Elitist Non-dominated Sorting Genetic Algorithm-Based Heuristic for Optimizing Rail Freight Transportation

Vinay V. Panicker, C. S. Aryadutt and K. P. Anoop

Abstract Transportation services are crucial for any supply chain as these services deliver raw and intermediate materials to manufacturers and deliver finished goods to retailers and end customers. The present work focuses on the freight transportation adopted by an Indian food grain supply chain. The problem can be considered as a single-source multiple-destination distribution-allocation problem. Three penalty factors have been introduced to quantitatively represent the risk of incurrence of demurrage cost, to match time of supply and release of food grains and to maintain uniform capacity utilization throughout the network of depots. A multi-objective model is formulated with an objective to minimize these penalty factors. The model is solved using elitist non-dominated sorting genetic algorithm (NSGA II)-based heuristic. The solutions obtained prove the fast-converging nature of NSGA II algorithm.

Keywords Food grain supply chain · Multi-objective optimization
Non-dominated sorting genetic algorithm (NSGA II)

1 Introduction

Freight transport is crucial for the economic growth and development of a country. Transportation services are very much needed for any supply chain as the point of production and point of consumption rarely coincides. It is a fact that the cost of logistics is very high in India. The improvement in transportation sector and better

V. V. Panicker (✉) · C. S. Aryadutt · K. P. Anoop
Department of Mechanical Engineering, National Institute of Technology Calicut,
Campus P.O, Kozhikode, Kerala, India
e-mail: vinay@nitc.ac.in

C. S. Aryadutt
e-mail: aryadutt93@gmail.com

K. P. Anoop
e-mail: kpanoop05@gmail.com

H. Vasudevan et al. (eds.), *Proceedings of International Conference on Intelligent Manufacturing and Automation*, Lecture Notes in Mechanical Engineering,
https://doi.org/10.1007/978-981-13-2490-1_58

coordinated development of railways, roads and waterways are the strategies to be adopted to reduce the cost.

As rail transportation is more energy efficient than road, movement of freight via the rail could reduce logistics costs considerably. This work focuses on the freight transportation of food grains by an Indian food grain supply chain. The organization considered in this study owns depots at different states. Since these depots are horizontally collaborated, excess grains are stored at different locations. Since the food grains are moved from one source depot (consignor) to different destination storage depots (consignee), the problem can be considered as a single-source multiple-destination distribution-allocation problem.

Recently, few studies have been done in the food grain supply chain under consideration to optimize the freight transportation. A mathematical model for the determination of the optimal freight rate and the corresponding intermodal terminal location is proposed in [1]. A mathematical model for intra-state transportation of food grains incorporating flexibility in choosing economic mode of transport is developed in [2]. A mixed integer non-linear programming model to minimize the total cost which includes transportation, inventory and operational costs was developed and solved in [3].

Most of the works have considered single grain transportation problem. To overcome this drawback, a mixed integer linear programming model is developed in [4] to minimize the inventory holding cost and transportation cost for multiple food grains. Reference [5] proposed an integer non- linear programming and solved using exact method to formulate a monthly distribution-allocation plan for food grains by minimizing the total penalty value. In this work, a single-objective model priority was given for various penalty factors considered.

Further, a multi-objective model was developed for freight optimization of food grain supply chain and solved using multi-objective simulated annealing (MOSA)-based meta-heuristic in [6]. MOSA is applied to obtain a set of non-dominated plans for the distribution and allocation of food grains. The present work is an extension of this work by using NSGA-based meta-heuristic to solve the problem.

2 Problem Description

Food grains transported from a distant-source depot which is to be distributed and allocated to the set of destination storage depots under consideration. The distant sources are treated as a single source. Hence, the problem type considered can be called as a single-source multiple-destination distribution-allocation problem.

The food grain organization in this study has to distribute the allocated quantity from the source to the warehouses by meeting their demand. As per the provisions of Indian Railway, this organization can combine the demands of its two warehouses and order a full-train load. But Indian railway has put restrictions on the possible combinations of the warehouses based on the distance between them. So the food grains arrive at destination in full rake or half rake depending on warehouse demand.

It is the responsibility of this consignee organization to free the wagons within the allotted free time. For the extra time taken to unload the food grains, penalty (demurrage cost) have to be paid to Indian Railway by the consignee.

In order to quantify the problem, penalty-based approach is adopted in this work. The three penalty factors introduced are, namely rake penalty factor, weekly penalty factor and capacity utilization penalty factor.

