OGUNSEMI, A., MCCALL, J., KERN, M., LACROIX, B., CORSAR, D. and OWUSU, G. 2022. Facility location problem and permutation flow shop scheduling problem: a linked optimisation problem. In *Fieldsend, J. (ed.) GECCO'22 companion: proceedings of 2022 Genetic and evolutionary computation conference companion, 9-13 July 2022, Boston, USA, [virtual event]*. New York: ACM [online], pages 735-738. Available from: https://doi.org/10.1145/3520304.3529033

Facility location problem and permutation flow shop scheduling problem: a linked optimisation problem.

OGUNSEMI, A., MCCALL, J., KERN, M., LACROIX, B., CORSAR, D. and OWUSU, G.

2022

Facility Location Problem and Permutation Flow shop Scheduling Problem: A Linked Optimisation Problem

Akinola Ogunsemi
The Robert Gordon University
a.ogunsemi@rgu.ac.uk

John McCall
The Robert Gordon University
j.mccall@rgu.ac.uk

Mathias Kern
BT Applied Research
mathias.kern@bt.com

Benjamin Lacroix
The Robert Gordon University
b.m.e.lacroix@rgu.ac.uk

David Corsar
The Robert Gordon University
d.corsar1@rgu.ac.uk

Gilbert Owusu
BT Applied Research
gilbert.owusu@bt.com

ABSTRACT

There is a growing literature spanning several research communities that studies multiple optimisation problems whose solutions interact, thereby leading researchers to consider suitable approaches to joint solution. Real-world problems, like supply chain, are systems characterised by such interactions. A decision made at one point of the supply chain could have significant consequence that ripples through linked production and transportation systems. Such interactions would require complex algorithmic designs. This paper, therefore, investigates the linkages between a facility location and permutation flow shop scheduling problems of a distributed manufacturing system with identical factory (FLPPFSP). We formulate a novel mathematical model from a linked optimisation perspective with objectives of minimising facility cost and makespan. We present three algorithmic approaches in tackling FLPPFSP; Non-dominated Sorting Genetic Algorithm for Linked Problem (NSGALP), Multi-Criteria Ranking Genetic Algorithm for Linked Problem (MCRGALP), and Sequential approach. To understand FLPPFSP linkages, we conduct a pre-assessment by randomly generating 10000 solution pairs on all combined problem instances and compute their respective correlation coefficients. Finally, we conduct experiments to compare results obtained by the selected algorithmic methods on 620 combined problem instances. Empirical results demonstrate that NSGALP outperforms the other two methods based on relative hypervolume, hypervolume and epsilon metrics.

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, Mathias Kern, Benjamin Lacroix, David Corsar, and Gilbert Owusu. 2022. Facility Location Problem and Permutation Flow shop Scheduling Problem: A Linked Optimisation Problem. In

1 INTRODUCTION

Linked optimisation problem explains the concept of joint optimisation task of multiple problems. Real-world problems, like supply chains, are characterised by interactions, where subproblems features are linked [2]. Hence, an optimal solution for each subproblem might not guarantee optimal overall solutions [12].

In distributed manufacturing, all production tasks are shared among multiple factories to achieve economic benefits [8]. An example is distributed permutation flow shop scheduling (DPFSP), an extension of classical permutation flow shop scheduling problem [8], which is applied in fields like petrochemical processing, automobile manufacturing, and cell manufacturing. DPFSP only considers scheduling operation and ignores factory costs. This paper introduces a new variation of DPFSP with factory costs. In this paper, we assess the relationship between FLP and PFSP, and adopt three algorithmic approaches to tackle them.

2 PROBLEM BACKGROUND

[9] first introduces six mixed integer linear programming (MILP) models for DPFSP. Likewise, [4] tackles DPFSP with genetic algorithm. Recently, [8] presents a knowledge-based multi-objective memetic optimisation algorithm to address sustainable DPFSP. Similarly, [14] addresses distributed assembly permutation flow shop scheduling problem with genetic algorithm. Adapting FLP to DPFSP creates a linked optimisation problem with two classical problems (FLP and PFSP). This forms a novel model of facility location problem and permutation flow shop scheduling problem (FLPPFSP).

