Received July 17, 2020, accepted July 28, 2020, date of publication August 4, 2020, date of current version August 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3014134

Greedy-Based Non-Dominated Sorting Genetic Algorithm III for Optimizing Single-Machine Scheduling Problem With Interfering Jobs

CHEN-YANG CHENG[1], SHIH-WEI LIN[2,3,4,*], POURYA POURHEJAZY[1], KUO-CHING YING[1], SHU-FEN LI[5], AND YING-CHUN LIU[6]

[1]Department of Industrial Engineering and Management, National Taipei University of Technology, Taipei 10608, Taiwan
[2]Department of Information Management, Chang Gung University, Taoyuan City 33302, Taiwan
[3]Department of Neurology, Linkou Chang Gung Memorial Hospital, Taoyuan City 33305, Taiwan
[4]Department of Industrial Engineering and Management, Ming Chi University of Technology, New Taipei City 24301, Taiwan
[5]Department of Industrial Engineering and Management, National Chin-Yi University of Technology, Taichung City 41170, Taiwan
[6]Hsinchu Science Park Burau, Ministry of Science and Technology of Taiwan, Hsinchu 30016, Taiwan

Corresponding author: Kuo-Ching Ying (kcying@ntut.edu.tw)

[*]*(Shih-Wei Lin is co-first author.)*

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST108-2221-E-027-025/MOST109-2410-H-182-009-MY3, and in part by the Linkou Chang Gung Memorial Hospital under Grant CMRPD3G0011.

ABSTRACT Given the importance of production planning and control in the design of flexible services and manufacturing systems, scheduling problems with interfering jobs are much-needed optimization tools to respond to heterogeneous and fluctuating market demands in a timely fashion. This study contributes to the scheduling literature developing an effective multi-objective (M-O) metaheuristic to solve the Single-machine Scheduling Problems with Interfering Jobs (SSP-IJs). Integrating a local search-based mechanism into the evolutionary search procedure, a Greedy-based non-dominated sorting genetic algorithm III (GNSGA-III) is proposed that effectively explores multi-objective solution environments. Various performance indicators within extensive numerical tests are used to compare the performance of the GNSGA-III with that of the best-performing benchmark algorithm in the literature developed to solve the SSP-IJs. Statistical tests verify that the developed multi-objective optimization algorithm is superior with respect to various performance indicators. Applications of the developed solution approach are worthwhile topics to help advance multi-objective optimization problems.

INDEX TERMS Scheduling, interfering jobs, multi-objective optimization, non-dominated solutions, metaheuristics.

I. INTRODUCTION

Operational flexibility is instrumental in advanced manufacturing and supply chain systems [1]. Flexible production planning and control systems help quick, and effective response to demand surges in the volatile market where the proliferation of products and services has increased supply chain operational complexities [2]. Taking the manufacturing sector as an example, big companies provide various business-to-business solutions and assign the tasks to different management units that must compete for using the same resources with different goals. This operational complexity is not limited to the production sector and may happen in the service and administration environments when different units' operational goals are conflicting. In this situation, multifaceted performance indicators are required to effectively optimize operations from a system-wide perspective.

Production scheduling with interfering job sets is a relatively new topic that addresses conflicting operational goals. Developed by [3], the single-machine scheduling problem with interfering jobs (SSP-IJs) is a prime example of a production situation where the available resource is used to fulfill various customer requirements. References [4], [5] analyzed the complexity of job interference in a single-machine production environment, highlighting the need for applying Multi-objective (M-O) optimization

The associate editor coordinating the review of this manuscript and approving it for publication was Zhiwu Li.

approaches that simultaneously address various sets of conflicting goals. Although M-O optimization in supply chain and operations management is in the growing stage of development [6], M-O scheduling problems are relatively limited. From the existing studies, researchers extended the SSP-IJs problem, integrating inventory and batch delivery decision variables into the production scheduling problem [7], [8]. Reference [9] extended SSP-IJs accounting for time-window and unavailability intervals and solved it using a dynamic programming approach. Scheduling problems with interfering job sets were also studied in other production environments, like job-shop [10], parallel machines [11], and flowshop [12]. Fewer studies developed M-O solution algorithms to optimize this highly intractable scheduling extension. For an exhaustive review of the M-O scheduling studies, we refer the readers to [13].

The SSP-IJs is recognized to be NP-hard when considering Total Weighted Completion Time (TWCT), lateness, or makespan as one of the optimization objectives [14]. Several studies presented polynomial-time approximation methods aiming to reduce the computational time to obtain exact solutions to small-scale SSP-IJs problems [15]–[17]. More recently, [18] introduced a variety of the shortest processing time algorithm to address the SSP-IJs, yielding acceptable solutions in a small fraction of the computational time required by complete enumeration methods to solve small-scale test instances. Given the wide real-world implications of SSP-IJs [14], state-of-the-art solution algorithms are needed to obtain dependable solutions for industrial-scale applications of the scheduling problems with interfering jobs. To address this issue, reference [19] developed heuristic approaches to solve the SSP-IJs problem minimizing total weighted completion time (TWCT) and the maximum lateness considering large-scale instances; these objectives are crucial for production efficiency and service quality objectives, respectively, and a decrease in one objective value may increase that of the other objective. Comparing the results obtained by the forward WSPT-EDD heuristic with that of a time-indexed MIP formulation solved by exact methods, reference [19] showed that their approach generates high-quality non-dominated solutions to SSP-IJs. To contribute to the understudied literature of SSP-IJs, our study develops a hybrid local-global search M-O metaheuristic to further improve the solutions obtained by the mentioned benchmark algorithm. Inspired by a well-known heuristic, Iterated Greedy (IG), the Greedy-based Non-dominated Sorting Genetic Algorithm III (GNSGA-III) is developed to provide divergent and high-quality non-dominated solutions to the M-O scheduling situations with interfering jobs.

The remainder of this manuscript begins with elaborating on the mathematical formulation and the proposed solution algorithm in Section 2. Section 3 provides numerical analysis including an introduction to the test instances and performance measures, algorithm calibration, test results, and statistical analysis. This study is concluded in Section 4, providing the major findings and suggestions for possible future research directions.

II. RESEARCH METHOD

A. MODEL FORMULATION

Let assume a production environment with I interfering job sets $(S_1, S_2, \ldots, S_I)$ with $n_1, n_2, \ldots, n_I$ jobs, respectively, where a total of $N = \sum_{s=1}^{I} n_s$ jobs are to be processed on a single machine. In this situation, the disjoint interfering sets of jobs – which are associated with different goals – have to compete for using the same machine. Deterministic processing and due times, P_j^s and d_j^s, are associated with job $j \in \{1, 2, \ldots, n_s\}$ from set S_s. Given the general assumptions of the $1|inter|ND(\sum w_j C_j, L_{\max})$ problems [19], the objective is to find the best sequence of jobs with the desired trade-offs of non-dominated solutions for TWCT $(\sum w_j c_j)$ and maximum lateness $(L_{\max})$ objectives. To analyze this trade-off, let assume a set of feasible solutions considering two conflicting objectives $(I = 2)$. In this situation, $z_1(x)$ and $z_2(x)$ are the target values of the jobs in the set S_1 and S_2, respectively, and x^* is a near Pareto-optimal solution if no other solutions are dominating. The following indices, parameters, and decision variables are defined to formulate the problem.

Indices, sets, and parameters

j, k	Job index, $j, k \in \{1, 2, \ldots, N\}$
s	Job set index, $s \in \{1, 2, \ldots, I\}$
t	Time index discretized time into periods $1, \ldots, T$ where period t ends at the time t.
N	Total number of jobs, $N = n_1 + n_2$ where there are n_1 jobs in set 1, and n_2 jobs in set 2
D	The overall due time
P_j^s	The processing time of job j from set s
d_j^s	Due time of job j from set s
w_j^s	Weight of job j from set s

Decision variables

$x_{j,t}$	A binary time-indexed variable $= 1$ if the job j is processed at the time t; $= 0$, otherwise.

