Chapter 6
Cohort Intelligence for Solving Travelling Salesman Problems

As demonstrated previously, the performance of the Cohort Intelligence Algorithm (CI) algorithm was quite satisfactory for solving combinatorial problem such as Knapsack Problem (KP) [1]. The purpose of this chapter is to further demonstrate:

1. The ability of the CI methodology solving classic NP-hard combinatorial problem such as the Traveling Salesman Problem (TSP). In all 9 small sized test cases (14 to 29 cities) of the TSP from the TSPLIB [2] were solved.
2. In addition to the original CI approach incorporated with a *roulette wheel selection approach* [3], two different approaches such as a best *behavior selection approach* and a *random behavior selection approach* have been incorporated. In the best behavior selection approach every candidate follows the best behavior in the cohort. In the latter approach candidates randomly select any behavior in the cohort.
3. In order to jump out of possible local minima and further make the cohort saturate to global minimum, a generic approach of perturbation and further accepting worst behaviors was successfully incorporated.

The results highlighted the simplicity of the CI algorithm as well as robustness of the solution with the three approaches. It also underscored that the CI incorporated with the roulette wheel selection approach more realistically resembles the competitive and interactive learning behavior of the cohort candidates, which eventually makes the cohort successful. In addition, it also demonstrated that always following the best behavior/solution may make the cohort to saturate faster; however may make the cohort stuck into local minima. The encouraging results may help solve the real world problems with increasing complexity as the TSP can be further generalized to a wide variety of routing and scheduling problems [4].

A.J. Kulkarni et al., *Cohort Intelligence: A Socio-inspired Optimization Method*,
Intelligent Systems Reference Library 114, DOI 10.1007/978-3-319-44254-9_6

6.1 Traveling Salesman Problem (TSP)

The Travelling Salesman Problem (TSP) is a classic combinatorial NP Hard problem [5–8]. It includes N cities and one salesman. One of the N cities is considered as origin city. The salesman must start from origin city, visit all the remaining cities exactly once and must return to the origin city. The goal is to find the minimum cost (distance, time, etc.) of route/path that the salesman should follow. It is represented as $f(\mathbf{v}) = f(v_O, \ldots v_i, \ldots v_N, v_O)$ where, v_O is the origin city and v_i represents any intermediate city and $\mathbf{v} = (v_O, \ldots v_i, \ldots v_N, v_O)$ is the route the salesman follows or the order in which the salesman visits the N cities.

6.1.1 *Solution to TSP Using CI*

In the context of CI algorithm presented in Chap. 2 the edges in the route $\mathbf{v} = (v_O, \ldots v_i, \ldots v_N, v_O)$ are considered as the characteristics/qualities which decide the overall cost of the route. The procedure begins with the initialization of number of cohort candidates C, number of variations t and the route $\mathbf{v}^c$ of every candidate $c,\ (c = 1, \ldots, C)$.

In the cohort of C candidates, every individual candidate $c,\ (c = 1, \ldots, C)$ has a route $\mathbf{v}^c = (v_O^c, \ldots v_i^c, \ldots v_N^c, v_O^c)$ which is the order in which the candidate visits the N cities. The origin city v_O^c of the route of every candidate $c,\ (c = 1, \ldots, C)$ was fixed. The remaining cities were arranged randomly in the route. This way C routes $(\mathbf{v}^1, \ldots, \mathbf{v}^c, \ldots, \mathbf{v}^C)$ are formed. And associated costs are calculated $\mathbf{F}^C = \left\{f(\mathbf{v}^1), \ldots, f(\mathbf{v}^c), \ldots, f(\mathbf{v}^C)\right\}$. The following procedure is explained in the context of the *roulette wheel selection* approach.

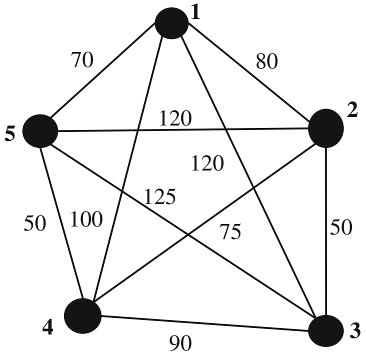
(a) In the context of roulette wheel approach, the probability p^c of selecting the route $\mathbf{v}^c$ of every associated candidate $c,\ (c = 1, \ldots, C)$ is calculated.

