A Multi-Objective Memetic Algorithm for Vehicle Resource Allocation in Sustainable Transportation Planning

Hoong Chuin Lau   Lucas Agussurja   Shih-Fen Cheng
School of Information System
Singapore Management University
{hclau, lagussurja, sfcheng}@smu.edu.sg

Pang Jin Tan
DHL Supply Chain
Singapore
PangJin.Tan@dhl.com

Abstract

Sustainable supply chain management has been an increasingly important topic of research in recent years. At the strategic level, there are computational models which study supply and distribution networks with environmental considerations. At the operational level, there are, for example, routing and scheduling models which are constrained by carbon emissions. Our paper explores work in tactical planning with regards to vehicle resource allocation from distribution centers to customer locations in a multi-echelon logistics network. We formulate the bi-objective optimization problem exactly and design a memetic algorithm to efficiently derive an approximate Pareto front. We illustrate the applicability of our approach with a large real-world dataset.

1 Introduction

Having contributed significantly to economic growth, increasing employment, and the creation of a global marketplace, the logistics sector is now leading in efforts and setting trends and standards on environmental sustainability. This development has been triggered by the assessment report from the Intergovernmental Panel on Climate Change 2007 [Metz *et al.*, 2007], in which it was claimed that the transportation sector, which includes both freight and passenger transportation, accounts for 14% of the global greenhouse gas (GHG) emission. In response to the need to reduce greenhouse gas emissions, green logistics, which focuses on how to reduce the environmental effects of transportation, is attracting great interest in research.

From the managerial decision support point of view, there is an increasing emphasis in the literature on supply chain management dealing with environmental issues under the emerging concept of green supply chain management (GrSCM), which ranges from the green design of products to green operations. Srivastava conducted a comprehensive review on GrSCM which focuses on green operations Srivastava. Topics related to green operations are broad, and could potentially cover decision makings that are strategic, tactical, and operational in nature. In this paper, we study a tactical planning problem involving vehicle allocation in a multi-echelon logistics network with environmental considerations.

2 Related Works

As introduced earlier, research work on GrSCM can be classified according to the level of decision making, which can be strategic, tactical, or operational. At the strategic level, which mainly deals with the design of supply chain networks, researchers have explored the impacts of a wide range of cost elements. For example, the CO_2 emission cost by vehicle weight is considered by [Elhedhli and Merrick, 2012], and they demonstrate that such consideration would significantly change the configuration of the supply chain. Separately, Wang *et al.* investigate the trade-off between the total cost of supply chain and the environmental impact [Wang *et al.*, 2011], and come up with the conclusion that the investment on environmental protection is most beneficial when the demand level is high. A wide variety of mathematical frameworks are utilized depending on the complexity of the formulation. For example, a Lagrangian heuristic and a multi-objective mixed-integer formulation are utilized by the former and the latter examples respectively. From the literature, we also see examples using enhanced genetic algorithms (e.g., see [Chang, 2010]).

At the tactical level, network designs are given and the research questions are mostly related to the planning of fleets and the routing of vehicles in the fleet. For example, a vehicle routing problem constrained by both the fuel tank capacity and the availability of fuel stations is studied in [Erdoğan and Miller-Hooks, 2012]. This problem is modeled as a mixed integer linear program and solved heuristically. In another example with multiple objectives on environmental consideration and overall efficiency [Siu *et al.*, 2012], it is shown that by using genetic algorithms, Pareto optimality could be achieved in over 90% of the problem instances.

In this paper, we are interested in studying multi-objective tactical planning problem in the GrSCM domain. In particular, we would like to consider both service levels and energy consumption for a given three-tier logistics network. Although there is much past work on multi-objective routing problem, none can be directly applied to our problem, due to its three-tier structure. To handle the complexity of the problem, we implemented a meta-heuristic algorithm to solve the vehicle allocation problem in the GrSCM. This tactical planning problem is motivated based on an actual industrial case study and to the best of our knowledge, it has not been explored by other researchers. Our main interest is to efficiently generate Pareto frontier and use it to understand the trade-off between service level and energy requirement given a set of demand and resource constraints.