3 PROBLEM FORMULATION

3.1 Linked Problem Perspective

A linked optimisation problem P of n connected problems is;

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

$$p_\iota = \left\{ x^\iota_{\{x^1_*,\cdots,x^n_*\}\backslash x^\iota_*}, f^\iota_{\{x^1_*,\cdots,x^n_*\}\backslash x^\iota_*}, c^\iota_{\{x^1_*,\cdots,x^n_*\}\backslash x^\iota_*} \right\} \tag{2}$$

In Eq. 2, x^ι_* denotes candidate solution in x^ι. $\{x^1_*, \cdots, x^n_*\}\backslash x^\iota_*$ are feasible solutions for other problems in P which affect p_ι. $x^\iota_{\{x^1_*,\cdots,x^n_*\}\backslash x^\iota_*}$ is a search space of problem p_ι. $f^\iota_{\{x^1_*,\cdots,x^n_*\}\backslash x^\iota_*} : x^\iota \rightarrow \mathbb{R}$ denotes the objective function. $c^{\iota}_{\{x^1_*,\cdots,x^n_*\}\backslash x^{\iota}_*}$ denotes constraints set of p_ι. $D = \{D^X, D^F, D^C\}$ connects the problems in Eq. 3.

$$D^X_{\iota j}/D^F_{\iota j}/D^C_{\iota j} = \begin{cases} 1, & \text{if } x^{\iota}_* \ changes\ x^j/f^j/c^j \\ 0, & Otherwise \end{cases} \tag{3}$$

3.2 Formulation of FLPPFSP

FLP considers subset of facilities to service customers demands. A matrix of size $l{\times}K$ defines cost d_{kj} to fulfil each job. The assignment of job $j \in J$ to a facility k is depicted by y_{kj} where $y_{kj} \in \{0, 1\}$. A fixed cost α_k is incurred when a facility is selected to fulfil a job. FLP seeks x_{FLP}= $\{x_1, \cdots, x_K\} \in \{0, 1\}^K$ that minimises;

$$minf^1(x_{FLP}) = \sum_{k=1}^{K} \alpha_k x_k + \sum_{k=1}^{K}\sum_{j=1}^{l} d_{kj} x_k\, y_{kj} \tag{4}$$

Subject to;

$$\sum_{k=1}^{K}\sum_{j=1}^{l} y_{kj} = 1 \tag{5}$$

$$x_k, y_{kj} \in \{0, 1\} \quad \forall j, k \tag{6}$$

Constraint 5 ensures that each job is fulfilled by one facility. Constraint 6 defines the decision variables. PFSP is defined by $l \times m$ matrix of machine processing times and seeks to find permutation $\mathbf{x}_{PFSP} = (\mathbf{x}_1, \cdots, \mathbf{x}_l)$ that minimises makespan C_{max}.

$$minf^2(x_{PFSP}) = C_{max} \tag{7}$$

FLPPFSP simultaneously minimises the facility cost and makespan. Setup times, preemption or interruption are not considered. FLPPFSP linked model is as follows;

$$\begin{cases} minf^1(x_{FLP}) = \sum_{k=1}^{K} \alpha_k x_k + \sum_{k=1}^{K}\sum_{j=1}^{l} d_{kj} x_k\, y_{kj} \\ minf^2_{x_{FLP}}(x_{PFSP}) = \max\left\{C_{x_k}(x_{PFSP})\right\}_{k=1}^{K} \end{cases} \tag{8}$$

4 PROPOSED APPROACH

4.1 Genetic Components

We use binary based encoding for FLP and permutation based encoding for PFSP. Each algorithm in Algorithm 1 generates its population separately and applies genetic processes on them. In Algorithms 2 and 3, each random solution of FLP instantiates PFSP to generate a corresponding random solution. Half uniform crossover HUX operator is used for FLP solutions and partially mapped crossover PMX for PFSP solutions. BitFlip mutation is used for updating FLP solutions and permutation swap mutation for PFSP solutions. Both algorithms in Algorithm 1 use tournament selection. Offspring generation is the same in Algorithms 2 and 3. First, generate **n** offspring of FLP solutions using HUX and BitFlip operators. Next, instantiate PFSP with each offspring and generate N random solutions for PFSP. Then, perform crossover and mutation operations on N solutions of PFSP and generate **n** offspring. Next, evaluate **n** offspring and select best offspring from **n** offspring of PFSP and pair with each offspring of FLP.

4.2 Sequential Approach

Sequential approach solves FLPPFSP in sequence. This approach solves linked problems in hierarchical structure usually between two decision makers [6] [7]. In Algorithm 1, algorithm A_{FLP} solves problem p_{FLP}, selects the best solution x^*_{FLP} then, instantiates p_{PFSP} once. Next, algorithm A_{PFSP} solves instantiated p_{PFSP} and then select best solution $\mathbf{x}^*_{PFSP}$ and returns best solution pair.

