Chapter 9
Solution to the Selection of Cross-Border Shippers (SCBS) Problem

Nomenclature

T	Number of periods in the planning horizon
I	Number of containers
J	Number of shippers
K	Number of different types of goods
i	Containers, $i = 1, 2, \ldots, I$
j	Shippers, $j = 1, 2, \ldots, J$
k	Type of goods, $k = 1, 2, \ldots, K$
Δ_k	Set of containers of type k
α_{jk}	A binary parameter = 1, if shipper j can handle the containers of type k = 0, otherwise
a_j	Fixed cost of choosing shipper j
b_{ij}	Variable cost of shipping container i through shipper j. If a shipper cannot handle the kind of goods in a container, then the variable cost is set to a high value
c_j	Maximum capacity of shipper j
e_i	Volume of container i
p_{ij}	Expected time of processing container i through shipper j
D_i	Due date of container i
F	Fund available
w_1	Weight assigned to the goal of fund constraint
w_{2i}	Weight assigned to the goal of due date for container i
w_{31}	Penalty for exceeding the limit of non-compliant shippers
w_{32}	Reward for using fewer than allowable number of non-compliant shippers
θ_j	A binary parameter = 1, if shipper j complies with cross-border regulations = 0, otherwise
θ^0	Maximum allowable number of non-compliant shippers
$\hat{c}_{jt}$	Maximum capacity of shipper j in period t
$\hat{F}_t$	Fund available in period t
t	Index for periods, $t = 1, 2, \ldots, T$
N_v	Number of decision variables in the problem

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_9

N_c	Number of constraints in the problem
N_{in}	Number of tested instances
d_1^- and d_1^+	Deviational variables associated with the fund constraint in the one-period setting
d_{1t}^- and d_{1t}^+	Deviational variables associated with the fund constraint in period t
d_{2i}^- and d_{2i}^+	Deviational variables associated with every container i, $i = 1, 2, \ldots, I$
d_3^- and d_3^+	Deviational variables associated with the selection of the non-compliant shippers θ^0

In this chapter, we demonstrate the ability of Cohort Intelligence (CI) methodology to solve problem of optimal selection of cross-border shippers and cargo assignments [1]. The problem includes various constraints related to due dates, processing times, fund availability, and shippers' compliance. We formulate and solve the multi-period instance of this problem as well. The performance of the CI method is compared to that of Integer Programming (IP) solution obtained using CPLEX and to specifically developed multi-random-start local search (MRSLS) method.

9.1 Selection of Cross-Border Shippers (SCBS) Problem

Cross-border shippers are major players in international trade and transportation [1, 2]. With the ever-changing standards of international compliance, international shippers of imported and exported goods must comply with an increasing number of regulatory constraints. The selection of shippers with cross-border compliance/non-compliance emerged as an important problem after the North American Free Trade Agreement (NAFTA) [3] became functional in 1994 which considerably increased cargo traffic between Canada, the United States and Mexico. Selecting compliant cross-border shippers helps avoid frustrating shipment delays at border check points and also results in transportation cost savings. In this section, we examine the problem of a company that must meet a number of goals by selecting shippers for the purpose of transporting containerized cargo across borders. The company must rely on shippers that can be either compliant or non-compliant. On the one hand, a compliant shipper is more costly to use; however, it allows shorter delivery times as it can facilitate the smooth transit of cargo through the border. On the other hand, a non-compliant shipper, while cheaper to use, may take longer delivery times of the cargo to the customer's destination as it may experience inspection slowdowns at the border. The elements involved in selection of a cross-border shipper problem include the total cargo volume to be transported to customers, the total funds available over the planning period, the ability of shippers

to handle special types of goods, anticipated delivery due dates, processing times for the particular good through the shipper, the type of shipper to use, etc. The mathematical formulations of the single- and multi-period problems are discussed below.

