Report Econometric Institute 2012-24
October 2012

Evaluation of scalarization methods and NSGA-II/SPEA2 genetic algorithms for multi-objective optimization of green supply chain design

Corné van der Plas, Tommi Tervonen*, Rommert Dekker

Econometric Institute, Erasmus University Rotterdam, The Netherlands

Abstract

This paper considers supply chain design in green logistics. We formulate the choice of an environmentally conscious chain design as a multi-objective optimization (MOO) problem and approximate the Pareto front using the weighted sum and epsilon constraint scalarization methods as well as with two popular genetic algorithms, NSGA-II and SPEA2. We extend an existing case study of green supply chain design in the South Eastern Europe region by optimizing simultaneously costs, CO_2 and fine dust (also known as PM - Particulate Matters) emissions. The results show that in the considered case the scalarization methods outperform genetic algorithms in finding efficient solutions and that the CO_2 and PM emissions can be lowered by accepting a marginal increase of costs over their global minimum.

Keywords: Multiple objective programming, Supply chain management, Green logistics, Integer programming

*Corresponding author: Econometric Institute, Erasmus University Rotterdam, PO Box 1738, 3000 DR Rotterdam, The Netherlands.

Email addresses: `vdplas@gmail.com` (Corné van der Plas), `tervonen@ese.eur.nl` (Tommi Tervonen), `rdekker@ese.eur.nl` (Rommert Dekker)

1. Introduction

Design of (intercontinental) supply chains is a well established topic in operations research (Melo et al., 2009). The models developed can be used to give advice on where to locate distribution centres and which transportation links to use. Traditionally cost is used as the sole optimization criterion. As companies are more and more concerned about environmental issues, there is an increasing need to take other objectives such as greenhouse gas emissions into account in these models.

Environmentally conscious supply chain design is part of green logistics that has recently received a considerable amount of attention in the literature (see e.g. Sbihi and Eglese, 2007; Zheng and Zhang, 2010; Dekker et al., 2012). Most models for traditional supply chain design apply a single objective optimization methods (Melo et al., 2009). Some studies have applied Multi-Objective Optimization (MOO) models for supply chain design; a bi-objective model for locating hazardous waste and routing has been proposed by Alumur and Kara (2007), a four-objective one by Ioannis and Giannikos (1998), and a model integrating risks with the costs by Nema and Gupta (1999). For green supply chain design, Quariguasi Frota Neto et al. (2008) balanced costs against the environmental impact. They argued that an improvement of sustainable logistics is often only possible with a negative return on investment. The same authors presented a framework for optimizing simultaneously both the costs and two environmental impact criteria by converting heuristically the multi-objective optimization problem into multiple single objective ones (Quariguasi Frota Neto et al., 2009). Harris et al. (2011) evaluated a supply chain on its overall logistics costs and CO_2 emissions by taking into account the problem structure (e.g. the number of depots) and different freight utilization ratios; the study involved both operational and strategic decisions. Wang et al. (2011) considered a supply chain design decision involving costs and CO_2 emissions.

In all the previously mentioned green supply chain design studies the multi-objective part was treated in a simplified way, mostly by applying a single MOO method. Moreover, most studies considered only two objectives, which is an "easy" multi-objective case. In this paper we study which MOO methods perform best for realistic green supply chain models with at least three objectives. We consider *a posteriori* approaches to MOO problems where a good approximation of a representative set of pareto optimal solutions is generated first, and then a smaller set (possibly of size one) of these solutions is chosen for further consideration, often with the help of a multi-criteria decision aiding method. We evaluate applicability of four *a posteriori* MOO methods: the weighted sum and epsilon constraint scalarization methods and two evolutionary algorithms: NSGA-II and SPEA2. We have chosen these methods as on the one hand the two scalarization ones are the simplest ways to do MOO (Ehrgott, 2005), and on the other hand NSGA-II and SPEA2 are among the most applied genetic algorithms, which itself is an increasingly popular approach for MOO.

The methods are evaluated on their applicability for green supply chain design by re-analyzing an existing study that considers designing a white goods supply chain for the South Eastern European region. The study considers three objectives: costs, CO_2 and PM emissions. The original model by Mallidis et al. (2012) solved the three objectives individually. This paper extends the original study by optimizing simultaneously all three criteria of the Multi-Objective Mixed Integer Linear Programming (MOMILP) problem and showing the trade-offs between the objectives.

The rest of the paper is organized as follows. Section 2 introduces the methods, Section 3 describes the case study, and Section 4 contains details of its implementation with the given methods. Section 5 presents the study results and Section 6 concludes.

2. Methods

We consider MOO problems in supply chain design in which M objectives are to be simultaneously optimized subject to J inequality and K equality constraints. The decision variables can be both discrete and continuous ones. Obviously in non-trivial cases there exists more than a single efficient solution, often an innumerable amount. The general formulation of such problems is:

$$\begin{array}{lll} \text{Minimize} & f_1(x) & \\ \text{Minimize} & f_2(x) & \\ & \ldots & \\ \text{Minimize} & f_M(x) & \\ \text{Subject to} & g_j(x) \geq 0 & j = 1, 2, ..., J \\ & h_k(x) = 0 & k = 1, 2, ..., K \end{array} \tag{1}$$

In this study we consider different methods for solving (1); scalarization methods that transform the multiple objective problem into multiple single objective ones, and heuristic genetic algorithms that generate directly an estimation of the whole Pareto front.