The mixed-integer programming (MIP) formulation of the $1|inter|ND(\sum w_j C_j, L_{\max})$ problem, developed by [19], is provided in the following.

$$z_1 = Min\left(\sum_{j=1}^{n}\sum_{t=1}^{T-p_j+1} w_j\left(t-1+p_j\right)x_{j,t}\right)$$
$$z_2 = Min\,(L_{\max}) \tag{1}$$

Subject to :

$$\sum_{t=1}^{T-p_j+1} x_{j,t} = 1,\ \forall j \in \{1,\ldots,N\} \tag{2}$$

$$\sum_{j=1}^{n}\sum_{s=\max(0,t-p_j+1)}^{t} x_{j,s} \le 1,\quad \forall t \in \{1,\ldots,T\} \tag{3}$$

Generation t of GNSGA-III (Structured reference points Z_s, and parent population P_t)

Begin {

1. $P_{t+1} = \phi$; $i = 1$;
2. $Q_t \leftarrow$ Crossover and Mutation (P_t);
3. $G_t \leftarrow$ IG (P_t);
4. $R_t = P_t \cup Q_t \cup G_t$;
5. $F_1, F_2 ..., F_L$ = Non-dominated sorting (R_t);
6. while $|P_{t+1}| < N$: $P_{t+1} = P_{t+1} \cup F_i$ and $i = i + 1$;
7. if $|P_{t+1}| = N$: break;
8. else: $F_l \leftarrow F_i$;
9. $N - |P_{t+1}| \leftarrow$ reference-based-selection (Z_s, F_l);

} **Return** P_{t+1}

FIGURE 1. The GNSGA-III computational procedure.

IG-based mechanism (P_t, d)

begin
  $\boldsymbol{\pi}^{original} \leftarrow \mathrm{RWS}(P_t)$;
  //Destruction phase
  $\boldsymbol{\pi}^{removed} = \varnothing$;
  for $j = 1{:}d$:
    Randomly select jobs from $\boldsymbol{\pi}^{original}$ *and add to* $\boldsymbol{\pi}^{removed}$;
    $\boldsymbol{\pi}^{removed} \leftarrow \boldsymbol{\pi}^{removed} \cup \{\boldsymbol{\pi}^{removed}\}$;
  $\boldsymbol{\pi}^{partial} \leftarrow \boldsymbol{\pi}^{original} - \boldsymbol{\pi}^{removed}]$;
  //Construction phase
  while $\boldsymbol{\pi}^{removed}$:
    $\boldsymbol{\pi}^{partial} \leftarrow$ *Insert* $\boldsymbol{\pi}^{removed}{}_{[1]}$ *to all positions of* $\boldsymbol{\pi}^{partial}$, *and select the best*;
  else $\boldsymbol{\pi}^{new} \leftarrow \boldsymbol{\pi}^{partial}$
  $Q_t \leftarrow \boldsymbol{\pi}^{new}$
return Q_t

FIGURE 2. A Generic Procedure of the IG-based Mechanism.

$$\sum_{t=1}^{T-p_j+1} (t - 1 + p_j) x_{j,t} - d_j \leq L_{\max}, \quad \forall j \in \{1, \ldots, N\} \tag{4}$$

$$x_{j,t} \in \{0, 1\}, \quad \forall j \in \{1, \ldots, N\}, \forall t \in \{1, \ldots, T\} \tag{5}$$

In this formulation, $d_j^1 = D$ for $\forall j \in \{1, \ldots, n_1\}$, where $D \geq \sum_{j=1}^{n_1} p_j^1 + \sum_{k=1}^{n_2} p_k^2$, and $w_j^2 = 0$ for $\forall j \in \{1, \ldots, n_1\}$. In this formulation, D is introduced to guarantee that the jobs in the first set will not influence the objective value associated with the second job set. The first objective of (1) minimizes the total weighted completion time, and the second objective seeks to maximize the total lateness value, which is conflicting in nature with the first objective. That is, the better values for one objective may worsen that of the other objective function. Constraint (2) guarantees that each job is processed at a certain time slot. Constraint (3) restricts the machine to process only one job at a given time. Constraints (4) calculates the maximum lateness value, and constraints (5) specifies that decision variables are of binary type.

B. PROPOSED SOLUTION ALGORITHM

M-O optimization initially formed around a priori preference articulation where the pre-ordering of the objectives is decided before optimizing the problem [20]. Advancing the M-O optimization to a whole new level, [21] introduced the Non-dominated Sorting Genetic Algorithm (NSGA) to solve bi-objective problems providing a set of Pareto-optimal points rather than a single solution; this approach enables the decision-maker to undertake tradeoffs amongst optimum solutions before arriving at a final decision. The basic NSGA algorithm was later on improved by including elitism into the search procedure (NSGA-II; [22]). Most recently, [23] incorporated reference points into the non-dominated solutions in NSGA-III algorithm, empowering the algorithm to maintain solution diversity throughout the search procedure. The main difference distinguishing NSGA-III's computational procedure from NSGA-II is in the selection procedure where the crowding distance operator is replaced with a multi-step niche-preservation approach [23].

NSGA-II and -III algorithms are successfully applied in solving M-O optimization problems in various contexts, like transportation [24], [25], manufacturing [26], [27], and healthcare [28], [29], among the other application areas. More particularly, the adjusted NSGA-III has appeared to outperform the state-of-the-art solution methods developed for solving many-objective problems [30], generating more diverse near-non-dominated solutions [23]. Given that the scheduling problems with interfering job sets may have to deal with more than two conflicting operational goals, NSGA-III is preferred to NSGA-II for this particular application area.

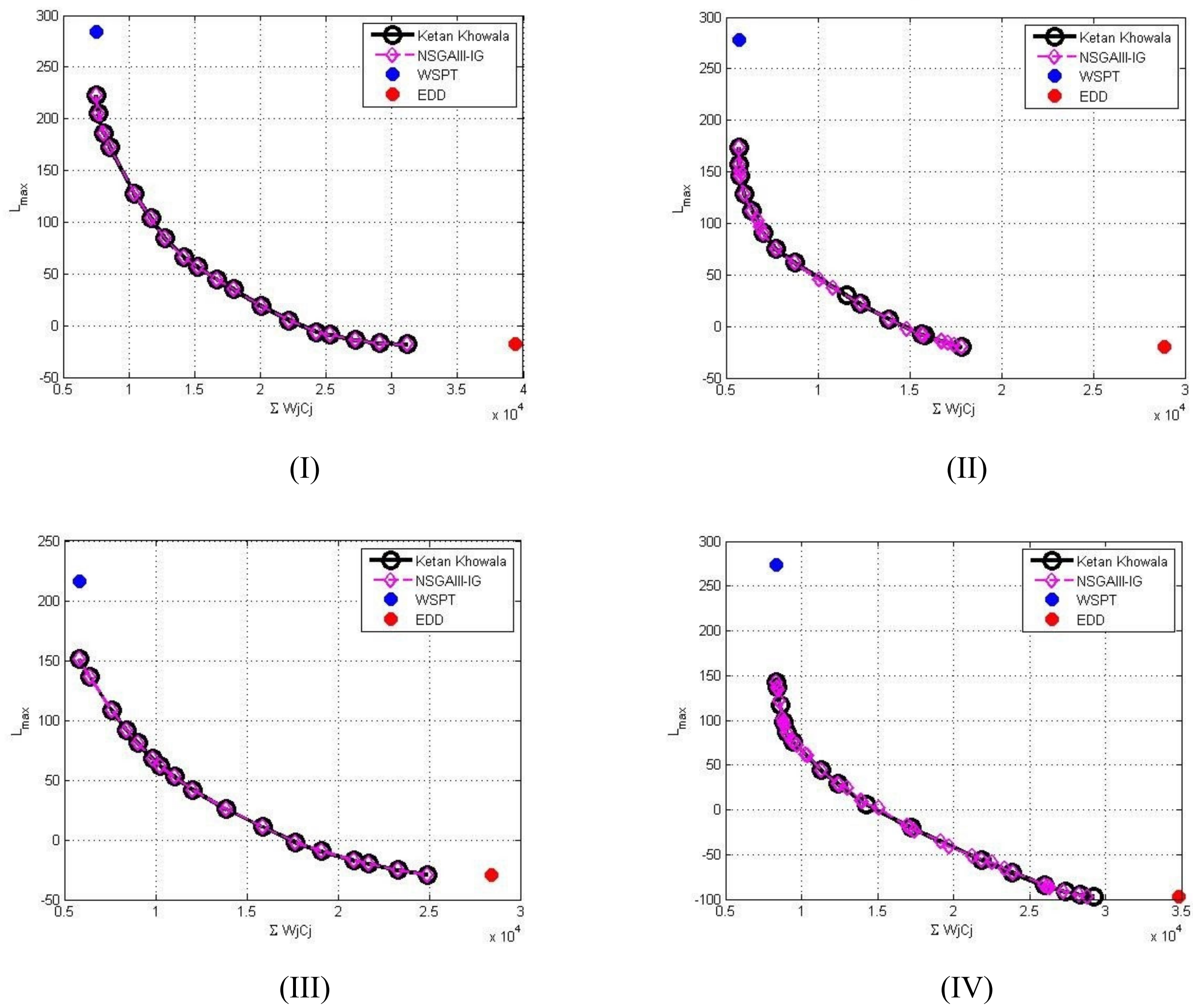

FIGURE 3. **Non-dominated solution benchmarks over 20-20 test instances.**

Inspired by the IG algorithm, developed by [31], the GNSGA-III developed in our study integrates a local search mechanism consisting of destruction and construction, neighborhood search, and acceptance operators as a part of the evolutionary process best-suited to search for high quality (near)-non-dominated solutions in the scheduling problems with interfering jobs. The computational steps at a given iteration of GNSGA-III are summarized in Figure 1. This procedure continues for a certain number of iterations, the stopping criterion. The major computational elements are briefly explained in the following.