The paper is organized as follows: Section 3 describes the problem definition followed by detailed explanation of the meta-heuristic approach used in our problem in section 4. Next, we discuss different scenarios and interpret the trade-off between solutions. A conclusion will be given in the final section.

3 Problem Formulation

We are given a fixed set of vehicles which have varying capacities and fuel efficiencies. In general, the higher the capacity of the vehicle, the lower the fuel efficiency (this holds in general, but not always). It is not immediately clear what would make a good solution since if one utilizes the truck with high capacity more, one needs to make fewer trips but at the same time could potentially consume more fuel. In addition, we are given a set of demand points, and for each demand point, we are given the tonnage required and a specific time window for which the required tonnage must arrive. The time horizon is 24 hours, representing a typical day. The distribution network we consider is a three-echelon one, where goods are to be distributed from *regional distribution centers* (RDCs) to *distribution centers* (DCs), and finally to various demand points. If necessary, it is also possible to send goods directly from RDCs to demand points. In addition, DCs have storage/processing capacity constraints, and in this paper, we model this as a capacity constraint on the incoming tonnage to a DC. The decision is to assign dedicated trucks to serve links between pairs of locations. Each truck is allowed to make multiple trips but is only allowed to serve a given link. The objectives are twofold: 1) maximize the service level, which is defined as the tonnage served at demand points (not exceeding requested amount); 2) minimize the total fuel consumed by all service trucks.

3.1 A Mixed Integer Linear Programming Model

We first attempt to formulate the truck assignment problem as a mixed integer linear programming (MILP) model, which comes with two objective functions. Assume that the problem is defined on a directed graph $G = (V, E)$, such that $V = I \cup J$ and $I = I_1 \cup I_2$, where I_1, I_2 and J are the sets of RDCs, DCs, and demand points respectively. Let E denotes set of all links that allow loaded trips. In practice, nodes from higher echelon can only make loaded trips to nodes belonging to lower echelons, thus E can be formally defined as $E = \{(i_1, i_2)|i_1 \in I_1, i_2 \in I_2\} \cup \{(i_1, j)|i_1 \in I_1, j \in J\} \cup \{(i_2, j)|i_2 \in I_2, j \in J\}$. For every link $(i, j) \in E$, D_{ij} denotes the distance from locations i to j. For simplicity, we assume that distances are symmetric, i.e., $D_{ij} = D_{ji}$. Do note that although the graph is asymmetric by definition, an emptied truck has to move back from j to i if it's scheduled to serve more demands along the same link (such movements are implicit and paired with loaded trips, therefore they are not included in set E).

The capacity at each DC $i \in I_2$ is denoted as w_i. Each demand point $j \in J$ is characterized by a three-tuple (d_j, S_j, F_j), denoting the tonnage required, earliest, and latest delivery times respectively. Let K be the set of available vehicles, and for each vehicle $k \in K$, we denote by c_k its capacity, and e_k its fuel efficiency (fuel consumed per unit distance). We assume that all vehicles share the same average speed of $\mathcal{V}$. For each vehicle assigned to link $(i, j) \in E$, it can make at most N_{ij} number of trips. We now define the decision variables as follows:

- δ_{ijk}: a binary variable indicating whether vehicle k is assigned to serve link $(i, j) \in E$ (1 is true, 0 otherwise).
- n_{ijk}: number of *loaded* trips made by vehicle k using link (i, j). Since each vehicle is dedicated to only one link, to make n_{ijk} deliveries, k has to traverse link (i, j) n_{ijk} times, while returning on link (j, i) $(n_{ijk}-1)$ times.
- β_{ijk}: tonnage flowing from i to j using vehicle k.
- t_{ijk}: the time when vehicle k first departs from location i and move to j.