(b) Every candidate $c,\ (c = 1, \ldots, C)$ using roulette wheel approach selects to follow a certain route $\mathbf{v}^{(c)}$ of some other candidate (c), i.e. it incorporates an edge from within $\mathbf{v}^{(c)} = \left(v_O^{(c)}, \ldots v_i^{(c)}, \ldots v_N^{(c)}, v_O^{(c)}\right)$ and incorporates into its existing route $\mathbf{v}^c = \left(v_O^c, \ldots v_i^c, \ldots v_N^c, v_O^c\right)$. Following a route $\mathbf{v}^{(c)} = \left(v_O^{(c)}, \ldots v_i^{(c)}, \ldots v_N^{(c)}, v_O^{(c)}\right)$ means incorporating an edge from it into its existing route $\mathbf{v}^c = \left(v_O^c, \ldots v_i^c, \ldots v_N^c, v_O^c\right)$. More specifically, an edge $(v_i^{(c)}, v_{i+1}^{(c)})$ from within $\mathbf{v}^{(c)} = \left(v_O, \ldots v_i^{(c)}, \ldots v_N^{(c)}, v_O\right)$ is selected. The positions of the cities of this edge in the route $\mathbf{v}^c = \left(v_O, \ldots v_i^c, \ldots v_N^c, v_O\right)$ of candidate c are identified. Then the position of the city which is farther from the city of origin v_O^c in the route is swapped with the position of the city immediately after the city which is closer to origin v_O^c. In other words, the positions of the cities of this edge

$\left(\text{i.e.}\ \left(v_i^{(c)}, v_{i+1}^{(c)}\right)\right)$ in the route $\mathbf{v}^c = (v_O, \ldots v_i^c, \ldots v_N^c, v_O)$ of candidate c are identified (say a and b). Then cities at positions $a + 1$ and b are swapped, which makes the edge $\left(v_i^{(c)}, v_{i+1}^{(c)}\right)$ a part of the route $\mathbf{v}^c$.

(c) In this way, every candidate c, $(c = 1, \ldots, C)$ forms t new independent routes with associated costs $\mathbf{F}^{c,t} = \left\{f(\mathbf{v}^c)^1, \ldots, f(\mathbf{v}^c)^j, \ldots, f(\mathbf{v}^c)^t\right\}$, $(c = 1, \ldots, C)$ and further selects the best cost route amongst them. This makes the cohort available with C updated routes with their costs represented as $\mathbf{F}^C = \left\{f^*(\mathbf{v}^1), \ldots, f^*(\mathbf{v}^c), \ldots, f^*(\mathbf{v}^C)\right\}$.

This process continues until saturation, i.e. every candidate finds the same route and does not change for successive considerable number of learning attempts. The above discussed procedure of solving TSP using CI algorithm is illustrated in Figs. 6.1, 6.2 and 6.3 with a 5 city TSP, $C = 3$, origin city v_O as 1, the corresponding route $\mathbf{v}^c$.



$$\begin{bmatrix} - & 80 & 120 & 100 & 70 \\ 80 & - & 50 & 75 & 120 \\ 120 & 50 & - & 90 & 125 \\ 100 & 75 & 90 & - & 50 \\ 70 & 120 & 125 & 50 & - \end{bmatrix}$$