2.1. Scalarization methods

Scalarization methods transform the original MOO problem (1) into a single-objective one that can be solved with standard optimization techniques. The two most basic scalarization techniques are the weighted sum and the epsilon constraint methods (Ehrgott, 2005). In the weighted sum method, the multiple objectives are transformed into a single one through a convex combination using M non-negative weights w_m:

$$\begin{aligned} &\text{Minimize} && F(x) = \sum_{m=1}^{M} w_m f_m(x) && \\ &\text{Subject to} && g_j(x) \geq 0 && j = 1, 2, ..., J \\ & && h_k(x) = 0 && k = 1, 2, ..., K \end{aligned} \tag{2}$$

The Pareto front is obtained by solving repeatedly (2) with different values for the weights w_m. Note that the solutions to this new problem can only be in the convex region of (1) (Ehrgott, 2005). Unlike the weighted sum, the epsilon constraint method can find solutions also in the non-convex regions of (1) by optimizing only one, say μ, of the M original objectives while considering the other objectives as constraints:

$$\begin{aligned} &\text{Minimize} && f_\mu(x), && \\ &\text{Subject to} && f_m(x) \leq \varepsilon_m && m = 1, 2, ..., M \text{ and } m \neq \mu \\ & && g_j(x) \geq 0 && j = 1, 2, ..., J \\ & && h_k(x) = 0 && k = 1, 2, ..., K \end{aligned} \tag{3}$$

In this transformation, different ε_m combinations yield possibly different efficient solutions. Drawback of the epsilon constraint method is its exponential computational complexity with respect to the number of objectives M (Laumanns et al., 2006).

2.2. NSGA-II

The Non-dominated Sorting Genetic Algorithm-II has been proposed for estimating meta-heuristically the Pareto fronts in MOO problems (Deb et al., 2002). It incorporates a fast non-dominated sorting algorithm to identify Pareto optimal solutions, and a diversity preservation mechanism for maintaining a well-spread Pareto front. The notation used in the genetic algorithms is presented in Table 1.

The non-dominated sorting algorithm compares iteratively pairs of alternatives to identify multiple domination fronts. A domination count n_p and the set of dominated solutions S_p are used to identify each front. The first front contains the solutions for which $n_p = 0$. For the second front, the n_p for each member of S_p is decreased by one. The members that now have $n_p = 0$ belong to the second front. This procedure is repeated for all fronts.

Table 1: Notation used in describing the genetic algorithms

Input:	P_t	*current population*
	$\bar{P}_t$	*archive set*
	N	*population size*
	$\bar{N}$	*archive size*
	$\mathcal{F}_i$	*i-th front*
	T	*maximum number of generations*
Output:	$\boldsymbol{B}$	*non-dominated set*

The crowded-comparison operator ($\succ_n$) is used for diversity preservation. The operator uses the solutions' crowding distances for ordering them. The total crowding-distance of a solution is the sum of its individual objectives' distances, that in turn are the absolute normalized differences between the solution and its closest neighbors ($\pm\infty$ at the extreme solutions). The crowded-comparison operator ($\succ_n$) ensures a uniform spread-out of the front throughout the various stages of the algorithm. In order for a solution to be preferred to another one it needs a better rank (i.e. it has to belong to a better non-domination front) or to have a larger crowding distance in case the two are of equal rank.

The main loop of NSGA-II starts with the initialization of a random parent population P_0 sorted based on the non-domination. First the offspring Q_0 of size N will be created using the usual binary tournament selection, recombination and mutation operators (Deb et al., 2002; Altiparmak et al., 2006). Algorithm 1 describes the procedure for the t-th generation. First the parents and their offspring ($R_t = P_t \cup Q_t$) are combined to obtain a population with size $2N$. Then the new population R_t is sorted according to their non-domination degrees. Elitism is ensured because the current as well as the previous members are included in R_t. The new population (P_{t+1}) will be filled with the best fronts (first $\mathcal{F}_1$, then $\mathcal{F}_2$, etc.), until the size of the next front ($\mathcal{F}_l$) is larger than the number of free slots in P_{t+1}. To have exactly N members in the new population and to maintain diversity (i.e., that a good spread of solutions is maintained in the obtained solution set), the front $\mathcal{F}_l$ is ordered based on the solutions' crowding distances and the first $N - |P_{t+1}|$ (the number of open slots) solutions will be added to P_{t+1}. The process is iterated until a stopping criterion is met.

2.3. SPEA2

The Strength Pareto Evolutionary Algorithm 2 (SPEA2) (Zitzler et al., 2001) is an improvement over the original SPEA (Zitzler and Thiele, 1999). The whole procedure is described in Algorithm 2. SPEA2 starts with an initial population (P_0) and an empty archive set ($\bar{P}_0 =$). In each iteration first the fitness value of each solution in the current

Algorithm 1 A single NSGA-II iteration for constructing the t-th generation)

```
R_t = P_t ∪ Q_t
F = fast-non-dominated-sort(R_t)
P_{t+1} = ∅ and i = 1
while |P_{t+1}| + |F_i| ≤ N do
  crowding-distance-assignment(F_i)
  P_{t+1} = P_{t+1} ∪ F_i
  i = i + 1
end while
Sort(F_i, ≻_n)
P_{t+1} = P_{t+1} ∪ F_i[1 : (N − |P_{t+1}|)]
Q_{t+1} = make-new-pop(P_{t+1})
t = t + 1
```

population is computed, followed by the environmental selection applied to generate a new archive set. If the stopping criteria are not met, the algorithm continues with selecting individuals for the next generation using binary tournament selection, and then applies genetic operators on the new generation.

Algorithm 2 Procedure of SPEA2 algorithm (Zitzler et al., 2001)

```
initialize P_0; P̄_0 = ∅.
for t = 1 → T do
  Calculate fitness for P_t and P̄_t.
  Environmental selection:  P̄_{t+1} = non-dominated solutions from  P_t ∪ P̄_t
  while |P̄_{t+1}| > N̄ do
    Reduce P̄_{t+1} by means of truncation operator.
  end while
  while |P̄_{t+1}| < N̄ do
    Fill P̄_{t+1} with dominated solutions from P̄_t.
  end while
  Binary tournament selection w/ replacement on P̄_{t+1} to fill mating pool
  Apply recombination and mutation operators to obtain P_{t+1}
end for
```

To avoid situations where population members dominated by the same members of the archive have the same fitness value, SPEA2 takes into account both the number of dominating and dominated solutions in computing the raw fitness of a solution. However, the raw fitness does not discriminate sufficiently non-dominated solutions, and the final fitness is composed of the raw fitness and additional density information computed through an adapted k-nearest neighbor algorithm.