9.1.1 Single Period Model

Defining the decision variables as

$$x_{ij} = \begin{cases} 1 & \text{if container } i \text{ is shipped through shipper } j \\ 0 & \text{otherwise} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{if shipper j is chosen} \\ 0 & \text{otherwise} \end{cases}$$

leads to the following formulation:

$$Min\, w_1 d_1^+ + \sum\nolimits_i w_{2i} d_{2i}^+ + w_{31} d_3^+ - w_{32} d_3^- \tag{9.1}$$

$$\sum\nolimits_i e_i x_{ij} \le c_j \quad \forall\, j \tag{9.2}$$

$$x_{ij} \le y_j \quad \forall\, i,j \tag{9.3}$$

$$\sum\nolimits_j a_j y_j + \sum\nolimits_i \sum\nolimits_j b_{ij} x_{ij} + d_1^- - d_1^+ = F \tag{9.4}$$

$$\sum\nolimits_j p_{ij} x_{ij} + d_{2i}^- - d_{2i}^+ = D_i \quad \forall\, i \tag{9.5}$$

$$\sum\nolimits_j \left(1 - \theta_j\right) y_j + d_3^- - d_3^+ = \theta^0 \tag{9.6}$$

$$x_{ij} = 0 \,\forall i \in \Delta_k \quad \text{and} \quad \alpha_{jk} = 0 \tag{9.7}$$

$$\sum\nolimits_j x_{ij} = 1 \quad \forall\, i \tag{9.8}$$

$$x_{ij} \in \{0,1\},\ y_j \in \{0,1\} \quad \forall\, i,j,t \tag{9.9}$$

$$d_1^-, d_1^+ \ge 0, \quad d_{2i}^-, d_{2i}^+ \in \{0,1\}, \quad d_3^-, d_3^+ \ge 0 \tag{9.10}$$

The objective function in Eq. 9.1 represents the deviational variables to be optimized associated with the goal constraints given in Eqs. 9.4–9.6. Constraints in Eq. 9.2 represent the 'volume capacity' constraints which ensure that the total volume of containers assigned to the particular shipper does not exceed its maximum capacity. Constraints in Eq. 9.3 forces $y_j = 1$ when a shipper is selected. Constraint in Eq. 9.4 represents the 'fund availability' goal constraint which

ensures that the total expenditure should not exceed the available fund. The first term in Eq. 9.4 represents the fixed costs associated with the selected shippers and the second term represents the variable costs for shipping the containers through a particular shipper. The 'due date delivery' goal constraints in Eq. 9.5 ensure that every container should be delivered to the customer on or before the stipulated delivery date. Constraint in Eq. 9.6 ensures that number of shippers selected should not exceed the maximum allowable non-compliant shippers. Constraints in Eq. 9.7 ensure that a container is not shipped through a shipper that cannot handle the type of goods in the container. Constraints in Eq. 9.8 ensure that each container is shipped through exactly one shipper.

9.1.2 Multi Period Model

We define the following (binary) decision variables:

$$\hat{x}_{ijt} = \begin{cases} 1 & \text{if container } i \text{ is shipped through shipper } j \text{ in period } t \\ 0 & \text{otherwise} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{if shipper j is chosen} \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{y}_{jt} = \begin{cases} 1 & \text{if shipper j is chosen in period } t \\ 0 & \text{otherwise} \end{cases}$$

The integer linear programming is

$$Min \sum_t w_1 \hat{d}_{1t}^+ + \sum_i w_{2i} d_{2i}^+ + w_{31} d_3^+ - w_{32} d_3^- \tag{9.11}$$

$$\sum_i e_i \hat{x}_{ijt} \le \hat{c}_{jt} \quad \forall j, t \tag{9.12}$$

$$\hat{x}_{ijt} \le \hat{y}_{jt} \quad \forall i, j, t \tag{9.13}$$

$$\hat{y}_{jt} \le y_j \quad \forall j, t \tag{9.14}$$

$$\sum_j a_j \hat{y}_{jt} + \sum_i \sum_j b_{ij} \hat{x}_{ijt} + \hat{d}_{1t}^- - \hat{d}_{1t}^+ = \hat{F}_t \quad \forall t \tag{9.15}$$

$$\sum_t \sum_j (p_{ij} + t) \hat{x}_{ijt} + d_{2i}^- - d_{2i}^+ = D_i \quad \forall i \tag{9.16}$$

$$\sum_j (1 - \theta_j) y_j + d_3^- - d_3^+ = \theta^0 \tag{9.17}$$

$$\hat{x}_{ijt} = 0 \quad \forall\, t, \quad i \in \Delta_k \quad \text{and} \quad \alpha_{jk} = 0 \tag{9.18}$$

$$\sum_{j} \sum_{t} \hat{x}_{ijt} = 1 \quad \forall\, i \tag{9.19}$$

$$\hat{x}_{ijt} \in \{0,1\}, \quad y_j \in \{0,1\}, \quad \hat{y}_{jt} \in \{0,1\} \quad \forall\, i,j,t \tag{9.20}$$

$$d_{1t}^{-}, d_{1t}^{+} \geq 0, \quad d_{2i}^{-}, d_{2i}^{+} \in \{0,1\}, \quad d_{3}^{-}, d_{3}^{+} \geq 0 \tag{9.21}$$