1) REFERENCE POINTS

This study employs the solution approach proposed by [19] to determine the extreme points for structuring the reference points. For this purpose, the $1|inter|\sum w_jC_j = K_{\min}, L_{\max}$ and $1|inter|\sum w_jC_j, L_{\max} = Y_{\min}$ problems are solved separately. Sorting the jobs associated with the first (S_1) and the second objective (S_2) based on the WSPT and EDD dispatching rules, respectively, the resulting optimum solutions are the extreme points positioning at the two ends of the Pareto-front. The lines connecting the extreme points with the ideal values define the "reference lines" which will be used in the selection mechanism of the GNSGA-III algorithm. Given the desired number of tradeoffs, which needs to be determined by the decision-maker, and the "reference lines", a structured reference point, Z_s will be the outcome of this procedure.

2) INITIALIZATION AND SOLUTION REPRESENTATION

The initialization procedure consists of generating random solutions to fill the primary population. Organizing the job sets in a single vector $[S_1, S_2, \ldots, S_I]$, integer permutation encoding is used to represent the random solutions. The initialization procedure continues by evaluating the population members. Known as the natural selection in the Genetic Algorithm [32], the evaluation procedure in NSGA-III considers domination and the reference points to keep the fittest

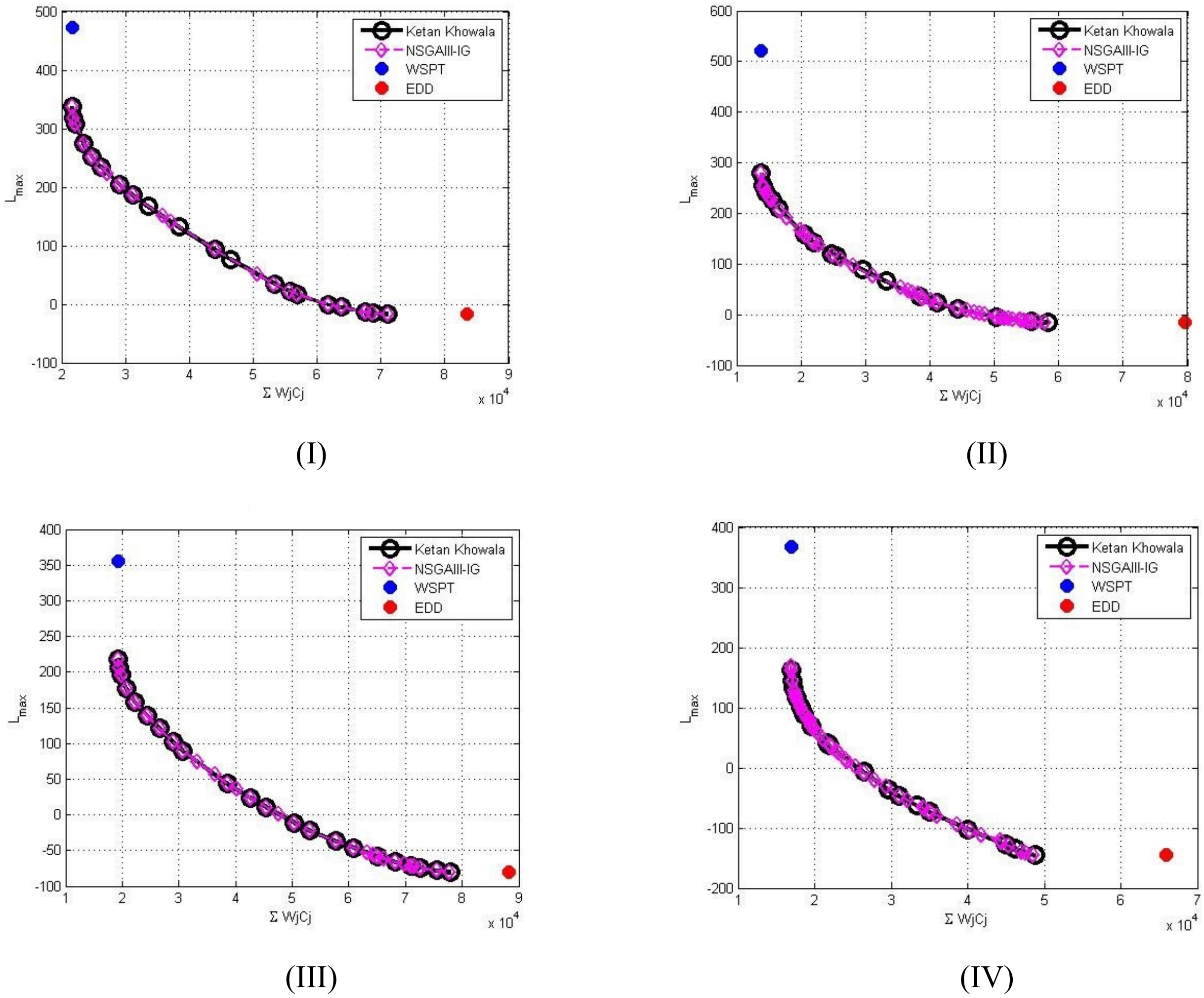

FIGURE 4. Non-dominated solution benchmark over 30-30 test instances.

TABLE 1. Calibration Results of the GNSGA-III Algorithm (Best in Bold).

Parameter	Phase I			Phase II			Phase III		
	A	B	C	D	E	F	G	H	I
d	1	5	10	5	5	5	5	5	5
i	$10\times N$	$10\times N$	$10\times N$	$10\times N$	$20\times N$	$30\times N$	$30\times N$	$30\times N$	$30\times N$
c	2	2	2	2	2	2	2	4	6
m	1	1	1	1	1	1	1	4	6
Average	14.37	**22.10**	20.63	21.03	22.10	**23.33**	**22.00**	21.63	21.40

solutions; the former measure evaluates superiority of a specific solution to the others concerning the objective values, and the later metric ensures maximum diversity amongst the solutions situated on the same frontier.

3) CROSSOVER AND MUTATION MECHANISMS

As a first step to the recombination procedure, solution pairs must be selected from the parent set P_t. Roulette Wheel Selection (RWS) is used for this purpose, where the likelihood of selecting a certain solution is proportionate to its objective values relative to the rest of the parent set, P_t members. The decision on the type of crossover and mutation mechanisms depends on the way solutions are encoded. From the existing crossover and mutation methods in the literature [33], the Partially-Matched Crossover (PMX) developed by [34] is applied to fill the offspring population Q_t. Breaking

