“Scientia Iranica”

An extended version of adaptive large neighborhood search for a relief commodities distribution network design under uncertainty

Ali Mahmoudi[1], Mohammad Shafiee[1], Davood Shishebori[1]*

[1]Department of Industrial Engineering; Yazd University; 89195-741 Yazd, Iran

E-mail: Shishebori@yazd.ac.ir (Corresponding author)

Supplementary file

Part (A). Simulated Annealing Algorithm(SA)

In the first step of algorithm, specific algorithm parameters like initial temperature, minimum and maximum iterations have to be initialized (amount of these parameters are shown in the appendix). Then, in first iteration, a feasible solution is created randomly (x_0) and its function value assesses ($f(x_0)$) as well. By established some operations on current solution, new solution (x_{new}) is generated and its function value assesses again ($f(x_{new})$). At this stage, two types of conditions will arise: (1) new solution is better than previous solution which in that situation new solution will be accepted and is replaced instead of previous solution ($x_0 = x_{new}$), (2) new solution will be accepted by a possibility. Under the second condition, a number is generated between [0,1] randomly, then according to a probability function, the new solution will be accepted or rejected. In this study, Boltzmann distribution function is used (Eq. (1)).

$$probability = e^{-\frac{f(x_{new})-f(x_0)}{kT}} \tag{1}$$

In Eq. (1), k refers to Boltzmann's constant [1]. And finally, by adjusting temperature next iteration will be started. The following chromosome structure (Fig. 1) is used to describe problem in SA algorithm [2]. Three operators are used to generate new solution in this algorithm, including (1) swapping, (2) insertion, and (3) reversion.

(i). **Swapping:** in this method, two locations on the chromosome are selected and swapped (Fig. 2).

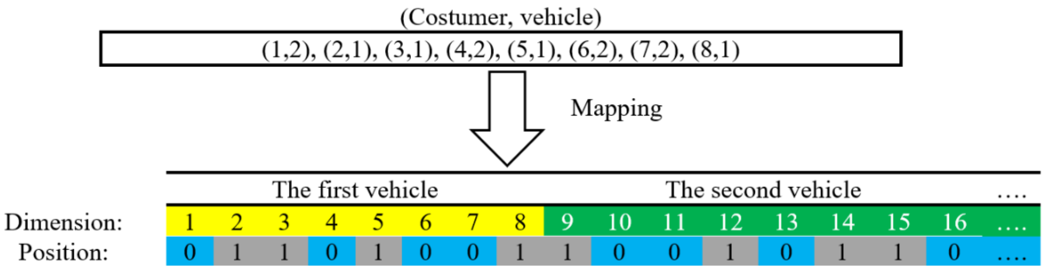


Figure 1. Chromosome structure used in the algorithm

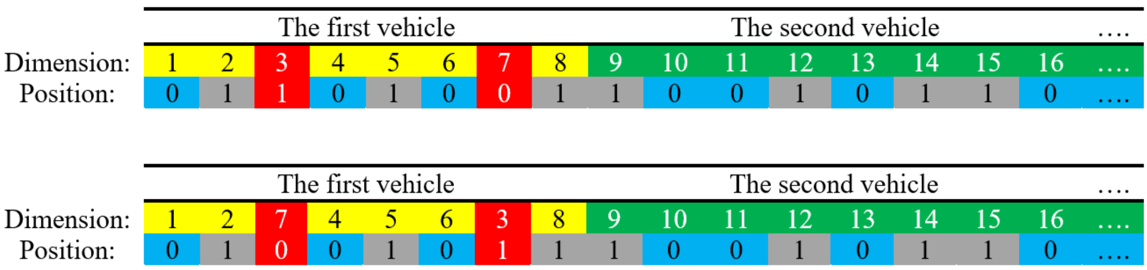


Fig. 2 Swapping method

(ii). **Insertion**: in this method, one node is selected randomly and then it is located on the best position on the chromosome (Fig. 3).