The two objective functions are defined below:

1. Maximize the service level, defined as the total served tonnage at demand points:
$$\sum_{j \in J} \sum_{i \in I} \sum_{k \in K} \beta_{ijk}.$$
2. Minimize the fuel consumption of all loaded trips, which is a proxy for total fuel consumption:
$$\sum_{(i,j) \in E} \sum_{k \in K} n_{ijk} D_{ij} e_k.$$

Subject to the following constraints:

1. Number of traveled trips for a link (i, j) is positive only if a vehicle is assigned to it; M is a large number.
$$0 \le n_{ijk} \le M\delta_{ijk} \ , \ (i, j) \in E, k \in K.$$
2. Each vehicle assigned to at most one link in the network:
$$\sum_{(i,j) \in E} \delta_{ijk} \le 1 \ , \ k \in K.$$
3. Served flow cannot exceed available vehicle capacity:
$$\beta_{ijk} \le n_{ijk} c_k \ , \ (i, j) \in E, k \in K.$$
4. Upper bound on the number of trips made on link (i, j):
$$n_{ijk} \le N_{ij} \ , \ (i, j) \in E, k \in K.$$
5. Outgoing tonnage at DCs cannot exceed incoming tonnage:
$$\sum_{i_1 \in I_1} \sum_{k \in K} \beta_{i_1 i_2 k} \ge \sum_{j \in J} \sum_{k \in K} \beta_{i_2 j k} \ , \ i_2 \in I_2.$$
6. Incoming tonnage cannot exceed DC capacity:
$$\sum_{i \in I_1} \sum_{k \in K} \beta_{ijk} \le w_j \ , \ j \in I_2.$$

7. Constraint on earliest delivery time at demand points:

$$\delta_{ijk} S_j \leq t_{ijk} + \frac{D_{ij}}{\mathcal{V}} \ , \ i \in I, j \in J, k \in K.$$

8. Constraint on latest delivery time at demand points:

$$t_{ijk} + (2n_{ijk} - 1)\frac{D_{ij}}{\mathcal{V}} \leq \delta_{ijk} F_j \ , \ i \in I, j \in J, k \in K.$$

9. Served tonnage should not exceed demand:

$$\sum_{i \in I} \sum_{k \in K} \beta_{ijk} \leq d_j \ , \ j \in J.$$

10. Domains of decision variables:

$$\delta_{ijk} \in \{0, 1\} \ , \ n_{ijk} \in \mathbb{Z}_{\geq 0},$$
$$\beta_{ijk} \geq 0, t_{ijk} \in \mathbb{R}_{\geq 0} \ , \ (i,j) \in E, k \in K.$$

3.2 Max-Flow Problem with Variable Capacity

The problem described above contains an interesting combinatorial subproblem when we restrict ourself to consider just the first objective function (i.e. maximizing the service level). The problem becomes one of constructing a flow network with the best maximum flow, where the nodes are fixed, but we are given flow capacity (in term of vehicles) as resources to be distributed among the links. Each vehicle will contribute a certain amount of flow capacity when assigned to a link, depending on the capacity of the vehicle and the number of trips it can make on that link, due to the inherent constraint of the link. When assigned to a link adjacent to a demand point, the number of possible trips is further restricted by the time window specified at the demand point, whichever is the lower. Hence, when assigned to different links, it is likely that a vehicle will contribute different amount of flow capacity. This, and the discrete nature of the vehicles make the problem nontrivial.

This problem consists of two levels of decision making. At the lower level, given an assignment of vehicles to links, we can construct a flow network corresponding to the resulting distribution network. The capacity of a link in the flow network represents the maximum number of goods that can be sent through that link. Maximizing the service level in the distribution network is then equivalent to finding the maximum flow of the flow network. This is a well-studied problem, and efficient solutions are readily available. Given a flow network, the value of its maximum flow is unique and equivalent to the maximum service level achievable on the corresponding distribution network. On the higher level, the assignment of vehicles to links is another decision problem, which is combinatorial in nature. We are required to find an assignment of vehicles to links such that the corresponding resulting flow network has the highest maximum flow among all possible vehicles assignment. Such flow problems, with variable capacity sizing, have been considered in the literature, see [Huang *et al.*, 2002] for example.