- Rake penalty factor

Quantifies the relative risk in allotting a full rake to a depot over a half rake in terms of the incurrence of demurrage cost.

- Weekly penalty factor

Quantifies the priority of a particular week in a month over others. It is defined as the ratio of the outflow in a month to the outflow during a particular week in that month. So, the week with relatively more outflow will be served so as to minimize the weekly penalty factor.

- Capacity utilization penalty factor

The distribution of food grains across the storage depots is made uniform by incorporating capacity utilization penalty factor. It is the ratio of the storage capacity of a depot to the existing stock level there. By minimizing capacity utilization penalty factor, the depots with minimum capacity utilization can be given preference.

These penalty factors quantify the three objectives of the project. The project focuses on obtaining a monthly rake allocation plan for eight warehouses under consideration by minimizing the three penalty factors. The consignee organization has to allocate the food grains in such a way that it should meet demand at all warehouses and should not exceed the storage capacity of the warehouses. In a month, four weeks are considered, and in each week, only one allocation is possible. In some situations, the food grains shipped from source may exceed the demand; hence, the model should be flexible to accommodate that. By approaching the problem as multi-objective optimization problem, the expected result is Pareto-optimal solutions. So the organization can choose from a set of solutions depending on the priority of the objectives.

3 Solution Methodology

A method to solve multi-objective optimization problems and find multiple Pareto-optimal solutions is proposed in [7] named as non-dominated sorting algorithm (NSGA I). NSGA I algorithm is faster compared to other evolutionary algorithms for obtaining multiple Pareto–optimal solutions. Later, modified NSGA I, called as NSGA II, was proposed as in [8] which is computationally faster than NSGA I. NSGA II is based on a non-dominated sorting approach which uses an explicit mechanism to preserve diversity among solutions. NSGA II-based heuristic is proposed to minimize the supply chain cost and to improve responsiveness of supply chain in [9]. NSGA II algorithm is applied to multi-objective parallel machine scheduling problem in [10]. A NSGA II-based heuristic is developed to solve the problem under consideration.

3.1 Steps Involved in the Heuristic

The steps involved in NSGA II algorithm are given below:

Step 1: Initially, a random population P_0 of size N is created
Step 2: The population is sorted into different domination levels (F_i) and calculate the crowding distance for each solution in various fronts.
Step 3: Solutions are randomly grouped to N pairs in such a way no pair have same solutions. Crowded tournament selection operation is carried out in each pair to get the best solution.
Step 4: Crossover and mutation operations are carried out on resulting population to generate offspring population $Q_{0.}$
Step 5: Combine parent and offspring population to create R_t and perform non-dominated sorting and crowding distance calculation on R_t.
Step 6: Set new population $\mathrm{P}_{t+1} = \Phi$, and set a counter $i = 1$. Until (abs $(P_{t+1}) +$ abs $(F_i)) < N$, perform $P_{t+1} = P_{t+1} \cup F_i$ and $i = i + 1$.
Step 7: When (abs $(P_{t+1}) +$ abs $(F_i)) > N,$ fill the remaining slots based on crowding distance to obtain population $P_{1.}$
Step 8: Pair the population and perform crowded tournament selection, crossover and mutation operations on the obtained population to get population Q_1.
Step 9: Repeat steps 5, 6, 7 and 8 till the stopping criteria is met.

3.2 Solution Representation

The solution for the problem considered is represented by 8 × 4 matrix, where 8 rows denote the eight warehouses, and 4 columns denote the four allocations possible for the month. Hence, the solution is considered as a matrix as in [11] of dimension 8 × 4. Table 1 represents a sample solution. It is assumed that in a week, only one allocation is possible, and hence in a month, four allocations are possible. Allocations are done in terms of half rakes.

In Table 1, each cell in matrix can be filled by 0, 1 or 2, where 0 represents no allocation, 1 represents half-rake allocation and 2 represents full-rake allocation for a particular warehouse in a particular week. The total quantity transported from the source must be allocated meeting demand of a warehouse subject to storage capacity limitations. Half-rake allocation occurs as a combination.

Table 1 Solution representation

Warehouse	Week 1	Week 2	Week 3	Week 4
WH1	1	2	2	0
WH2	2	1	2	0
WH3	0	0	0	2
WH4	0	0	2	0
WH5	0	1	0	0
WH6	1	0	0	0
WH7	0	2	2	1
WH8	0	2	0	1

3.3 Operators Used

The NSGA II algorithm-based heuristic mainly uses the following six operators such as non-dominated sorting, crowding distance calculation, crowded tournament selection, crossover, mutation and divorce operators.