Cost matrix

Fig. 6.1 Illustrative example with 5 cities

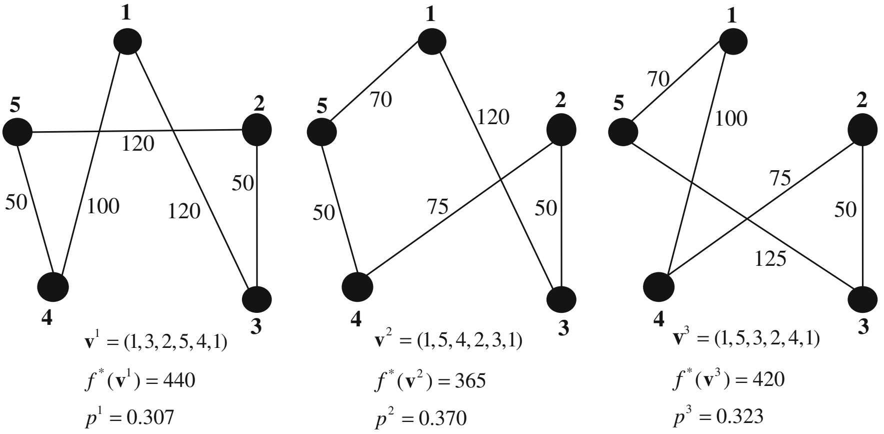

1
5
2
120
50
100
120
50
4
3
$\mathbf{v}^1 = (1,3,2,5,4,1)$
$f^*(\mathbf{v}^1) = 440$
$p^1 = 0.307$
1
5
70
120
2
50
75
50
4
3
$\mathbf{v}^2 = (1,5,4,2,3,1)$
$f^*(\mathbf{v}^2) = 365$
$p^2 = 0.370$
1
70
5
100
2
75
50
125
4
3
$\mathbf{v}^3 = (1,5,3,2,4,1)$
$f^*(\mathbf{v}^3) = 420$
$p^3 = 0.323$

Fig. 6.2 Illustrative example with 5 cities (candidate solutions)

Route of candidate 1: $\mathbf{v}^1 = (1,3,2,4,5,1)$, Route of candidate 2: $\mathbf{v}^2 = (1,5,4,2,3,1)$

$t = 1$: edge sampled from within $\mathbf{v}^2$ using roulette wheel selection: $(1,5)$

The cities are identified in $\mathbf{v}^1$: $(\boxed{1},3,2,\boxed{5},4,1)$

Update the route $\mathbf{v}^1$ by incorporating the edge $(1,5)$: $(1,\boxed{5},2,\boxed{3},4,1)$

Cost of route $f(\mathbf{v}^1)^1$: 430

$t = 2$: edge sampled from within $\mathbf{v}^2$ using roulette wheel selection: $(4,2)$

The cities are identified in $\mathbf{v}^1$: $(1,3,\boxed{5},4,\boxed{2},1)$

Update the route $\mathbf{v}^1$ by incorporating the edge $(4,2)$: $(1,3,2,\boxed{4},\boxed{5},1)$

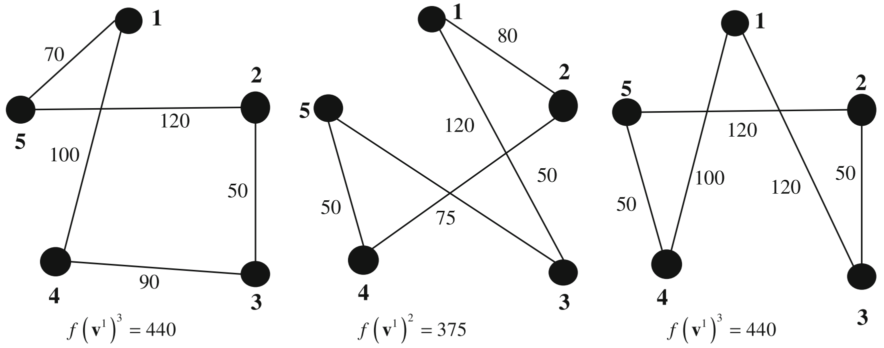
Cost of route $f(\mathbf{v}^1)^2$: 375

$t = 3$: edge sampled from within $\mathbf{v}^2$ using roulette wheel selection: $(5,4)$