A fixed size archive is used for the environmental selection phase, which starts by including in the new archive all the non-dominated solutions from the union of the current population and the old archive. Then, if the size of the archive is exactly the correct one ($|\bar{P}_{t+1}| = \bar{N}$), the environmental selection is completed. Else, the archive set is filled with the best dominated solutions from the previous archive or an archive truncation procedure is invoked that iteratively removes solutions until the size is equal to the predefined one. The solutions with minimum distances to other solutions, as defined with the k-th nearest neighbor procedure, are eligible for removal.

The main differences between SPEA2 and NSGA-II are the diversity assignment, replacement and archiving (Liefooghe et al., 2009). In NSGA-II the crowding-distance is used to maintain a well-spread Pareto front whereas SPEA2 applies the k-nearest neighbor approach. In addition, NSGA-II uses elitist replacement whereas SPEA2 applies a general replacement strategy, and SPEA2 uses an archive set whereas NSGA-II does not. The two algorithms are similar in that both use binary tournament as their selection method.

3. Case study: supply chain design of white goods in South-Eastern Europe

The study considers a multinational company that aims to serve a specific geographical area (market) in the South East Europe region, trading various products with similar characteristics (e.g. white goods, furniture, etc.). For this supply chain, all cargo is transported from one distant major loading point to one of the entry points. These entry points are either international ports or other major transportation nodes and therefore have no capacity limitations. Through these entry points, the goods are transported to a distribution center for container deconsolidation purposes and the regional markets are served from there onwards. The demand is centred in these markets, so that they are the last stage of the supply chain. An example such supply chain network is presented in Figure 1.

In designing the supply chain network, decisions have to be made concerning the selection of entry points, the choice of transport means, the location of distribution centers and the determination of the associated flows. Finally there is also an option to lease or outsource the transportation.

The supply chain design model has the following optimization criteria: (1) the total costs including transportation/handling costs per TEU (Twenty feet Equivalent Unit; the size of a container), operational costs of distribution centers and the total amount of emissions generated from the above supply chain operations separately for (2) CO_2 and (3) PM

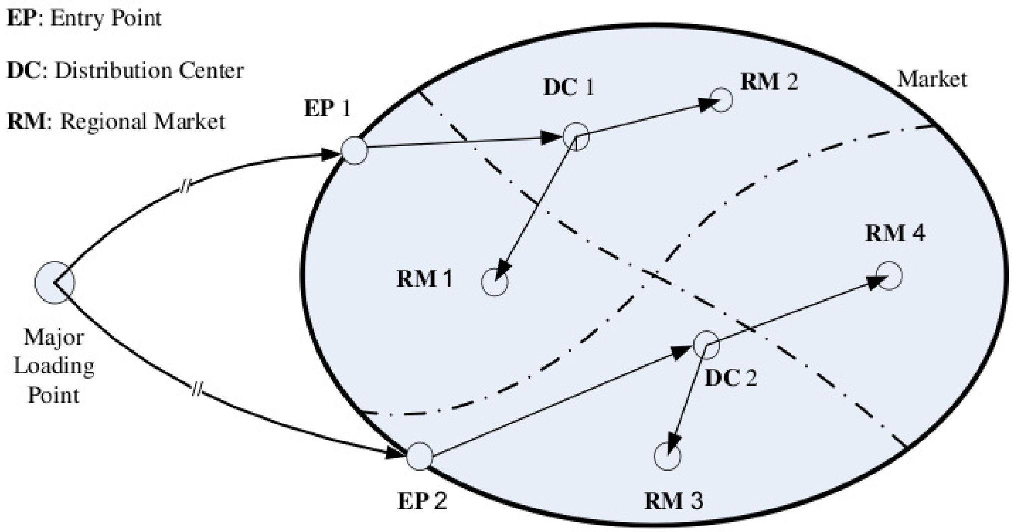

Figure 1: An example supply chain network for the case study (Mallidis et al., 2012).

emissions.

Two realistic options are considered for the supply chain designs: in *option A* the transportation is outsourced to a third party logistics provider while in *option B* the transportation is leased. The distribution centers are leased through medium time-horizon contracts for both options. The following assumptions are made to model the supply chain: (i) rail services utilize the public railway network. A block train is utilized when the number of TEUs exceeds a specific number, which results in a discount cost per TEU transported; (ii) the outsourced transportation as well as the storage together with the deconsolidation/consolidation costs are charged per TEU based on spot market prices; and (iii) the trucks of a third party logistics provider (outsourcing transportation option) will transport cargo flows of other customers in the return haul of the trip, while in the leasing option trucks are exclusively utilized and thus will return empty or almost empty (e.g. carrying commercial returns, and/or packaging material). Therefore, the transportation outsourcing results in lower emissions, albeit at somewhat higher costs.

The supply chain in the case study focuses on transporting white goods in the South Eastern Europe market that includes Bulgaria, Romania and Macedonia. There is a planning horizon of one year, and the replenishment orders are set on a monthly basis. To obtain realistic data a market share of 20% is assumed of the real annual sales of white goods. The major loading point (origin of the white goods) is the Port of Shanghai and we consider three different entry points: Ports of Thessalonki, Varna, and Constanta. There are 16 potential distribution centers located on entry points (Thessaloniki, Varna and Constanta) and regional market's capital. There are 15 regional markets considered, satisfying the demand of the entire region.

The goods are first shipped from Shanghai to Gioia Tauro (Italy) with typical mother vessels (6000 TEU), where the containers are transshipped onto feeder vessels that deliver them at the entry points. The annual demand for the region of Bulgaria, Romania and Macedonia is estimated using (i) the estimated annual demand for the same products in Greece, and (ii) the ratio of each country's region GDP related to that of Greece.

To transport the goods from the entry points to the distribution centers, they use either: (i) electrical trains in the Bulgarian and Romanian rail routes, as well as the route from Thessaloniki to Skopje; (ii) diesel trains in the route of Thessaloniki to Kulatu/Promachon (national border); or (iii) heavy duty trucks in all routes (using truck types EURO III or IV, V, VI as option). The higher the EURO type, the higher the cost per truck and the lower the PM emissions. The CO_2 emissions are hardly effected by the EURO type. To transport the goods from the distribution centers to the regional markets, heavy duty trucks are used too since delivery trucks are too small to serve the regional market retail stores. The choice of the port of entry and the number of distribution centers determine the transport route as well as the possibilities of consolidation and using electric trains. E.g. one may choose for importing all demand through the Port of Thessaloniki and transporting everything from there.

