OGUNSEMI, A., MCCALL, J., ZAVOIANU, C. and CHRISTIE, L.A. 2024. Cost and performance comparison of holistic solution approaches for complex supply chains on a novel linked problem benchmark. In *Proceedings of the Genetic and evolutionary computation conference 2024 (GECCO'24), 14-18 July 2024, Melbourne, Australia*. New York: Association for Computing Machinery (ACM) [online], pages 1327- 1335. Available from: https://doi.org/10.1145/3638529.3654163

Cost and performance comparison of holistic solution approaches for complex supply chains on a novel linked problem benchmark.

OGUNSEMI, A., MCCALL, J., ZAVOIANU, C. and CHRISTIE, L.A.

2024

Cost and Performance Comparison of Holistic Solution Approaches for Complex Supply Chains on a Novel Linked Problem Benchmark

Akinola Ogunsemi
National Subsea Centre, Robert Gordon University
a.ogunsemi1@rgu.ac.uk

John McCall
National Subsea Centre, Robert Gordon University
j.mccall@rgu.ac.uk

Ciprian Zavoianu
National Subsea Centre, Robert Gordon University
c.zavoianu@rgu.ac.uk

Lee A. Christie
National Subsea Centre, Robert Gordon University
l.a.christie@rgu.ac.uk

ABSTRACT

Modern supply chains are complex structures of interacting units exchanging goods and services. Business decisions made by individual units in the supply chain have knock-on effects on decisions made by successor units in the chain. Linked Optimisation Problems are an abstraction of real-world supply chains and are defined as a directed network where each node is a formally defined optimisation problem, and each link indicates dependencies. The development of approaches to holistically solve linked optimisation problems is of high significance to decarbonisation as well as building robust industrial supply chains resilient to economic shock and climate change. This paper develops a novel linked problem benchmark (IWSP-VAP-MTSP) integrating Inventory Warehouse Selection Problem, Vehicle Assignment Problem and Multiple Traveling Salesmen Problem. The linked problem represents tactical and operational supply chain decision problems that arise in inventory location and routing. We consider three algorithmic approaches, Sequential, Nondominated Sorting Genetic Algorithm for Linked Problem (NSGALP) and Multi-Criteria Ranking Genetic Algorithm for Linked Problem (MCRGALP). We generated 960 randomised instances of IWSP-VAP-MTSP and statistically compared the performance of the proposed holistic approaches. Results show that MCRGALP outperforms the other two approaches based on the performance metrics used, however, at the expense of greater computational time.

CCS CONCEPTS

• **Applied computing → Supply chain management**.

KEYWORDS

linked optimisation, genetic algorithm, multi-criteria decision-making, scheduling and planning

ACM Reference Format:
Akinola Ogunsemi, John McCall, Ciprian Zavoianu, and Lee A. Christie. 2024. Cost and Performance Comparison of Holistic Solution Approaches for Complex Supply Chains on a Novel Linked Problem Benchmark. In *Genetic and Evolutionary Computation Conference (GECCO '24), July 14–18, 2024, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3638529.3654163

GECCO '24, July 14–18, 2024, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-0494-9/24/07.
https://doi.org/10.1145/3638529.3654163

1 INTRODUCTION

Modern supply chains are complex structures of interacting units exchanging goods and services. Business decisions made by individual units in the supply chain have knock-on effects on decisions made by successor units in the chain. Linked optimisation problems are abstraction of real-world supply chains and are defined as a directed network where each node is a formally defined optimisation problem, and each link indicates dependencies as studied in [23]. Specifically, the solution to a parent problem places conditions on the solution set, objective function and constraints of dependent problems. Developing resilient and sustainable methods for solving linked optimisation problems holistically is crucial to decarbonisation and to building strong industrial supply chains.

However, the theoretical investigation of computational intelligence methods for tackling linked problems poses a new challenge as there are no complex computational analysis provided to give insights into the interactions between the different sub-problems [5]. We, therefore, offer two research outcomes: 1. creating benchmark problems by linking existing benchmark sets. For example, given benchmark sets, say travelling salesman problem (TSP) and knapsack problem (KP) problems, and a semantic process for how a TSP benchmark solution may give rise to a KP benchmark instance, we can set up a supply chain benchmark where solutions to the linked problems are combined i.e., (s_1, s_2) where s_1 is a solution to a TSP and s_2 is a solution to the KP instantiated from s_1. 2. adopting holistic approaches that consider the dependency relationships between linked problems. Specifically, investigating to what extent the choice of solution for sub-problem p_1 affects the value available to solvers of the instantiated sub-problem p_2.

Concatenation is a common strategy used in [14], where two problems - job shop scheduling problem (JSSP) & vehicle routing problem (VRP) were tackled using a cooperative coevolutionary approach. By concatenation, we mean embedding n problems in a given algorithm while simultaneously considering the different configurations of the problems in a linked structure shown in Figure 1. The approach uses known and/or specially designed operators to enhance the algorithm performance further to explore complex search spaces relating to the linked problem [18]. Multitasking op-

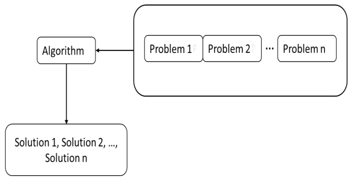


Figure 1: Concatenation Approach

timisation also uses similar strategy. Multi-factorial evolutionary algorithm (MFEA) is one of the newly proposed solvers for multi-factorial optimisation (MFO) that takes advantage of the genetic complementarity across tasks. However, this algorithm ignores the connection between the optimisation tasks [35]. Each task represents a contributing factor that affects the population evolution in the algorithms designed for MFO problems [33].

This paper investigates a supply chain problem that integrates Inventory Warehouse Selection Problem, Vehicle Assignment Problem and Multiple Traveling Salesmen Problem (IWSP-VAP-MTSP). The rest of the paper is organised as follows. Section 2 gives a brief background of similar work in the study of IWSP-VAP-MTSP. Section 3 presents the formulation of the IWSP-VAP-MTSP. Section 4 describes our algorithmic approaches. In Section 5, we provide details about our experimental setup and discuss results. Lastly, Section 6 concludes and presents future works.

2 PROBLEM BACKGROUND

IWSP-VAP-MTSP refers to a class of optimisation problems that involve the integration of inventory management, warehouse selection for demand fulfilment, vehicles assignment for delivery and multiple routing of vehicles to customer locations. IWSP-VAP-MTSP is an extension of the ILRP studied in the literature. So far, different variations of ILRP have been explored in the literature. For instance, Serna et al. [30] paper investigates the application of Inventory Routing Problem (IRP) for the distribution of goods between multiple clients and multiple suppliers using genetic algorithm. Likewise, Saif et al [29] consider how the integration of inventory, location, and routing decisions (ILRP) may impact supply chain performance. The authors adopt vendor managed inventory (VMI) strategy and design an improved genetic algorithm (IGA) to solve the problem. Similarly, Perez et al [27] present a multi-objective based metaheuristic algorithm to tackle a bi-objective supply chain problem involving location-routing problem and supplier selection problem. They attempt to minimise the total costs on the entire chain, and to maximise suppliers' equipment effectiveness [27].