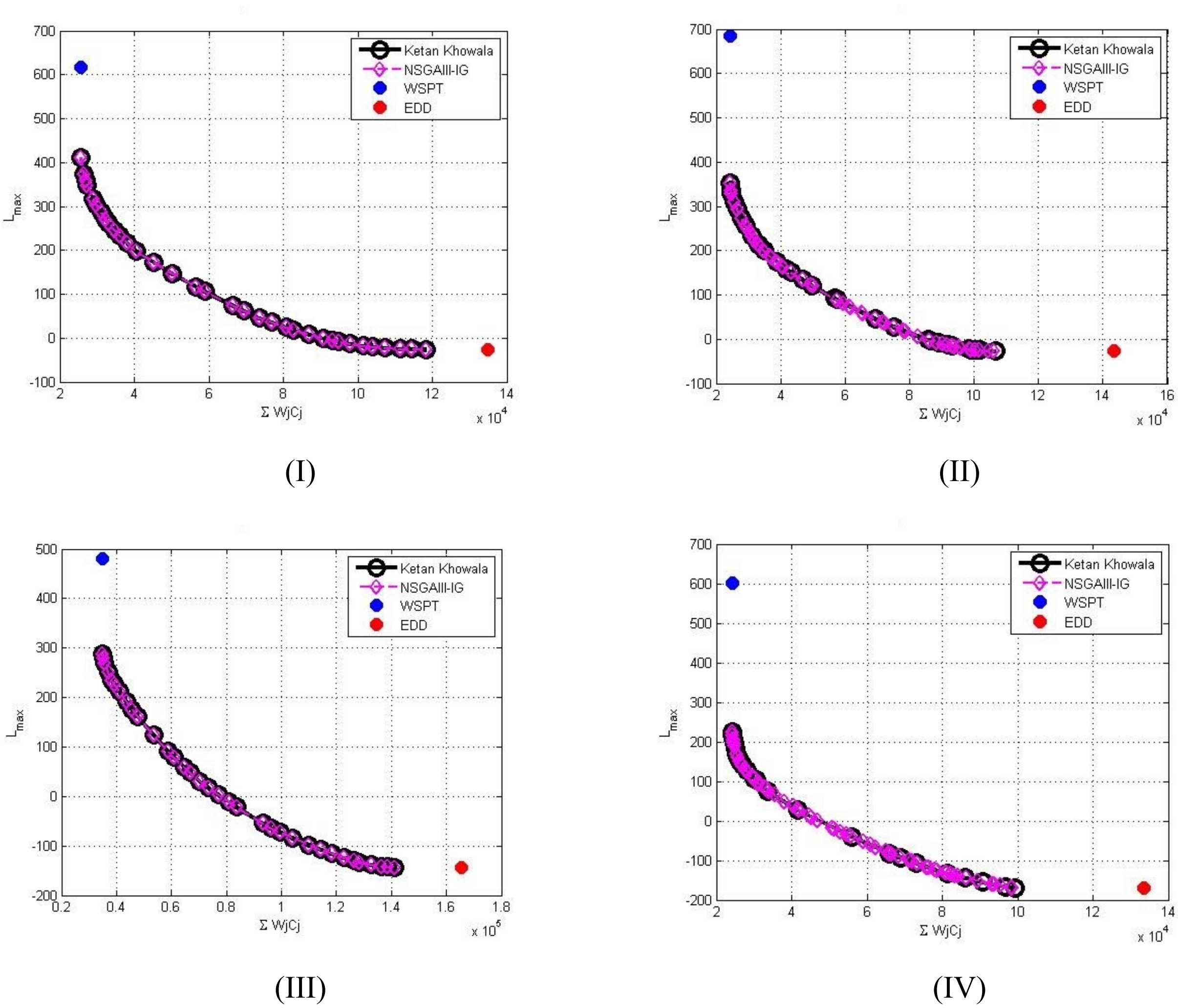

FIGURE 5. **Non-dominated solution benchmark over 40-40 test instances.**

the selected parent solutions into three sections using two random crossover points, PMX forms the first offspring with the first and third sections of the first parent and the second section of the second parent. The remaining sections of the parents form the second offspring. Next, a random job-swap may take place as the mutation operator considering a certain probability. The recombination process is completed after fixing the offspring solutions considering the scheduling problem constraints and replacing the repeated jobs.

4) IG-BASED MECHANISM

The IG-based mechanism selects non-dominated solutions from the parent set, P_t, and applies destruction-construction and neighborhood search procedures to generate new non-dominated solutions, G_t. The greedy concept triggers iterative modifications in the solution to search for better (near)-non-dominated solutions. The pseudocode of the IG-based mechanism is provided in Figure 2 with the major steps described in the following:

Step 1. Apply RWS to select a solution, $\pi^{original}$ from the parent set P_t based on the rank of the objective value.

Step 2. Destruct $\pi^{original}$ by extracting d random jobs; sort the removed jobs to form $\pi^{removed} = [J_1, J_2, \ldots, J_d]$. On this basis, the resulting partial sequence $\pi^{partial}$ includes $n - d$ jobs.

Step 3. Apply the construction procedure by iteratively inserting $\pi^{removed}$ jobs into $\pi^{partial}$. For this purpose, extract the first job, J_1 from $\pi^{removed}$, and insert it into the $n - d + 1$ available positions in $\pi^{partial}$ and calculate the resulting objective values. Apply the nondominated procedure to select the best partial sequence.

Step 4. Continue step 3 to insert the remaining jobs, i.e. $J_2, \ldots, J_d$ until no jobs are left in $\pi^{removed}$. Save the resulting solution, π^{new} in G_t.

5) SELECTION MECHANISM

Given the parent population P_t, offspring population Q_t, and the set of neighborhood solutions G_t at the t^{th} generation of

TABLE 2. **Results Comparison Over 20-20 Test Instances (Best in Bold).**

Indicator	Algorithm	Instance I	Instance II	Instance III	Instance IV
A	GNSGA-III	17.4	**20.8**	17.6	**27.4**
	WSPT-EDD	17.4	15.2	17.6	16.8
B	GNSGA-III	**17.4**	**20.2**	**17.6**	**26.8**
	WSPT-EDD	16.8	9.0	16.0	11.8
B/A	GNSGA-III	**1.0000**	**0.9733**	**1.0000**	**0.9817**
	WSPT-EDD	0.9647	0.5918	0.9059	0.7083
Ω	GNSGA-III	**0.5097**	**0.6902**	**0.5269**	**0.6844**
	WSPT-EDD	0.4903	0.3098	0.4731	0.3156
C	GNSGA-III	**0.0000**	**0.0267**	**0.0000**	**0.0183**
	WSPT-EDD	0.0353	0.4082	0.0941	0.2917
γ	GNSGA-III	**0.0000**	**0.9661**	**0.0000**	**0.7734**
	WSPT-EDD	0.1529	14.8160	5.7412	33.6552
Δ	GNSGA-III	**1.3213**	**0.5630**	**0.4954**	0.7064
	WSPT-EDD	**1.3213**	0.5675	0.4976	**0.6587**

TABLE 3. **Results Comparison Over 30-30 Test Instances (Best in Bold).**

Indicator	Algorithm	Instance I	Instance II	Instance III	Instance IV
A	GNSGA-III	**22.6**	**50.0**	**27.4**	**44.6**
	WSPT-EDD	21.2	21.4	24.2	21.0
B	GNSGA-III	**22.6**	**48.0**	**27.4**	**42.2**
	WSPT-EDD	14.6	14.0	19.6	13.8
B/A	GNSGA-III	**1.0000**	**0.9622**	**1.0000**	**0.9484**
	WSPT-EDD	0.7146	0.6555	0.8030	0.6695
Ω	GNSGA-III	**0.6045**	**0.7749**	**0.5878**	**0.7513**
	WSPT-EDD	0.3955	0.2251	0.4122	0.2487
C	GNSGA-III	**0.0000**	**0.0228**	**0.0000**	**0.0436**
	WSPT-EDD	0.2854	0.3354	0.1970	0.3200
γ	GNSGA-III	**0.0000**	**3.7158**	**0.0000**	**3.4112**
	WSPT-EDD	15.1383	46.8273	7.0742	28.2472
Δ	GNSGA-III	0.4926	0.9215	**0.5141**	0.7159
	WSPT-EDD	**0.4875**	**0.5634**	0.5281	**0.6084**

GNSGA-III, a total of $2Pop + |G_t|$ members have resulted. In the last computational step of the current iteration, $R_t = P_t \cup Q_t \cup G_t$ members are to be ranked, and selected such that the total population size of the $(t+1)^{th}$ generation is larger than the pre-specified size$|R_t| \geq N$. Given the non-dominated frontiers, $F_1, F_2, \ldots, F_L$ the solution quality is considered as the main metric to populate the new parent population in the $(t+1)^{th}$ generation, P_{t+1}. In this situation, the first l frontier members are directly selected for the next generation $(F_1 \cup F_2 \ldots \cup F_l)$, while an additional mechanism is needed to select from the members of the $(l+1)^{th}$ frontier to fill the $Pop - |F_1 \cup F_2 \ldots \cup F_l|$ vacancies in P_{t+1}. For this purpose, the distance from the adjacent reference line from the structured reference points Z_s is used as the selection gauge to maintain the diversity of the non-dominated solutions.

Considering the classified population and the reference points Z_s, the objective values associated with the population members, which are scaled differently should be normalized. In the normalized solution space, the reference point with a reference line in close proximity to a given population member is considered to be associated with that member.

TABLE 4. Results Comparison Over 40-40 Test Instances (Best in Bold).

