Ozean Journal of Applied Sciences 2(2), 2009
ISSN 1943-2429


# Deterministic Approach to Path Optimization Problem

Aykut Kentli*, Ali Fuat Alkaya
Marmara University, Engineering Faculty, Mechanical Engineering Department, Istanbul, Turkey
*E-mail: akentli@eng.marmara.edu.tr

**Abstract:** *By the automation of manufacturing, production of a good becomes a bunch of steps and sequences like; bolt sequencing, cutting, drilling and welding process steps. Most of these sequencing type engineering problems can be modeled as traveling salesman problems (TSP), which is widely known in operation research area. Even though much kind of methods and algorithms are developed for TSP, a few of them are applied to engineering problems. This study aims to use a deterministic approach (record-to-record travel algorithm) to solve TSP modeled engineering problems. Bolt sequencing and drilling path problems are chosen from literature as test problems. Results show performance of the proposed algorithm*

**Keywords:** *Path Optimization, TSP, Deterministic Approach, Record-to-Record Travel*

## INTRODUCTION

Engineering problems can be modeled using computer simulation. Determining optimal solutions using such models can be very difficult, due to the large number of process sequences and associated parameter settings that exist. This has forced researchers to develop heuristic strategies to address such optimization problems (bolt assembly sequencing, drilling sequences).

Bolt assembly sequencing is one of the earliest problems considered as TSP problem in manufacturing engineering. Assembly planning requires some knowledge and reasoning method to generate assembly sequences. The heuristics used typically depend upon the assembly and the type of operation that is involved. Huang et.al. (1997) has used GA to solve two sample bolt assembly problems. Sinanoglu et.al. (2005) has introduced a new strategy using NN. Ben-Arieh and Kramer (1994) developed a two-stage methodology to consistently generate all feasible assembly sequences in consideration of the various combinations of subassembly operations.

Drilling is the new engineering problem that researchers have formulated as TSP on mainly this decade. Due to the point-to-point tool movement in drilling, a considerable amount of the processing time is spent on moving the table from one location to another. So, drilling path optimization can save machining time.

Despite the importance of drilling path optimization, few researchers worked on this problem in the literature. First, Kolahan and Liang (1996) have formulated the problem as TSP and worked on applying Tabu search algorithm to solve the problem and then, they extended their research to a more complex case (Parker et.al., 2000). El-Midany et.al. (2007) have considered the application to EDM drilling. Ghaiebi and Solimanpur (2007) have used ant colony algorithm. Krishnaiyer and Cheraghi (2006) used same algorithm and they proposed a web base system. Zhu (2006) has used particle swarm algorithm and then, Zhu and Zhang (2007) has extended the research and applied to new sample problems. Onwubolu and Clerc (2004) have proposed a usage of Particle Swarm algorithm. Sigl and Mayer (2005) has considered CNC drilling route and used Evolutionary algorithm with 2-Opt heuristic. Qudeiri et.al. (2007) have used GA to generate new optimum NC code.

This study introduces new applications of a deterministic approach for different process design optimization problems, called record-to-record travel with local exchange moves (RRTLEM) algorithm (Li et.al., 2007). Computational results are reported. RRTLEM algorithm provide a deterministic approach to such problems (bolt assembly sequencing, drilling sequences). Hereafter the paper is organized as follows. Section 2 describes the problems of interest. Section 3 presents the used methodology. Section 4 provides the results. Section 5 includes conclusions and scope for future works.

## METHODOLOGY

RRTLEM is a local search heuristic. In TSP variants, for a local search a general method is to apply exchanges of edges or nodes (Croes (1958), Waters (1987)). For exchanging two nodes, we considered two alternatives; we can either insert a node into a different place on the route, called 1-0 Exchange move, or we can exchange two nodes, called 1-1 Exchange. Exchanging two edges is called 2-Opt move.