The cities are found: $(1,3,2,\boxed{5},\boxed{4},1)$

No changes in the route by including edge $(5,4)$: $(1,3,2,5,4,1)$

Cost of route $f(\mathbf{v}^1)^3$: 440


1
2
5
4
3
70
120
100
50
90
$f(\mathbf{v}^1)^3 = 440$
1
2
5
4
3
80
120
50
50
75
$f(\mathbf{v}^1)^2 = 375$
1
2
5
4
3
120
100
120
50
50
$f(\mathbf{v}^1)^3 = 440$

$$\mathbf{F}^{1,3} = \left\{ f(\mathbf{v}^1)^1, f(\mathbf{v}^1)^2, f(\mathbf{v}^1)^3 \right\} = \{430, 375, 440\}$$

$$f^*(\mathbf{v}^1) = 375$$

Fig. 6.3 Illustrative example with 5 cities (variations obtained)

3.1(a) The probability p^c of each candidate $c, (c = 1, \ldots, 3)$ is calculated. The calculated probability values are presented in Fig. 6.2.

3.1(b) Using roulette wheel selection approach, assume that candidate 1 decides to follow candidate 2. An edge is selected randomly from within the route $\mathbf{v}^2$ and incorporated it into $\mathbf{v}^1$ forming a new route.

3.1(c) In such manner $t = 3$ new routes are formed. It is represented in Fig. 6.3 along with the associated route cost vector $\mathbf{F}^{1,3} = \left\{ f(\mathbf{v}^1)^1, f(\mathbf{v}^1)^2, f(\mathbf{v}^1)^3 \right\}$

and the selected best route with cost $f^{*}(\mathbf{v}^{1})$. It is presented in Fig. 6.3 In this way, candidates 2 and 3 also follow certain candidate's route and update their own routes. It makes the cohort available with 3 updated routes with costs $\mathbf{F}^{3} = \left\{f^{*}(\mathbf{v}^{1}), f^{*}(\mathbf{v}^{2}), f^{*}(\mathbf{v}^{3})\right\}$.

This process continues until saturation (convergence) i.e. every candidate finds the same route and does not change for successive considerable number of learning attempts.

In the context of the illustration provided in Figs. 6.1 and 6.2, in case of the best behavior selection approach every candidate in the cohort will follow candidate 2 as it has the minimum cost the current learning attempt. And the in case of the random behavior selection approach, every candidate selects a candidate randomly and follows its behavior. The results of the CI approach solving the TSP are discussed in the next section.

6.2 Results and Discussion

The CI algorithm discussed in Chap. 2 applied for solving the TSP was coded in MATLAB 7.7.0 (R2008b) and simulations were run on a Windows platform using i3-M380, 2.53 GHz processor speed with 3 GB RAM. The number of candidates C and variations t were chosen to be 5 and 5, respectively. In all, nine cases of the TSP [2] with number of cities varying from 14 to 29 were solved. Every case was solved 20 times. In the earlier version of CI [3], the CI candidates used roulette wheel approach for the selection of the behaviour in the cohort to follow. In addition to it, the CI algorithm here was successfully implemented with the best behavior selection and random behavior selection approach. The results are summarized in Table 6.1 with representative saturation history plots of the cohort with 5 candidates are presented in Fig. 6.4a–c.

It could be understood that, the cohort with roulette wheel selection approach saturated/converged in every run solving every case of the TSP to a marginally better solution than the best behaviour and random behaviour selection approach. In addition, even though the computational cost (time and function evaluations) of the CI with roulette wheel approach was more than CI with the best behaviour approach incorporated, the standard deviation (SD) exhibited comparatively more robustness (refer to Fig. 6.5b). The inherent probabilistic nature of the roulette wheel selection approach helped the algorithm explore a better solution from within the cohort. Also in some of the runs, similar to the feasibility-based rule developed in [9, 10], this approach necessarily helped the CI candidates jump out of local minima by following worse behaviour. It is important to mention here that the overall tendency to improve by competition and interaction ensured the saturation to an optimal solution. Furthermore, as presented in Table 6.1 and Fig. 6.5a, c the best behavior selection approach was found to be computationally (time and function evaluations) cheaper; however, the cohort stuck into local minima and did not yield a better