To be more precise, given an instance of the problem defined on the directed graph $G = (V, E)$, and an assignment of vehicles to links, we can construct the flow network $G' = (V', E')$, such that $V' = V \cup \{0\} \cup \{\emptyset\} \cup I'_2$ is the nodes of the flow network. It consists of the additional source node 0, sink node $\emptyset$, and duplicates of the DC nodes I'_2. For each $i \in I_2$, there is a corresponding $i' \in I'_2$. The links of the flow network consist of the following, $E' = \{(0,i)|i \in I_1\} \cup \{(i_1, i_2)|i_1 \in I_1, i_2 \in I_2\} \cup \{(i,j)|i \in I_1, j \in J\} \cup \{(i,i')|i \in I_2, i' \in I'_2\} \cup \{(i',j)|i' \in I'_2, j \in J\} \cup \{(j,\emptyset)|j \in J\}$. The capacity of each link in E' is defined as follows. Each link $(0,i), i \in I_1$ going from the source to an RDC node has infinity capacity. Each vehicle $k \in K$ assigned to the link $(i_1, i_2) \in E, i_1 \in I_1, i_2 \in I_2$ will contribute $N_{i_1 i_2} c_k$ capacity to the link $(i_1, i_2) \in E'$ in the flow network. Each vehicle $k \in K$ assigned to a link $(i,j) \in E, i \in I_1, j \in J$ will contribute $\min\{N_{ij} c_k, \lfloor (F_j - S_j) c_k / (2D_{ij}/\mathcal{V}) \rfloor + 1\}$ capacity to the link $(i,j) \in E'$, where the second term is the maximum flow allowed by the time window constraint. Similarly, each vehicle $k \in K$ assigned to link $(i,j) \in E, i \in I_2, j \in J$ will contribute capacity given by the same expression to the link $(i',j) \in E'$. Each link $(i,i') \in E', i \in I_2, i' \in I'_2$ has the capacity equals to w_i, the holding capacity of DC i. Finally, each link $(j,\emptyset) \in E', j \in J$ has the capacity equals to d_j the amount of goods required by demand point j.

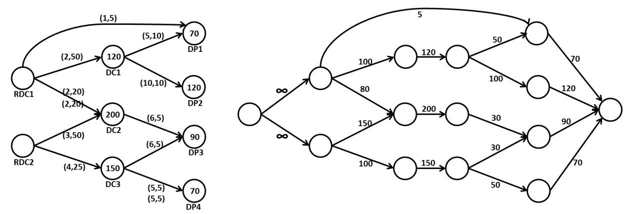


Figure 1: (Left) The tuple on a link represents the maximum number of trips and vehicle capacity respectively. The number on a DC is its holding capacity, and the number on a demand point is the requested quantity. (Right) The flow network constructed from the assignment, with max flow of 215, providing 61.43% service level.

It is not difficult to see that the max flow of the constructed network gives the maximum achievable service level given the assignment. Figure 1 shows an example of an assignment of vehicles to links and the corresponding flow network. In our solution approach, we will be using the property described here to decompose the problem.

4 Solution Approach

As shown above, our problem can be formulated as a bi-objective MILP model. Since the two objectives are potentially conflicting, a standard approach is to represent the trade-off between two conflicting objectives as a Pareto front, which is essentially a plot of all non-dominated solutions. One way to generate the Pareto front is to fix a range of service levels and for each service level, solve the MILP model to minimize the fuel consumption. Unfortunately, such exact approach is not scalable. For even a small model with just 1 RDC, 2 DCs and 10 demand points, it takes on average more than 16 hours to obtain a single solution point.

As the generation of Pareto fronts is computationally intractable for multi-objective optimization problems of practical size, past research has to rely on different heuristic approaches to approximately solve the problem (e.g., see [Bringmann *et al.*, 2011]). An important class of such approaches is the class of evolutionary algorithms (EA) which depends on an evolving population of candidate solutions. EA is particularly popular among researchers in the field since a pool of feasible solutions are available at all times, and solutions will keep improving over time.

As our problem comes with a special structure (a variant of the fixed capacity max-flow problem), which we embed into our algorithm design. The resulting algorithm is a combination of local search algorithm which considers the problem structure, and the EA which maintains and evolves the solution population. In the EA literature, such combination is commonly termed as a memetic algorithm (see [Moscato, 1989] for a recent survey). Our design of the memetic algorithm is presented next.

