# Optimal Path by Application of Practicle Swarm Optimization with Differential Evolution: A Research

Ramesh Kait
Department of Computer Science and Application
Kurukshetra University, Kurukshetra

**Abstract**

**The sense of optimization in terms of optimal solution is mainly dependant on the technology and the framework used. Evolutionary algorithms and meta-heuristic algorithms have the potential to carry out better results. In this paper we have combined the behaviour of DE and PSO to produce optimal result w.r.t no of iteration and total distance. And after executing the entire system with this new feature we can conclude that this hybrid algorithm will produce better and early result as compared to the individual executions. The hybrid will perform in 20+ iteration and DE in 200+, showing that the system will produce optimal and better results. In the paper we have given a comparative study of these procedures.**

## 1. Differential Evolution

As DE has the quality of GA and is actually inspired by the natural process, so during the proper execution of this algorithm we met many types of steps named differently in different papers. Like GA this also have the capacity to produce diversified solution every time we execute it hence this algorithm have the power of crossover and mutation a these two are the main and most important factor in producing next generation from the previous one. As individual this algorithm works in a context of producing optimal result while taking care of the population size & number of iterations [1]. It executes these steps same as that of normal GA context, but change occur while updating the elements (off-springs), these should be quite more attractive and impactful. DE can therefore also be used on optimization problems that are not even continuous noisy change over time, etc.
DE optimizes a problem by maintaining a population of candidate solutions and by creating new candidate solutions after combining existing ones according to its simple formulae, and then keeping whichever candidate solution has the best value or fitness on the optimization problem at hand. In this way the optimization problem is treated as a blind box that only provides a measure of quality given a candidate solution and the gradient is therefore not needed [2].

## 2. PSO

The popularity of the PSO framework in various scientific communities is due to the fact that on one hand it can be realized and, if necessary, can be adapted to further needs easily, but on the other hand shows in experiments good performance result w.r.t the quality of the obtained solution and the time needed to obtain it. By adapting its parameters, users easily and successfully control the swarm's behaviour with respect to "exploration" ("searching where no one has searched before") and "exploitation" ("searching around a good position") [3]. The local best of a particle is the best position with respect to f this particle has encountered so far. In contrast to evolutionary algorithms, the individuals of the swarm population cooperate by sharing information about the search space via the global attractor, which is the best position any particle has found so far. The particles move in time-discrete iterations. PSO is inspired by the flocking and schooling patterns of birds and fish, Particle Swarm Optimization (PSO) was invented by Russell Eberhart and James Kennedy in 1995 [4]. Originally, these two started out developing computer software simulations of birds flocking around food sources, and then later realized how well their algorithms worked on optimization problems.

## 3. Process of DE

Mutation: Mutation operator is the prime operator of DE and it is the implementation of this operation that makes DE different from other Evolutionary algorithms. The mutation operation of DE applies the vector differentials between the existing population members for determining both the degree and direction of perturbation applied to the individual subject of the mutation operation. The mutation process at each generation begins by randomly selecting three individuals in the population [5].
Crossover: Once the mutation phase is complete, the crossover process is activated. After the mutation operation, the crossover operation is performed to mate the two vectors $X_i^g$ and $V_i^g$ to form the offspring $U_i^g=[u_{i,1}^g, u_{i,2}^g, u_{i,3}^g, \ldots\ldots u_{i,D}^g]$ with the $U_{i,d}^g$ determined as:

$$U_{i,d}^g = \begin{array}{ll} v_{i,d}^g, & \text{if } r_d <= CR \text{ or } d = r_n(i) \\ x_{i,d}^{g-1}, & \text{if } r_d > CR \text{ and } d != r_n(i) \end{array}$$

Where rd ∈ [0, 1] is a randomly generated real value for the dth dimension and rn(i)∈{1, 2, …, D} is a randomly chosen dimension to ensure that at least one dimension of the $u_i^g$ from $v_i^g$ . The parameter *CR* is called crossover rate.
Selection: The selection scheme of DE also differs from that of other EAs [6]. The population for the next generation is selected from the individual in current population and its corresponding trial vector according to the rule.