Indicator	Algorithm	Instance			
		I	II	III	IV
A	GNSGA-III	**25.2**	**51.6**	**33.2**	**65.4**
	WSPT-EDD	25.0	22.2	32.8	25.2
B	GNSGA-III	**25.2**	**47.2**	**33.2**	**60.6**
	WSPT-EDD	22.4	15.6	31.2	19.8
B/A	GNSGA-III	**1.0000**	**0.9129**	**1.0000**	**0.9278**
	WSPT-EDD	0.8848	0.7308	0.9529	0.7822
Ω	GNSGA-III	**0.5353**	**0.7441**	**0.5161**	**0.7537**
	WSPT-EDD	0.4647	0.2559	0.4839	0.2463
C	GNSGA-III	**0.0000**	**0.0628**	**0.0000**	**0.0595**
	WSPT-EDD	0.1152	0.2692	0.0471	0.2178
γ	GNSGA-III	**0.0000**	**11.7048**	**0.0000**	**9.7338**
	WSPT-EDD	0.6970	43.7319	0.8235	34.2364
Δ	GNSGA-III	**0.5347**	**0.7111**	**0.4847**	**0.7313**
	WSPT-EDD	0.5391	0.7347	0.4858	0.8015

TABLE 5. Results Comparison Over 50-50 Test Instances (Best in Bold).

Indicator	Algorithm	Instance			
		I	II	III	IV
A	GNSGA-III	**29.0**	**64.4**	**38.6**	**85.8**
	WSPT-EDD	28.4	26.6	36.6	33.2
B	GNSGA-III	**29.0**	**57.4**	**38.2**	**76.2**
	WSPT-EDD	21.8	21.4	32.0	28.4
B/A	GNSGA-III	**1.0000**	**0.8885**	**0.9905**	**0.8899**
	WSPT-EDD	0.7589	0.8138	0.8657	0.8412
Ω	GNSGA-III	**0.5820**	**0.7251**	**0.5508**	**0.7386**
	WSPT-EDD	0.4180	0.2749	0.4492	0.2614
C	GNSGA-III	**0.0000**	**0.0707**	**0.0095**	**0.0837**
	WSPT-EDD	0.2411	0.1862	0.1343	0.1588
γ	GNSGA-III	**0.0000**	**25.9826**	**0.3857**	**13.1277**
	WSPT-EDD	5.3984	39.4799	6.0672	29.4296
Δ	GNSGA-III	**0.5041**	0.7293	0.5041	0.7664
	WSPT-EDD	0.5131	**0.6826**	**0.4909**	**0.7132**

Keeping the track of reference point association counts, the algorithm applies a niching method to select the best $(l+1)^{th}$ frontier members to fill all vacant population slots while ensuring maximum diversity. For more details on the reference-based selection mechanism, we refer the authors to [23].

III. NUMERICAL ANALYSIS

A. TEST INSTANCES

This study applies the test instances developed by reference [19]. In addition to the symmetric workloads, where S_1 and S_2 sets consist of 20-20, 30-30, 40-40, and 50-50 jobs in the original test dataset, asymmetric job sets of 10-30 and 30-10 are developed to provide additional insights into the interfering production situations. For this purpose, random processing times are generated following a $U[1, 20]$ distribution. Besides, random weight and due time are assigned to every job considering $U[1, 10]$ and $U[P(L-R/2), P(L+R/2]$ distributions, respectively. Considering $P = 0.5P_1 + P_2$, $L \in \{0.5, 0.7\}$, $R \in \{0.4, 0.8\}$, four operating configurations, I : $[0.3P, 0.7P]$, II : $[0.1P, 0.9P]$, III : $[0.5P, 0.9P]$ and IV : $[0.3P, 0.1P]$ are considered for each of the instance

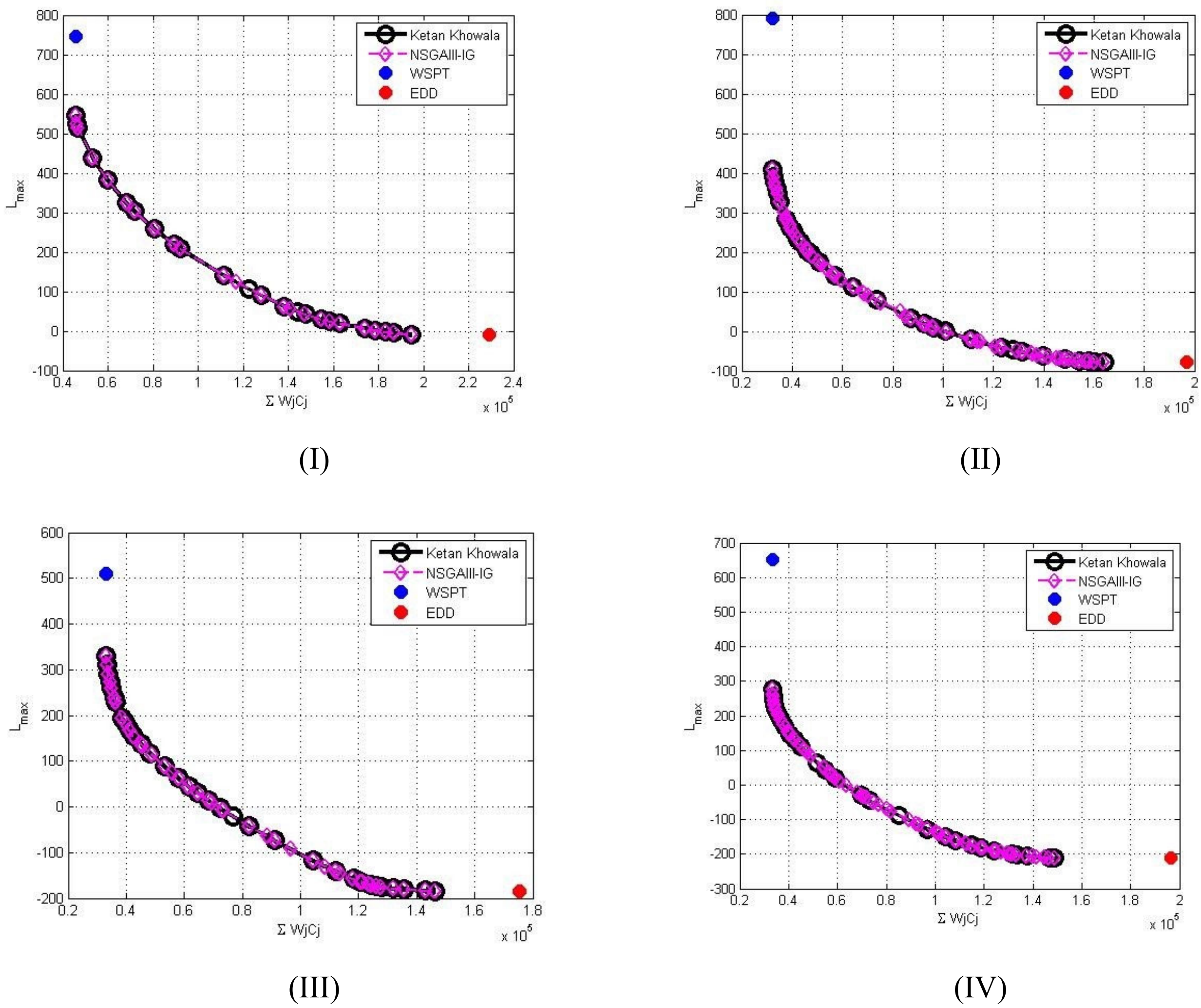

FIGURE 6. Non-dominated solution benchmark over 50-50 test instances.

categories. Given 20 instances for each of the workload categories, a total of 120 are considered for the numerical experiments.

B. PERFORMANCE MEASURES

This study considers the set of performance indicators suggested by [18], [35], [36] to compare the benchmark algorithms. These indicators have been widely used in multi-objective optimization literature, and considering them all together provides a more comprehensive evaluation of the algorithms' performance.