The difference in transportation times are insignificant since the major part of the total lead of maritime transportation is almost the same for all network realizations. Therefore, no inventory holding costs are included in this model. For more information on the cost and emission calculations, see Mallidis et al. (2012, 2010).

4. Implementation of the MOO model

The complete multi-objective optimization model consists of 3 objectives (costs in 1000e/y, CO_2 and PM emissions in tn/y), 2553 continuous variables, 1280 integers, 1625 constraints and 11124 non-zeros. The complete MOMILP model is included in Appendix A. For each method, the pareto front was estimated separately for options A (outsourcing transportation) and B (leasing transportation). As the study considers 4 discrete options with integer variables for the distribution center choices, the pareto front is non-continuous and expected to be non-convex as well. Trade-offs between costs, CO_2 and PM emissions are established by using different transport means (train vs truck), (de) consolidating cargo and by reallocating markets to other ports.

4.1. Scalarization methods

The Weighted Sum (WSM) and Epsilon Constraint Methods (ECM) were implemented using Lindo Lingo software. In order to obtain a front for the WSM, we used all weight combinations from 0.02 to 1 with step size 0.02, resulting in 1326 MILP problems to solve. For the ECM, we solved $9^3 = 729$ problems optimizing one of the objective while constraining the other two each with 9 different ε_m values equally spaced within an interval formed from the minimum and maximum values of the corresponding criterion of the three single-criterion optimization problems. These intervals of extreme values are, for Option A, costs: $[843.7, 1032.7]$, CO_2: $[535.1, 570.6]$, PM: $[2.7, 14.8]$; and for Option B, costs: $[825.5, 956.8]$, CO_2: $[537.9, 621.4]$, and PM: $[4.4, 27.6]$. Note that for both scalarization methods the different parameterizations can lead to equal efficient solutions, and consequently cardinality of the final solution set can be smaller than the amount of problems solved.

4.2. Genetic algorithms

We used an existing implementation of NSGA-II and SPEA2 in the ParadisEO framework (Cahon et al., 2004; Liefooghe et al., 2007) which requires the user to specify the method parameters, chromosome encoding and to implement the initialization and evaluation procedures as well as the desired genetic operators. Source code of these implementation parts specific to our case study is freely available at `https://github.com/CornePlas/MOOSCN`.

4.2.1. Chromosome encoding

We used priority-based encoding (Gen and Cheng, 1997) for the network structure and flows similarly to Altiparmak et al. (2006). With priority-based encoding the length of a chromosome is equal to the number of sources (J) plus the number of depots (K). If the supply chain consists of multiple stages, each part of the chromosome represents a single stage. To decode the chromosome to the real transportation flows, the highest priority is selected first for each part. The position of this priority integer determines which source/depot is selected, and then the corresponding transporation link with with the lowest costs is utilized. This is repeated for all $|J| + |K|$ priorities. Note that the resulting flow is either the demand of the depot or the maximum capacity of the source.

In our case study we have to implement a representation for two stages. Apart from the transportation flows, we have to represent the transportation modes as well. To this end, we add integers from zero to four to the chromosomes to represent each possible mode. Then we connect the index of the transportation mode with the same index in the priority-based part of the chromosome, i.e., the integer (transportation mode) on the i-th place belongs to the priority on the i-th place starting from the first priority. This means we have the following representation:

1. Integers $(0-4)$ of size $|EP|+|DC|$ representing transportation modes for the first stage;
2. Permutation of size $|EP|+|DC|$ with priorities to set-up the transportation tree for the first stage;
3. Integers $(0-3)$ of size $|DC|+|RM|$ representing the transportation modes for the second stage;
4. Permutation of size $|DC|+|RM|$ with priorities to set-up the transportation tree for the second stage.

The overall chromosome length is $(|EP|+|DC|)+(|EP|+|DC|)+(|DC|+|RM|)+(|DC|+|RM|) = (3+16)+(3+16)+(16+15)+(16+15) = 100$. This means that we can represent each possible solution with 100 integers and no additional encoding is necessary for the binary decision variables.

4.2.2. Genetic operators

Not all genetic operators can be applied on the different parts of the chromosome. The parts that represent the transportation modes are just integers in the range from zero to four, and default crossover and mutation operators can be used. The parts that represent the network flows are priority-encoded and require application of operators compatible with the permutation encoding: partially mapping, order or position based crossovers.

For the transportation parts, i.e. the transportation modes, we use the uniform crossover operator in which each integer of a parent has a certain chance to be exchanged for the corresponding offspring one (in our case 50%). For the priority parts of the chromosome, we use order crossover operator that changes the order of the priorities in a particular chromosome, but maintains unique priorities. Thus, for example, the first child takes consecutive integers from the first parent, but these will be re-ordered in the order they appear in the second parent. The integers outside the selected part remain the same.

We used the same mutation operators for the different parts. These are the swap, shift and TwoOpt mutation ones. The swap mutation swaps two components of a chromosome, shift mutation shifts two components of a chromosome, and TwoOpt mutation samples two random points and then takes the reverse order of the integers between the two points.

Table 2: Running times of tests instances with comparable pareto front sizes. Weighted Sum (WSM) was performed with 1326 optimization runs and Epsilon Constraint Method (ECM) with 729. Both NSGA-II and SPEA2 had 10^4 generations, 180 size population, and SPEA2 had an achieve bound to a maximum size of 100.

Method	Option	Pareto front size	Time
WSM	A	91	1h55m46s
WSM	B	103	1h11m07s
ECM	A	105	1h03m35s
ECM	B	83	28m38s
NSGA-II	A	96	10m53s
NSGA-II	B	103	10m00s
SPEA2	A	100	8m01s
SPEA2	B	100	7m42s

5. Results

Both genetic algorithms were run with a stopping criterion of reaching 10^4 generations. We also assessed that the results did not change significantly when running for 10^6 generations. The running times on both options for WSM, ECM, and the two genetic algorithms are presented in Table 2. From the computational point of view the genetic algorithms are significantly faster, but for normal supply chain design scenarios where the multi-objective optimization model is solved only once the time difference is practically insignificant.