### 3.1. Hybrid Code

The pseudo code of the Hybrid DE and PSO (DE-PSO):

```
Initialize the population
For i = 1 to N (Population size) do
        Select r1, r2, r3 Î N randomly
        // r1, r2, r3 are selected such that r1_ r2 _ r3//
        For j = 1 to D (dimension) do
                Select jrand ÎD
                If (rand () < CR or j = jrand)
                // rand () denotes a uniformly distributed random
                number between 0 and 1//
                $U_{ji,\,g+1} = x_{r1,\,g} + F *(x_{r2,\,g} - x_{r3,\,g})$
                End if
                If $(f(U_{ji,g+1}) < f(X_{ji,g}))$ then
                $X_{ji,g+1} = U_{ji,g+1}$
                Else
                PSO activated
                Find a new particle using equations (2) and (3).
                (Let this particle be $TXji$ )
                        If ( $f(TXji) < f(Xji,g)$ ) then
                         $X_{ji,g+1} = TX_{ji}$
                        Else $X_{ji,g+1} = X_{ji,g}$
                        End if
                End if
        End for
End for.
```

### 3.2. Applied PMX Crossover

The differential evolution's 'crossover' technique is similar to that of the genetic algorithm. In fact the differential evolution uses the partially-mapped crossover, but the selection of the dominant parent is determined by the variable CR [7].

For example, to perform X −Y , X will be used as a 1×N row matrix and Y will become an N×N identity matrix. First, the values of Y s elements are reversed and placed in an N ×1 column matrix. The column matrix is expanded into a N × N identity matrix with a 1 being placed in the column that represents the original value of the column matrix. All other columns in the row contain a 0. The matrices are multiplied to produce a new 1×N row matrix shown. Since the values in the permutations are integers the scalar multiplication of the mutation factor F also needed reconsideration. This operation was defined as a sweep over the elements in the intermediate solution and swapping the element value with another random element if a randomly generated number was less than F.

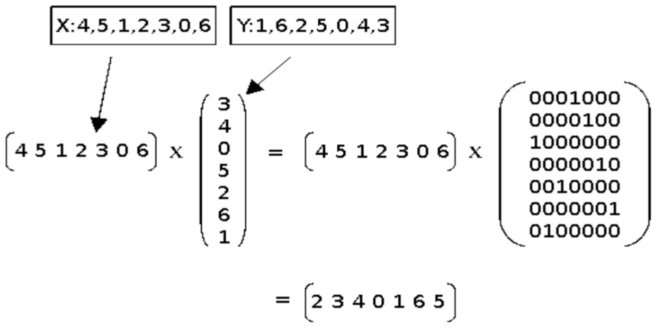
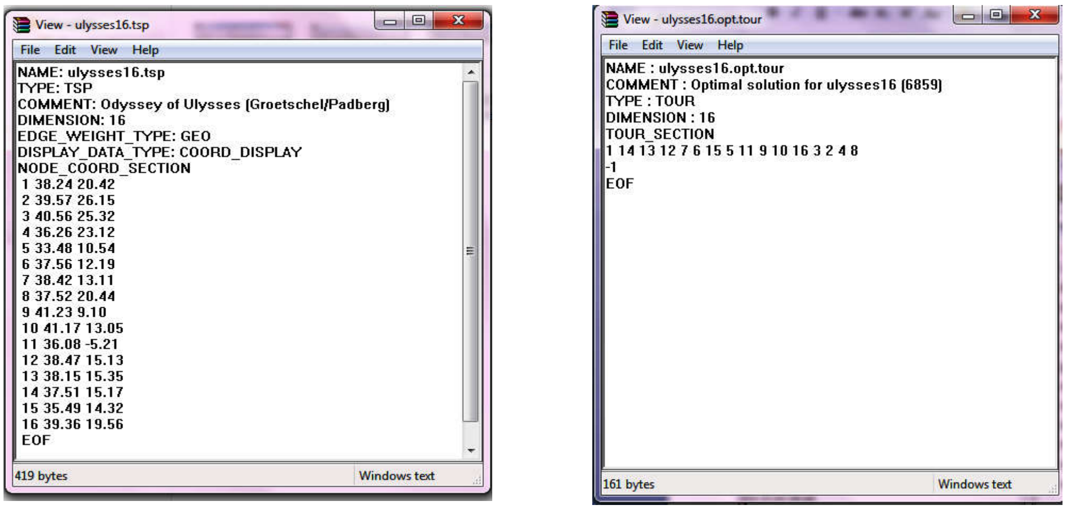

