flipping for searching the continuum (Price and Storn, 1997). Consider, for example to change a binary 16(10000) into a binary 15(01111) with bit-wise flipping would require inverting all the five bits. In most bit flipping schemes a mutation of this magnitude would be rare. Alternately under addition, 16 become 15 simply by adding -1. Adopting addition as the mutation operator to restore the adjacency of nearby points is not, however, a panacea. Then the fundamental question concerning addition would be how much to add. The simple adaptive scheme used by DE ensures that these mutation increments are automatically scaled to the correct magnitude. Similarly DE uses a non-uniform crossover in that the parameter values of the child vector are inherited in unequal proportions from the parent vectors. For reproduction, DE uses a tournament selection where the child vector competes against one of its parents.

The overall structure of the DE algorithm resembles that of most other population based searches. The parallel version of DE maintains two arrays, each of which holds a population of NP, D-dimensional, real valued vectors. The primary array holds the current vector population, while the secondary array accumulates vectors that are selected for the next generation. In each generation, NP competitions are held to determine the composition of the next generation. Every pair of vectors (X_a, X_b) defines a vector differential: $X_a - X_b$. When X_a and X_b are chosen randomly, their weighted differential is used to perturb another randomly chosen vector X_c. This process can be mathematically written as $X'_c = X_c + F\,(X_a - X_b)$. The scaling factor F is a user supplied constant in the range ($0 < F \leq 1.2$). The optimal value of F for most of the functions lies in the range of 0.4 to 1.0 (Price and Storn, 1997). Then in every generation, each primary array vector, X_i is targeted for crossover with a vector like $X^{'}_c$ to produce a trial vector X_t. Thus the trial vector is the child of two parents, a noisy random vector and the target vector against which it must complete. The non-uniform crossover is used with a crossover constant CR, in the range $0 \le CR \le 1$. CR actually represents the probability that the child vector inherits the parameter values from the noisy random vector. When CR = 1 for example every trial vector parameter is certain to come from $X^{'}_c$. If, on the other hand, CR=0, all but one trial vector parameter comes from the target vector. To ensure that X_t differs from X_i by atleast one parameter, the final trial vector parameter always comes from the noisy random vector, even when CR = 0. Then the cost of the trial vector is compared with that of the target vector, and the vector that has the lowest cost of the two would survive for the next generation. In, all just three factors control evolution under DE, the population size, NP; the weight applied to the random differential, F; and the crossover constant, CR. The pseudo-code for DE is given in the fourth section.

Choosing NP, F, and CR is seldom difficult and some general guidelines are available. Normally, NP ought to be about 5 to 10 times the number of parameters in a vector. As for F, it lies in the range 0.4 to 1.0. Initially F = 0.5 can be tried then F and/or NP is increased if the population converges prematurely. A good first choice for CR is 0.1, but in general CR should be as large a possible (Price and Storn, 1997). Among DE's advantages are its simple structure, ease of use, speed and robustness. Already, DE has been successfully applied for solving several complex problems and is now being identified as a potential source for accurate and faster optimization.

4.Results and discussions

The pseudo code of the DE algorithm used in the present study is given below.

- Initialize the values of D, NP, CR and F
- Initialize all the vectors of the population randomly between a given lower bound (LB) and upper bound (UB).

```
for i =1 to NP
 for j =1 to D
   (X_i,0 )_j = LB +  random number( UB - LB )
```

- Evaluate the cost of each vector. Cost here is the area of the shell-and-tube heat exchanger for the given design configuration, calculated by a separate function using Bell' method.

```
for  i =1 to NP
C_i = cal_area( )
```

- Find out the vector with the lowest cost i.e. the best vector so far.

```
C_min = C_1  and best =1
  for  i =2 to NP
 if (C_i < C_min)
   then  C_min = C_i  and best = i
```

- Perform mutation , cross-over, reproduction and evaluation of the objective function until there is no further improvement in the lowest cost

```
do
  {    for i =1 to NP
       {
```

- For each vector X_i (target vector), select three distinct vectors X_a, X_b and X_c, randomly from the current population (primary array) other than the vector X_i

```
do
  {
```

```
        r1 = random number *NP
        r2 = random number *NP
        r3 = random number *NP

    }   while (r1=i) OR (r2=i) OR (r3=i) OR (r1=r2) OR (r2=r3) OR (r1=r3)
```

- Perform mutation and create a noisy vector $X_{n,i}$,
  $X_{n,i} = X_{a,i} + F(X_{b,i} - X_{c,i})$
- Perform cross-over for each target vector X_i with its noisy vector $X_{n,i}$ and create a trial vector , $X_{t,i}$. If CR =0 inherit all the parameters from the target vector X_i, except the last one which should be from $X_{n,i}$, else depending on a random number generated inherit either from X_i or $X_{n,i}$ as given below.

```
    for n=1 to D
      r=random number*1
     if(CR==0) {
                    for j=1 to (D-1)
                    Xt,j  = Xi,j    }
           else  {
                  for j=1 to D   {
                  r=random number*1
                   if(r<CR)     Xt,j =  X n,j
                     else          Xt,j   =  Xi,j   }   }
```