Equation 9.11 represents the deviational variables to be optimized. Constraint in Eq. 9.12 represents the volume capacity constraints. Constraint in Eq. 9.14 forces $y_j = 1$ whenever shipper j is selected. Constraint in Eq. 9.13 forces $\hat{y}_{jt} = 1$ when shipper j is selected in period t. Constraint in Eq. 9.15 represents the 'fund availability' goal constraint. The 'due date delivery' goal constraints in Eq. 9.12 ensure that every container should be delivered to the customer on or before the stipulated delivery date. Constraint in Eq. 9.17 ensures that number of shippers selected does not exceed the maximum allowable non-compliant shippers. Constraint in Eq. 9.14 ensures that a container is not shipped through a shipper that cannot handle the type of goods in the container. Constraint in Eq. 9.19 ensures that every container is shipped through exactly one shipper on a particular period.

9.2 Numerical Experiments and Results

The CI procedure is now applied to solve the Selection of Cross-border Shippers (SCBS) problem discussed in Sect. 9.1. The procedure is coded in MATLAB 7.7.0 (R2008B). In addition, the simulations are run on a Windows platform using an Intel Core2 Quad CPU, 2.6 GHz processor speed and 4 GB memory capacity. In total, 8 distinct cases presented in Table 9.1 are solved for the single-period version and 18 cases presented in Table 9.2 are solved for the multi-period version of the problem. For every case, 10 instances are generated and every instance is solved 10 times using the CI method. The associated CI parameters such as the number of candidates S and the number of variations Y are chosen to be 3 and 10, respectively.

For all the considered problem cases, the number of different types of goods is set equal to $K = 5$. The size of the set of containers Δ_k for every type of good $k = 1, 2, \ldots, K$ chosen for every problem is listed in Tables 9.1 and 9.2 The value of α_{jk} randomly chosen to be either 0 or 1 such that each good $kk = 1, 2, \ldots, K$ is handled by at least one shipper $j = 1, 2, \ldots, J$. Each shipper $j, j = 1, 2, \ldots, J$, is randomly chosen to be either compliant $(\theta_j = 1)$ or noncompliant $(\theta_j = 0)$. The maximum allowable number of non-compliant shippers θ^0 are considered to be equal to the number of non-compliant shippers.

Furthermore, the fixed costs a_j for compliant and non-compliant shippers are uniformly generated from within the intervals $[100, 150]$ and $[150, 250]$,

Table 9.1 Results for test problems (single period)

J, I	Δ_k	N_v	N_c	N_{in}	IP	CI performance			% gap MRSLS versus IP	% gap MRSLS versus CI
					CPU time (s)	Avg sol % gap	CPU time (s)	SD (CPU time)		
5, 41	15, 10, 8, 5, 3	625	621	10	0.4867	2.5042	2.8276	1.0843	5.4395	4.5194
6, 47	17, 12, 9, 6, 3	869	873	10	0.4695	2.9914	2.2563	2.3568	3.5952	1.9336
8, 64	22, 18, 8, 10, 6	652	586	10	0.8511	4.1836	2.7992	0.7857	5.2474	2.8775
4, 132	40, 32, 25, 15, 20	800	666	10	0.5818	5.7075	7.3635	3.9597	10.1657	5.0322
8, 91	40, 35, 25, 15, 17	922	829	10	1.2820	4.6123	6.6243	2.4293	11.0711	6.0294
3, 143	50, 40, 20, 15, 18	1442	1297	10	5.6789	6.1791	10.6124	5.0364	10.9465	5.8111
8, 900	350, 250, 150, 100, 50	5349	904	10	42.9057	5.9488	30.4846	10.0735	11.1418	4.7671
8, 965	400, 250, 130, 100, 85	9662	8695	10	55.9057	5.9132	37.1572	7.2841	16.7324	8.2755

