American Journal of Applied Sciences 8 (2): 107-112, 2011
ISSN 1546-9239
© 2010 Science Publications

Optimization of Capacitated Vehicle Routing Problem by Nested Particle Swarm Optimization

[1]Karuppusamy Kanthavel and [2]Paruchuri Prasad
[1]Department of Mechanical Engineering,
Anna University of Technology, Coimbatore, Tamilnadu, India
[2]Department of Mechanical Engineering,
P.S.G. College of Technology, Coimbatore, Tamilnadu, India

Abstract: Problem statement: Vehicle routing problem determines the optimum route for each vehicle as a sequence of visiting cities. The problem has been defined as NP-hard and exact solution is relatively difficult to achieve for real time large scale models. Though several attempts to solve the problem were made in the literature, new approaches may be tried to solve the problem to further reduce computational efforts. **Approach:** In this context this study focuses on maximum utilization of loading capacity and determines the optimum set of vehicle routes for Capacitated Vehicle Routing Problem (CVRP) by a Nested Particle Swarm Optimization (NPSO) technique. The algorithm is implemented as Master PSO and slave PSO for the identification of candidate list and route sequence in nested form to optimize the model. **Results:** Benchmarking data set of capacitated vehicle routing is considered for the evaluations. The total distance of set vehicle route obtained by the new approach is compared with the best known solution and other existing techniques. **Conclusions/Recommendations:** The NPSO produces significant results and computational performance than the existing PSO algorithms. This newly proposed NPSO algorithm develops the vehicle schedule without any local optimization technique.

Key words: Capacitated Vehicle Routing Problem (CVRP), Nested Particle Swarm Optimization (NPSO), Travelling Salesman Problem (TSP), metaheuristic techniques, local optimization

INTRODUCTION

The Capacitated Vehicle Routing Problem (CVRP) exists in various real-time cases with constraints of vehicle capacity, number of visits, time windows for the customers and route length. The CVRP problem (Dantzig and Ramser, 1959) was first formulated as a NP-hard problem and it attracted several researchers' attention to propose exact, heuristic and metaheuristic techniques. A constructive heuristics was proposed (Clarke and Wright, 1964), which integrates cities on maximum saving theory to determine the minimum travel distance of a vehicle in the model. The CVRP attracted further attention from the researchers (Cordeau *et al*., 2002; Lysgaard *et al*., 2004), the model was formulated and well defined.

In real time logistics transportation is the key operation, model building and developing solution techniques for CVRP is the basic step to solve complex models. In enumeration CVRP requires very high computational operations and time to find the optimal solution hence it is declared as NP-hard problem (Haimovich *et al*., 1988) and solved by various metaheuristics. Metaheuristics are originally defined as solution methods that orchestrate an interaction between local improvement procedure and higher level strategies to create a process capable of escaping from local optima and performing a robust search for a solution space (Glover and Kochenberger, 2003), which produces result more close to the optimum value with less time. The CVRP model was solved by simulated annealing (Alfa *et al*., 1991), Tabu Search (Thaillard, 1993; Gendreau *et al*., 1994), Genetic Algorithms (Potvin and Bengio, 1996; Baker and Ayechew, 2003; Sarabian and Lee, 2010; Nazif and Lee, 2010), Ant Colony Optimization (Bullnheimer *et al*., 1999) and recently by discrete and classical version of Particle swarm optimization (Chen *et al*., 2006; Ai and Kachitvichyanukul, 2007) techniques. These metaheuristic techniques implemented along with local optimization to obtain the results, but as a further improvement, exclusive classical PSO without local optimization is implemented in this study.