RRTLEM is a hybrid of record-to-record travel (RRT) and local search moves. RRT is a deterministic variant of Simulated Annealing (SA), developed by Dueck (1993) and it is shown that the quality of the computational results obtained so far by RRT is better than SA (Figure 1). RRT starts by a generated initial solution, *s*. *Record* is defined as the cost of best solution (*bs*) observed so far. *Deviation* is defined as a predefined percentage of *Record*. It is deterministic because, a neighbor solution *s'* replaces current solution only if its cost is less than *Record+Deviation*.

```
RRT
s := GenerateInitialSolution()
bs := s
Record := f(bs)
Deviation := Record x p
while termination conditions not met
        s' := PickNextNeighbor(N(s))
        if f(s') < Record + Deviation
                s := s'
                if f(s') < Record
                        bs := s'
                        Record := f(s')
                        Deviation := Record x p
                endif
        endif
endwhile
```

Figure 1. Record-to-record Travel Algorithm

The core of RRTLEM consists of simple iterative statements (Figure 2). How RRT concept is embedded in local search moves is given in Figures 3-5 in detail.

```
Create neighbor list, NL(i)  for each node i.
cs := Create initial solution using Convex-Hull and Or-Opt
for a predefined number of times
   apply 1-0 Exchange move with RRT (uphill moves allowed)
   apply 1-1 Exchange move with RRT (uphill moves allowed)
   apply 2-Opt Exchange move with RRT (uphill moves allowed)
endfor
cs:= bs
apply 1-0 Exchange move with RRT (only downhill moves)
apply 1-1 Exchange move with RRT (only downhill moves)
apply 2-Opt Exchange move with RRT (only downhill moves)
```

Figure 2. RRT with Local Exchange Moves (RRTLEM)

```
for each node i in the current solution cs
   savings := -∞
   for each edge j whose one end is in NL(i)
         obtain s' by inserting node i between edge j
         if f(s') < f(cs)
                  cs := s'
                  if f(s') ≤ f(bs)
                           bs := s', dev := f(bs) x rate
                  endif
                  continue with the next node
         endif
         if f(s') ≥ f(s) and f(s) - f(s') ≥ savings
                  store j as maximum saving edge (mes := j)
         endif
   endfor
   if f(s) – savings ≤ f(bs) + dev and uphill moves allowed
         cs := the solution obtained by inserting node i between mes
   endif
endfor
```

Figure 3. 1-0 Exchange Move with RRT

```
for each node i in the current solution cs
   savings := -∞
   for each node j which is in NL(i)
         obtain s' by exchanging the places of nodes i and j
         if f(s') < f(cs)
                  cs := s'
                  if f(s') ≤ f(bs)
                           bs := s', dev := f(bs) x rate
                  endif
                  continue with the next node
         endif
         if f(s') ≥ f(s) and f(s) - f(s') ≥ savings
                  store j as maximum saving node (msn := j)
         endif
   endfor
   if f(s) - savings ≤ f(bs) + dev and uphill moves allowed
         cs := the solution obtained by exchanging nodes i and msn
   endif
endfor
```

Figure 4. 1-1 Exchange Move with RRT

In 1-0 Exchange move with RRT, 1-1 Exchange move with RRT and 2-Opt Exchange move with RRT as the initial solution, we use the solution given Convex-Hull constructive heuristic and Or-Opt improvement heuristic (Or (1976), Stewart (1977)). During the search process, current solution (*cs*) is the route from which new solutions (*s'*) are obtained and best solution (*bs*) is the route whose cost is the best among the solutions created. Cost of solution *s*, *f(s)*, denotes the total route cost using the sequence stored in *s*. When uphill moves are allowed for a local search move, *cs* can be assigned a new solution whose cost is worse than cost of *cs* but better than cost of *bs* plus *dev*. *dev* is a predefined percentage (*rate*) of the cost of *bs*, and whenever a new *bs* is found, *dev* is also updated. Uphill moves create the opportunity for *cs* to escape from trapping local minima. After executing iterative statements, starting from the *bs*, we apply each local search move once more. The idea is inspired from the study (Li et.al., 2007), but important modifications are added that will improve the performance. Note that in our algorithm, there are two parameters that must be fine tuned in order to obtain best performance. These are the number of iterations, *noi*, and percentage value, *rate*, that is used to calculate the deviation.

```
for each edge i in the current solution cs
    savings := -∞
    for each edge j whose one end is in NL of both ends of i
        obtain s' by reconnecting the edges i and j in the alternative way
        if f(s') < f(cs)
                cs := s'
                if f(s') ≤ f(bs)
                        bs := s', dev := f(bs) x rate
                endif
                continue with the next node
        endif
        if f(s') ≥ f(s) and f(s) - f(s') ≥ savings
                store j as maximum saving edge (mse := j)
        endif
    endfor
    if f(s) – savings ≤ f(bs) + dev and uphill moves allowed
        cs := the solution obtained by reconnecting the edges i and mse in
    the alternative way
    endif
endfor
```