Another aspect we included in the problem is the concept of vehicle assignment to fulfil demands. This problem seeks to assign the right vehicle so that the overhead costs of using the vehicles is minimised [2]. Many studies have investigated fleet operation scheduling and assignment. One of them is seen in Solos et al [32] work. They apply a stochastic variable neighbourhood algorithm to solve shift scheduling problem of tank trucks.

There are many real-world scenarios applicable to multiple travelling salesmen problem (MTSP). Examples are school bus routing problem, and the pickup and delivery problem. In Vali et al's [34] study, authors explore constraint programming (CP) to formulate and solve MTSP based on interval variables, global constraints and domain filtering algorithms. Similarly, Camci et al [6] investigate a travelling maintainer problem (TMP) based on a generalised formulation of TSP. TMP seeks to find the best route for maintainers that minimises the travel, maintenance, and expected failure cost for all cities. A genetic algorithm and particle swarm optimisation solutions were proposed for comparison in the TMP study.

In IWSP-VAP-MTSP, determining the optimal number of warehouse and inventory, vehicle assignment and obtaining the best multiple permutations of tours are the decisions that must be taken simultaneously. In tackling the IWSP-VAP-MTSP, we need to identify how the three problems (IWSP, VAP and MTSP) are connected. The integration of IWSP, VAP & MTSP causes complexity in designing appropriate algorithms for solving the problem.

3 PROBLEM FORMULATION

3.1 Problem Statement

IWSP-VAP-MTSP is a mix of tactical and operational supply chain decision problems which involves the integration of warehouse selection and inventory decisions, vehicle assignment decision and routing decision. The problem linkages introduce non-linearity and can be defined as follows: demands of J retailers have to be serviced by a chosen subset of K potential warehouses, in a single period. The problem considers a single product where each warehouse is limited by capacity to fulfil all demands. Once a retailer's demand $j, j = 1, 2, \cdots, J$ is assigned to a warehouse $k, k = 1, 2, \cdots, K$, it cannot be transferred to another warehouse as it must be serviced by the assigned warehouse. Furthermore, decisions have to be made to replenish inventory once retailer's demands are fulfilled. The total cost associated with the inventory and warehouse selection comprises of cost relating to warehouse fixed cost, transportation cost, working inventory cost and safety stock cost. Here, each retailer has different variance and demands. Consider a truckload trucking company operating a fleet of trucks contracted to fulfil demands at different retailers' locations. Each truck can be loaded with multiple retailers' demands but the capacity of each truck limits the number of retailers it can service. The amount of retailers assigned to each truck determines the tour of the individual truck. The decision made by the fleet company incorporates three decision levels, i.e., the decisions of what vehicles should be assigned to selected warehouse k, what retailers assigned to warehouse k should be serviced by vehicle v and what tour would provide the shortest traveling distance for vehicle v to fulfil delivery of retailer j demands. The assignment problem seeks to determine the number of vehicles which perform the tasks of fulfilling the demands. We assume that the total cost for the assignment task includes the cost of vehicles assigned to each warehouse and the cost of fulfilling the demands which amounts to the transportation cost attributed to the warehouse. The routing decision results to a multiple travelling salesmen problem where each truck must find the shortest path at which retailers' demands are fulfilled within the delivery time window. The routing decision determines the costs associated with the operations of the vehicles in fulfilling the retailers' demands.

Let a linked optimisation problem be P and n be number of connected problems. Let D describes the connectedness of the problems and represents an adjacency matrix of the linkages between problems in P. P can be defined as;

$$P = \{p_1, p_2, \cdots, p_n, (D)\} : \quad p_\iota \in P \quad and \quad \iota = 1, \cdots, n \tag{1}$$

The mathematical formalism for linked problem is available in [23]. IWSP-VAP-MTSP is modeled from a linked problem perspective based on Figure 2. Solution x_{IWSP} of IWSP serves as input for VAP and modifies the original solution representation, $\mathrm{x}_{VAP} = (\mathrm{x}_0, \mathrm{x}_1, \cdots, \mathrm{x}_{J-1})$ of VAP to $\mathrm{x}_{VAP\{x_{IWSP}\}} = (\mathrm{x}_0, \mathrm{x}_1, \cdots, \mathrm{x}_{K-1})$, such that, $k = 0, 1, \cdots, K-1$, where $\mathrm{x}_k = (\rho_0, \rho_1, \cdots, \rho_{J-1})$, $j = 0, 1, \cdots, J-1$. Each integer $\rho_j \in \{0, 1, 2, \cdots, V\}$ corresponds to a vehicle assigned to warehouse k and index j in x_k represents a retailer. $\rho_j = 0$ implies that no retailer j is serviced by vehicle v in warehouse k. Next, the solution of VAP is injected as inputs for MTSP. This creates sub-tours where, each tour corresponds to a vehicle. One or more tours could be associated with a selected warehouse. Once the solution for MTSP is determined, the solution feeds back to the VAP to modify its objective function. Likewise, the solution for VAP in turn, feeds back to IWSP to modify its objective function.

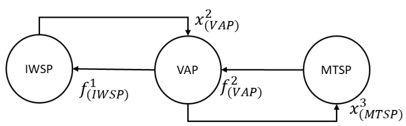


Figure 2: IWSP, VAP and MTSP Linkages

3.2 Formulation of IWSP-VAP-MTSP

We establish a linked optimisation model of IWSP-VAP-MTSP based on three criteria; total warehouse and inventory cost, total vehicle cost and total traveling distance.

3.2.1 Inventory-Warehouse Selection Problem (IWSP). IWSP is a variant of FLP and inventory location model involving a selection of warehouses to service set of customers demands at minimum cost and making inventory decision to minimise total inventory cost. IWSP includes warehouse operational and fixed costs, transportation costs between warehouses and retailers, and safety and cycle stock costs for each warehouse. Notations for decision problem is defined in Table 1. The formulation of IWSP is described below according to [27]. The model considers an aggregated function with given set of K potential warehouses $\mathbb{F} = \{\mho_0, \mho_1, \cdots, \mho_{K-1}\}$ such that, $\mho_k \in \mathbb{F}, k = 0, 1, \cdots, K-1$ and J retailers to be assigned to a subset of K warehouses which seeks to minimise $f^1(x_{IWSP})$. IWSP's solution is denoted as $x_{IWSP} = \{x_1, \cdots, x_K\} \in \{0, 1\}^K$.

$$\begin{aligned} min f^1(x_{IWSP}) = & \sum_{k=0}^{K-1} \alpha_k x_k + \sum_{k=0}^{K-1} \sqrt{2 \cdot D_k \cdot OC_k \cdot HC_k} \\ & + \sum_{k=0}^{K-1} Z_p \cdot HC_k \cdot \sqrt{LT_k} \cdot \sqrt{\pi_k} + \sum_{k=0}^{K-1} \sum_{j=0}^{J-1} c_{kj} x_k y_{kj} \end{aligned} \tag{2}$$

Subject to;

$$\sum_{k=1}^{K} y_{kj} = 1 \quad \forall j = 1, 2, \cdots J \tag{3}$$

$$\sum_{j=0}^{J-1} d_j \cdot y_{kj} = D_k \quad \forall k = 0, 1, 2, \cdots K-1 \tag{4}$$

$$\sum_{j=0}^{J-1} \mu_j \cdot y_{kj} = \pi_k \quad \forall k = 0, 1, 2, \cdots K-1 \tag{5}$$