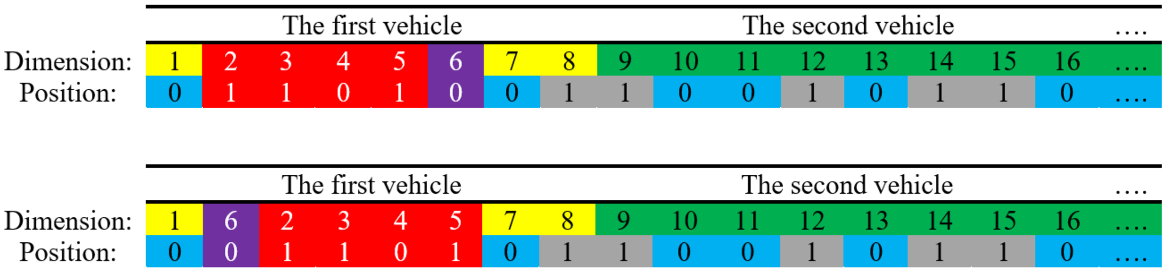


Fig. 3 Insertion method

(iii). **Reversion**: in this method, two locations on the chromosome are selected and the order of nodes are Reversed (Fig. 4).

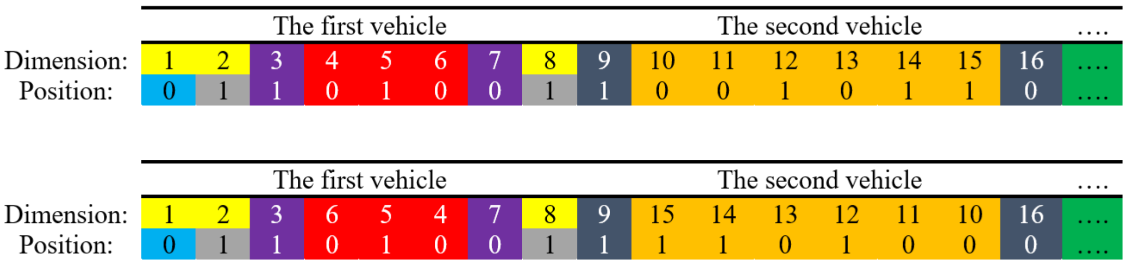


Fig. 4 Reversion method

Part (B). Genetic Algorithm (GA)

The following values (Table. 1) are used for the parameters of the algorithms ,and for more information about GA refer to Tasan and Gen, Hosseininasab and Shetab-Boushehri, and Kazemi, Mahdavi Mazdeh studies [3-5].

Table 1. Parameter settings for algorithms

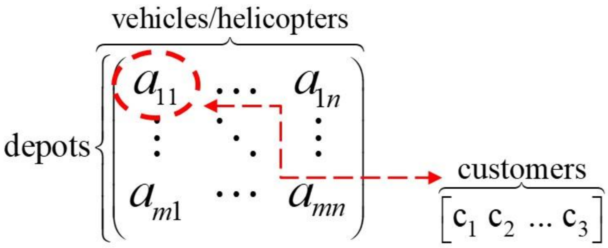
ALNS		**GA**						**SA**		
α_1	20	**Number of scenarios**	**Size of problems**	**Max_Iter**	**Npop**	**Pc**	**Pm**	**Initial temperature (T_0)**	**Max_Iter**	**Max_Sub_Iter**
α_2	10	1	Small	55	35	0.3	0.06	100	1000	10
α_3	2		Medium	75	45	0.45	0.2	100	2000	15
β	0.1		Large	95	55	0.6	0.2	100	3000	25
		2	Small	110	65	0.5	0.3	100	1000	10
			Medium	120	75	0.55	0.35	100	2000	15
			Large	170	85	0.8	0.35	100	3000	25
		3	Small	200	95	0.7	0.35	100	1000	10
			Medium	240	105	0.85	0.4	100	2000	15
			Large	280	115	0.9	0.45	100	3000	25

Part (C). Adaptive large neighborhood search (ALNS)

The steps of the ALNS algorithm are presented in the following seven sections.

1. Generate an initial feasible solution.

In this study, each generated solution is represented as a cell array. An example of generated solutions is shown below.



(C.1)

The columns represent the number of the vehicles in the depots, the rows represent the depots and a_{ij} stands for the customers receiving service from vehicle *j* from depot *i*.

2. Assign a weight to each removal (destruction) and insertion (repair) operator

In the first iteration of the algorithm, due to insufficient information about the performance of the operators, the weight of all the operators is considered the same. Therefore, the probability of selecting the *k* operator can be calculated from Eq. (C.2).

$$P_k = \frac{W_k}{\sum_{i=1}^{n} W_i} \quad \text{(C.2)}$$

For destruction operators, n = 8, and for repair operators, n = 3.