Figure 5. 2-opt Move with RRT

Neighbor list (*NL*) is an important topic that should be explained. During local searches, it is more rational to try exchanges between nodes or edges that are away from each other at most a reasonable amount of distance. For example, for exchanging two nodes, it is not necessary to exchange the node at the most left and the node at the most right. To avoid from these unprofitable moves, we build neighbor lists for each node.

## CASE STUDIES

This section explains the test problems. These problems are selected among literature studies. The complexity of the problem is considered as a criterion in choosing problems. Each following case has different size in path length and number of nodes. Chosen problems are 10 bolt assembly problem, 14 hole drilling problem and 442 point PCB drilling problem.

### *10 Bolt Assembly*

This problem is chosen as a benchmark problem because Huang et.al. (1997) have also solved this problem by using GA. In the case, a metal plate with have 10 bolts at locations shown in Figure 6 is considered. Bolts are to be inserted into the pre-drilled holes. The objective is to minimize the movement while passing over and inserting a bolt into each hole.

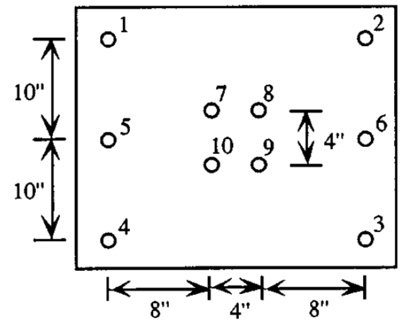


Figure 6. Metal plate with ten holes.

***14 hole drilling***

This case is also chosen as a test case in literature (Zhu (2006), Zhu and Zhang (2007)). Figure 7 shows the part and hole locations. Locations of 14 holes are scattered according to previous case.

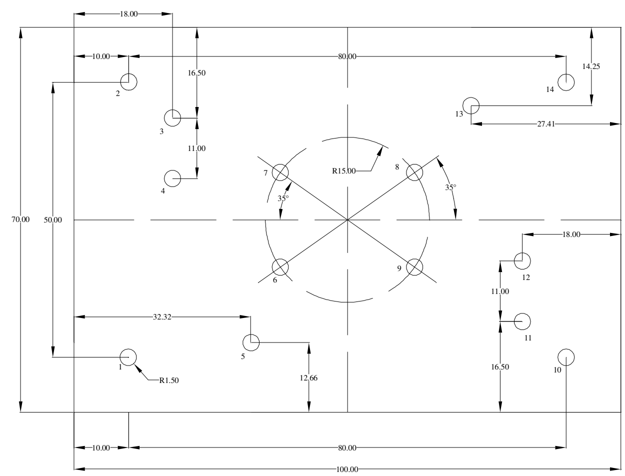

Figure 7. 14 hole-part drawing

***PCB442***

PCB drilling is one of the mostly used benchmark problems in TSP. They could be found in TSP Library (TSPLIB, 2009). There are over a hundred benchmark data with their best solutions obtained up to now. PCB442 is chosen in this study as a benchmark problem (Figure 8).

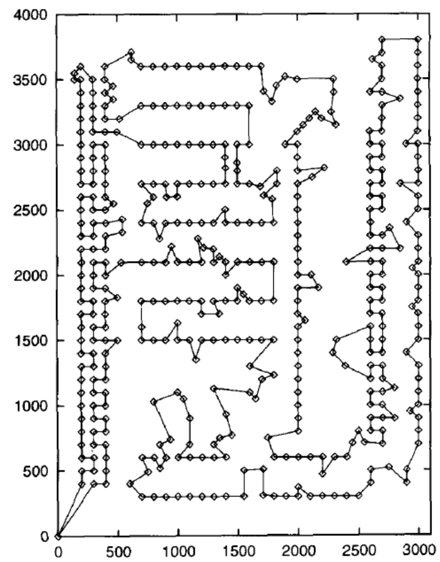

Figure 8.  A solution of PCB442 (Schneider et.al., 1996)

## RESULTS

As we pointed out in the previous sections, the heuristic is tested on three different type engineering problems.  In this section, we summarize the results and compare our findings with studies in literature.