X:4,5,1,2,3,0,6
Y:1,6,2,5,0,4,3
(4 5 1 2 3 0 6) x (3 4 0 5 2 6 1) = (4 5 1 2 3 0 6) x (0001000 0000100 1000000 0000010 0010000 0000001 0100000)
= (2 3 4 0 1 6 5)

Fig 1: Implementation of Permutation Matrix Operations

## 4. Experiments and validations

### 4.1. TSP Base

For the proposed work verification and validation we have gone through the site tsplib.com. We obtained a complete dataset and the best and optimal solution [8]. The site helped us in finding the result more close to the already presented result for the same dataset. Users made new datasets and place their results on this website or may have the option to reuse the dataset present their and their result may be placed publically if they have much better result.



```
NAME: ulysses16.tsp
TYPE: TSP
COMMENT: Odyssey of Ulysses (Groetschel/Padberg)
DIMENSION: 16
EDGE_WEIGHT_TYPE: GEO
DISPLAY_DATA_TYPE: COORD_DISPLAY
NODE_COORD_SECTION
 1 38.24 20.42
 2 39.57 26.15
 3 40.56 25.32
 4 36.26 23.12
 5 33.48 10.54
 6 37.56 12.19
 7 38.42 13.11
 8 37.52 20.44
 9 41.23 9.10
 10 41.17 13.05
 11 36.08 -5.21
 12 38.47 15.13
 13 38.15 15.35
 14 37.51 15.17
 15 35.49 14.32
 16 39.36 19.56
 EOF
```

```
NAME : ulysses16.opt.tour
COMMENT : Optimal solution for ulysses16 (6859)
TYPE : TOUR
DIMENSION : 16
TOUR_SECTION
1 14 13 12 7 6 15 5 11 9 10 16 3 2 4 8
-1
EOF
```

Fig 2 : ulysses16.tsp Dataset and its Optimal Path

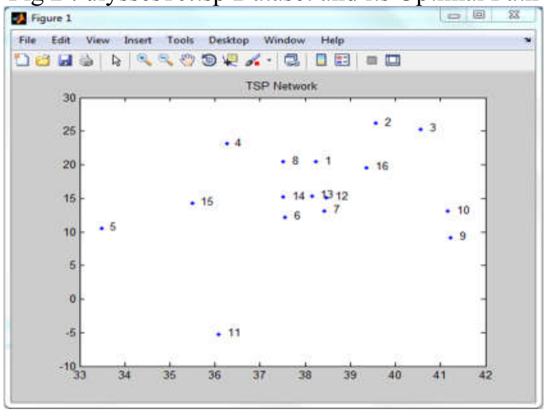


Fig 3 : Illustration of Cities Of The Datasets

This figure presents the optimal path and the optimal result for the cities to 6859. It actually produces result. The next figure demonstrate the node placement for this dataset ranking from 1-16. In this research we have an idea of presenting it more attractive by showing the comparison of core DE with our proposed work with PSO. Here we placed the same fitness function as that of the hybrid, but while doing so we found a watchable comparison.