5.1. Computational results

The Pareto fronts of the MOMILP model for the two options A and B were estimated through the two scalarization methods and the two genetic algorithms. Figures 2 and 3 present 2d plots of subsets of the efficient solutions for the cost/CO_2 and cost/PM criteria. The plotted solutions are the efficient ones for the particular option regarding the two objectives, and may therefore contain solutions that are dominated by solutions from the other option. The figures show that the scalarization methods outperform the genetic algorithms in finding efficient solutions for both options A and B. Also, the spread of the solutions from the scalarization methods is comparable to that of NSGA-II. The spread of solutions of SPEA2 is worse than that of the scalarization methods as is shown in Figure 2. NSGA-II converged towards the pareto fronts for cost and CO_2 (Figure 2), but not for the third objective, PM (Figure 3). Note that there are considerably less efficient solutions in Figure 2 than in Figure 3 as more solutions are dominated when the PM dimension is left out than when the CO_2 one is.

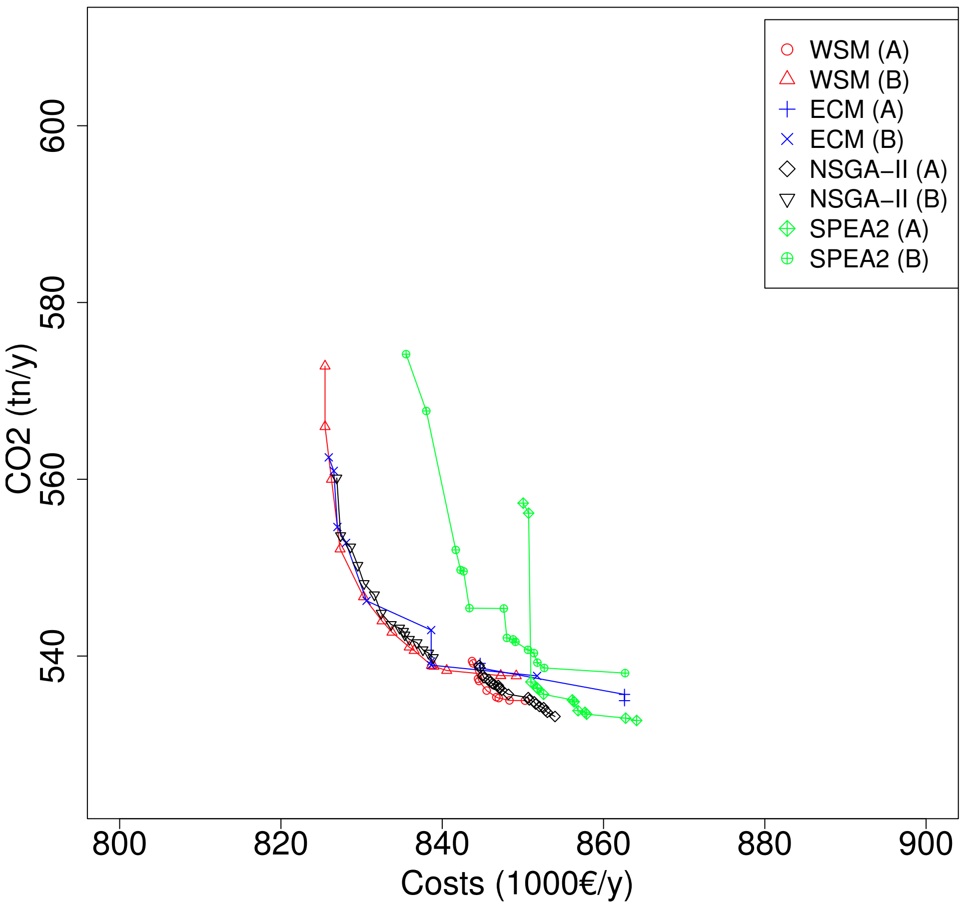

Figure 2: 2d plot of costs vs CO_2 for a subset of efficient solutions as estimated with the Weighted Sum (WSM) and Epsilon Constraint Methods (ECM), NSGA-II and SPEA2, both for options A and B.