### *10 Bolt Assembly*

Cost of obtained bolt assembly path (shown in Figure 9) is 93.25. On the other hand, Arora et.al has found 74.6274. It should be mentioned that assembly path is accepted open in their setup. Thus, distance between node 1 and 2 is omitted. But, robot arm or worker hand should come back to first point to assemble new product. So, this distance should be added to total cost. Total cost becomes 94.6274.

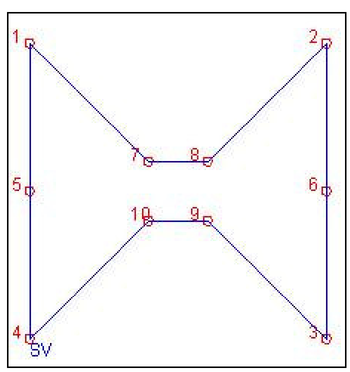


Figure 9.  Solution for bolt assembly

### *14 hole drilling*

Drilling sequence for the work-piece (shown in Figure 10) is obtained as 290.40. This path cost is smaller than many results in literature (shown in Table 1).

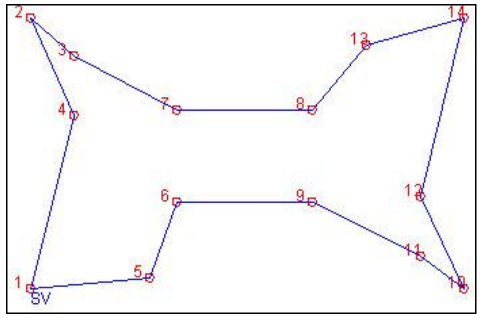


Figure 10. Solution for hole drilling.

Table 1. Comparison of our findings with literature and optimum.

| PROBLEM | COST | | |
|---|---|---|---|
| | Our Approach | Literature | Best Found |
| Bolt | 93.25 | 94.6274 (Huang et.al., 1997) | 93.25 |
| Drilling | 290.40 | 291 (Zhu (2006), Zhu and Zhang (2007) | 280 |
| PCB442 | 52327.81 | 55609.48 (Murakoshi and Sato, 2007) | 50778 |

***PCB442***

Cost of drilling path for PCB442 is obtained as 52327.81 (Figure 11). There are a lot of studies on this problem, so exact optimum path (cost is 50778) is already obtained. But our, approach gives an approximate solution in an acceptable execution time.

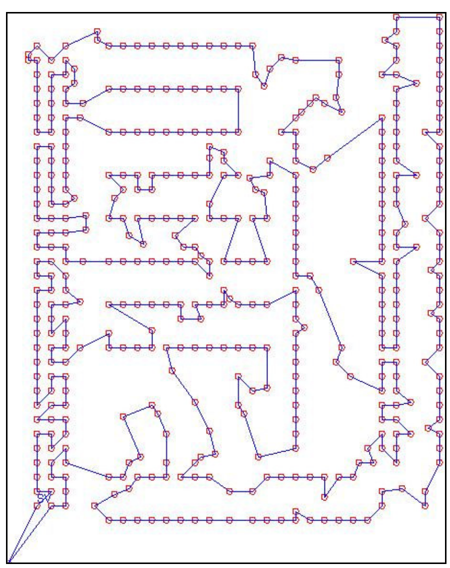

Figure 11. Solution for PCB442

As a result, deterministic approaches give an acceptable solution to engineering problems (as shown in Table 1), especially for smaller scale problems. It could be easily concluded that using artificial intelligence is not unique solution for every case. Our approach is giving results with 3% percent deviation from exact optimum in worst cases and most of time; it is giving better results than the other approaches.

## CONCLUSION

This study has showed that deterministic approaches are giving accurate solutions as much as evolutionary algorithms and hybrid approaches for sequence type engineering problems. Furthermore, algorithm does not need to test for different times to validate its efficiency due they always find the same solution. It is believed that usefulness of the deterministic approach will be easily understood if application to different kind problems is accomplished. It is also recommended as a further study to work on the adjustment of the RRTLEM parameters to get much more accurate results for large scale problems.