Table 9.2 Results for test problems (multi period)

T, J, I	Δ_k	N_v	N_c	N_{in}	IP	CI Performance			% gap MRSLS versus IP	% gap MRSLS versus CI
					CPU time (s)	Avg sol % gap	CPU time (s)	SD (CPU time)		
3, 5, 41	15, 10, 8, 5, 3	625	621	10	0.2293	4.2318	8.9319	2.5987	17.1880	13.7941
3, 6, 47	17, 12, 9, 6, 3	869	873	10	0.1778	2.6832	8.0473	3.5242	85.6929	82.4845
4, 3, 64	22, 18, 8, 10, 6	840	840	10	0.1107	2.1852	6.7515	2.5531	7.7782	5.5640
2, 4, 132	40, 32, 25, 15, 20	1181	1178	10	0.2932	5.0093	20.3730	6.3161	12.6099	8.9235
3, 3, 91	40, 35, 25, 15, 17	825	822	10	0.1560	2.5276	15.3088	2.9411	19.9835	19.1615
8, 3, 143	50, 40, 20, 15, 18	1032	1030	10	0.2230	2.1348	17.0945	3.4270	44.8077	42.6552
3, 4, 900	350, 250, 150, 100, 50	11104	11104	10	0.8642	5.6406	36.2087	7.8925	19.3918	14.4520
4, 8, 965	400, 250, 130, 100, 85	9662	8695	10	5.2541	3.8738	74.6240	14.1965	11.1537	7.2439
4, 25, 1000	300, 250, 200, 150, 100	27029	26026	10	40.232	6.7197	79.1075	18.0798	29.4734	23.8698
8, 3, 2871	900, 700, 600, 500, 171	66730	66734	10	9.9120	4.7621	73.0579	5.7850	53.4205	47.9887
8, 3, 3876	1400, 1000, 800, 500, 176	87946	87952	10	18.044	11.4148	108.0878	35.0878	52.4043	37.0317
5, 37, 1954	800, 500, 300, 200, 154	316461	316556	10	210.10	5.9746	113.6585	40.0708	102.0578	91.7036
9, 47, 1521	600, 400, 300, 121, 100	542795	543019	10	376.31	10.1997	93.0901	22.9238	55.1360	48.6519

(continued)

Table 9.2 (continued)

T, J, I	Δ_k	N_v	N_c	N_{in}	IP	CI Performance			% gap MRSLS versus IP	% gap MRSLS versus CI
					CPU time (s)	Avg sol % gap	CPU time (s)	SD (CPU time)		
8, 15, 6576	3000, 2000, 800, 500, 276	883635	883705	10	528.07	11.5074	141.5355	10.6661	174.4285	139.5145
8, 15, 5286	2000, 1000, 986, 800, 500	567372	567434	10	352.62	9.8089	130.3254	11.6471	129.1648	103.3162
8, 13, 5479	300, 250, 200, 150, 100	493533	493586	10	399.50	10.8206	133.1693	23.9897	134.3845	112.7871
8, 8, 4954	1400, 1000, 800, 500, 176	276901	276923	10	128.39	11.5375	71.7315	16.1202	153.8094	128.5710
8, 26, 3249	900, 800, 700, 600, 249	581894	582007	10	410.17	10.2553	133.7576	10.8573	157.9798	135.3840

respectively. The variable costs b_{ij} of shipping container i through shipper j for compliant and non-compliant shippers are uniformly generated from within the interval $[20, 50]$ and $[50, 80]$, respectively. Similarly, The funds available F and $\hat{F}_t$ are uniformly generated from within the interval $\left[max\left(b_{ij}\right) + \frac{I}{4}, max\left(b_{ij}\right) + \frac{I}{2}\right], i = 1, 2, \ldots, I$ and $j = 1, 2, \ldots, J$. The maximum capacities c_j and $\hat{c}_{jt}$ and the volumes e_i, $i = 1, 2, \ldots, I$, are uniformly generated from within the interval $[200, 900]$ and $[10, 25]$, respectively. In addition, the expected processing times p_{ij} of container i through shipper j for both compliant and non-compliant shippers are uniformly selected from within the interval $[T/2, T]$ and $[1, T]$, respectively. Finally, the due dates $D_i, i = 1, 2, \ldots, I$, are randomly generated from within $\left[min\left(p_{ij}\right), \left(max\left(p_{ij}\right) + 1\right)\right], i = 1, 2, \ldots, I$ and $j = 1, 2, \ldots, J$. Note that all the goals are considered equally important and are assigned weights equal to 1.