Table 6.1 Summary of CI performance solving the TSP

Problem name	Cities (N)	Reported optimum [2]	Behavior selection approaches	Cohort candidates' saturated/converged route costs	Closeness to reported optimum (%)	Average function evaluations (FE)	Time (sec)	Accepted route cost $f^*(\mathbf{v})$			SD
								Best	Mean	Worst	
Burma14	14	30.8785	Best	[31.2321, 31.2321, 31.2321, 31.2321, 31.2321]	1.1451	105,000	2.7	31.2321	33.2283	35.6267	2.225
			Roulette	[30.8785, 30.8785, 30.8785, 30.8785, 30.8785]	0.0000	1,575,000	30.88	30.8785	30.8785	30.8785	0.00
			Random	[31.3712, 31.3712, 31.3712, 31.3712, 31.3712]	1.5956	2,625,000	102.1	31.3712	31.9577	32.5442	0.830
P01	15	284.3809	Best	[363.295, 363.295, 363.295, 363.295, 363.295]	9.7578	120,000	3.8	363.295	352.9675	342.640	14.605
			Roulette	[284.381, 284.381, 284.381, 284.381, 284.381]	0.0000	1,320,000	41.67	284.381	284.381	284.381	0.00
			Random[a]	[284.381, 284.381, 360.830, 323.392, 323.392]	0.0000	3,000,000	69.36	284.381	284.381	284.381	0.00
Ulysses16	16	74.1087	Best	[84.0071, 84.0071, 84.0071, 84.0071, 84.0071]	13.3565	120,000	4.02	84.0071	84.2699	84.5327	0.372
			Roulette	[73.9876, 73.9876, 73.9876, 73.9876, 73.9876]	0.1634	2,400,000	85.50	73.9876	74.026	74.3079	0.099
			Random[a]	[83.4726, 79.5323, 76.9698, 80.4994, 77.2778]	3.8606	3,000,000	102.7	76.9698	77.7545	78.5392	1.110

(continued)

Table 6.1 (continued)