Corresponding author: Karuppusamy Kanthavel, Department of Mechanical Engineering, Anna University of Technology, Coimbatore, Tamilnadu, India

CVRP model: Capacitated Vehicle Routing Problem (CVRP) was defined (Cordeau *et al*., 2002; Lysgaard *et al*., 2004) as a Set of n customers served from the common depot or warehouse of 0 , for a non negative q_i customer demand by N number of vehicles of having capacity of Q and distance or cost of C_{ij} between two nodes of i and j by vehicle k. The objective of CVRP is to determine optimum route schedule which minimizes the distance or cost with the following constraints:

- Each customer is served exactly once by exactly one vehicle
- Each vehicle starts and ends its route at the warehouse
- The total length of each route must not exceed the constraint
- The total demand of any route must not exceed the capacity of the vehicle

MATERIALS AND METHODS

The CVRP mathematical model was formulated based on previous study (Bodin *et al*., 1983) to explain the objective function, vehicle schedule with constraints. The objective function is expressed in Eq. 1, which aims to minimize the sum of set the of routes visiting by the vehicles in the model. The customer is exactly visited only once by a vehicle and it is ensured by the Eq. 2 and 3, the vehicle visit between two customers is assigned as X_{ij}^{k} =1 otherwise "0" to obtain the objective function. The vehicle tour starts at warehouse, visit customers on sequence and finish with the warehouse. In this tour, the vehicle needs to visit the cities continuously and Eq. 4 ensures the continuity visit among the cities in the route:

$$\text{Min}\sum_{k=1}^{N}\sum_{i=0}^{n}\sum_{j=0}^{n}C_{ij}^{k}X_{ij}^{k} \tag{1}$$

Subject to:

$$\sum_{k=1}^{N}\sum_{i=0}^{n}X_{ij}^{k}=1,j=1,2,....,n \tag{2}$$

$$\sum_{k=1}^{N}\sum_{j=0}^{n}X_{ij}^{k}=1,i=1,2,....,n \tag{3}$$

$$\sum_{i=0}^{n}X_{it}^{k}-\sum_{j=0}^{n}X_{tj}^{k}=0,k=1,2,.....,N;t=1,2,.....n \tag{4}$$

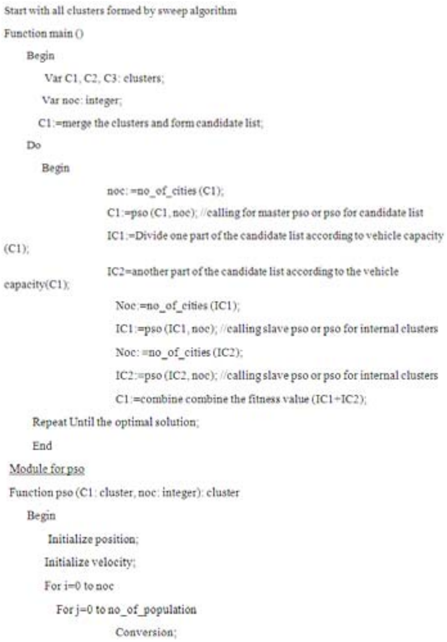
Nested particle swarm optimization:

Particle swarm optimization: Particle swarm optimization is a metaheuristic introduced (Kennedy and Eberhart, 1995) based on the simulation of the social behavior of birds within a flock. The advantages of PSO are ease in implementation compared to other evolutionary methods and only few parameters need to be adjusted in the algorithm. Similar to other evolutionary algorithms, PSO also has a fitness function that takes the particle's position and assigns to it a fitness value. The position with the minimum fitness value in the entire run is called as social or global best. Each particle keeps track of its minimum fitness value, called its cognitive or local best. Each particle is initialized with a random position and random velocity. The velocity of the particle, each of dimensions, is accelerated towards the global best and its own local best. The inertia weight has a well balanced mechanism with flexibility to enhance and adapt to both global and local exploration abilities.

The detailed study of applications, technique and models are presented (Kennedy *et al*., 2001; Clerc, 2006) to further enlightened the scope of PSO. The CVRP benchmarking problem was solved by hybrid discrete PSO (Chen *et al*., 2006) with Simulated Annealing (SA) algorithm. A classical version of PSO (Ai and Kachitvichyanukul, 2007) was introduced to solve CVRP and further (Ai and Kachitvichyanukul, 2009) proposed PSO based on GLNPSO to solve CVRP by two different solution representations namely; SR-1 and SR-2 and also local optimization is incorporated on PSO solutions. In another PSO study, Chen demonstrated the power of PSO by getting exact solution for up to 67 cities and inexact solution for above 67 cities cases and however these methods consumes more computational time. Further in PSO (Ai and Kachitvichyanukul, 2009) the SR-1 and SR-2 was proposed with local improvement technique, SR-2 produced relatively better result and it attained moderate consistency in producing exact result for the data set.