3. The roulette wheel mechanism is used to select the operators in each iteration of the algorithm.

4. Destruction of the current solution

5. Repairing of the destruction solution obtained from step 4

6. Check the condition for accepting a new answer

If the new solution is stronger than the current one, it is accepted and replaced the current solution. Otherwise, the new solution will be accepted with a predetermined probability. The Boltzmann's distribution function (Eq. C.3) is used to calculate the probability of accepting a new solution.

$$P = e^{\left(-\frac{f^{new} - f^{curent}}{T}\right)} \tag{C.3}$$

The basis of the ALNS algorithm is the SA. Therefore, in Eq. (C.3), T represents the initial temperature considered in the algorithm. In each repetition, T is reduced by a factor of $\delta \in (0,1)$

7. Update the weight and the probability of each operator

The weight and the probability of selecting each of the operators are updated using Eqs. (C.4) and (C.5). Points α_1, α_2, and α_3 are assigned to the operators according to their performance in each iteration.

$$\Pi_i^{r+1} = \Pi_i^r + \begin{cases} \alpha_1 & \text{If the new solution is better than the best solution} \\ \alpha_2 & \text{If the new solution is better than the current solution} \\ \alpha_3 & \text{If the new solution is not better than the current solution} \end{cases} \tag{C.4}$$

$$w_i^{r+1} = w_i^r \times (1-\beta) + \beta \times \Pi_i^r / N_i \tag{C.5}$$

Parameter β in the equation of weights controls the static ($\beta \in (0,1)$). When β is close to 0, it indicates that the operator weight is significantly dependent on its past performance, and, if it is close to 1, the weights are updated with the latest score they obtained in the previous iteration. Also, N_i is the number of times operator i is used up to r. Then, using Eq. (C.2), the probability of each operator is calculated.

8. Check the stop conditions

(D). Other findings

The result of last 20 runs for three metaheuristic algorithms is demonstrated in the figures 5-8.

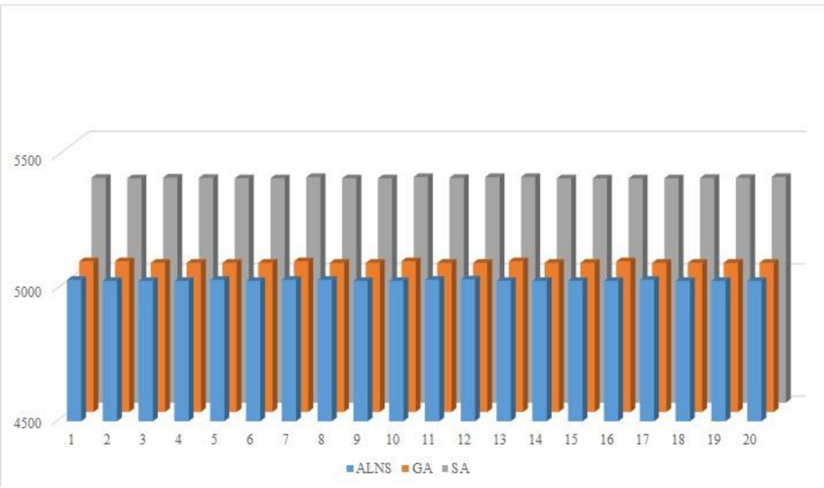


Fig. 5 Comparison of the performances of the suggested algorithms in the last 20 runs (Objective values- problem number:1)

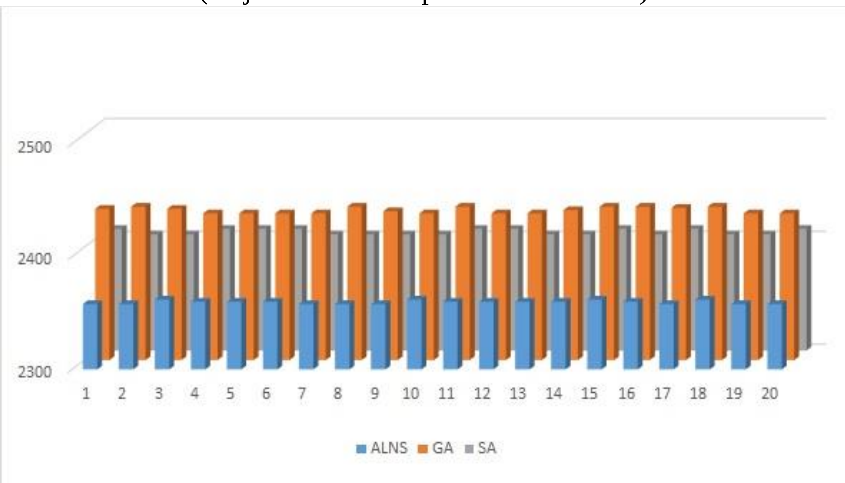


Fig. 6 Comparison of the performances of the suggested algorithms in the last 20 runs (Objective values- problem number:15)