Table 1: IWSP Decision Variables

Var	Description
α_k	Fixed and operational costs of selected warehouse k
K	Number of candidate warehouse
J	Number of retailers
x_k	Binary variable indicates 1 if warehouse k is selected and 0 otherwise
y_{kj}	Binary variable indicates 1 if a selected warehouse k is assigned to fulfil retailer j and 0 otherwise
c_{kj}	Cost of transportation to fulfil retailer j demands by warehouse k
HC_k	Holding cost per unit product at warehouse k
OC_k	Ordering cost per unit product at warehouse k
D_k	Average demand at warehouse k
w_k	Demand capacity of warehouse k
LT_k	Lead time to supply warehouse k
π_k	Demand variance of warehouse k
μ_j	Demand variance of retailer j
d_j	Average demand of retailer j
Z_p	Value of the accumulated standard normal distribution with a probability p related to the service level

$$D_k \leq w_k \quad \forall k = 1, 2, \cdots K-1 \tag{6}$$

$$x_k, y_{kj} \in \{0, 1\} \quad \forall j, k \tag{7}$$

Constraint 3 ensures that each retailer is assigned to only one warehouse. Constraint 4 - 5 adds the average demand and demand variance of the allocated retailers to each warehouse respectively. Constraint 6 guarantees that demands at warehouse k are less than or equal to its capacity. Constraint 7 defines the decision variables.

3.2.2 Vehicle Assignment Problem (VAP). VAP is a decision problem of assigning heterogeneous fleet of V vehicles to fulfil demands from distributed warehouses to J retailers. VAP is modeled according to the vehicle cost modelling in [29] which includes dispatching and routing costs. The dispatching cost is a fixed cost and the routing cost is associated with the travelled distance. The vehicles differ in capacities and the assignment has to satisfy all demands serviced by the distributed warehouses within a single period. We seek to find the assignment of J retailers' demands $\mathrm{x}_{VAP} = (\mathrm{x}_0, \mathrm{x}_1, \cdots, \mathrm{x}_{J-1})$, such that, $j = 0, 1, \cdots, J-1$ and $\mathrm{x}_j \in \{1, 2, \cdots, V\}$, that minimises the costs associated with the use of vehicles to fulfil demands. Table 2 shows the decision problem variables.

$$min f^2(\mathrm{x}_{VAP}) = \sum_{v=1}^{V} \sum_{j=0}^{J-1} \mathrm{C}_{vj} \cdot z_{vj} + \sum_{v=1}^{V} C_v \tag{8}$$

Subject to:

$$\sum_{j=0}^{J-1} z_{vj} \geq 0 \quad \forall v = 1, 2, \cdots V \tag{9}$$

$$\sum_{j=0}^{J-1} d_j \cdot z_{vj} \leq r_v \quad \forall v = 1, 2, \cdots V \tag{10}$$

$$z_{vj} \in \{0, 1\} \quad \forall j, v \tag{11}$$

Constraints 9 ensures that each retailer is assigned to exactly one vehicle, Constraints 10 ensures that the total units of demands of retailers assigned to each vehicle does not exceed the vehicle's capacity and Constraints 11 defines the decision variables.

Table 2: VAP Decision Variables

Var	Description
C_v	Fixed cost of vehicle usage - drivers' salaries, highway tolls, insurance and license fees. Costs are in monetary units per tour.
C_{vj}	Variable costs of vehicle usage and routes - fuel, maintenance and routing costs. Costs are measured in monetary units.
V	Number of vehicles
J	Number of retailers where $j = 0, 1, \cdots, J-1$
d_j	Average demand of retailer j
r_v	Capacity of vehicle v
z_{vj}	Binary variable that indicates 1 if a vehicle v is assigned to fulfil retailer j and 0 otherwise

3.2.3 Multiple Travelling Salesman Problem (MTSP). Travelling salesman problem (TSP) is one of the famous classical optimisation problems which determines a tour that minimises the total distance traveled by a salesman [28]. Multiple Travelling Salesman Problem (MTSP) is an extension of TSP where, a set of routes is assigned to m salesmen relating to J cities. Each city must be included in exactly a tour while requiring a minimum total traveled distance [31][34]. Let $L = l_0, l_1, \cdots, l_{J-1}$ be a set of cities to be visited. Each $l_j \in [a_j, b_j]$ is defined by coordinates in a 2-dimensional plane. The distance $\mathbf{d}^t_{ji}$ between two cities in sub-tour t is an Euclidean distance and the objective function, $f^3(\mathbf{x}_{MTSP})$, seeks to minimise the total distance traveled on T sub-tours and $\mathbf{x}_{MTSP} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T)$, such that, $\mathbf{x}_t \subset L$ and $\emptyset = \bigcap_{t=1}^{T} \mathbf{x}_t$. We express $f^3(\mathbf{x}_{MTSP})$ in Equation 12:

Table 3: MTSP Decision Variables

Var	Description
T	Number of tours where $t = 1, 2, \cdots, T$
J	Number of retailers/Customers where $j = 0, 1, \cdots, J-1$
$\mathbf{d}^t_{ij}$	Travel distance in kilometers from retailer's location i to location j
$\mathbf{y}^t_{ij}$	Binary variable indicates 1 if salesman visits location j from location i in tour t and 0 otherwise
z^t_{kj}	Binary variable indicates 1 if salesman visits location j from a starting point k in tour t and 0 otherwise
$\mathbf{z}^t_{jk}$	Binary variable indicates 1 if salesman returns to the starting point from location j in tour t and 0 otherwise

$$min f^3(\mathrm{x}_{MTSP}) = \sum_{t=1}^{T}\sum_{i=0}^{J-1}\sum_{j=0,i\neq j}^{J-1} \mathrm{d}^t_{ij}\mathrm{y}^t_{ij} + \sum_{k=1}^{K}\sum_{t=1}^{T}\sum_{j=0}^{J-1}(\mathrm{d}^t_{kj}\mathrm{z}^t_{kj} + \mathrm{d}^t_{jk}\mathrm{z}^t_{jk}) \tag{12}$$

Subject to:

$$\sum_{i=0}^{J-1}\sum_{j=0,i\neq j}^{J-1} \mathrm{y}^t_{ij} \geq 0 \quad \forall t = 1, 2, \cdots T \tag{13}$$

$$\sum_{k=1}^{K}\sum_{j=0}^{J-1} \mathrm{z}^t_{kj} = 1 \quad \forall t = 1, 2, \cdots T \tag{14}$$

$$\sum_{k=1}^{K}\sum_{j=0}^{J-1} \mathrm{z}^t_{jk} = 1 \quad \forall t = 1, 2, \cdots T \tag{15}$$

$$\mathrm{y}^t_{ij}, \mathrm{z}^t_{kj}, \mathrm{z}^t_{jk} \in \{0, 1\} \quad \forall j, t, k \tag{16}$$

Constraint 13 allows a single visit and returns back to the starting location. Constraints 14 - 15 ensures that a visit from a starting point and a visit back to the starting point occurs once per tour. Constraints 16 defines the decision variables.