The above-stated operators are explained below:

Non-Dominated Sorting. This operator classifies the solutions in the population to various non-domination levels. The solutions in the population that are not dominated by any other solutions are put in front one, solutions that are dominated by one solution is put in front two, and so on. No solution is better than any other solution in that front. We calculate two entities to find non-dominated fronts:

(i) n_i, the number of solutions which dominate the solution i, and
(ii) S_i, a set of solutions which the solution i dominates.

Identify the solutions with n_i value equal to 0 which forms the first non-dominated front is put in list F_1. For each solution in the current front, visit each member (j) in its set S_i and reduce its n_j count by one. After this, again check for the solutions with n_i value zero and put this in list F_2. Then, continue this process using the newly identified front as our current front till all the solutions are classified into non-dominated fronts.

Crowding Distance. The crowding distance for a particular solution gives an estimate of the density of solutions surrounding it.

This quantity can be considered as the size of largest cuboid enclosing a point i without including any other point in the population. It is identified as the largest rectangle in case of two-objective optimization problem. Figure 1 shows the case of a two-objective optimization problem, where $f1$ and $f2$ represent two objectives, and j represents the point for which crowding distance is calculated. Arrange all the solutions in a front F in ascending or descending order for each objective. For the boundary solutions, the crowding distance value is assigned as infinity. For rest of the solutions, crowding distance is calculated using the following formula:

$$d_{i_j^m} = d_{i_j^m} + \frac{f_m^{j+1} - f_m^{j-1}}{f_m^{\mathrm{max}} - f_m^{\mathrm{min}}} \tag{1}$$

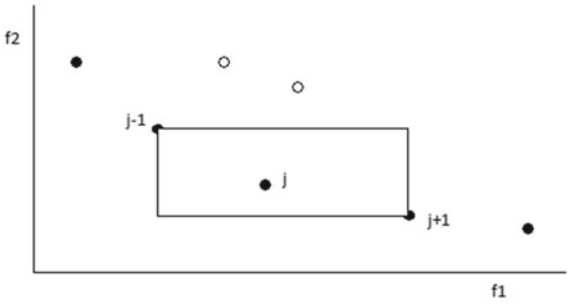

Fig. 1 Crowding distance representation for two-objective optimization problem

where $d_{i_j^m}$ represents the crowding distance for solution i, and j denotes the position of solution i in the sorted array for objective m; f_m^j represents the objective value of solution in position j in sorted array for objective m and $f_m^{\max}$ and $f_m^{\min}$ represents the maximum and minimum values of objective function value for objective m, respectively.

The distance for solution i is sum of the values calculated using the above formula for each objective.

Crowded Tournament Selection. The crowded comparison operator ($\angle_c$) helps in the selection process at various stages of the algorithm to obtain a uniformly spread out Pareto-optimal front. For every individual solution i in the population, the following two attributes are calculated—(1) Non-domination rank (i_{rank}) and (2) Local crowding distance (i_{distance}). A solution i wins a tournament with another solution j: If ($i_{\text{rank}} < j_{\text{rank}}$) or (($i_{\text{rank}} = j_{\text{rank}}$) and ($i_{\text{distance}} > j_{\text{distance}}$)).

CrossOver. The crossover operation is carried out based on crossover probability. Random numbers are generated for each solution. If the generated random number is within the crossover probability, then that solution enters the mating pool. The solutions are paired, and crossover operation is performed on each pair. Seven cases of crossover are performed on each pair of parents.

Case (1): The first column of parents is interchanged.
Case (2): The first two columns of parents are interchanged.
Case (3): The first three columns of parents are interchanged.
Case (4): The second column of parents is interchanged.
Case (5): The third column of parents is interchanged.
Case (6): The second and third columns of parents are interchanged.
Case (7): The second and fourth columns of parents are interchanged.

Each pair of parents results in seven pair of children after crossover which may be feasible or not. If feasible solutions are obtained in any case, then that case of crossover is chosen, and in case of no feasible solutions in any of seven cases, the case in which the solutions can be made feasible with less complexity is chosen.

Divorce Operator. When two parents are not able to give feasible solutions in any of the cases of crossover, the divorce operator removes this pair of parents from mating pool and selects two other parents randomly.