4.1 A Memetic Algorithm

The pseudo code of our Memetic Algorithm for Green logistics (MAG) is given in Algorithm 1. Besides the problem data introduced in the previous section, additional parameters are required to execute MAG: N_p denotes the size of population, N_o denotes number of offspring to be generated, G is number of generations, q is the probability that a parent is chosen from non-dominated solution, r_c is the point where the crossover should occur, and finally q_m is the mutation probability.

In Algorithm 1, the initial population is generated in line 1. For each new generation, each solution in the current population is first improved using local search (line 4). After all solutions are improved, both parents are sampled independently with parameter q (the probability that a parent must be chosen from the non-dominated set). The crossover then happens at r_c (line 9). The newly created offspring are then subject to mutations with probability q_m (lines 10 and 11). After N_o offspring are generated, the top N_p solutions are then selected as the next generation (line 15). The details of Algorithm 1 are explained next.

Algorithm 1: Memetic Algorithm for Green logistics (MAG).

Input: All problem-related data, N_p, N_o, G, q, r_c, q_m.
Output: Final population.

1 $P_0 \leftarrow \textsc{Init}(N_p)$
2 $t = 0, P_{t+1} \leftarrow \{\}$
3 **while** $t \leq G$ **do**
4     $P_t \leftarrow \bigcup_{p \in P_t} \{\textsc{LocalSearch}(p)\}$
5     $i = 1$
6   **while** $i \leq N_o$ **do**
7         $p_1 = \textsc{Sample}(P_t, q)$
8         $p_2 = \textsc{Sample}(P_t, q)$
9         $(p'_1, p'_2) = \textsc{CrossOver}(p_1, p_2, r_c)$
10         $p'_1 = \textsc{Mutate}(p'_1, q_m)$
11         $p'_2 = \textsc{Mutate}(p'_2, q_m)$
12         $P_{t+1} \leftarrow P_{t+1} \bigcup \{p'_1, p'_2\}$
13         $i = i + 2$
14   **end**
15     $P_{t+1} \leftarrow \textsc{Selection}(P_{t+1}, N_p)$
16     $t = t + 1$
17 **end**
18 **return** P_G

4.2 Solution Representation & Fitness Function

A solution is represented by an array A of size $|K|$, where each element $A[i]$ represents the link that vehicle i is assigned to. If vehicle i is not assigned to any link, a special code na is used. The domain of $A[i]$ is set K plus na.

A solution's fitness is a two-tuple which contains both the service level and the total fuel consumption. To compute both service level and fuel consumption, we need to first construct the flow network (where link capacities are derived from vehicle assignments) and then solve the max-flow problem. After solving the max-flow problem, the fuel consumption can be obtained based on the actual distances traveled by all vehicles on their respective links times their fuel efficiencies.

The construction of the flow network from vehicle assignment is given as follows:

1. Start with the same number of vertices as the number of (RDCs + DCs + Demand Points).
2. An edge exists in the flow network if there is at least one vehicle assigned to the corresponding link.
3. The capacity of an edge is the total capacity of vehicles assigned to the corresponding link.
4. Replace each vertex corresponding to a DC by a pair of vertices d and d', and create an edge from d to d' with capacity equals the capacity of the DC. Incoming edges to the DC go to d, outgoing edges from the DC go out of d'.
5. Add a source vertex and an edge from the source vertex to each RDC vertex with infinite capacity.
6. Add a sink vertex and an edge from each demand vertex to the sink vertex with capacity equals the demand at the corresponding demand demand point.

The algorithm used for computing maximum flows is the standard Ford-Fulkerson algorithm. It runs in time $O(|V|^3)$, $|V|$ being the number of vertices in the flow network. The computation of total fuel consumption is straightforward.

4.3 Initialization

The initialization procedure is given as follows:

1. Generate a random real number p between 0 and 1 (exclusive), which becomes the probability of a vehicle being assigned to a link.
2. Create a solution where each vehicle will be assigned to a (uniformly) randomly chosen link with probability p (in other words, with probability $(1 - p)$, the vehicle will be assigned na).
3. Repeat the above steps until the population size is reached.