Algorithm 1: *SEQUENTIAL*

$x^*_{FLP} \leftarrow A_{FLP}|p_{FLP}$;
$\mathbf{x}^*_{PFSP} \leftarrow A_{PFSP}|(p_{PFSP}, x^*_{FLP})$;
Result: $(x^*_{FLP}, \mathbf{x}^*_{PFSP})$

4.3 Nondominated Sorting Genetic Algorithm for Linked Problem (NSGALP)

Multi-objective approach adapts NSGA-II framework [3] and considers solutions of FLP and PFSP as a joint solution and their objective functions as bi-objective functions. See Algorithm 2.

Algorithm 2: *NSGALP*

$pop^0_{(FLP,PFSP)} \leftarrow$ Randomly initialise population ;
Fitness evaluation on $pop^0_{(FLP,PFSP)}$;
Assign fast non-dominated sort to $pop^0_{(FLP,PFSP)}$;
Apply crowding-distance assignment to $pop^0_{(FLP,PFSP)}$;
$t \leftarrow 0$;
while *Stopping criterion not met* **do**
  $\mathcal{R}^t_{(FLP,PFSP)} \leftarrow$ Select from $pop^t_{(FLP,PFSP)}$;
  $Q^t_{(FLP,PFSP)} \leftarrow$ Generate offspring from $\mathcal{R}^t_{(FLP,PFSP)}$;
  Fitness evaluation on $Q^t_{(FLP,PFSP)}$;
  $\mathbf{pop}^t_{(FLP,PFSP)} \leftarrow pop^t_{(FLP,PFSP)} \cup Q^t_{(FLP,PFSP)}$;
  Assign fast non-dominated sort to $\mathbf{pop}^t_{(FLP,PFSP)}$;
  Apply crowding-distance assignment to $\mathbf{pop}^t_{(FLP,PFSP)}$;
  $pop^{t+1}_{(FLP,PFSP)} \leftarrow$ Select survivor from $\mathbf{pop}^t_{(FLP,PFSP)}$;
  $t \leftarrow t + 1$;
end
Result: $\mathcal{F}^1_{(FLP,PFSP)}$

4.4 Multi-Criteria Ranking Genetic Algorithm for Linked Problem (MCRGALP)

MCRGALP uses Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) [13] as a comparator for tournament selection in 5 steps [11]. Step 1 normalises decision matrix in Eq. 9.

$$r_{i\iota} = \frac{f^{\iota}_i}{\sum_{i=1}^{|pop^t|}(f^{\iota}_i)^2} \tag{9}$$

Step 2 determines normalized weighted value $v_{i\iota}$ with given weight $w_\iota = (w_1, \cdots, w_n)$ in Eq. 10.

$$v_{i\iota} = r_{i\iota} * w_\iota \tag{10}$$

Step 3 identifies the ideal best solutions v^+_ι and ideal worst solutions v^-_ι in Eq. 11 and Eq. 12.

$$v^+_\iota = \left\{(\max v_{i\iota} | \iota \in I), (\min v_{i\iota} | \iota \in I'), i = 1, \cdots, |pop^t|\right\} = \left\{v^+_1, \cdots, v^+_n\right\} \tag{11}$$

$$v^-_\iota = \left\{(\min v_{i\iota} | \iota \in I), (\max v_{i\iota} | \iota \in I'), i = 1, \cdots, |pop^t|\right\} = \left\{v^-_1, \cdots, v^-_n\right\} \tag{12}$$

Algorithm 3: *MCRGALP*

$pop^0_{(FLP,PFSP)} \leftarrow$ Randomly initialise population ;
Fitness evaluation on $pop^0_{(FLP,PFSP)}$;
Assign score to each solution pair in $pop^t_{(FLP,PFSP)}$;
$t \leftarrow 0$;
while *Stopping criterion not met* **do**
  $pop^*_{(FLP,PFSP)} \leftarrow$ Get best n pairs of $pop^t_{(FLP,PFSP)}$;
  $Q^t_{(FLP,PFSP)} \leftarrow$ Generate offspring from $pop^*_{(FLP,PFSP)}$;
  Fitness evaluation on $Q_{(FLP,PFSP)}$;
  Assign score to each solution pair in $Q_{(FLP,PFSP)}$;
  $pop^t_{(FLP,PFSP)} \leftarrow pop^t_{(FLP,PFSP)} \cup Q^t_{(FLP,PFSP)}$;
  $pop^{t+1}_{(FLP,PFSP)} \leftarrow$ Get top N solution pairs with best score from $pop^t_{(FLP,PFSP)}$;
  $t \leftarrow t + 1$;
end
Result: $(x^*_{FLP}, \mathbf{x}^*_{PFSP})$