NPSO algorithm: The present study proposes two stage implementations of PSO to solve the given CVRP problem. Initially feasible individual clusters are formed using sweep algorithm and the clusters are analyzed for arriving of optimal vehicles routes. Since the clusters and the corresponding routes formed found to be not optimal, route optimization is carried out by merging multiple successive clusters. PSO is implemented to reorganize the clusters (Master PSO) as well as optimize each cluster for optimum route (Slave PSO) in nested form called as nested PSO (NPSO).

This NPSO algorithm is functioning in first stage as Master PSO and in second stage as slave PSO to feed candidate list and to formulate vehicle routes within the candidate list respectively. NPSO pseudo code is shown in Fig. 1. Initially cities are clustered by sweep algorithm to the individual route and multiple clusters are merged to form a candidate list. While clustering, the sweep algorithm prefers cities on polar coordinate order to fill maximum capacity of vehicle and during cluster formation last prioritized city may cause violation of loading constraint, however subsequent city preferred to ensure maximum filling condition of the vehicle. The multiple clusters formed by sweep algorithm are merged to form the candidate list as mentioned in the Fig. 2.



```
Start with all clusters formed by sweep algorithm
Function main ()
    Begin
       Var C1, C2, C3: clusters;
       Var noc: integer;
      C1:=merge the clusters and form candidate list;
    Do
       Begin
                    noc: =no_of_cities (C1);
                    C1:=pso (C1, noc); //calling for master pso or pso for candidate list
                    IC1:=Divide one part of the candidate list according to vehicle capacity
(C1);
                    IC2=another part of the candidate list according to the vehicle
capacity(C1);
                      Noc:=no_of_cities (IC1);
                      IC1:=pso (IC1, noc); //calling slave pso or pso for internal clusters
                      Noc: =no_of_cities (IC2);
                      IC2:=pso (IC2, noc); //calling slave pso or pso for internal clusters
                      C1:=combine combine the fitness value (IC1+IC2);
     Repeat Until the optimal solution;
     End
 Module for pso
 Function pso (C1: cluster, noc: integer): cluster
    Begin
       Initialize position;
       Initialize velocity;
       For i=0 to noc
          For j=0 to no_of_population
                    Conversion;
```

Fig. 1: NPSO pseudo code

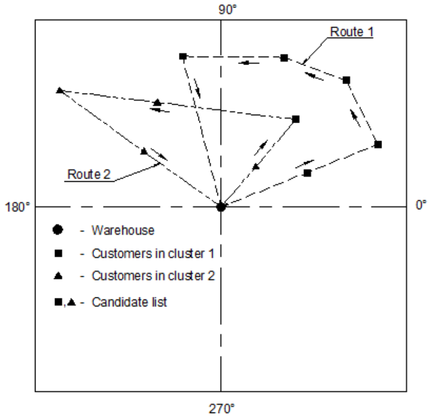


Fig. 2: Illustration of Candidate list and Route

Master PSO in initialization process selects candidate list and assigns random number to each city to form a particle as given in Table 1. In subsequent process random numbers are arranged in the ascending order on rank basis along with the corresponding particle as shown in Table 2 and in the candidate list particle are grouped for each vehicle based on the capacity say vehicle set 1, vehicle set 2. Slave PSO receives the each set information and solve as Travelling Salesman Problem (Srichandum and Rujirayanyong, 2010) to determine shortest distance of each route separately as listed in Table 3. As mentioned in Table 3, Step-1 and Step-2 particles are arranged on rank basis by slave PSO to calculate the required fitness value of shortest route of each vehicle.