The average CI solution for every case is compared with the associated Integer Programming (IP) solution obtained by using CPLEX. The IP could solve the single period problem up to number of shippers $j = 8$ and number of containers $I = 965$, i.e. with 9662 variables and 8695 constraints. The performance comparison of the IP and CI solution is presented in Tables 9.1 and 9.2 along with the graphical illustration in Figs. 9.1 and 9.2.

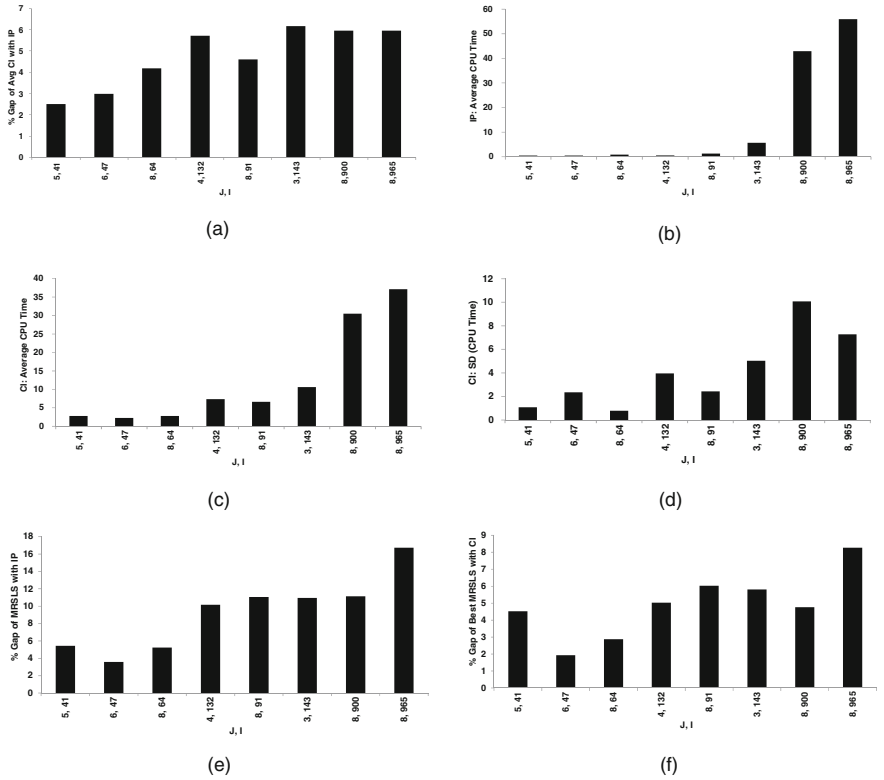

Fig. 9.1 Illustration of the CI, IP and MRSLS solution comparison (single period)