Step 4 calculates Euclidean distance from v^+_ι and v^-_ι.

$$S^+_i = \sqrt{\sum_{\iota=1}^{n} (v_{i_\iota} - v^+_\iota)^2} \quad i = 1, 2, \cdots, |pop^t| \tag{13}$$

$$S^-_i = \sqrt{\sum_{\iota=1}^{n} (v_{i_\iota} - v^-_\iota)^2} \quad i = 1, 2, \cdots, |pop^t| \tag{14}$$

Step 5 calculates performance score and ranks the solution pairs.

$$\mathcal{P}_i = \frac{S^-_i}{S^+_i + S^-_i} \; where \quad 0 \le \mathcal{P}_i \le 1 \tag{15}$$

5 EXPERIMENTS

We conduct experiments on the same computer environment with Intel Core i9, 2.4GHz, 32GB RAM, and Windows 10 Enterprise OS.

5.1 Benchmark Problems

We use Beasley [1] and Holmberg [5] FLP benchmarks and Taillard's PFSP benchmark [10]. We combined each FLP instance and PFSP instance and obtained 620 combined instances in total.

5.2 Exploratory Analysis of Problem Linkages

We evaluate 10000 random solutions for FLP. We then, instantiate PFSP for each FLP solution. Next, we evaluate 1000 random solutions for the instantiated PFSP and compute the mean value. Next, for each FLPPFSP instance, we determine the relationship between the sub-problems using Spearman's correlation coefficient. Figure 1 shows inconsistency across all problem instances.

5.3 Performance Metric

We use hypervolume (HV) [15], relative hypervolume RHV and multiplicative epsilon [16] metrics. We use a single reference point to compute HV and Pareto-optimal points for RHV and Epsilon.

5.4 Parameter Settings

Table 1 shows parameters used in our approaches. We set each weight w_ι for each decision criterion for the TOPSIS method to 0.5.

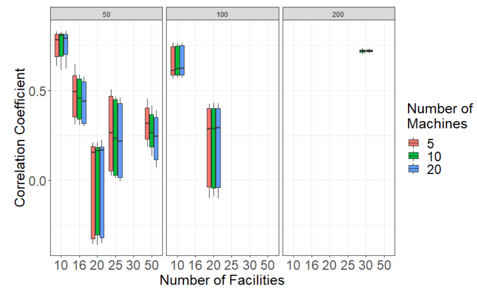


Figure 1: Correlation Analysis of FLP and PFSP

Table 1: Parameter Settings

Parameters	NSGALP	MCRGALP	SEQUENTIAL	
No. of Algorithms	1	1	2	
No. of Independent Run	100	100	100	100
Population Size	100	100	100	100
Max Evaluations	10000	10000	10000	10000
Mating Pool Size	100	100	-	-
Offspring Population Size	100	100	-	-
HUXCrossover	0.9	0.9	0.9	-
PMXCrossover	0.1	0.1	-	0.1
BitFlipMutation	0.8	0.8	0.8	-
PermutationSwapMutation	0.5	0.5	-	0.5

5.5 Experimental Results and Analysis

Table 2 shows the mean scores with best values highlighted in bold font. NSGALP significantly outperforms other approaches in RHV and epsilon but scores are relatively close in HV. Metrics are more biased towards NSGALP and are largely influenced by non-dominated points. Figure 2 shows three plots of mean computing time (milliseconds) against the performance metrics. The sequential approach achieved less computing time than the two other approaches. Furthermore, we consider performance in terms