The number of non-dominated solutions provided by a solution algorithm and the number of non-dominated solutions that are not identified by the other approaches in the benchmark tests are denoted by A and B indicators, respectively. Larger A and B values are desirable, and a B/A ratio close-to-one demonstrates the algorithm's strength in providing unique non-dominated solutions. Showing the relative exploration power, the dominance ratio, Ω_H, determines the percentage of non-dominated solutions found by a certain algorithm in the benchmark tests. Equation (6) calculates the Ω_H value, where larger values are desired, and $\Omega_H = 1$determines the absolute superiority of the algorithm H in finding non-dominated solutions.

$$\Omega_{H_k} = \frac{\left| P(\bigcup_i H_i) \backslash P(\bigcup_{i \neq k} H_i) \right|}{\left| P(\bigcup_i H_i) \right|} \tag{6}$$

where $\left| P(\bigcup_i H_i) \backslash P(\bigcup_{i \neq k} H_i) \right|$ determines the number of non-dominated solutions found by the algorithm H which are not explored by the other benchmark algorithms. Using Equation (7), $C(X, Y)$ helps compare the number of weak solutions in the Pareto-front of Y with that of the X algorithm ($x \succ y$), which determines the correctness of the algorithm. Smaller $C(X, Y)$ demonstrates the superiority of the X algorithm over the Y algorithm in obtaining strong solutions, where

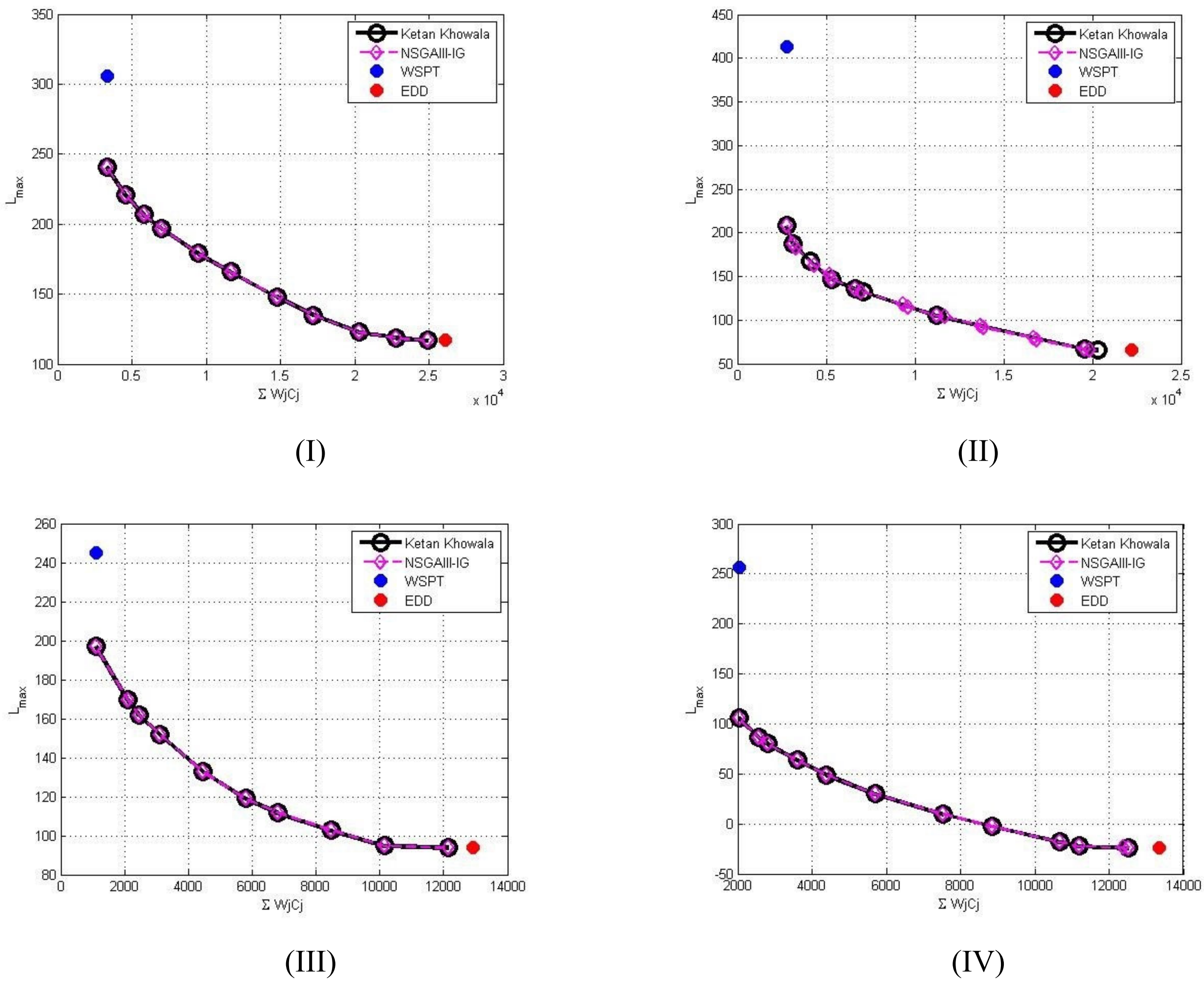

FIGURE 7. **Non-dominated solution benchmark over 10-30 test instances.**

$C(X, Y) = 1$ if all the Pareto-optimal solutions obtained by X dominates those obtained by the Y algorithm. Overall, smaller C specifies that the non-dominated solutions obtained by a certain algorithm are considered weaker.

$$C(X, Y) = \frac{|y \in Y, \exists x \in X : x \succ y|}{|Y|} \quad (7)$$

Comparing the difference between an average Pareto-front solution with the other members of the Pareto set, the diversity measure, Δ, demonstrates the algorithm's ability to provide diverse trade-offs; the more trade-offs there are, the more flexible the decision-making becomes. Smaller Δ is preferred, and close-to-zero values show a high diversity of the solutions of the Pareto set, and that the obtained results are uniformly distributed along with the net non-dominated solutions. Equation (8) calculates the Δ value, where d_f and d_l are the Euclidean distances between the two extreme solutions of the obtained non-dominated Pareto set by an algorithm with that of the net non-dominated Pareto set; the number of solutions in the obtained non-dominated Pareto set is demonstrated by N; d_i is the Euclidean distance between two successive solutions, i.e. solutions i and $i+1$ in the obtained non-dominated Pareto set; $\overline{d}$ is the calculated mean Euclidean distance over all the obtained non-dominated solutions.

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} \left| d_i - \overline{d} \right|}{d_f + d_l + (N-1)\overline{d}} \quad (8)$$

As a final performance indicator, the convergence power, γ, demonstrates the algorithm's ability to generate solutions close-or-similar to the net non-dominated solutions. Smaller γ values are desired, showing that the obtained non-dominated solutions by the algorithm are in closer average proximity to the net non-dominated solutions for Euclidean distances.

C. CALIBRATION OF GNSGA-III

A multi-step calibration test is applied to configure the GNSGA-III parameters. Each step consists of selecting the best value of one parameter while the rest of the parameter values remain constant. On this basis, the IG-based