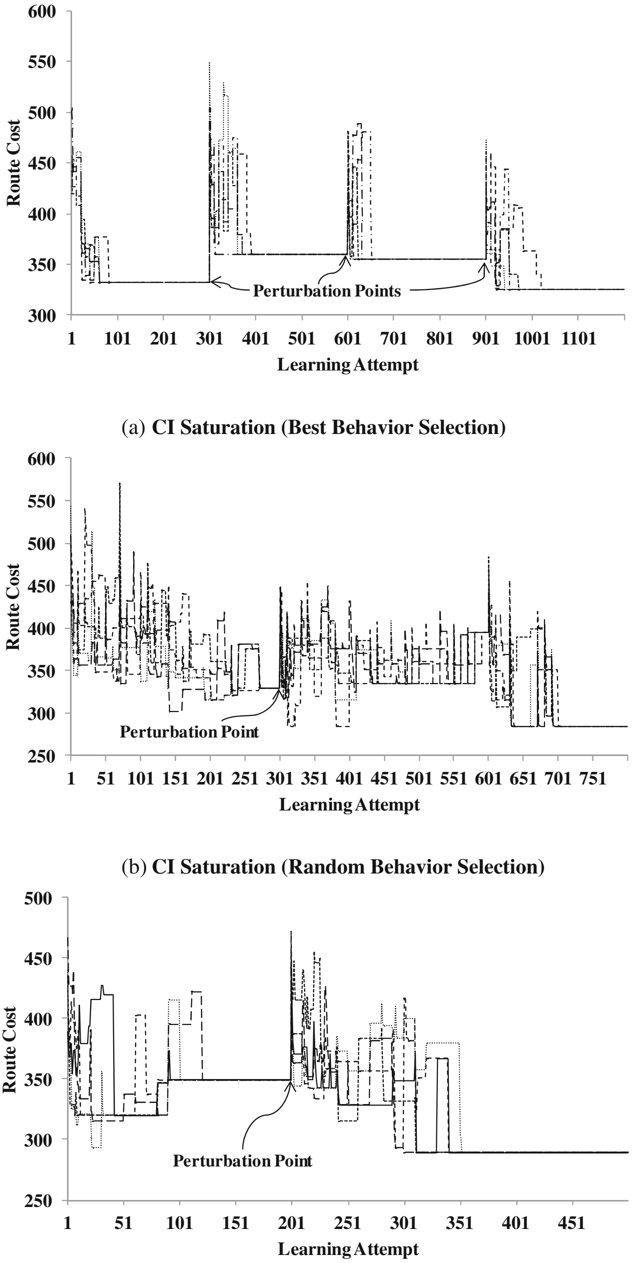
Problem name	Cities (N)	Reported optimum [2]	Behavior selection approaches	Cohort candidates' saturated/converged route costs	Closeness to reported optimum (%)	Average function evaluations (FE)	Time (sec)	Accepted route cost $f^*(\mathbf{v})$			SD
								Best	Mean	Worst	
Groetschel17	17	2085	Best	[2311, 2311, 2311, 2311, 2311]	10.8393	675,000	19.22	2311	2342	2373	43.84
			Roulette	[2085, 2085, 2085, 2085, 2085]	0.0000	2,025,000	70.74	2085	2086.4	2090	1.90
			Random	[2245, 2245, 2245, 2245, 2245]	7.6738	3,375,000	114.2	2245	2614.5	2984	522.6
Groetschel21	21	2707	Best	[3182, 3182, 3182, 3182, 3182]	17.5474	1,365,000	40.28	3182	3381	3580	281.4
			Roulette	[2707, 2707, 2707, 2707, 2707]	0.0000	2,100,000	72.16	2707	2707.6	2709	0.98
			Random	[2796, 2796, 2796, 2796, 2796]	32.8777	2,415,000	80.23	2796	2950.5	3105	218.5
Ulysses22	22	75.5975	Best	[96.1869, 96.1869, 96.1869, 96.1869, 96.1869]	27.2355	160,000	6.92	96.1869	97.895	99.6095	2.42
			Roulette	[75.5975, 75.5975, 75.5975, 75.5975, 75.5975]	0.0000	3,520,000	120.3	75.5975	76.607	77.8509	0.37
			Random[a]	[99.0993, 76.5557, 88.3879, 76.5557, 86.5087]	1.2675	3,520,000	121.4	76.5557	77.995	79.4341	2.04
Groetschel24	24	1272	Best	[1391, 1391, 1391, 1391, 1391]	9.3553	4,140,000	66.62	1391	1492	1467	53.74
			Roulette	[1272, 1272, 1272, 1272, 1272]	0.0000	4,320,000	70.42	1272	1282.6	1318	14.85
			Random	[1429, 1429, 1429, 1429, 1429, 1429]	12.3427	4,500,000	101.7	1429	1488.5	1548	84.14

(continued)

Table 6.1 (continued)

Problem name	Cities (N)	Reported optimum [2]	Behavior selection approaches	Cohort candidates' saturated/converged route costs	Closeness to reported optimum (%)	Average function evaluations (FE)	Time (sec)	Accepted route cost $f^*(\mathbf{v})$			SD
								Best	Mean	Worst	
Fri26	26	937	Best	[1156, 1156, 1156, 1156, 1156]	23.3724	5,062,500	154.3	1156	1180	1204	33.94
			Roulette	[937, 937, 937, 937, 937]	0.0000	5,062,500	154.3	937	941.6	959	8.38
			Random	[1085, 1085, 1085, 1085, 1085]	15.7951	5,062,500	154.3	1085	1210	1335	176.8
Bays29	29	9074	Best solution	[12665, 12665, 12665, 12665, 12665]	39.5746	350,000	13.65	12665	13013	13361	492.1
			Roulette	[9108.8, 9108.8, 9108.8, 9108.8, 9108.8]	0.3835	3,500,000	143.6	9108.8	9424.56	9654.3	192.26
			Random	[9122, 9122, 9122, 9122, 9122]	0.5290	5,250,000	200.2	9122	9341.5	9561	310.4