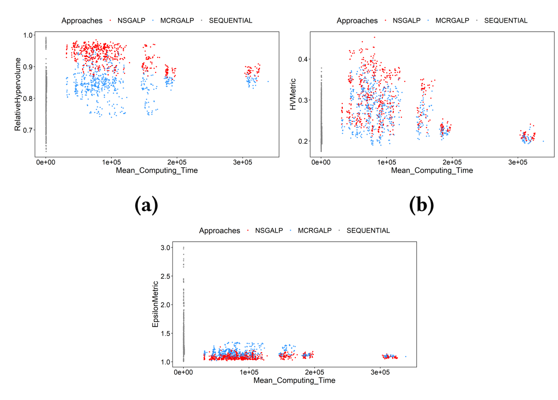


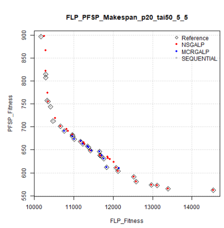
Figure 2: Mean Computing Time vs Performance Metrics

of correlation scores in Section 5.2. Figure 3a shows algorithmic performance based on instance with lowest correlation score (−0.26). Sequential perform poorly with high level of trade-off of makespan for factory cost. In Figure 3b, sequential approach produces the best Pareto approximation set on problem instance with median correlation score (0.4928). Figure 3c shows performance on problem

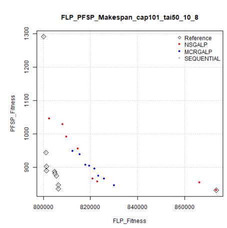
Table 2: Mean values of relative hypervolume, hypervolume and epsilon metrics of MCRGALP, NSGALP and SEQUENTIAL

Problem size	F	m	Relative Hypervolume			Hypervolume			Epsilon		
			MCRGALP	NSGALP	SEQUENTIAL	MCRGALP	NSGALP	SEQUENTIAL	MCRGALP	NSGALP	SEQUENTIAL
50	10	5	0.8454	**0.9559**	0.7381	0.2380	**0.2693**	0.2080	1.1344	**1.0737**	1.5233
50	10	10	0.8396	**0.9506**	0.7605	0.2190	**0.2481**	0.1985	1.1288	**1.0660**	1.3883
50	10	20	0.8457	**0.9589**	0.7969	0.2016	**0.2288**	0.1902	1.1058	**1.0575**	1.2646
50	16	5	0.9213	**0.9622**	0.8237	0.2982	**0.3128**	0.2620	1.0773	**1.0692**	1.8271
50	16	10	0.9159	**0.9580**	0.8282	0.2786	**0.2928**	0.2474	1.0726	**1.0639**	1.5983
50	16	20	0.9144	**0.9574**	0.8364	0.2612	**0.2749**	0.2348	1.0664	**1.0485**	1.4363
50	20	5	0.8317	**0.9371**	0.8155	0.3263	**0.3678**	0.3192	1.1984	**1.0922**	1.4377
50	20	10	0.8404	**0.9361**	0.8300	0.3005	**0.3349**	0.2961	1.1629	**1.0831**	1.3005
50	20	20	0.8497	**0.9419**	0.8489	0.2728	**0.3026**	0.2718	1.1310	**1.0689**	1.1956
50	25	5	0.8529	**0.9586**	0.7266	0.3551	**0.4003**	0.3014	1.1807	**1.0701**	2.5982
50	25	10	0.8870	**0.9563**	0.7995	0.3185	**0.3453**	0.2848	1.1196	**1.0608**	1.7053
50	25	20	0.8779	**0.9575**	0.7908	0.3021	**0.3315**	0.2698	1.1145	**1.0592**	1.5688
50	50	5	0.8613	**0.9553**	0.8006	0.3452	**0.3839**	0.3212	1.1506	**1.0704**	1.7663
50	50	10	0.8691	**0.9569**	0.8208	0.3084	**0.3406**	0.2911	1.1225	**1.0578**	1.4398
50	50	20	0.8703	**0.9571**	0.8299	0.2956	**0.3259**	0.2816	1.1191	**1.0546**	1.3180
100	10	5	0.8470	**0.9186**	0.7896	0.2655	**0.2879**	0.2472	1.1544	**1.1116**	1.3033
100	10	10	0.8461	**0.9164**	0.8115	0.2547	**0.2732**	0.2414	1.1425	**1.1077**	1.2456
100	10	20	0.8528	**0.9225**	0.8294	0.2332	**0.2522**	0.2265	1.1181	**1.0876**	1.1873
100	20	5	0.7869	**0.9169**	0.6988	0.3255	**0.3789**	0.2893	1.2822	**1.1293**	1.6547
100	20	10	0.7815	**0.9145**	0.7218	0.3067	**0.3588**	0.2833	1.2724	**1.1210**	1.5217
100	20	20	0.7875	**0.9157**	0.7390	0.2834	**0.3294**	0.2659	1.2426	**1.1200**	1.3849
200	30	10	0.8539	**0.8857**	0.8405	0.2229	**0.2312**	0.2194	1.1171	**1.1111**	1.2264
200	30	20	0.8557	**0.8864**	0.8513	0.2062	**0.2136**	0.2052	1.0989	**1.0822**	1.1695