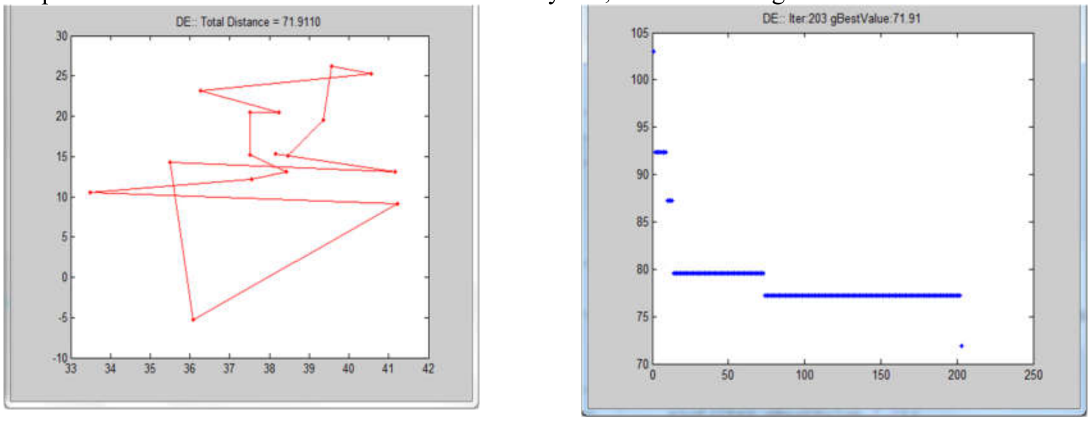


Fig 4 : DE optimal path & Fitness Value

### 4.2. Proposed DE-PSO Pseudo Code:

By the earlier snapshots we can say that DE itself produces a far better result than TSP does whether in terms of total distance or in terms of gBest value [9]. It got reduced by a great margin. So we could apply this hybrid format to DE for the purpose of reduction of these two values with respect to DE alone as in early figures. The algorithm got halt while the if condition goes unsatisfied, here we presented the merged part mainly the application of PSO in DE.

```
If $(f(U_{ji,g+1}) < f(X_{ji,g}))$ then
        $X_{ji,g+1} = U_{ji,g+1}$
        Else
                PSO activated:   Find a new particle using equations (1) and (2).
                (Let this particle be $TX ji$ )
                If ( $f(TX ji) < f(X ji,g)$ ) then
                        $X_{ji,g+1} = TX_{ji}$
                        Else
                        $X_{ji,g+1} = X_{ji,g}$
                End if
End if
```

**4.3. DE-PSO Fitness Function:**

As we have calculated the DE fitness value and at the end we have concluded that it is far better that the best fitness value presented on TSPLIB [10]. Here also we have the chance to recalculate the entire procedure to take care of the merged approach for DE-PSO. We are saying this approach provides a better result by comparing it with the DE fitness va.lue and total distance shown in the previous Snapshots. We have used the fitness function and calculate it from the second element of the file x.dmat just because we don't want NP problem to occur here. And we have called this fitness function to the .m file of hybrid approach. Following is the fitness function code applied in the hybridization:

```
 function d = fitness_function(x,dmat)
         n =length(x);
               d = 0; % Closed Path
                for k = 2:n
              d = d + dmat(x(k-1),x(k));
               end
```

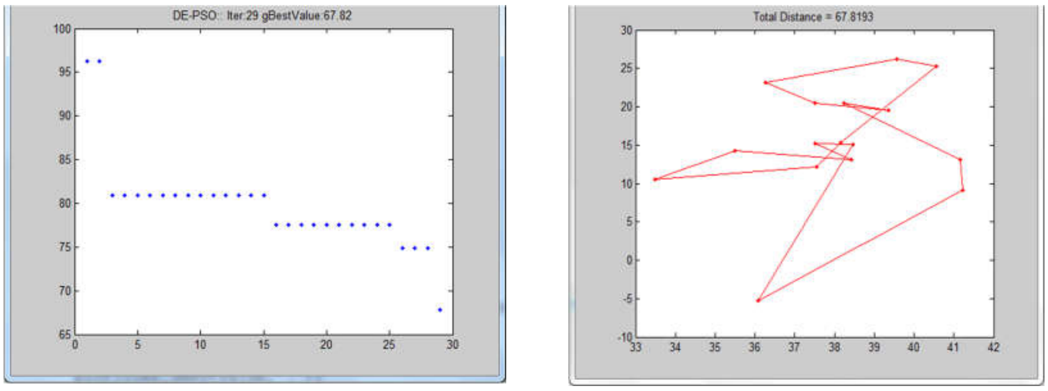


Fig 5 : gBest Value & Total Distance Taken By DE-PSO

These results are taken on the same dataset ulysses16 provided by TSPLIB. By watching all these results we can conclude that some way or the other the merged approach gives out better result in terms of total distance and required number of iteration. The comparative study can be concluded as following:

| **Known Optimal Distance : 73.3885 Popsize = 100, maxiter = 1000** | | |
|---|---|---|
| Algorithm | Optimal Distance | Iterations |
| DE | 71.91 | 203 |
| DE-PSO | 67.82 | 29 |