[a] Cohort failed to saturate

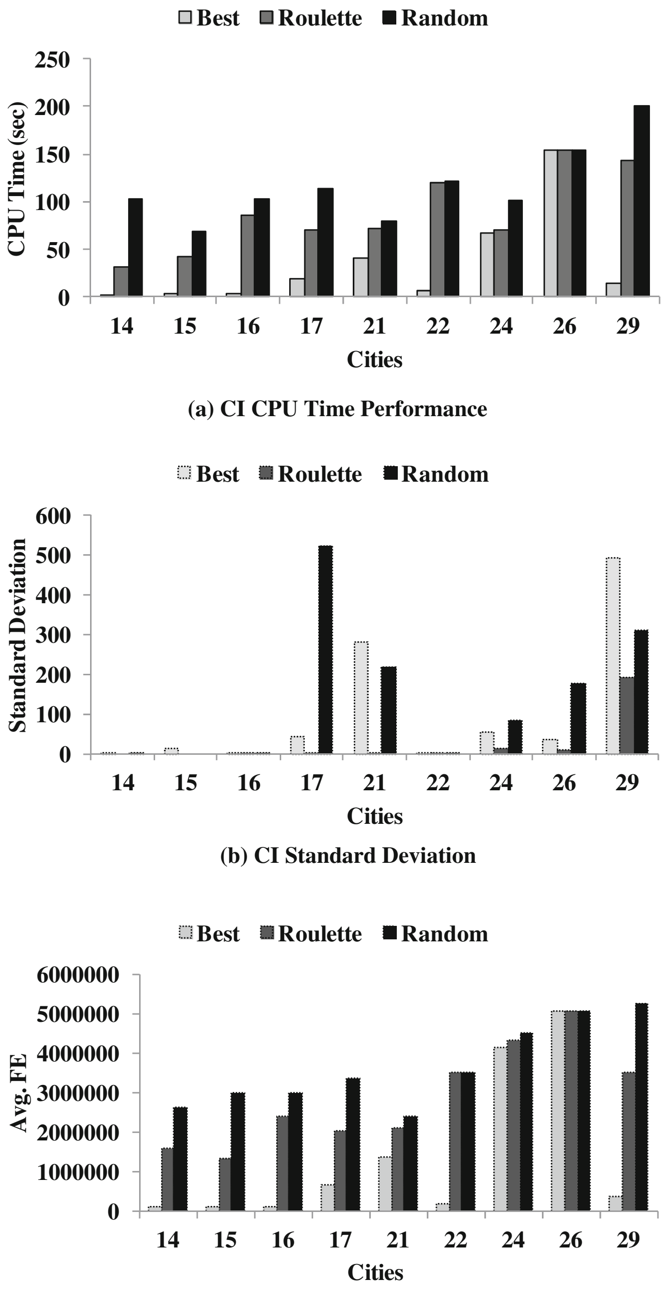


(a) **CI Saturation (Best Behavior Selection)**

(b) **CI Saturation (Random Behavior Selection)**

(c) **CI Saturation (Roulette Behavior Selection)**

Fig. 6.4 CI saturation history solving the TSP (P01, 5 candidates)



(a) CI CPU Time Performance

(b) CI Standard Deviation

(c) CI Average Function Evaluations

Fig. 6.5 Illustration of CI performance

solution. This happened because all the candidates in the cohort followed the best behaviour in every learning attempt, and the variation necessary for exploration and avoidance of the local minima was restricted. In the CI with random selection approach, following any candidate randomly forced the candidates explore the larger search space; however resulted into extremely slow saturation of the cohort behavior. Moreover as exhibited in Table 6.1, the saturation was not achieved in every run of the algorithm. The SD presented in Table 6.1 and Fig. 6.5a–c indicated that for solving all the cases the CI with roulette wheel selection approach was comparatively more robust. However, as the problem size increased, robustness of all the approaches was reduced. It was also observed that the computational cost of all the approaches increased with the increase in problem size.

Once the cohort behavior was saturated every candidate perturbed its individual solution. More specifically, the routes of all the candidates were randomly altered by changing the order of a certain number of cities in the vector $\mathbf{v}^c = \left(v_O^c, \ldots v_i^c, \ldots v_n^c, v_O^c\right)$ for every candidate c, $(c = 1, \ldots, N)$. It is important to mention here that the perturbation approach in [9, 10] required several parameters to be tuned which was completely avoided here. As evident in Fig. 6.4a–c, this approach was found to be helping the individual candidate's solution jump out of local minima and further saturates the cohort behavior to a significantly improved solution.