3.2.4 IWSP-VAP-MTSP. The solution variables are used in linking the inventory-warehouse planning, vehicle assignment and the routing problems. The linkages are achieved by injecting the solution variables x_{IWSP} from IWSP as input for VAP and then use the solution variables $\mathrm{x}_{VAP\{x_{IWSP}\}}$ from VAP as input for MTSP. We also account for the dependencies among the cost functions of the models simultaneously. We replaced the transportation cost $\sum_{k=0}^{K-1}\sum_{j=0}^{J-1} c_{kj}x_k y_{kj}$ in $f^1(x_{IWSP})$ with the cost function obtained in $f^2(\mathrm{x}_{VAP})$ as the total costs associated with using the vehicles. The cost function for VAP depends on the solution variable $\mathbf{x}_{MTSP\{\mathrm{x}_{VAP}\}}$ obtained for MTSP to compute the variable costs included in the total vehicle usage costs in VAP. The mathematical formulation of IWSP-VAP-MTSP is modeled in Equation 17.

$$\begin{cases} min f^1_{\mathrm{x}_{VAP}}(x_{IWSP}) = \sum_{k=0}^{K-1}\alpha_k x_k + \sum_{k=0}^{K-1}\sqrt{2\cdot D_k\cdot OC_k\cdot HC_k} \\ \quad + \sum_{k=0}^{K-1} Z_p\cdot HC_k\cdot\sqrt{LT_k}\cdot\sqrt{\pi_k} + f^2_{\mathbf{x}_{MTSP}}(\mathrm{x}_{VAP\{x_{IWSP}\}}) \\ min f^2_{\mathbf{x}_{MTSP}}(\mathrm{x}_{VAP\{x_{IWSP}\}}) = \sum_{k=0}^{K-1}\sum_{v=1}^{V}\sum_{j=0}^{J-1}\mathrm{C}_{kvj}\cdot z_{kvj}x_k + \sum_{v=1}^{V} C_v \\ min f^3(\mathbf{x}_{MTSP\{\mathrm{x}_{VAP}\}}) = \sum_{k=0}^{K-1}\sum_{v=1}^{V}\sum_{i=0}^{J-1}\sum_{j=0,i\neq j}^{J-1}\mathbf{d}^v_{kij}\mathrm{y}^v_{kij}x_k \\ \quad + \sum_{k=1}^{K}(\sum_{v=1}^{V}\sum_{j=0}^{J-1}(\mathbf{d}^v_{kj}\mathrm{z}^v_{kj} + \mathbf{d}^v_{jk}\mathrm{z}^v_{jk}))x_k \end{cases} \tag{17}$$

Subject to:

$$v = t \quad \forall t = 0, 1, \cdots, T, \forall v = 1, 2, \cdots V \tag{18}$$

$$\mathrm{C}_{kvj} = \sum_{j=0,i\neq j}^{J-1}\mathbf{c}^v_{kij}\mathbf{d}^v_{kij}\mathrm{y}^v_{kij}x_k \quad \forall i = 0, 1, \cdots, J-1, \forall v = 1, 2, \cdots, V, \forall k = 0, 1, \cdots K-1 \tag{19}$$

$$\mathrm{C}_{kvj} = \sum_{j=0}^{J-1}\mathbf{c}^v_{kij}\mathbf{d}^v_{kj}\mathrm{z}^v_{kj}x_k \quad \forall v = 1, 2, \cdots, V, \forall k = 0, 1, \cdots, K-1 \tag{20}$$

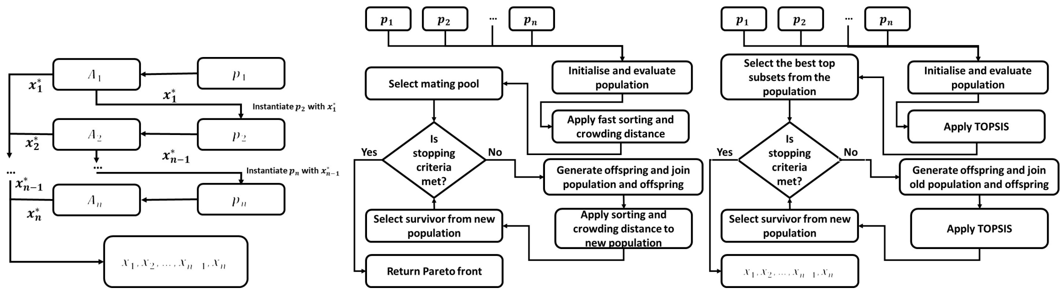
$$\mathbf{c}^v_{jk} = 0 \quad \forall j = 0, 1, \cdots J-1, \forall v = 1, 2, \cdots, V, \forall k = 0, 1, \cdots K-1 \tag{21}$$

$$z_{kvj}, \mathrm{y}^v_{kij}, \mathrm{z}^v_{kj}, \mathrm{z}^v_{jk} \in \{0, 1\} \quad \forall j, v, k \tag{22}$$

Constraints 3 - 7, 9 - 10 and 13 - 14 are used. Constraints 18 ensures that a tour t corresponds to a vehicle v. Constraints 19 - Constraints 20 calculate the variable cost of using vehicle v in fulfilling the demand of retailer j. $\mathbf{c}^v_{kij}$ represents variable cost (in monetary units per kilometer) of fulfilling retailer's j demands by vehicle v from warehouse k or retailer i. Constraints 21 ensures that no monetary cost is incurred on vehicle v returning back to the starting point. Constraints 22 defines the decision variables.

4 PROPOSED APPROACH

We consider three algorithmic frameworks discussed in [25] and [26]. These are Sequential approach, NSGALP and MCRGALP. Our rationale is to test these approaches on more complex linked optimisation problems in order to provide a systematic basis for a holistic study of interconnected or multiple optimisation problems. The sequential approach uses an optimal solution of the root problem as seen in Figure 3a and feeds it to the next problem as input to produce a solution. The sequential approach uses a hierarchical process. In a sequential approach, an algorithm is assigned to each



(a) Sequential Approach (b) NSGALP (c) MCRGALP

Figure 3: Algorithmic Approaches

problem in the linked structure. In this study, we consider an exact solution for IWSP using Gurobi solver [10] before solving the other problems with respective genetic algorithms. We adapt jMetal's source code [22] to the linked problem framework. Figures 3b and 3c depict generic concatenation methodology. The algorithm takes in all the problems and embeds generic solution representations. The algorithm also embeds a linked implementation concept that allows the interaction between the solutions of the related problems. During the interaction, a solution of the principal problem is fed to instantiate a dependent or set of dependent problems. See [26] for more details about the generic methodologies.

4.1 Genetic Components for IWSP-VAP-MTSP

The sequential algorithmic approach considers an exact solution based on Gurobi optimiser for the IWSP and two genetic algorithms, each for VAP and MTSP respectively. The Gurobi solver finds an exact solution in the sequential process for the IWSP and then feeds the solution to the next algorithm to allow for problem instantiation. The three problems have three different solution representations, and that uniquely differentiates the algorithms in terms of encoding and genetic operators used by each algorithm. In NSGALP and MCRGALP, we embed the different encodings and the genetic operators in a single algorithmic process.

4.2 Encoding

We use binary-based encoding for IWSP, integer encoding for VAP and permutation-based encoding for MTSP. The binary-based solution representation addresses two issues; warehouse selection and demand assignments to selected warehouses. The integer-based encoding assigns vehicles to selected warehouses and each vehicle is assigned to set of demands which is restricted to the demands associated to the warehouse a vehicle is assigned to. The permutation-based mechanism addresses a sequence of travel by each vehicle.