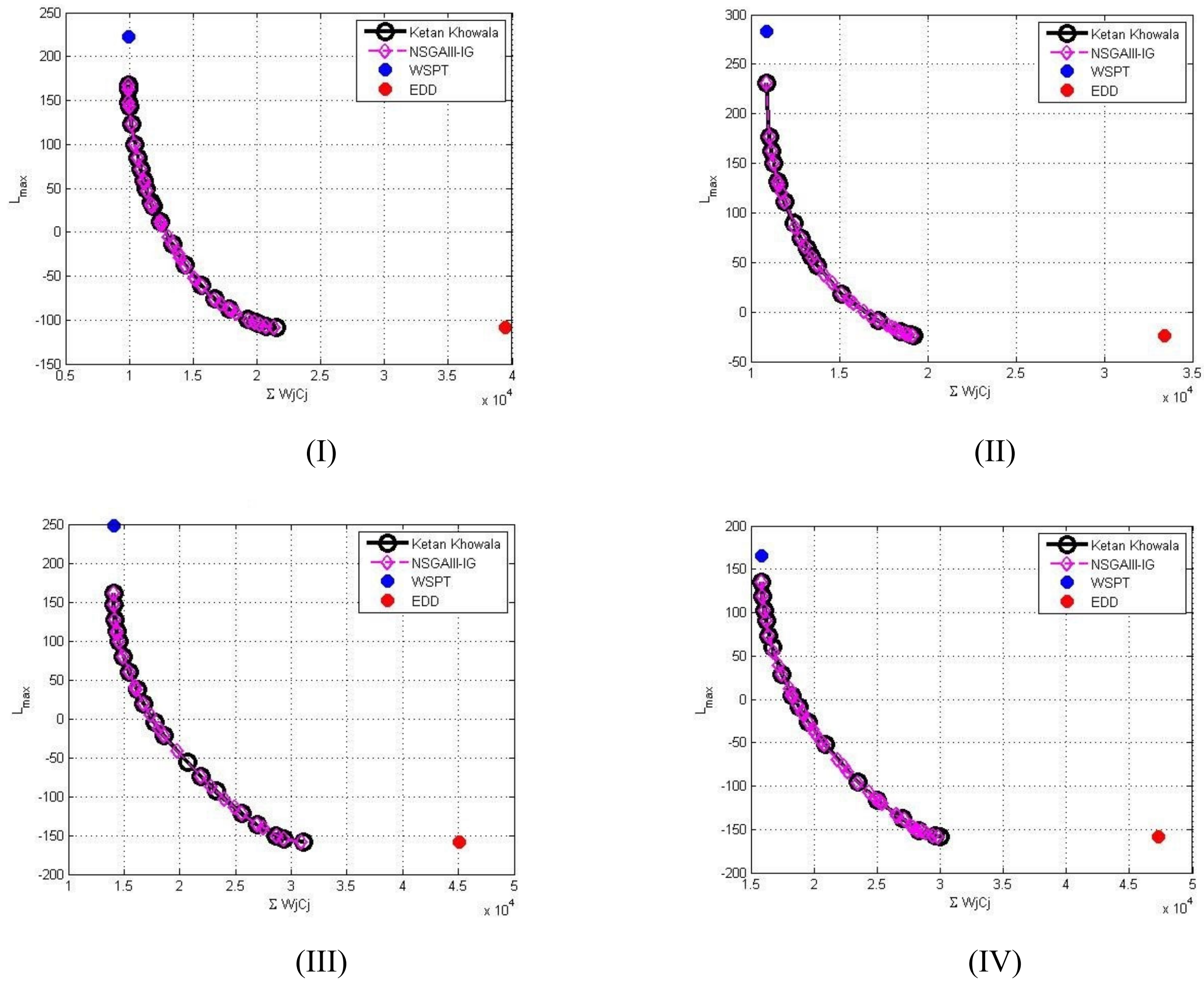

FIGURE 8. Non-dominated solution benchmark over 30-10 test instances.

mechanism's parameter d is calibrated in Phase I, followed by the iteration number, i, in Phase II, and, the crossover, c, and mutation, m, rates in Phase III. A total of 30 random test instances are used for the calibration tests, and the number of non-dominated solutions is considered as the performance indicator. Table 1 summarizes the procedure configuring the GNSGA-III parameters for the final experiments and results analysis.

D. RESULTS ANALYSIS

Experiments begin with solving the test instances that are configured by symmetric job sets. According to Tables 2-5, GNSGA-III is associated with larger A, B, B/A, Ω, and smaller C and γ in all of the test instances. More precisely, it is observed that the non-dominated Pareto set provided by GNSGA-III includes more solutions, much of which are unique in the benchmark tests and are not found by the other algorithms (see Figures 3-8). It is also observed that Ω values experience an increase when problems with larger workloads are considered; that is, GNSGA-III's superiority over the WSPT-EDD algorithm becomes more apparent when larger problems are sought. From another viewpoint, zero and close-to-zero C values confirm that the GNSGA-III algorithm results in a negligible number of weak solutions in the benchmark tests. In terms of solution diversity, the performance of the GNSGA-III is slightly better than the WSPT-EDD algorithm with the majority of the cases reporting equal and near-to-equal Δ values. Overall, GNSGA-III yields better solutions in the majority of the cases, as confirmed by the significantly smaller γ indicator.

Tables 6-7 summarize the test results for the instances with asymmetric job sets. The difference between the performance of GNSGA-III and WSPT-EDD algorithms becomes even more apparent when one of the interfering job sets is associated with higher workloads. GNSGA-III outperforms the WSPT-EDD algorithm over all the test instances considering A, B, B/A, Ω, C, and γ measures. The number of solutions and the unique solutions by each of the benchmark algorithms is visualized in Figures 7-8.

TABLE 6. Results Comparison Over 10-30 Test Instances (Best in Bold).

Indicator	Algorithm	Instance			
		I	II	III	IV
A	GNSGA-III	9.6	**11.8**	9.4	**10.2**
	WSPT-EDD	9.6	10.0	9.4	10.0
B	GNSGA-III	9.6	**11.8**	9.4	**10.2**
	WSPT-EDD	9.6	9.2	9.4	10.0
B/A	GNSGA-III	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	WSPT-EDD	**1.0000**	0.9192	**1.0000**	**1.0000**
Ω	GNSGA-III	0.5000	**0.5540**	0.5000	**0.5043**
	WSPT-EDD	0.5000	0.4460	0.5000	0.4957
C	GNSGA-III	0.0000	**0.0000**	0.0000	0.0000
	WSPT-EDD	0.0000	0.0808	0.0000	0.0000
γ	GNSGA-III	0.0000	**0.0000**	0.0000	0.0000
	WSPT-EDD	0.0000	18.7051	0.0000	0.0000
Δ	GNSGA-III	**0.3870**	**0.5748**	**0.4583**	0.5040
	WSPT-EDD	**0.3870**	0.6037	**0.4583**	**0.4904**

TABLE 7. Results Comparison Over 30-10 Test Instances (Best in Bold).

Indicator	Algorithm	Instance			
		I	II	III	IV
A	GNSGA-III	**31.2**	**29.2**	**30.8**	**35.2**
	WSPT-EDD	22.4	16.0	20.2	17.8
B	GNSGA-III	**30.2**	**29.0**	**30.4**	**34.6**
	WSPT-EDD	14.6	11.2	15.4	11.2
B/A	GNSGA-III	**0.9690**	**0.9933**	**0.9863**	**0.9842**
	WSPT-EDD	0.6481	0.6959	0.7661	0.6290
Ω	GNSGA-III	**0.6716**	**0.7212**	**0.6626**	**0.7552**
	WSPT-EDD	0.3284	0.2788	0.3374	0.2448
C	GNSGA-III	**0.0310**	**0.0067**	**0.0137**	**0.0158**
	WSPT-EDD	0.3436	0.3041	0.2239	0.3710
γ	GNSGA-III	**0.4683**	**0.1467**	**0.7097**	**0.5105**
	WSPT-EDD	13.9153	17.9386	14.6354	17.8289
Δ	GNSGA-III	0.6460	**0.6515**	**0.6587**	**0.5920**
	WSPT-EDD	**0.6179**	0.7702	0.7485	0.6878

As a final step to the numerical analysis, statistical test of significance is conducted to confirm our assertion that the developed GNSGA-III in this study is the best-performing algorithm developed to solve the $1|inter|ND(\sum w_jC_j, L_{\max})$ problem. Table 8 shows the test of significance with a negligible p-value for the A, B, B/A, Ω, C, and γ measures. Although the difference between the Δ values appears to be insignificant among the instances with symmetric job sets, GNSGA-III and WSPT-EDD perform equally good in terms of solution diversity with the majority of the Δ values about 0.5. It is also observed that GNSGA-III yields more diverse solutions when one of the conflicting objectives is associated with a higher workload.

IV. CONCLUSION

Flexible services and manufacturing require well-informed scheduling decisions that consider various customer needs and operational goals that may be conflicting in nature. This study contributes to the multi-objective scheduling problems literature, developing an effective metaheuristic to solve the $1|inter|ND(\sum w_jC_j, L_{\max})$ problem considering TWCT and maximum lateness performance indicators for interfering job sets.

Extensive numerical tests showed that GNSGA-III solutions are of high quality and comparable to the best-found solution obtained by the benchmark algorithm in the literature. When comparing to the WSPT-EDD algorithm,

TABLE 8. Statistical analysis considering various indicators.

Indicator	Algorithm	Ave.	Var.	DoF	*t-test*	*p-value (two-tail)*
A	GNSGA-III WSPT-EDD	20.8417 32.85	65.4789 390.9521	119	-8.2663	2.24E-13
B	GNSGA-III WSPT-EDD	16.6167 31.4333	63.0115 309.1384	119	-10.4832	1.30E-18
B/A	GNSGA-III WSPT-EDD	0.8042 0.9753	0.0307 0.0017	119	-10.7459	3.07E-19
Ω	GNSGA-III WSPT-EDD	0.3689 0.6310	0.0124 0.0124	119	-12.8732	2.73E-24
C	GNSGA-III WSPT-EDD	0.1942 0.0194	0.0305 0.001	119	11.2427	2.00E-20
γ	GNSGA-III WSPT-EDD	16.4391 2.9848	383.0934 70.4725	119	7.9341	1.30E-12
Δ	GNSGA-III WSPT-EDD	0.6232 0.6320	0.1813 0.1905	119	-0.7091	4.80E-01

the GNSGA-III algorithm yields more and better non-dominated solutions, providing the decision-makers with wider and more dependable trade-offs. The proposed metaheuristic offers more effectiveness compared to the WSPT-EDD algorithm as a constructive heuristic, at the expense of longer computational times. Overall, the multifaceted performance comparison and statistical analysis confirm that the developed GNSGA-III algorithm can be used as a strong benchmark M-O algorithm in the literature of SSP-IJs over both symmetric and asymmetric test instances.

