Classical MOEAs for solving a multi-objective problem of supply chain design and operation

ABSTRACT

This study focuses on a bi-objective mathematical programming formulation of the supply chain design and operation problem, which aims at simultaneously minimizing total costs and delays in order delivery. For small instances, a commercial solver (Gurobi Optimization) with different scalarizing techniques achieves a good representation of the real Pareto front. In the perspective of treating real size problems, NSGA-II and MOEA/D are applied to the same instances. Computational results highlight mitigated performances of both MOEAs and provide some insigts regarding future research paths for adapting MOEAs to such complex problems.

CCS CONCEPTS

• **Mathematics of computing → Combinatorial optimization**;
• **Applied computing → Supply chain management**;

KEYWORDS

Multi-objective supply chain optimization, MOEA/D, NSGA-II

ACM Reference Format:

. 2019. Classical MOEAs for solving a multi-objective problem of supply chain design and operation. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3321915

1 PROBLEM STATEMENT

In the last decades, the interest of production or logistics companies has been attracted by the study of supply chain (SC), to improve its profitability and operational efficiency. SC design and operation problems can be formulated as optimization problems that might involve multiple objectives. This study focuses on a bi-objective mathematical programming model introduced in a recent work [3] for a SC with three echelons (suppliers, factories, warehouses and customers), manufacturing several products (Figure 1). The first objective regards the total cost minimization, which includes investment (for plants and warehouses) and operation costs (production, storage and transportation). On the other hand, the second objective consists in minimizing the total delays in order delivery to customers, with respect to an established due date.

Accordingly, decision variables have to define the SC design and operation features, which implies determining how many factories

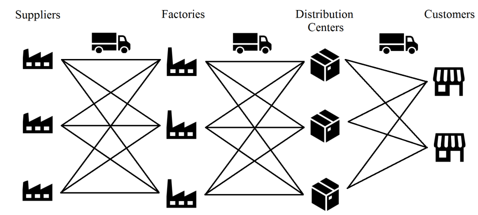


Figure 1: Illustration of a three echelon supply chain system

(in set J) and warehouses (in set K) to be opened, the quantities of products (set P) manufactured in each factory and the raw materials (set M) or product flows within the three echelons of the system, from suppliers (set I) to customers (set L). Constraints impose limited production capacities in the factories as well as storage capacities in both factories and warehouses. Furthermore, customer orders must be met (in terms of product quantity).

The above-mentioned formulation can be solved through a commercial solver, Gurobi, using different kinds of scalarizing techniques. But it is expected that, for real size examples, the problem complexity will not allow convergence in reasonable CPU times. Therefore, the objective of this study is to adapt classical Multi-Objective Evolutionary Algorithms (MOEAs) and to evaluate their performance for solving such problem.

2 MOEAS ADAPTATION

Since the problem is NP-hard, the implementations proposed in this work are rather canonical versions of MOEA/D [4] and NSGA-II [1]. MOEA/D uses a limited number of copies allowed in the replacement step, while parent selection is probabilistically performed either from the neighborhood of the target solution or within the whole population. An archive stores non-dominated solutions, using s-energy as a pruning criterion. Both algorithms use the same encoding technique, which must represent the whole SC structure and internal flows. The chromosome has to be divided into two sections, the first one regards factory/warehouse existence, while the second one encodes product or raw material flows within the three echelons. A simple binary encoding scheme is chosen for first section.

With respect to flow variables, the priority-based encoding mechanism proposed in [2] for multi-product transportation problems is adapted to the SC. This section has three parts, one for each echelon (Figure 2). Each section is a permutation where the position indicates a combination {node-product} (the node is either a source or destination) and the contained index is a priority for being scheduled. An edge entering or leaving the selected node is chosen according to an heuristic technique and the maximum possible quantity of product (among supply and demand) is assigned to the corresponding flow. Finally, the supply/demand of both the nodes involved are updated. Capacity constraints within

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3321915

Table 1: Computational results

	Instance 1		Instance 2		Instance 3		Instance 4		Instance 5	
Technique	$CPU-T$	HV_n	$CPU-T$	HV_n	$CPU-T$	HV_n	$CPU-T$	HV_n	$CPU-T$	HV_n
Gurobi Opt.	2880	1.0	2420	1.0	1640	1.0	980	1.0	1920	1.0
MOEA/D	907 (10.39)	0.80 (0.02)	903 (7.70)	0.86 (0.02)	897 (5.88)	0.90 (0.01)	901 (7.64)	0.93 (0.03)	904 (7.14)	0.64 (0.08)
NSGA-II	900 (7.77)	0.83 (0.01)	903 (5.45)	0.87 (0.01)	909 (5.0)	0.93 (0.01)	906 (8.34)	0.951 (0.01)	904 (8.34)	0.639 (0.08)