The shortest distance of each route calculated by slave PSO subsequently forwarded to the master PSO as a nested function. Master PSO updates the route distance and calculates overall fitness function, set of vehicle routes for each population by the Eq. 1. In each iteration local best and global best value are updated to achieve the optimum solution. Updating Particle velocity is governed by Eq. 5 and the new position of the particle is controlled by Eq. 6 in both master PSO and slave PSO. The optimum sets of vehicle route are only possible to obtain in the merged condition of clusters or with the best candidate list. In Fig. 2 formation of cluster, candidate list and route formation are illustrated.

PSO formula: The PSO formula implemented in NPSO for mater PSO and slave PSO algorithm is given below:

Table 1: Particle and generation of random numbers

	Candidate list							
	Route -1 and Route -2							
Particle	1.00	2.00	3.00	4.00	5.00	6.00	7.00	8.00
Position	0.87	0.65	0.45	0.90	0.12	0.35	0.56	0.32
Rank	7.00	6.00	4.00	8.00	1.00	3.00	5.00	2.00

Table 2 Particle arranged in ascending order of random numbers

Candidate list								
	Route -1				Route -2			
Particle	5.00	8.00	6.00	3.00	7.00	2.00	1.00	4.00
Position	0.12	0.32	0.35	0.45	0.56	0.65	0.87	0.90
Rank	1.00	2.00	3.00	4.00	5.00	6.00	7.00	8.00

Table 3: Particle arranged for individual vehicle

	Route -1				Route-2			
Step -1								
Particle	8.00	6.00	3.00	5.00	2.00	1.00	7.00	4.00
Position	0.12	0.09	0.88	0.75	0.75	0.65	0.23	0.48
Rank	2.00	1.00	4.00	3.00	4.00	3.00	1.00	2.00
Step -2								
Particle	6.00	8.00	5.00	3.00	7.00	4.00	1.00	2.00
Position	0.09	0.12	0.75	0.88	0.23	0.48	0.65	0.75
Rank	1.00	2.00	3.00	4.00	1.00	2.00	3.00	4.00

Table 4: Comparison of computational result*

				Distance				Time (sec)		
Instance	Q	n	N	BKS	Chen	The Jin Ai	NPSO	Chen	The Jin Ai	NPSO
An33k5	100	32	5	661	661	661	661	32	13	8
An46k7	100	45	7	914	914	914	914	129	23	15
An60k9	100	59	9	1354	1354	1365	1354	309	40	25
Bn35k5	100	34	5	955	955	955	955	38	14	9
Bn45k5	100	44	5	751	751	951	751	134	20	14
Bn68k9	100	67	9	1272	1272	1274	1272	344	50	28
Bn78k10	100	77	10	1221	1239	1223	1221	429	64	36
En30k3	4500	29	3	534	534	534	534	28	16	8
En51k5	160	50	5	521	528	521	521	301	22	15
En76k7	220	75	7	682	688	682	682	527	60	35
Fn72k4	30000	71	4	237	244	237	237	398	53	50
Fn135k7	2210	134	7	1162	1215	1162	1162	1526	258	165
Mn101k10	200	100	10	820	824	820	820	874	114	56
Mn121k7	200	120	7	1034	1038	1036	1034	1734	89	80
Pn76k4	350	75	4	593	602	594	593	496	48	45
Pn101k4	400	100	4	681	694	683	681	978	86	90

*: Q-vehicle capacity, n- number of cities, N - number of vehicles, BKS is best known solution so far, Chen is the result obtained (Chen *et al.* ,2006) SR-2 is the result obtained by (Ai and Kachitvichyanukul, 2009) and NPSO is the result obtained in this study

$$\vec{v}_{i+1} = w\vec{v}_i + c_1\vec{r}_1 \times \left(\vec{p}_i - \vec{x}_i\right) + c_2\vec{r}_2 \times \left(\vec{p}_g - \vec{x}_i\right) \quad (5)$$

$$\vec{x}_{i+1} = \vec{x}_i + \vec{v}_{i+1} \quad (6)$$

$\vec{x}_i$ = Current position of the i-th particle in the swarm

$\vec{v}_i$ = Velocity of the i-th particle

$\vec{p}_i$ = Best position found by the i-th particle, local best

$\vec{p}_g$ = Best position found from the particle's neighborhood, global best

c_1, c_2 = Acceleration coefficient

$\vec{r}_1, \vec{r}_2$ = Random numbers uniformly chosen from (0,1) w

= Inertia weight

RESULTS

Benchmarking capacitated vehicle routing problem data set has been considered for performance evaluation of the algorithm. The algorithm is implemented through advanced Java language in NetBeans IDE 6.5 environment in PC with Intel P4, 3.4 GHz, 2 GB RAM. Each instance data set has run for five replications and the best results are tabulated.