4.4 Local Search

Local search is performed on each non-dominated solution in the current population as follows. A neighbor of a solution is another solution with the same array differing only in one position (one vehicle). First, a set of random neighbors is generated, the size of which is given as a parameter. Second, for each of superior neighbors in the set, where a superior neighbor is one that dominates the current solution, compute its distance from the incumbent. The superior neighbor with maximum *distance* will be chosen. If no superior neighbor can be found in the set, keep the incumbent solution. The distance between two solutions p_1 and p_2 is computed as:

$$\frac{1}{M}(p_1.\text{fuel} - p_2.\text{fuel})^2 + (p_1.\text{service} - p_2.\text{service})^2, \quad (1)$$

where $1/M$ is the normalizing factor. Repeat the above steps until maximum number of iterations is reached.

4.5 Crossover and Mutation

The crossover and mutation operators are described as follows:

1. With probability q, a parent is uniformly chosen from the set of non-dominated solutions of current population (P_t), otherwise it is uniformly chosen from the whole population.
2. Crossover is performed by exchanging at fixed point (denoted as r_c), segment 1 from p_1 and segment 2 from p_2 will form the first offspring, while segment 2 from p_1 and segment 1 from p_2 will form the second offspring. Note that feasibility is always maintain in crossover.
3. This operation is performed until the maximum number of offspring is reached (N_o).
4. The children are then added to the next population.
5. In the mutation step, each solution in the population has a fixed probability (q_m) of being mutated, where an element of its array is changed randomly.

4.6 Selection

Of all the newly generate offspring in the population P_{t+1}, the non-dominated solutions (against all other solutions in P_{t+1}) are chosen first. If there are still vacancies, the dominated solutions whose closest distance to the non-dominated set is smallest will be chosen first. The distance between two solutions is computed as in (1).

5 Experimental Results

In our empirical study, we perform two sets of experiments. The first set, using generated data with certain properties, aims to analyze the effectiveness and efficiency of our solution approach. This is done by comparing our solutions with the exact solutions obtained from running CPLEX. The second set, using real-world data obtained from a large logistics company, aims to demonstrate the capability of our solution in supporting real-world decision makers. In particular, we quantify the impact of two types of investment decisions that this company can make: 1) improving fuel efficiency of the fleet, and 2) relocating DCs).

The parameters used by all our experiments are given as follows: N=100; N_o=500; G=10; q=0.5; q_m=0.25; and r_c (i.e. the crossover point) is set at elements $\lfloor 0.75|K| \rfloor$ and $\lfloor 0.25|K| \rfloor$ for p_1 and p_2 respectively.

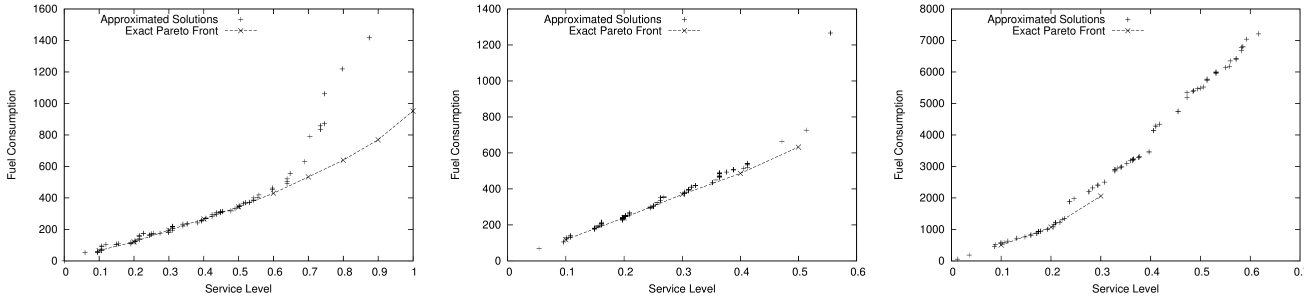
5.1 Run Time vs Optimality

We generate three sets of data that are used to compare our solution approach with the exact method. Each set of data describes a small distribution network of 1 RDC, 2 DCs, and 6 demand points. The differentiating features between the sets are the types and the number of vehicles used. The first set uses 34 vehicles with high capacities and high fuel consumptions. The second set uses 68 vehicles with medium capacities and fuel consumptions, while the third set uses 270 vehicles with low capacities and fuel consumptions. In terms of computational efforts, the three sets are in increasing order of difficulty. To generate the exact solutions, we fix a range of service levels with interval of 0.1 as constraints, and for each service level, we solve the MILP optimally using CPLEX solver (on a machine with Intel i5@3GHz processor). The time limit is set to 10 hours for each data set. We believe that this is a reasonable limitation, given that our solution approach returns its solutions within minutes on the same machine. The results are shown in Figure 2.