4.3 Initialisation

Initialisation is carried out by randomly generating a population of size N. In the sequential approach, the Gurobi solver produces an exact solution and feeds it to the next algorithm. The individual genetic algorithm in the sequential approach generate their population separately and apply the genetic search process to the population. This is quite different in NSGALP and MCRGALP. In Figures 3b and 3c, each approach randomly generates solutions for IWSP, VAP and MTSP respectively. After that, the three respective solutions are combined and the process is repeated to generate a population of combined solutions.

4.4 Genetic Operators

The genetic framework uses the same crossover and mutation operators for all three approaches, see Table 4. In the sequential approach, we do not require a genetic operator for the IWSP problem since we relied on an exact solution provided by the Gurobi solver [10]. For VAP, an integer-coded genetic algorithm is used which adopts integer SBX crossover and integer polynomial mutation operators to generate offspring. For MTSP, we use a permutation-coded genetic algorithm which utilises PMX and permutation swap mutation to update solutions. A tournament selection is employed for individual genetic algorithms in the sequential approach.

The procedures used for offspring generation are the same in NSGALP and MCRGALP. The procedure is outlined as follows; Generate offspring of IWSP solutions from mating pool using HUX crossover and BitFlip mutation operators. For each offspring generated, instantiate problem VAP and adopt integer SBX crossover and integer polynomial mutation operators. Then, use each offspring of VAP corresponding to each IWSP offspring to instantiate problem MTSP and randomly generate MTSP solutions. Next, perform crossover and mutation operations on the solutions of MTSP and generate offspring. Last, combine corresponding offspring of IWSP, VAP and MTSP together. We use crowded binary tournament selection for NSGALP. The crowded binary tournament operator is a modified version of the binary tournament selector that incorporates ranking and diversity using fast sort algorithm [7] and crowding distance computation [21]. MCRGALP embedded a multi-criteria algorithm (TOPSIS) [13] as a tournament selection operator to score each joint solution in a population to select a minimum of two parent pairs for mating. TOPSIS source code was obtained from [20] and adapted for the linked optimisation framework.

5 NUMERICAL EXPERIMENTS

We performed series of computational experiments to evaluate the performance of the proposed algorithmic approaches. The experiments are conducted on the same computer environment with Intel Core i9, 2.4GHz, 32GB RAM, and Windows 10 Enterprise OS. The three algorithmic approaches are implemented in Java. See [24].

5.1 Problems Instances

We generate 960 single period instances of the IWSP-VAP-MTSP problem inspired from the ILRP instances in Guerrero et al [9]. Features include: customers, warehouse and vehicles. Each instance is named as character 'p' followed by the corresponding problem size, number of vehicles, number of warehouses, a percentage of service level and the i-th number over 20 different instances. From [9], we consider Normal distribution for each level of demand d_j as $N[\mu_j, \sigma_j]$. μ_j where $\mu_j \in [5, 15]$ and $\sigma_j \in [0, 5]$. The coordinates for each warehouse and customer are randomly generated based on 100×100 square size. The distance between locations is determined by Euclidean distance metric. For each warehouse, fixed cost is randomly selected using Normal distribution with parameter set $\{(1000, 20), (5000, 100), (8000, 300)\}$. Warehouse capacity is determined by random selection from interval $[D/3, D]$ where $D = \sum_{j \in J} d_j$. For each vehicle, capacity is computed as $5 * b$ where b is a random value in the interval $[3, 15]$. The cost of using the vehicle is randomly selected from a set $\{350, 1000, 5000\}$. For inventory, the holding cost is randomly generated from interval $[0.03, 0.50]$, ordering cost from set $\{100, 250, 500\}$ and a range of 7 days lead time and service levels from set $\{50\%, 60\%, 70\%, 80\%, 90\%\}$. The benchmark set is available on request.

5.2 Experimental Settings

Table 4 contains fixed parameter settings which allow us to measure the behaviour of our approaches. The termination criterion is set to 5000 fitness evaluations for all algorithmic approaches. Each comparative algorithm was executed over 20 independent runs. We randomly selected a range of values for each problem instance between 0.5 - 0.9 for the crossover rate and 0.1 - 0.5 for mutation rate as used in [1] [4] [8] [11] [12] [15] [16] [17]. For the TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) method adopted in the MCRGALP approach, we set equal weightings for the fitness values of each sub-problem to 0.25. This is to ensure that the sub-problems have equal level of importance to maintain equal compromise. IWSP contains two constraints which are weighted at 0.05 each and VAP constraint is set to 0.10.

Table 4: Parameter Settings

Parameters	NSGALP	MCRGALP	SEQUENTIAL		
			A_1	A_2	A_3
No. of Run	20	20	-	20	20
Pop. Size	20	50	-	50	50
Max Eval.	5000	5000	-	5000	5000
Mating Pool Size	50	50	-	-	
Offspring Pop. Size	50	50	-	-	-
HUX	R[0.5 - 0.9]	R[0.5 - 0.9]	-	-	-
Integer SBX	R[0.5 - 0.9]	R[0.5 - 0.9]	-	R[0.5 - 0.9]	-
PMX	R[0.5 - 0.9]	R[0.5 - 0.9]	-	-	R[0.5 - 0.9]
BitFlip	R[0.1 - 0.5]	R[0.1 - 0.5]	-	-	-
Integer Polynomial	1.0	1.0	-	1.0	-
Perm. Swap	R[0.1 - 0.5]	R[0.1 - 0.5]	-	-	R[0.1 - 0.5]

5.3 Performance Measures

We consider four performance measures used in [26]. They include; hypervolume (HV) [37], relative hypervolume (RHV), inverted generational distance (IGD)[3] and multiplicative epsilon [38]. For each problem instance, we obtained a reference point $\mathbf{r}$ and a reference front $\mathbf{Z}$ as input parameters for metric computations.

5.3.1 Relative Hypervolume RHV. RHV measures the proportion of hypervolume achieved by individual approach. This is computed by dividing the hypervolume of approximations by individual approach by the hypervolume of the true Pareto front. A higher RHV indicates that approximations are closer to the true Pareto front.

$$RHV(\mathbf{Z}, A) = \frac{HV(A, \mathbf{r})}{HV(\mathbf{Z}, \mathbf{r})} \tag{23}$$

where $0 \leq RHV(\mathbf{Z}, A) \leq 1$

5.3.2 Hypervolume HV. HV considers the volume of the objective space dominated by an approximation set [36] bounded by a given reference point $\mathbf{r} \in \mathbb{R}^2$. Higher HV values indicate a better performance of the corresponding approaches.

5.3.3 Inverted Generational Distance IGD. IGD assesses the quality of approximations achieved by an algorithm to the Pareto front[3] by measuring how the approximations converge towards the true Pareto front. The smaller the IGD value, the closer the calculated front to the true Pareto front[19]. IGD is calculated as follows:

$$IGD(A, \mathbf{Z}) = \Big(\frac{1}{|\mathbf{Z}|} \sum_{i=1}^{|\mathbf{Z}|} \min_{a \in A} \mathbf{d}(\mathbf{z}, a)^2\Big)^{\frac{1}{2}} \tag{24}$$

where $\mathbf{d}(\mathbf{z}, a) = \sqrt{\sum_{\iota}^{n} (\mathbf{z}_\iota, a_\iota)}$ with a_ι being the ιth fitness value of point a from the approximations A and $\mathbf{z}_\iota$ being an ιth fitness value of point $\mathbf{z}$ from the true Pareto front $\mathbf{Z}$.