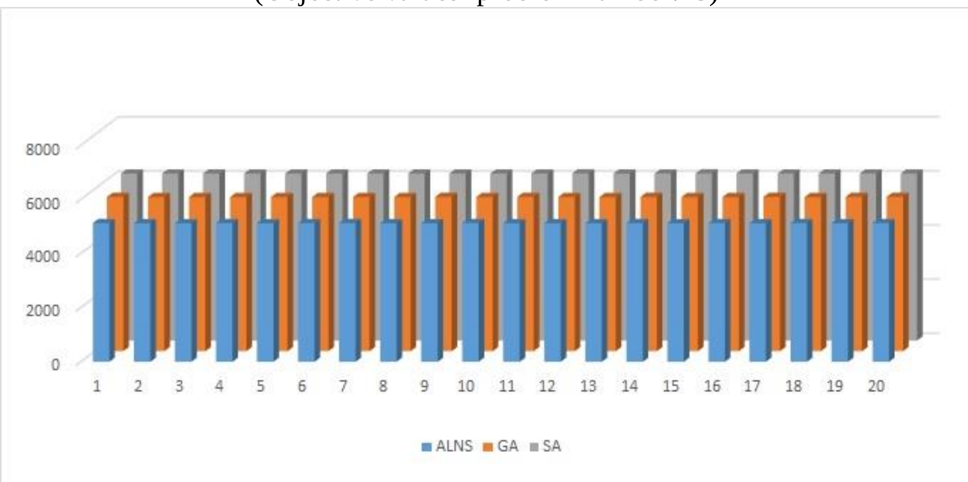


Fig. 7 Comparison of the performances of the suggested algorithms in the last 20 runs (Objective values- problem number:20)

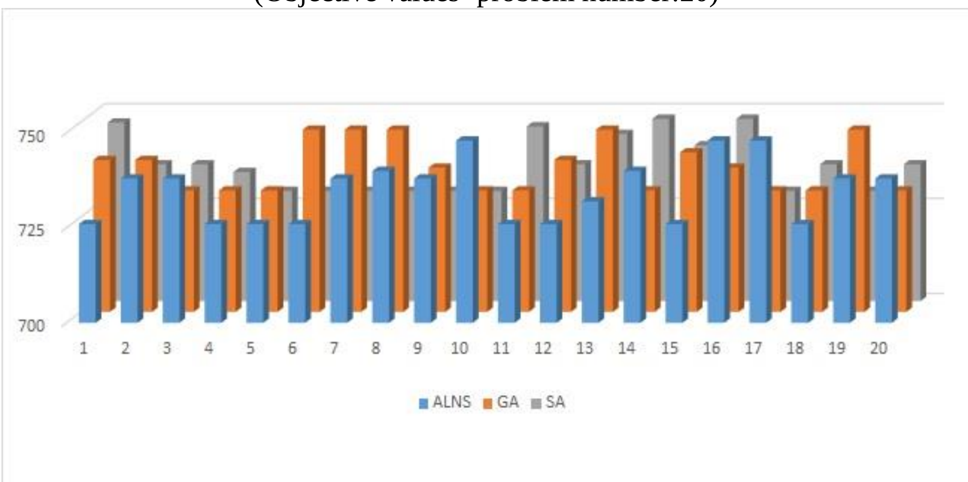


Fig. 8 Comparison of the performances of the suggested algorithms in the last 20 runs (Objective values- problem number:25)

1. Wei, L., et al., *A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints.* European Journal of Operational Research, 2018. **265**(3): p. 843-859.
2. Chen, A.-l., G.-k. Yang, and Z.-m. Wu, *Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem.* Journal of Zhejiang University-Science A, 2006. **7**(4): p. 607-614.
3. Tasan, A.S. and M. Gen, *A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries.* Computers & Industrial Engineering, 2012. **62**(3): p. 755-761.
4. Hosseininasab, S.-M. and S.-N. Shetab-Boushehri, *Integration of selecting and scheduling urban road construction projects as a time-dependent discrete network design problem.* European Journal of Operational Research, 2015. **246**(3): p. 762-771.
5. Kazemi, H., et al., *The integrated production-distribution scheduling in parallel machine environment by using improved genetic algorithms.* Journal of Industrial and Production Engineering, 2020: p. 1-14.