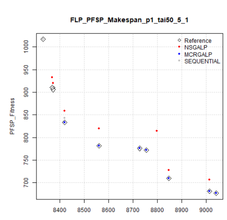
instance with the highest correlation score (0.86). The sequential approach is biased towards optimising the first problem and then produces sub-optimised solutions for the second problem. Considering the problem from the perspective of two companies in a supply chain. Results can offer guidance on the benefits/costs they are likely to experience in solving the overall problem. In Figures 3a and 3c, for sequential and NSGALP, both companies must consider the impact of optimising one problem over another, and decide if the costs/benefits are acceptable whereas, the MCRGALP tends to maintain a balanced compromise on both problems.



(a)



(b)



(c)

Figure 3: Example-Pareto Fronts of Algorithmic Approaches

6 CONCLUSION AND FUTURE WORK

Paper presents linked optimisation problem. We use NSGALP, MCRGALP, and SEQUENTIAL approaches and adapt them to linked optimisation framework. We then, conduct experiments to compare our methods on 620 problem instances. NSGALP outperforms MCRGALP and SEQUENTIAL in terms of metric scores. Other perspectives provide different rationale on algorithmic selection. For future work, appropriate performance metric should be explored to measure algorithmic performance without biasness towards a method. Also, the good performance of the NSGALP and MCRGALP results in sacrificing much computational time in searching for promising solution pair. It would be interesting to further explore some properties of the algorithms to improve efficiency.

REFERENCES

[1] J E Beasley. 1990. OR-Library: distributing test problems by electronic mail. *JORS* 41, 11 (1990), 1069–1072.

[2] M R Bonyadi, Z Michalewicz, and L Barone. 2013. The travelling thief problem: The first step in the transition from theoretical problems to realistic problems. In *2013 IEEE CEC*. IEEE, 1037–1044.

[3] K Deb, A Pratap, S Agarwal, and T A M T Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE TEC* 6, 2 (2002), 182–197.

[4] J Gao and R Chen. 2011. A hybrid genetic algorithm for the distributed permutation flowshop scheduling problem. *IJCIS* 4, 4 (2011), 497–508.

[5] K Holmberg, M Rönnqvist, and D Yuan. 1999. An exact algorithm for the capacitated facility location problems with single sourcing. *EJOR* 113, 3 (1999), 544–559.

[6] M Ibrahimov. 2012. *Evolutionary algorithms for supply chain optimisation.* Ph. D. Dissertation.

[7] F Legillon, A Liefooghe, and E Talbi. 2012. Cobra: A cooperative coevolutionary algorithm for bi-level optimization. In *2012 IEEE CEC*. IEEE, 1–8.

[8] C Lu, L Gao, W Gong, C Hu, X Yan, and X Li. 2021. Sustainable scheduling of distributed permutation flow-shop with non-identical factory using a knowledge-based multi-objective memetic optimization algorithm. *SEC* 60 (2021), 100803.

[9] B Naderi and R Ruiz. 2010. The distributed permutation flowshop scheduling problem. *COR* 37, 4 (2010), 754–768.

[10] E Taillard. 1993. Benchmarks for basic scheduling problems. *EJOR* 64, 2 (1993), 278–285.

[11] E Triantaphyllou, B Shu, S N Sanchez, and T Ray. 1998. Multi-criteria decision making: an operations research approach. *EEEE* 15, 1998 (1998), 175–186.

[12] D K S Vieira, G L Soares, J A Vasconcelos, and M H S Mendes. 2017. A genetic algorithm for multi-component optimization problems: the case of the travelling thief problem. In *ECECCO*. Springer, 18–29.

[13] A M Yaakob and A Gegov. 2016. Interactive TOPSIS based group decision making methodology using Z-numbers. *IJCIS* 9, 2 (2016), 311–324.

[14] X Zhang, Xiang-Tao Li, and Ming-Hao Yin. 2020. An enhanced genetic algorithm for the distributed assembly permutation flowshop scheduling problem. *IJBC* 15, 2 (2020), 113–124.

[15] E Zitzler and L Thiele. 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE TEC* 3, 4 (1999), 257–271.

[16] E Zitzler, L Thiele, M Laumanns, C M Fonseca, and V G Da Fonseca. 2003. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE TEC* 7, 2 (2003), 117–132.