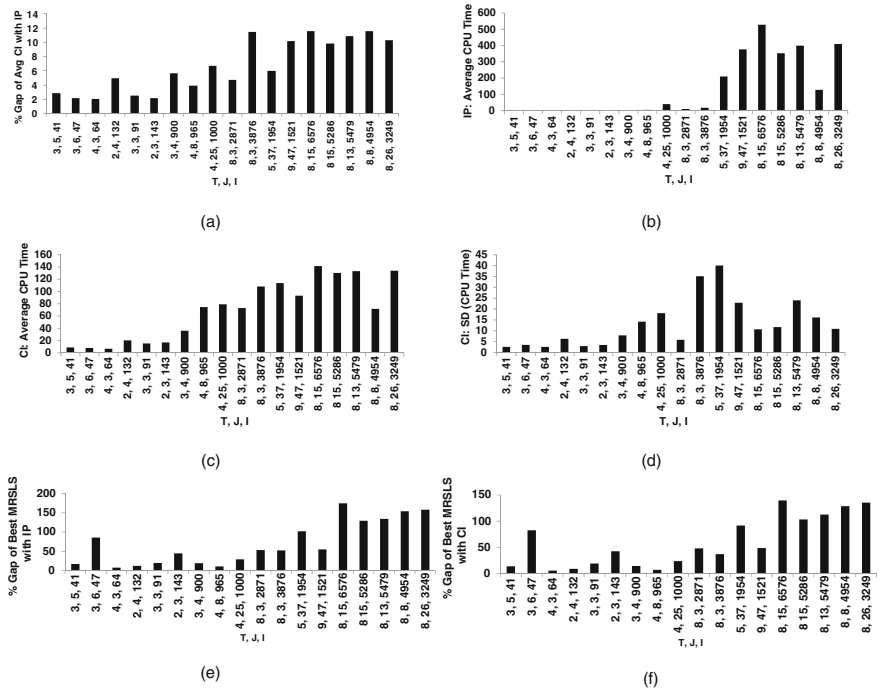

Fig. 9.2 Illustration of the CI and IP and MRSLS solution comparison (multi period)

It is evident from the results in Tables 9.1 and 9.2 and plot presented in Figs. 9.1a and 9.2a that for the smaller sized problems, the CI method could produce the solution comparatively closer, i.e. within 6 % of the reported IP solution. The difference gradually increased as the problem size grew; however, the maximum gap between the average CI solution and corresponding IP is noted to be within 12 % of the reported IP solution. Also, it is clear from Tables 9.1 and 9.2 and Figs. 9.1b, c and 9.2b, c that the CPU time for CI solving the problem with the smaller cases is more than the IP; however, the rate of increase is significantly lesser than that of IP. The increase in the time for CI is because the search space increased with problem size; however for every candidate the number of characteristics to be learnt in a learning attempt from the other candidate being followed did not change, which resulted in increased number of learning attempts and time to improve their individual behavior/solution and further reach the saturation/convergence. This is evident in Figs. 9.1d and 9.2d that the standard deviation (SD) of the CPU time for solving the problem increased with the increase in problem size.

In addition to the above, the performance of the CI method is also compared to a multi-random-start local search (MRSLS), which is carried out to find good solutions to both the single- and multi-period SCBS problem. The MRSLS implemented for this problem is similar in nature to the one used for finding solutions to the Sea Cargo Mix problem (see Chap. 8). Our MRSLS is again based on a

pair-wise interchange argument to generate a neighboring solution from the one currently being assessed. More specifically, an initial solution is first constructed. This solution specifies an assignment of containers to shippers. To construct an alternative solution from the existing one, two shippers are randomly selected. Then, for each shipper, a subset of cargoes that are currently assigned to this shipper are randomly chosen. In the new solution, the selected cargoes are interchanged (or swapped) among the two designated shippers. This process is continued in every successive learning attempt until a stopping criteria is met.

For each of the case problems considered (Tables 9.1 and 9.2), the MRSLS for the single- and multi-period SCBS problems is run 50 times with different initializations. Also, for a meaningful comparison, every MRSLS case is initialized to start in the neighborhood of the CI's starting point and is run for exactly the same time equal to the corresponding average CPU time the CI method takes to solve that case. Similar to the (SCM) problem, the acceptance of the resulting solution in every learning attempt depends on the following feasibility-based rules [4]: (1) if the existing solution is infeasible and the resulting solution has improved constraint violation, then the solution is accepted, (2) If the existing solution is infeasible and the resulting solution is feasible, then the solution is accepted, (3) if the existing solution is feasible and the resulting solution is also feasible and the objective function has improved objective, then the solution is accepted. If any of these conditions are not satisfied then the existing solution is retained and the resulting solution is discarded.

It is important to mention here again that for many of the MRSLS runs carried out to completion for the single- and multi-period case problems, only few solutions are feasible and most of them do not satisfy the feasibility conditions. This is because for every MRSLS run a solution is randomly initialized which could be infeasible and MRSLS further might not have been able to discover a feasible solution. Therefore, only the best of the feasible solutions obtained using MRSLS are considered for comparison with the CI. From Tables 9.1 and 9.2 as well as from Figs. 9.1a, e, f and 9.2a, e, f it can be seen that the rate of increase of the percentage gap between the solution obtained using MRSLS and that obtained using CPLEX is significantly more when compared to the rate of percentage gap increase between CPLEX and CI. In addition, the percentage gap between the solution obtained using MRSLS and CPLEX per case is also considerably larger than the one achieved for CPLEX versus CI. In other words, CI has performed significantly better than MRSLS in finding good solutions to the SCBS problem.