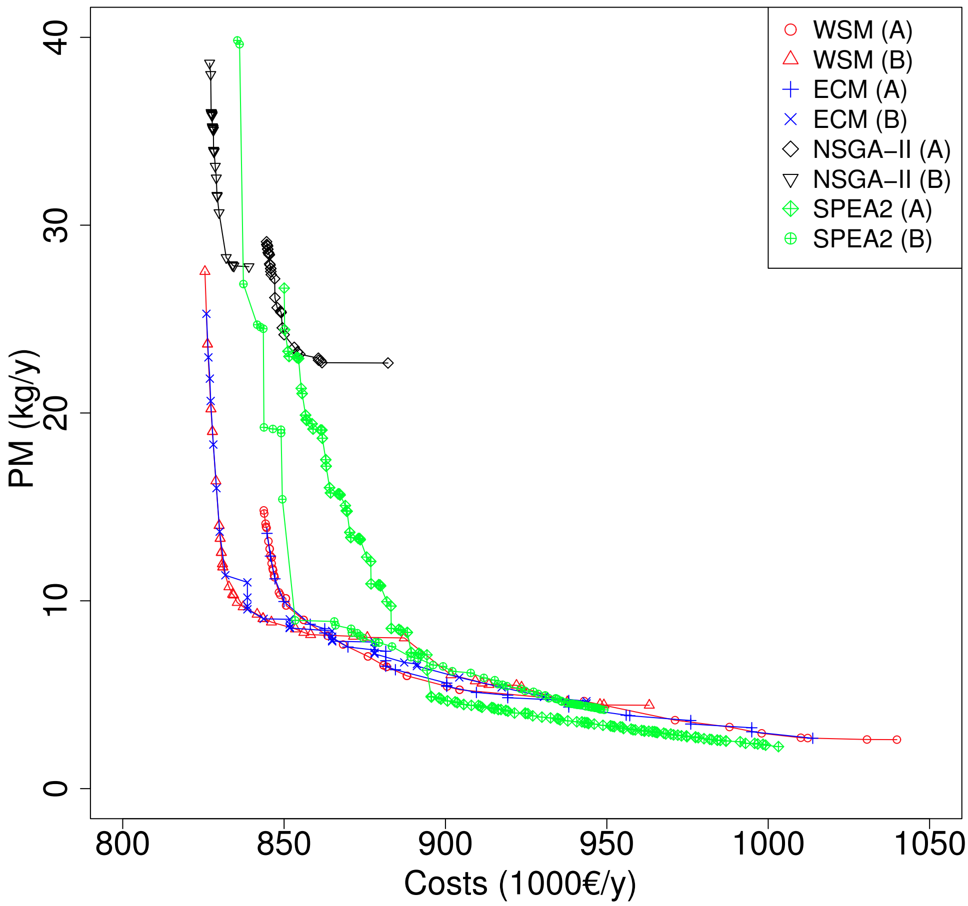

Figure 3: 2d plot of costs vs PM for a subset of efficient solutions as estimated with the Weighted Sum (WSM) and Epsilon Constraint Methods (ECM), NSGA-II and SPEA2, both for options A and B.

Taking into account that the pareto front is assumed to be non-convex, good performances of the scalarization methods and bad performances of the genetic algorithms is somewhat surprising. This might have to do with the chosen encoding although literature offers little proof to support the hypothesis. Although the running times of the genetic algorithms are considerably lower than those of the scalarization methods and the genetic algorithms' estimated fronts contain exactly the desired amount of solutions, their divergences from the optima are too large to be considered useful for our study. Therefore, for practical insights we will consider solely results from the weighted sum method.

5.2. Supply chain insights

The pareto front of both options estimated with the weighted sum method is presented as a heatmap in Figure 4. The heatmap gives especially good insights into the trade-offs between the three objectives. The cost and CO_2 objectives align pretty well: almost minimal CO_2 solutions can be obtained at little extra costs. This is in contrast with the observation of Quariguasi Frota Neto et al. (2008), who stated that one has to do investments to reduce CO_2 emissions. In most practical cases a supply chain redesign starts off with an existing non-optimal solution which is often 5 % or 10 % more expensive than the minimal cost solution, and therefore savings can be obtained both in terms of costs and CO_2 emissions.

However, minimizing PM emissions does involve much higher costs and may also lead to increased CO_2 emissions. Recall that the main option to reduce PM emission is to introduce EURO VI trucks even at the detriment of trains.These new trucks do not reduce CO_2 emissions much compared to other trucks and increase them compared to rail. Rail emissions do depend on the energy mix used for generating electicity. The solutions using option A (outsourcing transportation) perform better than the leased transportation (option B) in CO_2 emissions.

The case shows that the trade-offs are complex; it is not just about replacing one type of equipment by another one, as there are many related options. Hence one needs more than just the single-objective solutions.

The advantage of the heatmap also becomes clear when one realizes that models provide decision support rather than final decisions. Quite often many aspects other than the optimized objectives play a role in the actual decision making process (see e.g. **?**, for an overview of all aspects). This may well mean that the decision maker may be considering a non optimal solution. By representing theat solution in the heat map, one may identify which other improvements are possible from the given solution and how much these other soluitons will reduce costs or emissions.

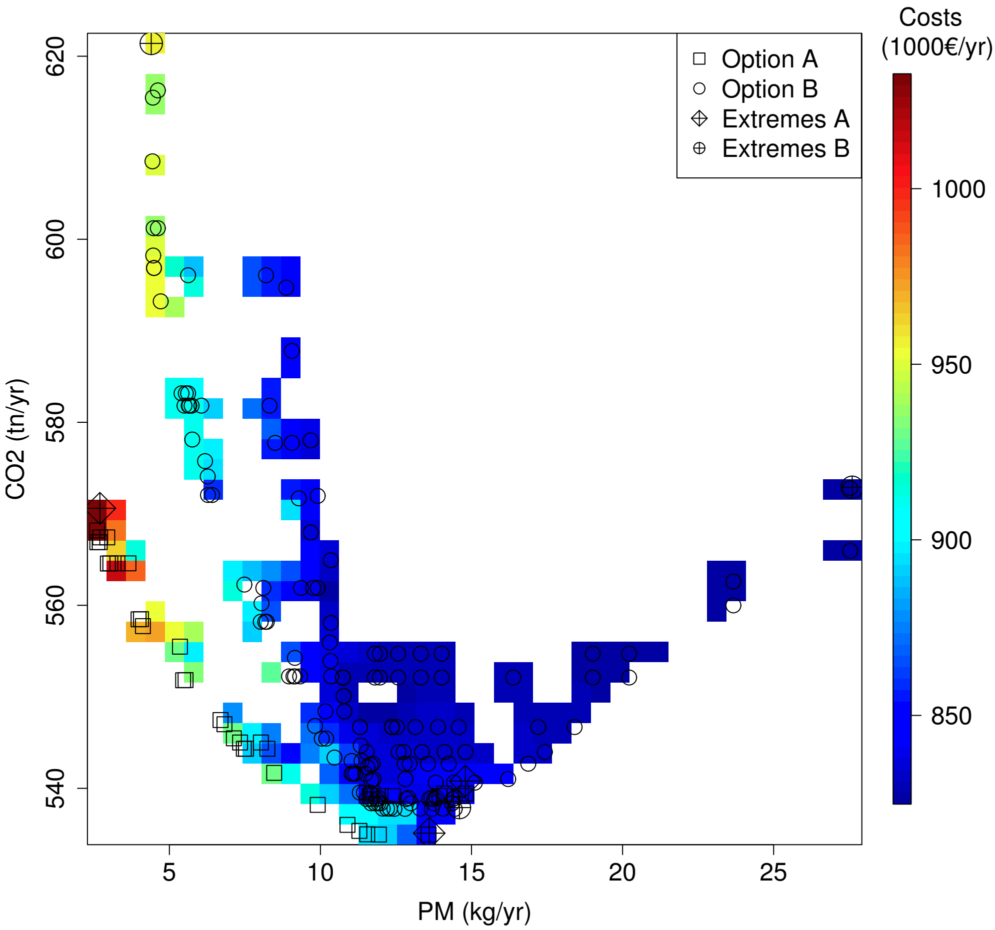

Figure 4: Heatmap of the efficient solutions for options A and B as obtained with the weighted sum method. The extreme (single-criterion optimum) solutions are marked with crossed points.