5.3.4 Multiplicative Epsilon ϵ. The Epsilon indicator gives a factor by which an approximation set is worse than another with respect to all objectives [38]. A lower Epsilon value corresponds to a better approximation set, regardless of the problem type.

$$epsilon(A, \mathbf{Z}) = \max_{\mathbf{z} \in \mathbf{Z}} \min_{a \in A} \max_{1 \leq \iota \leq n} epsilon(a_\iota, \mathbf{z}_\iota) \tag{25}$$

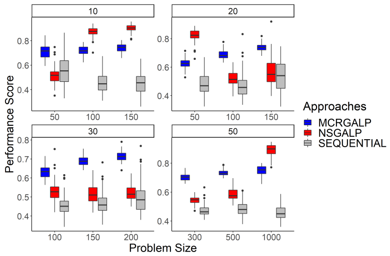
where $epsilon(a_\iota, \mathbf{z}_\iota) = a_\iota / \mathbf{z}_\iota$

5.4 Experimental Results and Analysis

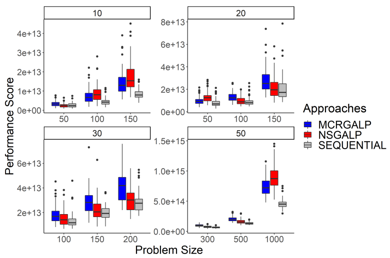
We consider the performance of our approach based on performance metrics and computational time. Furthermore, we explore specifically the sensitivity of our solutions to varied parameters.

5.4.1 Performance Metrics. Figure 4 shows the overall performance of our algorithmic approaches. Each sub-figure is partitioned by the number of warehouses and grouped by problem size. In Figure 4a, according to the RHV metric, the MCRGALP method mostly outshines both the NSGALP and the sequential approach, however, NSGALP performs better with most of the problem instances involving 10 warehouses. Similar level of performance is demonstrated in HV, IGD and Epsilon in Figures 4b, 4c and 4d respectively.

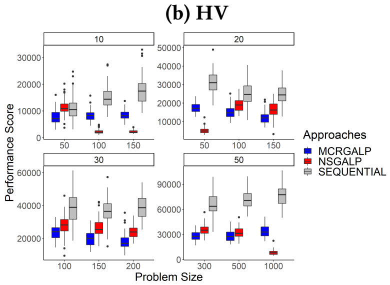
The approximations predicted by MCRGALP are closer to the Pareto front. This explains the quality of joint solutions selected by the MCRGALP. The NSGALP also shows prospect in the quality of combined solutions selection but not as good compared to the MCRGALP. The joint solutions selected by MCRGALP are more



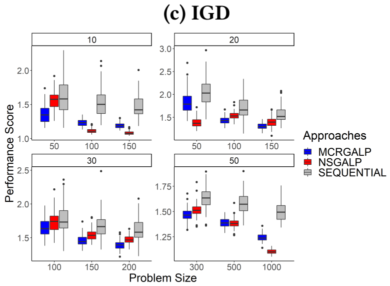
(a) RHV



(b) HV



(c) IGD



(d) Epsilon

Figure 4: Overall Performance of Algorithmic Approaches

dominant as explained empirically by a problem instance in Figure 5b. For the sequential approach, the algorithm is biased towards the first problem (i.e., IWSP), thereby sacrificing others due to the importance it places on the first problem for optimal solution. So, the sequential approach explores joint solution that converges towards the region of the first problem with minimal consideration towards other problems' search spaces. This performance is evident in figures 5a and 5b on how the solutions predicted by the algorithms have converged towards the Pareto front. MCRGALP has demonstrated its ability to select high quality joint solutions involving multiple problems without sacrificing one problem for another specifically, if individual problems are equally important.

We test the significant differences in algorithmic performance using Wilcoxon signed-rank at 0.05 significant level. Table 5 summarizes the corresponding p-values among the paired algorithms on instances grouped by problem size and number of warehouses. The highlighted bold fonts indicate that statistically, there are no significant differences in algorithmic performance. Table 5 shows huge significant differences in performance across the metrics. However, there are some exceptional cases where performance between paired algorithms are significantly similar. We have three exceptional cases in Epsilon, five in HV, one in IGD and RHV respectively. For instance, for problem size 50 involving 10 warehouses, the NSGALP and Sequential appeared to have significantly performed in a similar way across Epsilon, HV and IGD metrics.

5.4.2 Computational Time [ms]. We compare the computational time required by each algorithmic approach to solve the problem instances. This is shown in Figure 5c. Obviously, the Sequential requires less amount of time to obtain joint solutions while the NSGALP requires more time than other approaches.

Overall, the three algorithmic approaches have shown promising performance, however, MCRGALP is more exceptional. MCRGALP delivers high quality joint solutions capable of producing optimal overall solution. The level of performance is subjective and open to different interpretations. The importance placed on each decision depends on the perspective of supply chain decision makers.

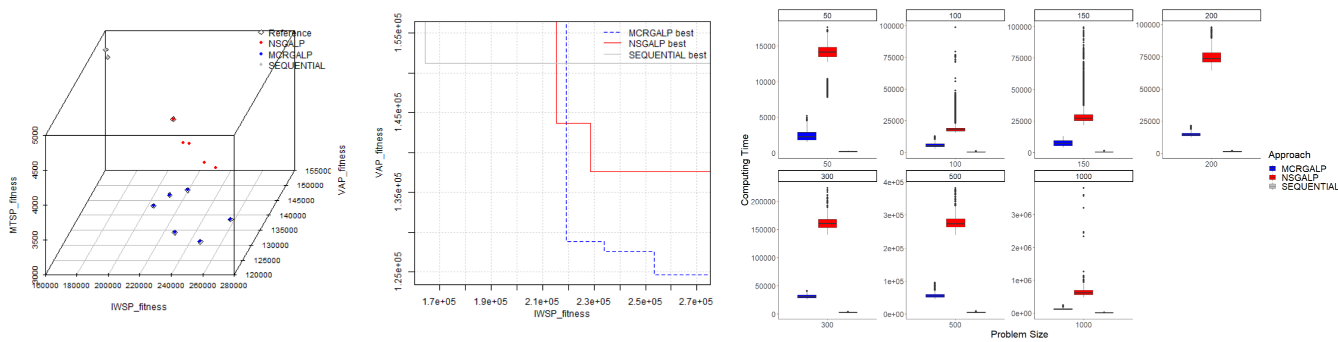
5.4.3 Sensitivity Analysis. We use generalised additive statistical model to further analyse the sensitivity of IWSP caused by the algorithmic behaviour to changes to some predictors (i.e., VAP objective function, MTSP objective function, crossover rate, and mutation rate). Figure 6 shows examples of algorithmic behaviour for predicting the objective function of IWSP in respect to changes in total vehicle cost, total traveling distance, crossover rate and mutation rate. For NSGALP and MCRGALP, changes in the respective objective functions of VAP and MTSP result to non-linear sensitivities in the IWSP objective function with variability of up to 40%. However, the Sequential method behaved indifferently in predicting the objective function of IWSP when changes are applied to the probability rates (i.e., crossover and mutation rates) compared to NSGALP and MCRGALP respectively.