- Perform selection for each target vector, X_i by comparing its cost with that of the trial vector, $X_{t,i}$; whichever has the lowest cost will survive for the next generation.
  $C_{t,i}$ = cal_area( )

```
  if (Ct,i < Ci )   newXi = Xt,i
    else             newXi  =  X i    }                    /* for i=1 to NP ends */
        }   while (  (oldCmin    - newCmin) < ∈  );
```

- Print the results.

The entire scheme of optimization of shell-and-tube heat exchanger design is performed by the DE strategy, while intermittently it is required to evaluate the heat transfer area or HE cost for a given design configuration. This task is accomplished through a separate function using Bell's method of heat exchanger design. Bell's method gives reasonable estimates of the shell-side heat transfer coefficient and pressure drop compared to Kern's method, as it takes into account the factors for leakage, bypassing, flow in window zone etc.

As a case study the following problem is considered (Sinnott, 1993).

Design a heat exchanger for the following duty: 20,000 Kg/hr of kerosene leaves the base of a side-stripping column at 200^o C and is to be cooled to 90^o C by exchange with 70,000 Kg/hr light crude oil coming from storage at 40^o C. The kerosene enters the exchanger at a pressure of 5 bar and the crude oil at 6.5 bar. A pressure drop of 1 bar is permissible on both the streams.

By performing the enthalpy balance, the heat duty for this case study is found to be 1510 KW and the outlet temperature of crude oil to be 78^o C. The crude is dirtier than the kerosene and so is assigned through the tube-side and kerosene to the shell-side. Using a proprietary program (HTFS, STEP5) the lowest cost design meeting the above specifications results in a heat transfer area of 55 m^2 (Sinnott, 1993). In this study DE is applied for the above problem and the program is executed for 3509 combinations of the key parameters: NP, F and CR.

The NP values are varied from 10 to 150 in steps of 5, and F, CR values are varied from 0 to1 in steps of 0.1. The global minimum HE area for the above heat duty is found to be 34.44 m^2. Out of the 3509 combinations of the key parameters, DE obtained the same global minimum heat transfer area in 1797 combinations as listed in Table-2. For a given NP, there are 121 combinations of the key parameters for F and CR (both varied from 0.0. to 1.0 in steps of 0.1). From Table-2 it can be seen that for NP=25, except the first eight combinations all other combinations (113) yield the lowest heat transfer area of 34.44 m^2. That is for NP=25 and F=0.0, there are three values of CR (0.8, 0.9, 1.0) which yield 34.44 m^2. Then for NP=25 and F=0.1 all the values of CR between 0.0 to 1.0 yield 34.44m^2. Then onwards, at NP=25 the same area is obtained for each value of F from 0.1 to 1.0 for all values of CR between 0.0 to 1.0. Similarly for other values of NP also the optimum combinations of F and CR are listed in Table-2. At NP=25 DE took 50 function evaluations and 0.06 seconds of CPU time (on a 266 MHz Pentium-II processor) to converge to this global optimum value. Since, for all other values of NP the function evaluations are more this value has been reported as the best combination.

Using GA also the same global minimum heat transfer area is obtained (34.44m^2) for the above problem but in a very narrow range of the GA parameters. By trial and error the crossover and mutation probabilities for this problem were found to be 80 and 10 percent respectively. The minimum area obtained using GA, with different population sizes (N) over 100 generations for 80 and 10 percent crossover and mutation probabilities respectively is shown in Fig. 2. From the figure it is observed that GA is finally converging to a heat transfer area of 40.08 m^2. The converged global minimum obtained by DE (34.44 m^2) is seen by GA in a very narrow range of the population size only (whereas DE obtained 34.44m^2 area in almost 50 % of the key parameter combinations). Thus DE is more likely to find a function's true global optimum. For a population size of 144, with 80 and 10 percent crossover and mutation probabilities respectively GA took 13 generations, 1728 function evaluations and 4.09 seconds of CPU time (on a 266 MHz Pentium-II processor) to obtain 34.44 m^2 area. The performance of DE and GA for this case study is compared in Table-3. From the table it can be seen that DE is almost 68 times faster than GA. And by using DE there is 98.5 % saving in the computational time compared to GA. Comparing the results of the proprietary program (HTFS, STEP5) with these algorithms (both DE and GA), there is 37.4 % saving in the heat transfer area for the case study considered. So for this case study, a shell-and-tube heat exchanger with 119 tubes of 1/2" O.D and 24 ft length in a 5/8" triangular pitch arrangement, with a single pass TEMA fixed tube sheet heat exchanger will have the lowest HE area, with 15% baffle cut and 20% baffle spacing. The tube-side and shell-side pressure drops for this configuration are 66.59 and 30.94 KPa respectively, and are well within the specifications.