The exact method manages to return all solution points on the first data set (Figure 2a), up to 0.5 service level on the second (Figure 2b), and only up to 0.3 service level on the third (Figure 2c). The inability of the exact method to generate enough points for a reasonably sized data justifies the use of heuristic method in this case. We observe that our method is able to give a good approximation of the exact solutions especially at 0.75 service level or less. At higher service level we see a drop in performance of the approach (Figure 2a). This is one issue that we will address in our future work. One possible extension that might overcome this problem is to focus on the part of the population that are around the given range. This will also give us a greater control on the population's evolution.

5.2 Real-World Case Study

The data used for the second set of experiments is based on an actual business operational problem. Here, the distribution network designed to deliver finished goods to customers is three-echelon. There are 5 regional distribution centers (RDCs) and 13 distribution centers (DCs) to serve more than 1800 demand points. Each demand point can be characterized by the total tonnage of the goods to be delivered to that particular demand point. Also, there are restrictions on the time periods for which the goods must arrive, failing which results in a case of service failure. In order to support the scale of operations, more than 5000 vehicles are deployed. There are 10 different types of vehicles being used, and they are of varying capacity and fuel efficiency. Quick review also shows that for the existing fleet, trucks with higher capacity tend to have a lower fuel efficiency. Moreover, trucks are assigned to the facilities (either RDCs, or DCs) which will then be used to serve specific *lanes* originating out from the facilities. With increasing environmental concerns, it is therefore important to develop an understanding on how the different strategies impact fuel consumption and service levels at the same time. This will help to design a truck assignment/deployment approach that balances both environmental and commercial concerns. Our results are shown in Figure 3.

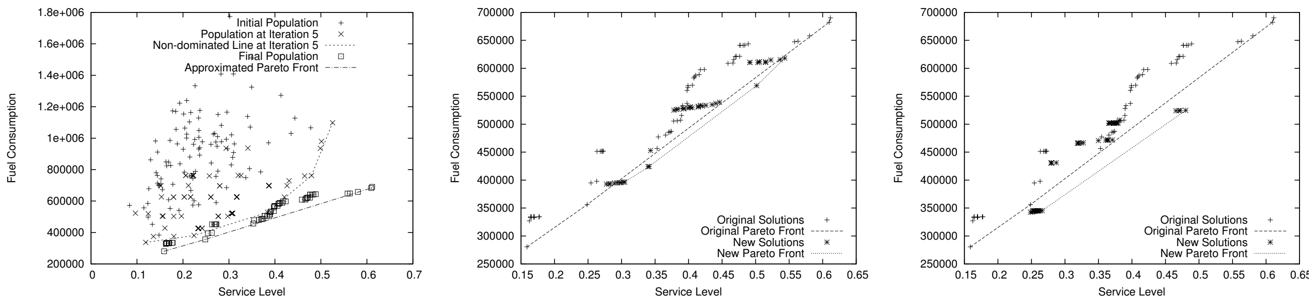


(a) Low number of vehicles with high capacities and fuel consumptions.

(b) Medium number of vehicles with medium capacities and fuel consumptions.

(c) High number of vehicles with low capacities and fuel consumptions

Figure 2: Comparing against the exact method.



(a) The performance of original setup.

(b) Increase vehicles' fuel efficiencies by 20%.

(c) Reduce the distances from DCs to demand points by an overall 10%.

Figure 3: Analysis using our method on a set of real-world operational data.