The PSO Parameters are chosen based on the available literature (Kennedy *et al*., 2001) and NPSO algorithm performance. The following parameters implemented for Master PSO: number of iterations = 5000, acceleration coefficients c_1 and c_2 = 2, inertia weight = 0.41, population=20 and Slave PSO: number of iterations = 2000, acceleration coefficients c_1 and c_2= 2, inertia weight = 0.41, population= 20.

In this benchmarking data set Euclidian distance of the cities are calculated with two decimal accuracy and same accuracy maintained for individual vehicle route. The total distance covered by all the vehicles is calculated and decimal value ignored for tabulation. The sixteen benchmarking data set of experimental (Chen *et al*., 2006; Ai and Kachitvichyanukul, 2009) results are listed in the Table 4 along with the computational time.

DISCUSSION

The results are compared with the Best Known Solution (BKS) of the data set. In this data the number of cities varies from minimum of 32 to maximum of 134 and number of vehicle varies from minimum of 3 to maximum 10. The comparison shows, Chen *et al.* (2006) results are matching with exact solution for seven instances and in remaining instances, a maximum deviation of 53 units in Fn135k7 instance is observed. Whereas Ai and Kachitvichyanukul (2009) study produced exact result for ten instances with maximum deviation of 11 units in An60k9 instance and NPSO produced exact result for all instances.

The computational performance analysis of algorithm is shown in the Table 5 in terms of minimum time, maximum time, mean and standard deviation. Chen *et al.* (2006) has consumed maximum time of 1743 second for Mn121k7 instance, Ai and Kachitvichyanukul (2009) has consumed maximum time of 258 second for Fn135k7 instance, NPSO consumed maximum of 165 second for Fn135k7 and in only 4 second extra time consumed for Pn101k4 instance.

Table 5: Computational performance analysis

Work	Min Time (sec)	Max Time (sec)	Mean	Standard Deviation
Chen *et al*. (2006)	28	1743	517.00	515.26
Ai and Kachitvichyanukul (2009)	13	258	60.00	60.56
NPSO	8	165	42.43	41.11

The existing PSO (Chen *et al*., 2006; Ai and Kachitvichyanukul, 2009) algorithm result shows the deviation of few instance with best Know solution and more (Chen *et al*., 2006) and moderate (Ai and Kachitvichyanukul, 2009) time consumption with the new proposed NPSO. The performance analysis indicated in Table 5, reveal the minimum range, minimum mean and minimum standard deviation is falling with NPSO among the other technique. The proposed nested operation of PSO is superior, eliminates local improvement technique while comparing to other exiting PSO algorithm.

CONCLUSION

The study focused on implementation of Nested Particle Swarm Optimization (NPSO) algorithm at two levels as master PSO and slave PSO for solving capacitated vehicle routing problem. The new NPSO algorithm was tested in terms of solution quality and computational performance by comparing with other published data. The NPSO solution quality and computational performance were found to better than the existing PSO algorithms which determined exact result with less computational time. As an extension, NPSO can be introduced to other variants of VRP models with suitable modification.

REFERENCES

Ai, J. and V. Kachitvichyanukul, 2009. Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem. Comput. Indus. Eng., 56: 380-387. DOI: 10.1016/j.cie.2008.06.012

Ai, T.J. and V. Kachitvichyanukul, 2007. A particle swarm optimization for the capacitated vehicle routing problem. Int. J. Logistics SCM Syst., 2: 50-55.