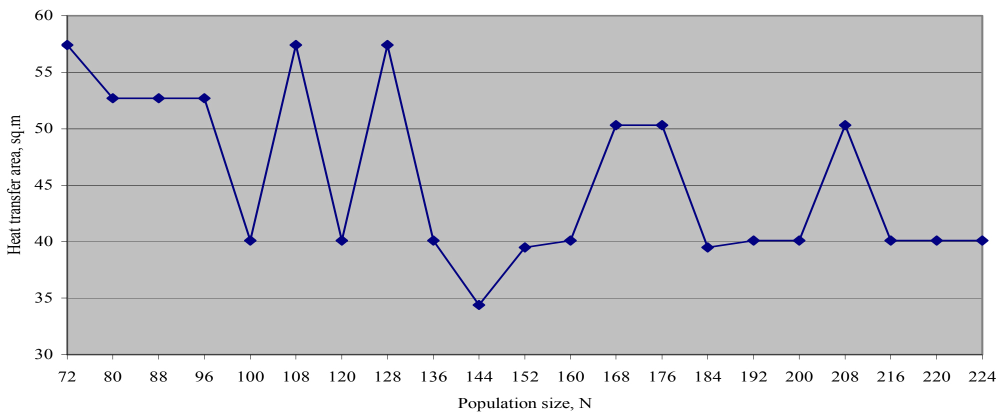

Figure 2. Effect of population size in GA

Table-3. Results of the optimal HED problem : DE and GA comparison

	DE	GA
CPU time	0.06 s	4.09 s
Function evaluations	50	1728
Global minimum heat transfer area	34.44 m^2	34.44 m^2
Converged minimum heat transfer area	34.44 m^2	40.08 m^2

5. Conclusions

This paper demonstrates the first successful application of Differential Evolution for the optimal design of shell-and-tube heat exchangers. A generalized procedure has been developed to run the DE algorithm coupled with a function that uses Bell's method of heat exchanger design, to find the global minimum heat exchanger cost. For a case study taken up the performance of DE and GA is compared. From this study we conclude that 1.The combinatorial algorithms such as GAs and DE provide significant improvement in the optimal designs compared to the traditional designs. 2. DE is significantly faster compared to GA and 3. DE is more likely to find a function's true global optimum.

Nomenclature

A_o	-	heat transfer area based on outer surface, m^2
A(X)	-	objective function heat transfer area, m^2
C_p	-	heat capacity at constant pressure, J/Kg.^{o}C.
CR	-	crossover constant
C(X)	-	objective function heat exchanger cost
F	-	weight applied to the random differential
F_t	-	LMTD correction factor
k	-	thermal conductivity, W/m ^{0}C
N	-	population size in GA
NP	-	population size in DE
Q	-	heat duty, W
ΔT_m	-	mean temperature difference
$U_{o.ass}$	-	assumed value of overall heat transfer coefficient based on outside area, W/m^2 ^{0}C
$U_{0,cal}$	-	calculated value of overall heat transfer coefficient based on outside area, W/m^2 ^{0}C
x	-	a design variable
X	-	a design configuration

greek symbols

ρ	-	density, Kg/m^3
μ	-	viscosity, Kg/m s

abbreviations

DE	-	Differential Evolution
GA	-	Genetic Algorithms
HE	-	Heat Exchanger
HED	-	Heat Exchanger Design
HTFS	-	Heat Transfer Flow Systems
LB	-	Lower Bound
LMTD	-	Log-Mean Temperature Difference
SA	-	Simulated Annealing
UB	-	Upper Bound

References

1. Babu, B. V., and Sastry, K. K. N. (1999). Estimation of heat transfer parameters in a trickle-bed reactor using differential evolution and orthogonal collocation, Comp. Chem. Engg., 23, pp 327 - 339.
2. Chaudhuri, P. D., Urmila, M. D. and Jefery, S.L. (1997). An automated approach for the optimal design of heat exchangers, Ind. & Engg. Chem. Res., Vol. 36, pp 3685 - 3693.
3. Deb, K. (1996). Optimization for engineering design : algorithms and examples, PHI. Pvt. Ltd.
4. Goldberg, D. E. (1989). Genetic algorithms in search, optimization and machine learning, Reading M.A., Addison Wesley.
5. Kirkpatrik, S., Gelatt, C. D. and Vechhi, M. P. (1983). Optimization by simulated annealing, Science, Vol. 220, No. 4568, pp 671-680.
6. Manish, C. T., yan Fu and Urmila, M. D. (1999). Optimal design of heat exchangers: a genetic algorithm framework , Ind. Engg. Chem.Res., Vol. 38, pp 456- 467.
7. Perry, R. H. and Don Green (1993). Perry's chemical engineers' hand book, McGHI Editions, Chemical Engg. series, 6th ed.
8. Price, K. and Storn, R. (1997). Differential evolution, Dr. Dobb's J., pp 18 - 24.
9. Rutenbar, R. A. (1989). Simulated annealing algorithms: an overview, IEEE Circiuts Devices Magazine, No. 100, pp 19-26.
10. Sastry, K. K. N., Behra, L., and Nagrath, I. J. (1998). Differential evolution based fuzzy logic controller for nonlinear process control. Fundamenta Informaticae: Special Issue on Soft Comput.(1998).
11. Sinnott, R. K. (1993). Coulson & Richardson's Chemical Engg., Vol. 6, 2nd ed., Pergamon.