6 CONCLUSION AND FUTURE WORK

This paper develops a novel linked problem benchmark (IWSP-VAP-MTSP) integrating Inventory Warehouse Selection Problem, Vehicle Assignment Problem and Multiple Traveling Salesmen Problem. Problem represents tactical and operational supply chain decision problems that arise in inventory location and routing. This study offers two contributions; creating benchmark problems by linking existing benchmark sets, and adopting variants of GAs as holistic solution approaches for complex supply chains. We generated 960 randomised instances of IWSP-VAP-MTSP to test and statistically compare the performance of the holistic solution approaches. Empirical results show that MCRGALP outperforms the other two methods based on the performance metrics used. In addition, we test for significant difference in algorithmic performance at 5% significance level. Empirical test shows huge significant differences in performance between the algorithms across the metrics but with

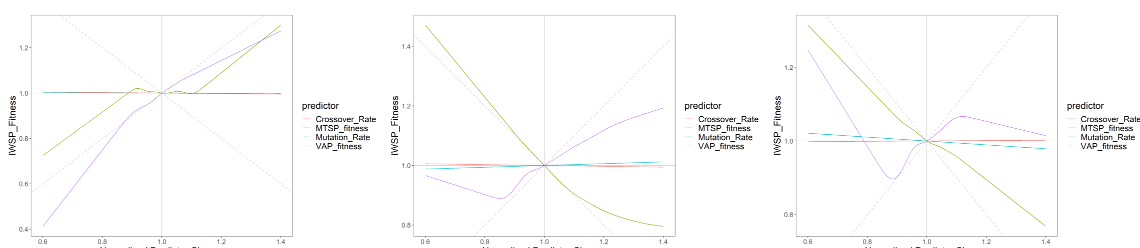
Table 5: Metrics Performance Test - p-values

Psize	K	Epsilon MCRGA Vs SEQ	Epsilon NSGALP Vs MCRGA	Epsilon NSGALP Vs SEQ	HV MCRGA Vs SEQ	HV NSGALP Vs MCRGA	HV NSGALP Vs SEQ	IGD MCRGA Vs SEQ	IGD NSGALP Vs MCRGA	IGD NSGALP Vs SEQ	RHV MCRGA Vs SEQ	RHV NSGALP Vs MCRGA	RHV NSGALP Vs SEQ
50	10	1.73E-10	1.63E-14	**4.87E-01**	5.23E-04	3.06E-06	**3.18E-01**	1.70E-07	5.92E-13	**2.37E-01**	1.19E-16	1.13E-25	5.06E-03
50	20	2.05E-06	1.63E-24	1.41E-26	1.75E-05	8.12E-06	4.95E-14	2.83E-26	9.39E-28	9.39E-28	4.04E-19	1.53E-27	1.01E-27
100	10	1.69E-24	2.96E-24	1.05E-27	2.00E-09	1.36E-02	1.55E-14	1.06E-21	1.09E-27	9.39E-28	1.42E-27	7.28E-27	9.39E-28
100	20	5.13E-20	5.35E-13	6.00E-09	1.53E-11	8.99E-07	**5.41E-02**	1.54E-19	1.13E-10	2.76E-10	6.97E-25	1.05E-27	4.06E-07
100	30	4.10E-04	5.86E-04	**7.60E-01**	2.34E-08	4.95E-03	7.35E-03	5.82E-21	3.05E-07	3.40E-12	9.09E-27	4.69E-23	1.34E-15
150	10	1.22E-26	2.19E-26	9.39E-28	9.06E-15	1.72E-03	1.29E-21	7.56E-26	9.39E-28	9.39E-28	9.39E-28	9.39E-28	9.39E-28
150	20	4.38E-23	2.96E-08	3.15E-11	1.61E-07	4.57E-05	**2.38E-01**	9.74E-26	4.60E-09	8.86E-18	7.56E-27	3.29E-20	**1.13E-01**
150	30	4.03E-15	2.37E-10	3.71E-07	9.99E-13	3.05E-07	**8.96E-02**	3.40E-24	1.05E-11	1.00E-16	3.47E-27	9.75E-28	4.12E-08
200	30	6.69E-17	2.39E-11	1.49E-09	1.13E-12	5.42E-09	**9.55E-02**	7.56E-27	9.56E-15	2.48E-25	6.54E-26	9.39E-28	2.24E-04
300	50	1.98E-19	2.30E-05	1.72E-14	1.11E-20	1.16E-10	6.77E-05	1.77E-27	3.40E-12	1.31E-26	9.39E-28	9.39E-28	4.21E-21
500	50	1.23E-23	**7.39E-01**	1.93E-23	4.20E-24	2.58E-10	4.32E-09	9.39E-28	3.25E-05	9.39E-28	9.39E-28	9.75E-28	3.32E-22
1000	50	1.01E-27	1.05E-27	9.39E-28	4.55E-25	4.29E-07	5.03E-27	1.01E-27	9.75E-28	9.39E-28	9.39E-28	3.22E-27	9.39E-28



(a) Fronts **(b) Empirical Attainment Function** **(c) Computing Time [ms] of Algorithmic Approaches**

Figure 5: Algorithmic Performance



(a) SEQUENTIAL **(b) NSGALP** **(c) MCRGALP**

Figure 6: Sensitivity Analysis on Problem Size 1000

the exception of some cases in Epsilon and HV metrics. We also consider the computing time of our approaches and discover that the Sequential approach requires relatively less time than NSGALP and MCRGALP to obtain an optimal joint solutions. For future work, MCRGALP and NSGASLP results in sacrificing much computational time in achieving quality joint solutions. It would be interesting to explore some algorithmic properties to further improve efficiency.

Also, one of the limitations not included in the study is the idea of a governing authority (such as environmental sustainability) with holistic goals derived from the solutions adopted by different units across the chain. For future interest, the consideration of holistic measures will provide an intellectual method for analysing external features underlying a holistic optimisation and by defining the external control in terms of the internal objectives.

REFERENCES

[1] Reginald Ankrah, Olivier Regnier-Coudert, John McCall, Anthony Conway, and Andrew Hardwick. 2018. Performance analysis of ga and pbil variants for real-world location-allocation problems. In *2018 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 1–8.

[2] Claudia Archetti and Martin Savelsbergh. 2009. The trip scheduling problem. *Transportation Science* 43, 4 (2009), 417–431.

[3] Leonardo CT Bezerra, Manuel López-Ibánez, and Thomas Stützle. 2017. An empirical assessment of the properties of inverted generational distance on multi- and many-objective optimization. In *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 31–45.

[4] Gladys Bonilla-Enriquez and Santiago-Omar Caballero-Morales. 2021. Search Algorithms on Logistic and Manufacturing Problems. (2021).

[5] Mohammad Reza Bonyadi and Zbigniew Michalewicz. 2016. Evolutionary computation for real-world problems. *Challenges in Computational Statistics and Data Mining* (2016), 1–24.

[6] Fatih Camci. 2014. The travelling maintainer problem: integration of condition-based maintenance with the travelling salesman problem. *Journal of the Operational Research Society* 65, 9 (2014), 1423–1436.