Table 1: Showing Comparative Study for Hybrid Approach

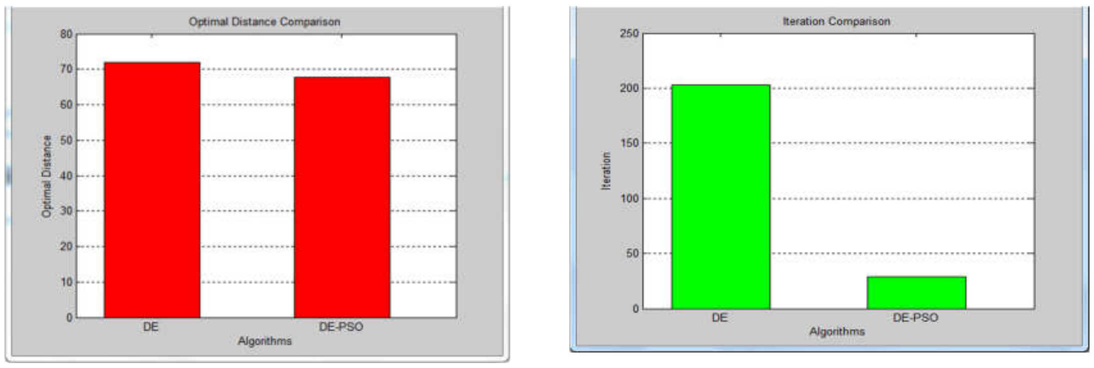


Fig 6 : Bar Graph Of Comparison w.r.t Distance &Iteration

The above figure shows the comparision between core DE algorithm and the hybrid DE-PSO algorithm which we have proposed w.r.t distance. By this graph also we can say that this hybrid syntax giving us result with slightly but lesser distance. And in the sight of optimality this dfference is also a good sign.

## 5. Conclusion & future scope:

The above shows the comparision of number of iterations taken by DE algorithm and hybrid DE-PSO algorithm. By this graph also we can say that this hybrid approach will give the result in drastic lesser number of iteration. Where DE is about to touch 200 and DE-PSO is somewhere on 25+ iterations. This research is done on one dataset and thus has a future work to carry out on different datasets already avialable and of'course with more constraints. The comparative study involves the no of iteration and total distance taken and by these two parameters the hybrid approach will provide a better result. This algorithm can be takes out for the work of other datasets with more number of cities. Th ecomplexity will increase with respect to the no of nodes and this algorithm somewhere or the other will help to reach optimility in number of iterations in total distance taken.

**Bibliography:**

[ 1. ] RAINER STORN, KENNETH PRICE. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11 (1997), 341–359.

[ 2. ] Swagatam Das, Member, IEEE, and Ponnuthurai Nagaratnam Suganthan, Senior Member, IEEE. Differential Evolution: A Survey of the State-of-the-Art. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, VOL. 15, NO. 1 (FEBRUARY 2011), 4-31.

[ 3. ] Chun-Yin Wu, Ching-Bin Zhang, Chi-Jer Wang. Topology Optimization of Structures Using Ant Colony Optimization. *GEC'09* (June 12-14 2009), 601-607.

[ 4. ] J.Kennedly, R.C.Eberhart. pactical swarm optimisation. *proc. in IEEE Internationalconference on neural network*, IEEE-Press (1995), 1942-1048.

[ 5. ] Kavitha Sooda, T. R. Gopalakrishnan Nair. A Comparative Analysis for Determining the Optimal Path using PSO and GA. *International Journal of Computer Applications (0975 – 8887)*, Volume 32, No.4 (October 2011).

[ 6. ] Lampinen, J. Liu and J. "A fuzzy adaptive differential evolution algorithm," Soft Comput. – A Fusion of Foundation, Methodologies (Apr 2005.), pp. 448–462.

[ 7. ] Claudio Comis Da Ronco, Ernesto Benini. GeDEA-II: A Simplex-Crossover Based Multi Objective Evolutionary Algorithm Including The Genetic Diversity As Objective. *GECCO'12 Companion* (July 7 Philadelphia, PA, USA), 619-620.

[ 8. ] Douglas G. Macharet, Armando Alves Neto, Vilar F. da Camara Neto, Mario F. M. Campos. An Evolutionary Approach for the Dubins' Traveling Salesman Problem with Neighborhoods. *ACM 978*, 7, 11 (July 2012), 377-384.

[ 9. ] Marek Antosiewicz, Grzegorz Koloch, Bogumił Kamiński. Choice of best possible metaheuristic algorithm for the travelling salesman problem with limited computational. *ISSN 2299-2634*, Vol. 7, No. 1 ( 2013), pp. 46-55.

[ 10. ] Fisher, R. A. and Yates, F. Statistical tables for biological, agricultural and medical research (3rd ed.). Oliver & Boyd, London. pp. 26–27., 1948.

[ 11. ] David Ardia, Kris Boudt, Peter Carl, Katharine M.Mullen and Brian G. Peterson. Differential Evolution with DEoptim An Application to Non-Convex Portfolio Optimization. *The R Journal* , Vol. 3, 1 (June 2011).