9.3 Conclusions

The emerging optimization technique of cohort intelligence (CI) is successfully applied to solve a cross-border shippers' problem. The results indicate that the accuracy of solutions to these problems obtained using CI is fairly robust and the

computational time is quite reasonable. Furthermore, the usefulness of CI in satisfactorily solving goal programming problems is also demonstrated.

The guiding principles of CI as an optimization procedure are grounded in artificial intelligence (AI) concepts. CI models the self-supervising behavior of a group of people seeking approximately the same goal. The self-supervising nature and rational behavior of the candidates among the cohort is illustrated along with the learning process that takes place among the candidates in order to further improve their individual characteristics/qualities. Furthermore, the inherent ability of the CI algorithm in handling complicated constraints lends to its applicability in solving real world complex problems. In addition, it is evident from the results that the variability as measured by standard deviation (SD) in the quality of solutions obtained using CI is commendable and remains almost stable as the problem size increases. This is because, even though the search space increases as the problem size increases, the number of characteristics in a learning attempt that need to be learnt by a candidate who is following the behavior of another candidate do not change. This results in an increase in the number of learning attempts in order to improve candidates' individual solutions and to finally reach the cohort's global solution.

Some limitations of the CI method should also be identified. The rate of convergence and the quality of the solution is dependent on the parameters such as the number of candidates and the number of variations. These parameters are derived empirically over numerous experiments and their calibration require some preliminary trials. It should also be observed that the number of characteristics attempted to adopt/learn is an important parameter when dealing with combinatorial optimization problems. As fewer characteristics are considered during the learning stage, this may delay the method's convergence rate significantly. The procedure may get stuck in the neighborhood of a local minimum, which may result into premature convergence. How to fine-tune the CI parameters and what to decide on the number of characteristics that needs to be learned by a candidate in every learning attempt can be done in an evolutionary and adaptive way as discussed in [5]. This may also help in increasing the accuracy of the solution as well as reducing the SD and overall performance of the algorithm. In addition, it should also be observed that the initial guess of the candidate solutions can affect the computational time of the algorithm. More specifically, if the initial candidate solutions are closer to the feasible region the chances of achieving saturation/convergence and reaching the optimal solution faster are high.

The paper also describes the application of a multi-random-start local search (MRSLS) that can be used to solve these three problems. The MRSLS implemented here is based on the interchange argument, a valuable technique often used in sequencing, whereby the elements of two adjacent solutions are randomly interchanged in the process of searching for better solutions. Our findings are that the performance of the CI is clearly superior to that of the MRSLS for many of the problem instances that have been solved.

As mentioned before, in agreement with the no-free-lunch theorem [6], any algorithm may not be directly applicable to solve all the problem types unless it can

be enhanced by incorporating some useful techniques or heuristics. The CI method may also benefit from certain performance-enhancing techniques when it is applied to different classes of problems. A mechanism to solve multi-objective problems is currently being developed, which can prove helpful in transforming the model's constraints into objectives/criteria (see [4, 6] for new development in this area). This can help reduce the dependency on the quality of the candidates' initial guess.

References

1. Kulkarni, A.J., Baki, M.F., Chaouch, B.A.: Application of the cohort-intelligence optimization method to three selected combinatorial optimization problems. Eur J Oper Res. (2015)
2. Li, Z., Bookbinder, J.H., Elhedhi, S.: Optimal shipment for an airfreight forwarder: formulation and solution methods. Transp. Res. Part C **21**, 17–30 (2012)
3. Wong, W.H., Lawrence, C.L., Hui, Y.V.: Airfreight forwarder shipment planning: a mixed 0-1 model and managerial issues in the integration and consolidation of shipments. Eur. J. Oper. Res. **193**, 86–97 (2009)
4. Deb, K.: An efficient constraint handling method for genetic algorithms. Comput. Methods Appl. Mech. Eng. **186**(2–4), 311–338 (2000)
5. Eiben, A.E., Smit, S.K.: Evolutionary algorithm parameters and methods to tune them. In: Hamidi, Y., et al. (eds.) Autonomous Search, pp. 15–36. Springer, Berlin (2011)
6. Patankar, N.S., Kulkarni, A.J., Tai, K., Ghate, T.D., Parvate, A.R.: Multi-criteria probability collectives. Int. J. Bio-Inspired Comput. **6**(6), 369–383 (2014)