Mutation. The mutation operation is carried out to incorporate the genetic changes that may occur over generations. So mutation probability is taken as a very low value. Random numbers are generated for each solution, and based on mutation probability, the solutions to undergo mutation are selected. In the selected solution, one cell is selected randomly and change is made into the selected cell in such a way that the solution remains feasible even after the change is made.

4 Results and Conclusion

The proposed heuristic was coded using MATLAB R2015b in a computer with Intel core i5 3.2 GHz processor and 8 GB RAM.

An initial population size of 8 and total allocated quantity of 24 is considered. The monthly demand, initial stock level and storage capacity are shown in Table 2. Table 3 shows the weekly penalty factor matrix. The number of iterations is set at 1000. The value of objective functions of the resulting population after iterations is shown in Table 4.

The solutions in the resulting populations show that some of the solutions are repeated. This is because of the fast-converging nature of the NSGA II algorithm.

Table 2 Problem instance

Warehouse	Demand	Initial stock level	Storage capacity
WH1	3	6	13
WH2	3	17	25
WH3	2	3	6
WH4	2	3	6
WH5	1	12	16
WH6	1	3	5
WH7	3	32	40
WH8	3	2	6

Table 3 Weekly penalty factor

	Week 1	Week 2	Week 3	Week 4
WH1	1	1	10	2
WH2	4	2	2	1
WH3	2	2	3	1
WH4	2	2	3	1
WH5	4	4	3	1
WH6	2	3	4	1
WH7	4	4	3	1
WH8	4	7	3	1

Table 4 Objective function values of solution

Solution No.	Objective 1	Objective 2	Objective 3
1	530	90	37
2	500	86	42
3	510	85	47
4	530	90	37
5	510	85	47
6	540	90	42
7	510	85	47
8	500	86	42

The computational time taken to solve the heuristic is less than ten seconds. A set of non-dominated solutions would provide the manager in the organization with more choice and thus improve the flexibility in decision making. The work can be further extended by considering multiple allocations in a week by incorporating necessary changes for the weekly penalty factor for the further allocations.

References

1. Dandotiya, R, Nath Banerjee, R., Ghodrati, B. and Parida, A.: Optimal pricing and terminal location for a rail–truck intermodal service–a case study, International Journal of Logistics Research and Applications, 14(5), pp. 335–349, 2011.
2. Maiyar, L. M, Thakkar, J. J., Awasthi, A., and Tiwari, M. K.: Development of an effective cost minimization model for food grain shipments, IFAC-PapersOnLine, 48(3), pp. 881–886, 2015.
3. Mogale, D. G., Kumar, S. K. and Tiwari, M. K.: Two stage Indian food grain supply chain network transportation-allocation model, IFAC-PapersOnLine, 49(12), pp. 1767–1772, 2016.
4. Tanksale, A., and Jha, J. K.: An effective heuristic for multi-period multi-foodgrain inventory transportation problem in India, INFOR: Information Systems and Operational Research, 54(4), pp. 355–379, 2016.
5. Manu, N.: Optimization of the movement of food grains in food corporation of India, M.tech Thesis, National Institute of Calicut, 2016.
6. Jerin, S.: Optimization of freight transportation in a food grain supply chain using multi objective simulated annealing based heuristic, M.tech Thesis, National Institute of Calicut, 2017.
7. Srinivas, N. and Deb, K.: Multiobjective optimization using nondominated sorting in genetic algorithms, Evolutionary computation, 2(3), pp. 221–248, 1994.
8. Deb., K ., Agrawal, S., Pratap, A. and Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II,In International Conference on Parallel Problem Solving From Nature (pp. 849–858). Springer, Berlin, Heidelber, 2000, September.
9. Chan, F.T ., Jha, A. and Tiwari, M. K.: Bi-objective optimization of three echelon supply chain involving truck selection and loading using NSGA-II with heuristics algorithm, Applied Soft Computing, 38, pp. 978–987, 2016.
10. Bandyopadhyay, . S. and Bhattacharya, R.: Solving multi-objective parallel machine scheduling problem by a modified NSGA-II, Applied Mathematical Modelling, 37(10), pp. 6718–6729, 2013.
11. Jawahar, . N. and Balaji, A. N.: A genetic algorithm for the two-stage supply chain distribution problem associated with a fixed charge, European Journal of Operational Research, 194(2), pp. 496–537, 2009.