6. Conclusions

In this paper we re-analyzed a case study of green supply chain design in the South Eastern Europe region by optimizing simultaneously costs, CO_2 and fine dust (also known as PM - Particulate Matters) emissions. We evaluated four multi-objective optimization methods in the context of the study: the weighted sum and epsilon constraint scalarization methods and two genetic algorithms, NSGA-II and SPEA2. The analyses showed significantly better estimation of the pareto front by the simpler scalarization methods. From the practical point of view, we found visualization of the 3-objective pareto front with a heat map to be possibly useful for communicating the involved trade-offs. Our analyses showed that by considering the complete pareto front instead of just the extreme solutions, the designed supply chain could achieve considerable reductions in CO_2 emissions with marginal cost increases over the global minimum.

The results of this study extend those of Mallidis et al. (2012) but are also different from what has previously been presented in the literature. We showed that the cost and CO_2 objectives align unlike what was reported by Quariguasi Frota Neto et al. (2008), who stated that one has to do additional investments to reduce CO_2 emissions. By analyzing a realistic case, we showed that when three objectives are taken into account the shape of the pareto front is irregular (and therefore not adhering to the assumptions made in purely numerical studies such as that of Wang et al., 2011), but can be efficiently estimated with scalarizing methods and visualized with a heat map. Also, the standard MOO techniques seem to be sufficient for green supply chain design and we do not expect the development of novel methodologies specifically for green logistics, such as in Quariguasi Frota Neto et al. (2009), to yield results relevant for practical decision support. Finally, it seems that genetic algorithms for MOO are inapplicable to problems similar to the case studied in this paper, and the lack of negative results in the genetic algorithm literature hints at the presence of a serious publication bias.

Our study is inherently limited in concentrating on a single case, and different problem structures will probably yield different results as the scalarization methods are not tractably able to find all efficient solutions (the epsilon constraint method) or to produce efficient solutions outside the convex region (the weighted sum method). However, although our problem contained a large amount of integer variables, the scalarization methods performed well. A possible explanation for this is the low dimensionality of the problem, and our view is that in such low dimensionality problems the scalarization methods provide a sufficient estimation of the pareto front for practical applications of MOO in environmentally conscious supply chain design.

References

Altiparmak, F., Gen, M., Lin, L., Paksoy, T., 2006. A genetic algorithm approach for multi-objective optimization of supply chain networks. Computers & Industrial Engineering 51 (1), 196–215.

Alumur, S., Kara, B. Y., 2007. A new model for the hazardous waste location-routing problem. Computers and Operations Research 34 (5), 1406–1423.

Cahon, S., Melab, N., Talbi, E.-G., 2004. ParadisEO: A Framework for the Reusable Design of Parallel and Distributed Metaheuristics. Journal of Heuristics 10 (3), 357–380.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computations 6 (2), 182–197.

Dekker, R., Bloemhof-Ruwaard, J., Mallidis, J., 2012. Operations research for green logistics - an overview of aspects, issues, contributions and challenges. European Journal of Operational Research 219 (3), 671–679.

Ehrgott, M., 2005. Multicriteria Optimization, 2nd Edition. Springer.

Gen, M., Cheng, R., 1997. Genetic algorithm and engineering design. Wiley.

Harris, I., Naim, M., Palmer, A., Potter, A., Mumford, C., 2011. Assessing the impact of cost optimization based on infrastructure modelling on co2 emissions. International Journal of Production Economics 131 (1), 313–321.

Ioannis, Giannikos, 1998. A multiobjective programming model for locating treatment sites and routing hazardous wastes. European Journal of Operational Research 104 (2), 333–342.

Laumanns, M., Thiele, L., Zitzler, E., 2006. An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. European Journal of Operational Research 169 (3), 932–942.

Liefooghe, A., Basseur, M., Jourdan, L., Talbi, E.-G., 2007. ParadisEO-MOEO: A Framework for Evolutionary Multi-objective Optimization. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (Eds.), Evolutionary Multi-Criterion Optimization. Vol. 4403. Springer Berlin / Heidelberg, pp. 386–400.

Liefooghe, A., Jourdan, L., Talbi, E.-G., 2009. A unified model for evolutionary multi-objective optimization and its implementation in a general purpose software framework. In: IEEE symposium on computational intelligence in multi-criteria decision-making (MCDM'09). pp. 88–95.

Mallidis, I., Dekker, R., Vlachos, D., Aug. 2010. Greening supply chains: Impact on cost and design. Econometric Institute Report EI 2010-39b, Erasmus University Rotterdam, Econometric Institute.

Mallidis, I., Dekker, R., Vlachos, D., 2012. The impact of greening on supply chain design and cost: a case for a developing region. Journal of Transport Geography 22, 118–128.

Melo, M. T., Nickel, S., da Gama, F. S., 2009. Facility location and supply chain management - a review. European Journal of Operational Research 196 (2), 401–412.

Nema, A. K., Gupta, S., 1999. Optimization of regional hazardous waste management systems: an improved formulation. Waste Management 19 (7), 441–451.

Quariguasi Frota Neto, J., Bloemhof-Ruwaard, J., van Nunen, J., van Hoek, E., 2008. Designing and evaluating sustainable logistics networks. International Journal of Production Economics 111 (2), 195–208.

Quariguasi Frota Neto, J., Walther, G., Bloemhof, J., van Nunen, J. A. E. E., Spengler, T., 2009. A methodology for assessing eco-efficiency in logistics networks. European Journal of Operational Research 193 (3), 670–682.