6.3 Concluding Remarks and Future Directions

For the first time emerging CI algorithm has been applied for solving combinatorial NP-hard problem such as the TSP with number of cities varying from 14 to 29. The rational and self supervising learning nature of the cohort candidates was successfully formulated and demonstrated along with the learning and improving qualities which further improved their individual behavior. The application of the CI methodology for solving combinatorial NP-hard problem such as the TSP is successfully demonstrated. The CI incorporated with the roulette wheel approach, best behavior selection as well as random behavior selection approaches was successfully presented. The results highlighted the overall simplicity of the algorithm as well as robustness of the solution with the roulette wheel approach. It also underscored that the CI incorporated with the roulette wheel selection approach more realistically resembles the competitive and interactive learning behavior of the cohort candidates, which eventually drove the cohort to marginally improved solution. Moreover, it is also demonstrated that always following the best behavior/solution may make the cohort to saturate faster; however may make the cohort stuck into local minima. In addition, in order to jump out of possible local minima and further make the cohort saturate to global minimum, a generic approach such as accepting worst behaviors was incorporated. The encouraging results may help solve the real world problems with increasing complexity as the TSP can be further generalized to a wide variety of routing and scheduling problems [4]. In addition, CI approach could be modified to make it solve Multiple TSP (MTSP) and Vehicle Routing Problem (VRP).

In addition to the advantages few limitations are also observed. A generic fine parameter tuning approach needs to be developed for selection of the parameters such as number of candidates *C* and number of variations *t*. In this chapter we have solved problems of sizes up to 29 cities. As the problem size was increased, the solutions obtained were less robust and the global minimum was found only intermittently. A possible solution to this problem is to develop a distributed CI approach wherein larger sized problems could be decomposed into smaller size problems and solve them independently. In this context, author see potential real world applications related to the distributed communication system such as, path planning of Unmanned Aerial vehicles (UAV) and addressing the ever growing traffic control problem using Vehicular ad hoc network (VANET).

References

1. Kulkarni, A.J., Shabir, H.: Solving 0–1 knapsack problem using cohort intelligence algorithm. Int. J. Mach. Learn. Cybern. (2014). doi 10.1007/s13042-014-0272-y
2. Reinelt, G.: TSPLIB—a travelling salesman problem library. ORSA J. Comput. **3**, 376–384 (1991)
3. Kulkarni, A.J., Durugkar, I.P., Kumar, M.: Cohort intelligence: a self-supervised learning behavior. In: IEEE International Conference on Systems, Man, and Cybernetics, Manchester, UK, pp. 1396–1400 (2013)
4. Somhom, S., Modares, A., Enkawa, T.: Competition-based neural network for the multiple travelling salesman problem with minmax objective. Comput. Oper. Res. **26**(4), 395–407 (1999)
5. Calvo, R.W., Cordone, R.: A heuristic approach to the overnight security service problem. Comput. Oper. Res. **30**(9), 1269–1287 (2003)
6. Christofides, N., Eilon, S.: An algorithm for the vehicle dispatching problem. Oper. Res. Q. **20**(3), 309–318 (1969)
7. Jati, G.K., Manurung, H.M., Suyanto, S.: Discrete cuckoo search for travelling salesman problem. In: IEEE International Conference on Computing and Convergence Technology, Seoul, Korea, pp. 993–997 (2012)
8. Luciana, B., Frana, P.M., Pablo, M.: A new memetic algorithm for the asymmetric travelling salesman problem. J. Heuristics **10**(5), 483–506 (2004)
9. Kulkarni, A.J., Tai, K.: A probability collectives approach with a feasibility-based rule for constrained optimization. Appl. Comput. Intell. Soft Comput. (2011), Article ID 980216
10. Kulkarni, A.J., Tai, K.: A probability collectives approach for multi-agent distributed and cooperative optimization with tolerance for agent failure. In: Czarnowski, I., Jedrzejowicz, P., Kacprzyk, J. (eds.) Agent Based Optimization, Studies in Computational Intelligence, vol. 456, pp. 175–201 Springer, Berlin-Heidelberg (2013)