Alfa, A.S., S.S. Heragu and M. Chen, 1991. A 3-opt based simulated annealing algorithm for vehicle routing problems. Comput. Indus. Eng., 21: 635-639. DOI: 10.1016/0360-8352(91)90165-3

Baker, B.M. and M.A. Ayechew, 2003. A genetic algorithm for the vehicle routing problem. Comput. Operations Res., 30: 787-800. DOI: 10.1016/S0305-0548(02)00051-5

Bodin, L., B.L. Golden, A. Assad and M.O. Ball, 1983. Routing and scheduling of the vehicle and crews: The state of the art. Comp. Operations Res., 10: 63-212.

Bullnheimer, R., F. Hartl and C. Strauss, 1999. An improved ant system algorithm for thevehicle routing problem. Annals Operation Res., 89: 319-328. DOI: 10.1023/A:1018940026670

Chen, A.L., G.K. Yang and Z.M. Wu, 2006. Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem. J. Zhejiang Univ. Sci., 7: 607-614. DOI: 10.1631/jzus.2006.A0607

Clarke, G. and J.V. Wright, 1964. Scheduling of vehicles from a central depot to a number of delivery points. Operations Res., 12: 568-581. DOI: 10.1287/opre.12.4.568

Clerc, M., 2006. Particle Swarm Optimization. 1st Edn., ISTE, London, ISBN: 1905209045, pp: 243.

Cordeau, J.F., M. Gendreau, G. Laporte, J.Y. Potvin and F. Semet, 2002. A guide to vehicle routing heuristics. J. Operat. Res. Soci., 53: 512-522. DOI: 10.1057/palgrave.jors.2601319

Dantzig, G.B. and J.H. Ramser, 1959. The truck dispatching problem. Manage. Sci., 6: 80-91. DOI: 10.1287/mnsc.6.1.80

Gendreau, M., A. Hertz and G. Laporte, 1994. A tabu search heuristic for the vehicle routing problem. Manage. Sci., 40: 1276-1290. DOI: 10.1287/mnsc.40.10.1276

Glover, F. and G.A. Kochenberger, 2003. Handbook of Metaheuristics. 1st Edn., Springer, USA., ISBN-10: 1402072635, pp: 570.

Haimovich, M., A.H.G. Rinnooy Kan and L. Stougie, 1988. Analysis of heuristics for vehicle routing problems. 1st Edn., Erasmus University Rotterdam, Rotterdam, pp: 11.

Kennedy, J. and R. Eberhart, 1995. Particle swarm optimization Proceeding of IEEE International Conference on Neural Network, (ICNN’95), Perth, Australia, pp: 1942-1948.

Kennedy, J., R. Eberhart and Y. Shi, 2001. Swarm Intelligence. 1st Edn., Morgan Kaufman Publishers, USA., ISBN-10: 1558605959, pp: 512.

Lysgaard, J., A.N. Letchford and R.W. Eglese, 2004. A new branch-and-cut algorithm for the capacitated vehicle routing problem. Mathem. Programm., 100: 423-445. DOI: 10.1007/s10107-003-0481-8

Nazif, H. and L.S. Lee, 2010. Optimized crossover genetic algorithm for vehicle routing problem with time windows. Am. J. Applied Sci., 7: 95-101. DOI: 10.3844/ajassp.2010.95.101

Potvin, J.Y. and S. Bengio, 1996. The vehicle routing problem with time windows Part II: Genetic Search. Informs J. Comput., 8: 165-172. DOI: 10.1287/ijoc.8.2.165

Sarabian, M. and L.V. Lee, 2010. A modified partially mapped multicrossover genetic algorithm for two-dimensional bin packing problem. J. Math. Stat., 6: 157-162. DOI: 10.3844/jmssp.2010.157.162

Srichandum, S. and T. Rujirayanyong, 2010. Production scheduling for dispatching ready mixed concrete trucks using bee colony optimization. Am. J. Eng. Applied Sci., 3: 7-14. DOI: 10.3844/ajeassp.2010.7.14

Thaillard, E., 1993. Parallel iterative search methods for vehicle routing problems. Networks, 23: 661-673. DOI: 10.1002/net.3230230804