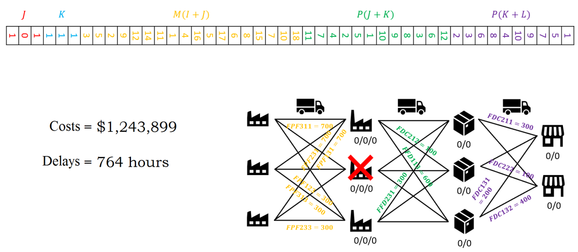
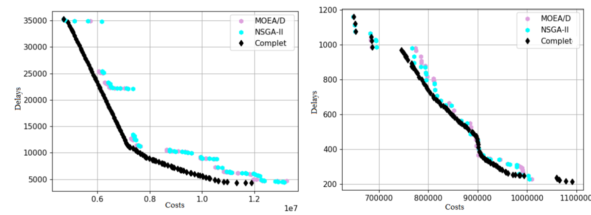


Figure 2: Encoding scheme



(a) Instance 3 **(b) Instance 4**

Figure 3: Complete fronts vs. MOEA/D vs. NSGA-II

factories/warehouses have to be respected in this step. This process is performed three times, in a backward fashion: customer demands allow to determine product units in warehouses, which determine the product flows from the factories and the manufactured quantities, etc. Note that this encoding scheme always preserves the feasibility of solutions. Regarding genetic operators, a two-point crossover is applied to the first (binary) section of the chromosome, while PMX is used for each of the three permutation separately.

3 EXPERIMENTATION AND RESULTS

Five small size instances ($|I| = |J| = |K| = |L| = |M| = |P| = 3$) were randomly generated and solved first with Gurobi with ϵ-constraints (using both total costs and delays as a constraint), and two scalarizing functions: Tchebycheff and Augmented Achievement (AASF). 51 solutions obtained with weight vectors {(0,1), (0.02, 0.98), ..., (1, 0)}, were generated for each instance and strategy. Gurobi converges for all instances in reasonable CPU times ($\leq$ 1 hour for 51 points, running on Intel Core i7-4770 3.40GHz, 16 Gb RAM). Then the obtained solutions were combined and filtered to build a reference set for evaluating MOEAs'performances. Both MOEAs were executed 11 times to study the median run. The population size and the generation number are set to 200 and 20,000 respectively, in order to have CPU times similar to those of Gurobi's shortest execution. The archive size is equal to 51. MOEA/D uses AASF where the neighborhood equals 10% of the population size, 2 copies are allowed for replacement and the probability of selecting parents within the population is 10%. Table 1 shows the median value and standard deviation of CPU time (s) and the normalized hypervolume (w.r.t. Gurobi's reference front).

The first observation is that both MOEAs show identical behaviors. Although no statistical test was performed, the hypervolumes obtained are quite similar and Figure 3 provides a clear illustration of how close are both approximated fronts. Besides, their performance is quite variable. Median hypervolumes range is from 64% to 95% of the "real" Pareto front. In some cases (instance 4), the solutions found reproduce quite closely the optimal front shape, while in others (instance 3), both MOEAs only identify some local fronts, showing poor convergence and diversity.

Discussion. The performances of MOEAs are not satisfactory enough on these small instances. Both MO search engines (dominance and decomposition) obtain similar results, so the solution encoding might be responsible for these trends. The heuristic rule used for edge selection was investigated (best cost or time, maximum regret, random) without resulting in significant improvements. However, an analysis of Gurobi's solutions identifies the reason of these convergence issues: the real Pareto front can be reached allowing smooth changes in some flow values, which are possible with the Mathematical Programming solver but not not with the chosen encoding scheme (always sends the maximum possible flow).

4 CONCLUSIONS

Bringing solution strategies to real-world problems is a main contribution of the multi-objective evolutionary optimization area. The preliminary experiments presented here provide some guidelines for future work on the SC design and operation problem. They highlight the need for developing adequate encoding schemes for complex industrial problems. Besides, hybridization between exact techniques and MOEAs looks promising, if well designed.

REFERENCES

[1] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation* 6, 2 (2002), 182–197.
[2] J. Lee, M. Gen, and K. Rhee. 2009. Network model and optimization of reverse logistics by hybrid genetic algorithm. *Computers & Industrial Engineering* (2009).
[3] Ángel D. Téllez Macías, Antonin Ponsich, Roman A. Mora Gutiérrez, and Eric A. Rincon García. 2018. Multi-objective optimization of the supply chain design and operation. In *Proceedings of the 6th International Conference Numerical and Evolutionary Optimization 2018.* Institut Politécnico Nacional, Mexico City, Mexico.
[4] Q. Zhang and H. Li. 2007. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation* (2007).