## REFERENCES


Ben-Arieh D. and Kramer B. (1994). Computer-aided process planning for assembly: generation of assembly operations sequence. *International Journal of Production Research*, 32 (3), 643-656.

Croes G. (1958). A Method for Solving Traveling-Salesman Problems. *Operations Research*, 6, 791-812.

Dueck G. (1993). New optimization heuristics: the great deluge algorithm and the record-to-record travel. *Journal of Computational Physics*, 104, 86-92.

EL-Midany T.T., Kohail A.M., Tawfik H. (2007). A Proposed Algorithm for Optimizing the Toolpoint Path of the Small-Hole EDM-Drilling. *Proceedings of Geometric Modeling and Imaging*, 25-32.

Ghaiebi H. and Solimanpur M. (2007). An ant algorithm for optimization of hole-making operations. *Computers and Industrial Engineering*, 52 (2), 308-319.

Huang M., Hsieh C., Arora J.S. (1997). A genetic algorithm for sequencing type problems in engineering design. *International Journal for Numerical Methods in Engineering*, 40, 3105-3115.

Kolahan F. and Liang M. (1996). A tabu search approach to optimization of drilling operations. *Computers and IndustrialEngineering*, 31 (1-2), 371-374.

Krishnaiyer K. and Cheraghi S.H. (2006). Ant algorithms: web-based implementation and applications to manufacturing system problems. *International Journal of Computer Integrated Manufacturing*, 19 (3), 264 – 277.

Li F., Golden B., Wasil E. (2007). A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem. *Computers and Operation Research*, 34 (9), 2734-2742.

Murakoshi K. and Sato Y. (2007). Reducing topological defects in self-organizing maps using multiple scale neighbourhood functions. *Biosystems*, 90 (1), 101-104.

Onwubolu G.C. and Clerc M. (2004). Optimal path for automated drilling operations by a new heuristic approach using particle swarm optimization. *International Journal of Production Research*, 42 (3), 473-491.

Or I. (1976). *Travelling Salesman Type Combinatorial Problems and Their Relation to the Logistics of Blood Banking*. Thesis (PhD). Northwestern University.

Parker S., Kolahan F., Liang M. (2000). Optimization of hole-making operations: a tabu-search approach. *International Journal of Machine Tools and Manufacture*, 40 (12), 1735-1753.

Qudeiri J.A., Yamamoto H., Ramli R. (2007). Optimization of operation sequence in CNC machine tools using genetic algorithm. *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, 1 (2), 272–282.

Schneider J., Froschhammer C., Morgenstern I., Husslein T., Singer J.M. (1996). Searching for backbones - an efficient parallel algorithm for the traveling salesman problem. *Computer Physics Communications*, 96 (2), 173-188.

Sigl S. and Mayer H.A. (2005). Hybrid Evolutionary Approaches to CNC Drill Route Optimization. *Proceedings of Computational Intelligence for Modeling, Control and Automation*, 1, 905-910.

Sinanoglu C. and Börklü H.R. (2005). An assembly sequence-planning system for mechanical parts using neural network. *Assembly Automation*, 25, 38-52.

Stewart W.R. (1977). A computationally efficient heuristic for the travelling salesman problem. *Proceedings of the 13th Annual Meeting of Southeastern TIMS*, Myrtle Beach, SC, USA, 75–83.

TSPLIB, Accessed 13 March 2009: http://www.iwr.uni-heidelberg.de/groups/comopt/ software/TSPLIB95/.

Waters C.D.J. (1987). A Solution Procedure for the Vehicle-Scheduling Problem Based on Iterative Route Improvement. *The Journal of the Operational Research Society*, 38 (9), 833-839.

Zhu G.-Y. (2006). Drilling Path Optimization Based on Swarm Intelligent Algorithm. *Proceedings of IEEE International Conference on Robotics and Biomimetics*, 193-196.

Zhu G.-Y. and Zhang W.-B. (2007). Drilling path optimization by the particle swarm optimization algorithm with global convergence characteristics. *International Journal of Production Research*, 46 (8), 2299-2311.