[7] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation* 6, 2 (2002), 182–197.

[8] Jian Gao, Rong Chen, and Wu Deng. 2013. An efficient tabu search algorithm for the distributed permutation flowshop scheduling problem. *International Journal of Production Research* 51, 3 (2013), 641–651.

[9] William Javier Guerrero, Caroline Prodhon, Nubia Velasco, and Ciro A Amaya. 2013. Hybrid heuristic for the inventory location-routing problem with deterministic demand. *International Journal of Production Economics* 146, 1 (2013), 359–370.

[10] Gurobi Optimization, LLC. 2023. Gurobi Optimizer Reference Manual. https://www.gurobi.com

[11] Irina Harris, Christine Mumford, and Mohamed Naim. 2009. The multi-objective uncapacitated facility location problem for green logistics. In *2009 IEEE Congress on Evolutionary Computation*. IEEE, 2732–2739.

[12] Jiangping Huang, Quanke Pan, and Qingda Chen. 2019. A hybrid genetic algorithm for the distributed permutation flowshop scheduling problem with sequence-dependent setup times. In *IOP Conference Series: Materials Science and Engineering*, Vol. 646. IOP Publishing, 012037.

[13] Ching-Lai Hwang and Kwangsun Yoon. 1981. Methods for multiple attribute decision making. In *Multiple attribute decision making*. Springer, 58–191.

[14] Maksud Ibrahimov. 2012. *Evolutionary algorithms for supply chain optimisation.* Ph. D. Dissertation.

[15] Abdullah Konak, Sadan Kulturel-Konak, and Lawrence Snyder. 2017. A multi-objective approach to the competitive facility location problem. *Procedia Computer Science* 108 (2017), 1434–1442.

[16] Chao Lu, Liang Gao, Wenyin Gong, Chengyu Hu, Xuesong Yan, and Xinyu Li. 2021. Sustainable scheduling of distributed permutation flow-shop with non-identical factory using a knowledge-based multi-objective memetic optimization algorithm. *Swarm and Evolutionary Computation* 60 (2021), 100803.

[17] Chao Lu, Yuanxiang Huang, Leilei Meng, Liang Gao, Biao Zhang, and Jiajun Zhou. 2022. A Pareto-based collaborative multi-objective optimization algorithm for energy-efficient scheduling of distributed permutation flow-shop with limited buffers. *Robotics and Computer-Integrated Manufacturing* 74 (2022), 102277.

[18] Sedigheh Mahdavi, Mohammad Ebrahim Shiri, and Shahryar Rahnamayan. 2015. Metaheuristics in large-scale global continues optimization: A survey. *Information Sciences* 295 (2015), 407–428.

[19] Jamie A Manson, Thomas W Chamberlain, and Richard A Bourne. 2021. MVMOO: Mixed variable multi-objective optimisation. *Journal of Global Optimization* 80, 4 (2021), 865–886.

[20] Danielle Martin. 2019. Github-topsis source. https://github.com/danigpam/topsis/tree/master/src/main/java/mcdm/topsis

[21] Pranab K Muhuri, Zubair Ashraf, and QM Danish Lohani. 2017. Multiobjective reliability redundancy allocation problem with interval type-2 fuzzy uncertainty. *IEEE Transactions on Fuzzy Systems* 26, 3 (2017), 1339–1355.

[22] AJ Nebro and JJ Durillo. 2014. Github-jmetal source. https://github.com/jMetal/jMetal

[23] Akinola Ogunsemi. 2022. *Holistic, data-driven, service and supply chain optimisation: linked optimisation.* Ph. D. Dissertation.

[24] A Ogunsemi. 2024. Github-LinkedProblemPramework. https://github.com/acson35/LinkedProblemFramework/tree/master

[25] Akinola Ogunsemi, John McCall, Mathias Kern, Benjamin Lacroix, David Corsar, and Gilbert Owusu. 2022. Facility location problem and permutation flow shop scheduling problem: a linked optimisation problem. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 735–738.

[26] Akinola Ogunsemi, John McCall, Mathias Kern, Benjamin Lacroix, David Corsar, and Gilbert Owusu. 2022. Job Assignment Problem and Traveling Salesman Problem: A Linked Optimisation Problem. In *Artificial Intelligence XXXIX: 42nd SGAI International Conference on Artificial Intelligence, AI 2022, Cambridge, UK, December 13–15, 2022, Proceedings*. Springer, 19–33.

[27] Rodolfo Eleazar Perez Loaiza, Elias Olivares-Benitez, Pablo A Miranda Gonzalez, Aaron Guerrero Campanur, and Jose Luis Martinez Flores. 2017. Supply chain network design with efficiency, location, and inventory policy using a multiobjective evolutionary algorithm. *International Transactions in Operational Research* 24, 1-2 (2017), 251–275.

[28] Gerhard Reinelt. 2003. *The traveling salesman: computational solutions for TSP applications*. Vol. 840. Springer.

[29] Ahmad Sayed Saif-Eddine, Mohammed Mostafa El-Beheiry, and Amin Kamel El-Kharbotly. 2019. An improved genetic algorithm for optimizing total supply chain cost in inventory location routing problem. *Ain Shams Engineering Journal* 10, 1 (2019), 63–76.

[30] Martin Dario Arango Serna, Julian Andres Zapata Cortes, and Daniela Gutierrez Sepulveda. 2015. Modeling the inventory routing problem (IRP) with multiple depots with genetic algorithms. *IEEE Latin America Transactions* 13, 12 (2015), 3959–3965.

[31] Yang Shuai, Shao Yunfeng, and Zhang Kai. 2019. An effective method for solving multiple travelling salesman problem based on NSGA-II. *Systems Science & Control Engineering* 7, 2 (2019), 108–116.

[32] Ioannis P Solos, Ioannis X Tassopoulos, and Grigorios N Beligiannis. 2016. Optimizing shift scheduling for tank trucks using an effective stochastic variable neighbourhood approach. *International Journal of Artificial Intelligence* 14, 1 (2016), 1–26.

[33] Zedong Tang and Maoguo Gong. 2019. Adaptive multifactorial particle swarm optimisation. *CAAI Transactions on Intelligence Technology* 4, 1 (2019), 37–46.

[34] Masoumeh Vali and Khodakaram Salimifard. 2017. A constraint programming approach for solving multiple traveling salesman problem. In *The Sixteenth International Workshop on Constraint Modelling and Reformulation*. 1–17.

[35] Yuan Yuan, Yew-Soon Ong, Abhishek Gupta, Puay Siew Tan, and Hua Xu. 2016. Evolutionary multitasking in permutation-based combinatorial optimization problems: Realization with TSP, QAP, LOP, and JSP. In *2016 IEEE Region 10 Conference (TENCON)*. IEEE, 3157–3164.

[36] Eckart Zitzler and Lothar Thiele. 1998. Multiobjective optimization using evolutionary algorithms—a comparative case study. In *International conference on parallel problem solving from nature*. Springer, 292–301.

[37] Eckart Zitzler and Lothar Thiele. 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE transactions on Evolutionary Computation* 3, 4 (1999), 257–271.

[38] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M Fonseca, and Viviane Grunert Da Fonseca. 2003. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on evolutionary computation* 7, 2 (2003), 117–132.