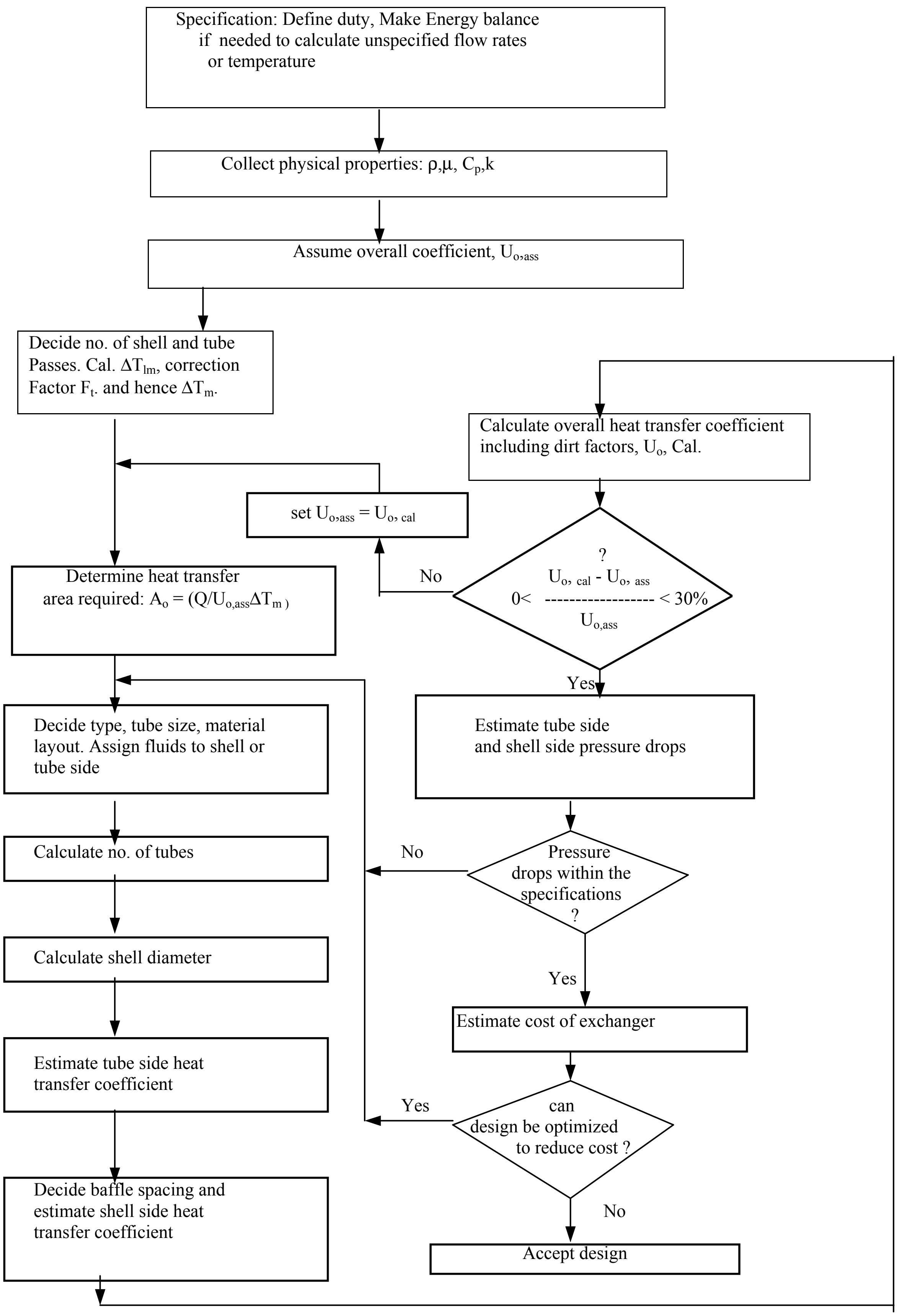

Fig. 1 Design procedure for shell-and-tube heat exchangers