Figure 3a shows the evolution of the algorithm on this data. It shows the states of the population at three different iteration points: at initialization, at iteration 5, and at final stage (iteration 10). One property that we observe is that our approach converges quickly to local optima. After iteration 10, additional iterations would not increase the quality of the population significantly. This is why we set the number of iteration to 10 for this particular case, giving us the balance between running time and the quality of solutions. It takes about one hour to complete 10 iterations.

We then make two separate modifications on the data, and re-run our algorithm on the two modified datasets, to see how sensitive the solutions are to these changes. The first modification is on the vehicles. For each type of vehicles we increase its fuel efficiency by 20%, and the result is given in Figure 3b, which shows that from around 0.27 to 0.55 service level, fuel consumptions are reduced by around 4.2%. It is safe to assume that the improvement will be quite uniform across the whole range of service levels. The reason why we are saving only 4.2% of fuel and not 20% is due to the bias towards service level in the fitness function, and we recognize it as a limitation of our current approach. The second modification is a reduction in the overall distance from DCs to demand points by 10%. Here, we ask the question: if it is possible to relocate the DCs such that the overall distance to the demands points are reduced by 10%, how much saving can we expect from fuel consumptions? The result, as plotted in Figure 3c, shows that we can expect a saving of around 9% across the whole range of service levels. Therefore, the solutions are much more sensitive to changes in connecting distances than to changes in fuel efficiencies. This is one demonstration of how our solution approach can be used to better guide decision makers in deciding the kind of investments to make and their impact to the company's performance in terms of service level and environmental sustainability.

6 Conclusions

With the increasing emphasis on sustainable transportation planning, there has been a lot of research on developing optimization models that can help planners with decision making. Inspired by a real world case study, we developed a mixed integer bi-criteria optimization model to solve a tactical planning problem in vehicle resource allocation. We then proposed a memetic algorithm to derive an approximate Pareto front for the problem, and demonstrated with different setups that the method allowed us to develop insights into how different scenarios have led to different conclusions. Future works will address some limitations of our current approach, which include: (1) Drop in performance in the higher range of service levels, and (2) Bias in the fitness function. We would also like to compare our approach to existing (meta-)heuristic multi-objective optimization methods.

References

[Bringmann *et al.*, 2011] Karl Bringmann, Tobias Friedrich, Frank Neumann, and Markus Wagner. Approximation-guided evolutionary multi-objective optimization. In *Twenty-Second International Joint Conference on Artificial Intelligence*, pages 1198–1203, 2011.

[Chang, 2010] Ying-Hua Chang. Adopting co-evolution and constraint-satisfaction concept on genetic algorithms to solve supply chain network design problems. *Expert Systems with Applications*, 37(10):6919 – 6930, 2010.

[Elhedhli and Merrick, 2012] Samir Elhedhli and Ryan Merrick. Green supply chain network design to reduce carbon emissions. *Transportation Research Part D: Transport and Environment*, 17(5):370 – 379, 2012.

[Erdoğan and Miller-Hooks, 2012] Sevgi Erdoğan and Elise Miller-Hooks. A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):100 – 114, 2012.

[Huang *et al.*, 2002] Simin Huang, Rajan Batta, and Rakesh Nagi. Variable capacity sizing and selection of connections in a facility layout. *IIE Transactions*, 2002.

[Metz *et al.*, 2007] B. Metz, O.R. Davidson, P.R. Bosch, R. Dave, and L.A. Meyer, editors. *Contribution of Working Group III to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, 2007.

[Moscato, 1989] Pablo Moscato. On evolutionary, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Technical Report Caltech Concurrent Computation Program Report (158-79)*, 1989.

[Siu *et al.*, 2012] Winson S.H. Siu, Chi-Kong Chan, and Henry C. B. Chan. Green cargo routing using genetic algorithms. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, volume 1, Hong Kong, March 2012.

[Srivastava, 2007] Samir K. Srivastava. Green supply-chain management: A state-of-the-art literature review. *International Journal of Management Reviews*, 9(1):53–80, 2007.

[Wang *et al.*, 2011] Fan Wang, Xiaofan Lai, and Ning Shi. A multi-objective optimization for green supply chain network design. *Decision Support Systems*, 51(2):262–269, 2011.