Sbihi, A., Eglese, R. W., 2007. Combinatorial optimization and Green Logistics. 4QR: A Quarterly Journal of Operations Research 5 (2), 99–116.

Wang, F., Lai, X., Shi, N., 2011. A multi-objective optimization for green supply chain network design. Decision Support Systems 51 (2), 262 – 269.

Zheng, L., Zhang, J., 2010. Research on Green Logistics System Based on Circular Economy. Asian Social Science 6 (11), 116–119.

Zitzler, E., Laumanns, M., Thiele, L., 2001. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich.

Zitzler, E., Thiele, L., 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Transactions on Evolutionary Computation 3 (4), 257–271.

Appendix A. Optimization model

Table A.1: Decision variables

Variable	Description
x_{ij}^{m}	number of TEU transported from node i to node j using transportation mode m =1,..,M.
z_{ij}	binary variable which indicates whether a block train is utilized or not in the route from node i to node j.
y_{j}^{w}	binary variable which indicates whether a distribution center of size w is leased at node j or not.

Table A.2: Model parameters

Parameter	Description
D_r	total demand at regional market r.
c_{ij}^{m}	cost of transporting a TEU from node i to node j using transportation mode m (node 0 is the major loading port).
c_{ij}^{bt}	block train transportation cost from node i to node j per TEU.
c_{j}^{dc}	deconsolidation/consolidation cost per TEU at a distribution center at node j (only in the option of outsourcing).
${}^{g}e_{ij}^{m}$	emissions of type g generated from transporting a TEU from node i to node j using transportation mode.
${}^{g}e_{ij}^{bt}$	emissions of type g generated during a block train trip from node i to node j.
L^{w}	capacity of a distribution center of size w (L^{w} is considered infinite).
t_{ij}^{m}	transportation time from node i to node j using transportation mode m.
h	holding cost per TEU.
N	represents the minimum TEU volume for charging a block train.
M_0	represents a very large constant.

Minimize total cost (TC):

$$
\begin{aligned}
TC = & \sum_{i\in EP}\sum_{m=1}^{M} c_{0i}^{m}x_{0i}^{m} + \sum_{i\in EP} c_{0i}^{bt}x_{0i}^{bt} \\
& + \sum_{i\in EP}\sum_{j\in DC}\sum_{m=1}^{M}\left(c_{ij}^{m}+c_{j}^{dc}\right)x_{ij}^{m} + \sum_{i\in EP}\sum_{j\in DC}\left(c_{ij}^{bt}+c_{j}^{dc}\right)x_{ij}^{bt} \\
& + \sum_{w}\sum_{j\in DC} f_{j}^{w}y_{j}^{w} + \sum_{j\in DC}\sum_{r\in RM}\sum_{m=1}^{M} c_{jr}^{m}x_{jr}^{m}
\end{aligned}
\tag{A.1}
$$

or Minimize total emissions (TE_g) of type g:

$$
\begin{aligned}
TE_g = & \sum_{i\in EP}\sum_{m=1}^{M} {}^{g}e_{0i}^{m}x_{0i}^{m} + \sum_{i\in EP} {}^{g}e_{0i}^{bt}x_{0i}^{bt} + \sum_{i\in EP}\sum_{j\in DC}\sum_{m=1}^{M} {}^{g}e_{ij}^{m}x_{ij}^{m} + \sum_{i\in EP}\sum_{j\in DC} {}^{g}e_{ij}^{bt}x_{ij}^{bt} \\
& + \sum_{j\in DC}\sum_{r\in RM}\sum_{m=1}^{M} {}^{g}e_{jr}^{m}x_{jr}^{m} + \sum_{j\in DC}\sum_{r\in DC} {}^{g}e_{jr}^{bt}x_{jr}^{bt} \qquad \forall g \in EG.
\end{aligned}
\tag{A.2}
$$

s.t.

Flow Constraints

$$\sum_{m=1}^{m} x_{0i}^{m} + x_{0i}^{bt} = \sum_{j\in DC}\sum_{m=1}^{M} x_{ij}^{m} + \sum_{j\in DC} x_{ij}^{bt}, \qquad \forall i \in EP \tag{A.3}$$

$$\sum_{i\in EP}\sum_{m=1}^{M} x_{ij}^{m} + \sum_{i\in EP} x_{ij}^{bt} = \sum_{r\in RM}\sum_{m=1}^{M} x_{jr}^{m} + \sum_{r\in RM} x_{jr}^{bt}, \qquad \forall j \in DC \tag{A.4}$$

$$\sum_{j\in DC}\sum_{m=1}^{M} x_{jr}^{m} + \sum_{j\in DC} x_{jr}^{bt} = D_r, \qquad \forall r \in RM \tag{A.5}$$

Capacity Constraints

$$\sum_{i\in EP}\sum_{m=1}^{M} x_{ij}^{m} + \sum_{i\in EP} x_{ij}^{bt} \le \sum_{w} L^{W} y_{j}^{w}, \qquad \forall j \in DC \tag{A.6}$$

$$\sum_{w} y_{j}^{w} \le 1, \qquad \forall j \in DC \tag{A.7}$$

Block Train Constraints

$$x_{0i}^{bt} M_0 z_{0i} \le 0, \qquad \forall i \in EP \tag{A.8}$$

$$x_{0i}^{bt} N z_{0i} \geq 0, \qquad \forall i \in EP \tag{A.9}$$

$$x_{ij}^{bt} M_0 z_{ij} \leq 0, \qquad \forall i \in EP, \forall j \in DC \tag{A.10}$$

$$x_{ij}^{bt} N z_{ij} \geq 0, \qquad \forall i \in EP, \forall j \in DC \tag{A.11}$$

$$x_{jr}^{bt} M_0 z_{jr} \leq 0, \qquad \forall j \in DC, \forall r \in RM \tag{A.12}$$

$$x_{jr}^{bt} N z_{jr} \geq 0, \qquad \forall j \in DC, \forall r \in RM \tag{A.13}$$

Non-Negativity Constraints

$$x_{ij}^{m} \geq 0 \tag{A.14}$$