Interfering jobs are prevalent in various production environments while scheduling problems reflecting this real-world situation are relatively limited. We feel that this scheduling feature requires significantly more development given its implications for modern production and operations management. Hybrid and parallel flowshops with interfering jobs are worthwhile research directions to pursue. Besides, scheduling problems with interfering jobs can be tested while considering mixed job-related constraints, like no-wait, blocking, as well as setup time features, to address case-specific operational needs. Given the complexities involved in the SSP-IJs problem, other multi-objective metaheuristics can be developed to improve the results obtained by the GNSGA-III algorithm to facilitate its industry-scale applications in various production conditions.

REFERENCES

[1] N. C.-A. Lee, E. T. G. Wang, and V. Grover, "IOS drivers of manufacturer-supplier flexibility and manufacturer agility," *J. Strategic Inf. Syst.*, vol. 29, no. 1, Mar. 2020, Art. no. 101594.

[2] P. Pourhejazy, J. Sarkis, and Q. Zhu, "Product deletion as an operational strategic decision: Exploring the sequential effect of prominent criteria on decision-making," *Comput. Ind. Eng.*, vol. 140, Feb. 2020, Art. no. 106274, doi: 10.1016/j.cie.2020.106274.

[3] K. R. Baker and J. C. Smith, "A multiple-criterion model for machine scheduling," *J. Sched.*, vol. 6, no. 1, pp. 7–16, Jan. 2003.

[4] A. Agnetis, P. B. Mirchandani, D. Pacciarelli, and A. Pacifici, "Scheduling problems with two competing agents," *Oper. Res.*, vol. 52, no. 2, pp. 229–242, Apr. 2004.

[5] C. T. Ng, T. C. E. Cheng, and J. J. Yuan, "A note on the complexity of the problem of two-agent scheduling on a single machine," *J. Combinat. Optim.*, vol. 12, no. 4, pp. 387–394, Oct. 2006.

[6] P. Pourhejazy and O. Kwon, "The new generation of operations research methods in supply chain optimization: A review," *Sustainability*, vol. 8, no. 10, p. 1033, Oct. 2016, doi: 10.3390/su8101033.

[7] Y. Yin, Y. Wang, T. C. E. Cheng, D.-J. Wang, and C.-C. Wu, "Two-agent single-machine scheduling to minimize the batch delivery cost," *Comput. Ind. Eng.*, vol. 92, pp. 16–30, Feb. 2016.

[8] Y. Yin, Y. Yang, D. Wang, T. C. E. Cheng, and C.-C. Wu, "Integrated production, inventory, and batch delivery scheduling with due date assignment and two competing agents," *Nav. Res. Logistics*, vol. 65, no. 5, pp. 393–409, Aug. 2018.

[9] Y. Yin, W. Wang, D. Wang, and T. C. E. Cheng, "Multi-agent single-machine scheduling and unrestricted due date assignment with a fixed machine unavailability interval," *Comput. Ind. Eng.*, vol. 111, pp. 202–215, Sep. 2017.

[10] A. Agnetis, P. B. Mirchandani, D. Pacciarelli, and A. Pacifici, "Nondominated schedules for a job-shop with two competing users," *Comput. Math. Organ. Theory*, vol. 6, no. 2, pp. 191–217, 2000.

[11] H. Balasubramanian, J. Fowler, A. Keha, and M. Pfund, "Scheduling interfering job sets on parallel machines," *Eur. J. Oper. Res.*, vol. 199, no. 1, pp. 55–67, Nov. 2009.

[12] M. Torkashvand, B. Naderi, and S. A. Hosseini, "Modelling and scheduling multi-objective flow shop problems with interfering jobs," *Appl. Soft Comput.*, vol. 54, pp. 221–228, May 2017.

[13] P. Perez-Gonzalez and J. M. Framinan, "A common framework and taxonomy for multicriteria scheduling problems with interfering and competing jobs: Multi-agent scheduling problems," *Eur. J. Oper. Res.*, vol. 235, no. 1, pp. 1–16, May 2014.

[14] J. Y.-T. Leung, M. Pinedo, and G. Wan, "Competitive two-agent scheduling and its applications," *Oper. Res.*, vol. 58, no. 2, pp. 458–469, Apr. 2010.

[15] D.-J. Wang, Y. Yin, S.-R. Cheng, T. C. E. Cheng, and C.-C. Wu, "Due date assignment and scheduling on a single machine with two competing agents," *Int. J. Prod. Res.*, vol. 54, no. 4, pp. 1152–1169, Feb. 2016.

[16] Z. Xingong and W. Yong, "Two-agent scheduling problems on a single-machine to minimize the total weighted late work," *J. Combinat. Optim.*, vol. 33, no. 3, pp. 945–955, Apr. 2017.

[17] D. Oron, D. Shabtay, and G. Steiner, "Single machine scheduling with two competing agents and equal job processing times," *Eur. J. Oper. Res.*, vol. 244, no. 1, pp. 86–99, Jul. 2015.

[18] C.-Y. Cheng, S.-F. Li, K.-C. Ying, and Y.-H. Liu, "Scheduling jobs of two competing agents on a single machine," *IEEE Access*, vol. 7, pp. 98702–98714, 2019.

[19] K. Khowala, J. Fowler, A. Keha, and H. Balasubramanian, "Single machine scheduling with interfering job sets," *Comput. Oper. Res.*, vol. 45, pp. 97–107, May 2014.

[20] C. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed. New York, NY, USA: Springer, 2007.

[21] N. Srinivas and K. Deb, "Muiltiobjective optimization using nondominated sorting in genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, Sep. 1994.

[22] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[23] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using Reference-Point-Based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014, doi: 10.1109/TEVC.2013.2281535.

[24] P. Pourhejazy, O. K. Kwon, and H. Lim, "Integrating sustainability into the optimization of fuel logistics networks," *KSCE J. Civil Eng.*, vol. 23, no. 3, pp. 1369–1383, Mar. 2019.

[25] B. Yu, Z. Peng, Z. Tian, and B. Yao, "Sailing speed optimization for tramp ships with fuzzy time window," *Flexible Services Manuf. J.*, vol. 31, no. 2, pp. 308–330, Jun. 2019.

[26] Y. Rao, R. Meng, J. Zha, and X. Xu, "Bi-objective mathematical model and improved algorithm for optimisation of welding shop scheduling problem," *Int. J. Prod. Res.*, vol. 58, no. 9, pp. 2767–2783, 2019.

[27] X. He, S. Dong, and N. Zhao, "Research on rush order insertion rescheduling problem under hybrid flow shop based on NSGA-III," *Int. J. Prod. Res.*, vol. 58, no. 4, pp. 1161–1177, Feb. 2020.

[28] D. Wang, F. Liu, Y. Yin, J. Wang, and Y. Wang, "Prioritized surgery scheduling in face of surgeon tiredness and fixed off-duty period," *J. Combinat. Optim.*, vol. 30, no. 4, pp. 967–981, Nov. 2015.

[29] H. Qiu, D. Wang, Y. Wang, and Y. Yin, "MRI appointment scheduling with uncertain examination time," *J. Combinat. Optim.*, vol. 37, no. 1, pp. 62–82, Jan. 2019.

[30] X. Bi and C. Wang, "An improved NSGA-III algorithm based on objective space decomposition for many-objective optimization," *Soft Comput.*, vol. 21, no. 15, pp. 4269–4296, Aug. 2017.

[31] R. Ruiz and T. Stützle, "A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem," *Eur. J. Oper. Res.*, vol. 177, no. 3, pp. 2033–2049, Mar. 2007.

[32] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms*. Hoboken, NJ, USA: Wiley, 2004.

[33] F. Alabsi and R. Naoum, "Comparison of selection methods and crossover operations using steady state genetic based intrusion detection system," *J. Emerg. Trends Comput. Inf. Sci.*, vol. 3, no. 7, pp. 1053–1058, 2012.

[34] D. E. Goldberg and R. Lingle, "Alleles, loci, and the traveling salesman problem," in *Proc. Int. Conf. genetic Algorithms Appl.*, vol. 154, 1985, pp. 154–159.

[35] K.-C. Ying, S.-W. Lin, and S.-Y. Wan, "Bi-objective reentrant hybrid flowshop scheduling: An iterated Pareto greedy algorithm," *Int. J. Prod. Res.*, vol. 52, no. 19, pp. 5735–5747, Oct. 2014.

[36] S.-W. Lin and K.-C. Ying, "A multi-point simulated annealing heuristic for solving multiple objective unrelated parallel machine scheduling problems," *Int. J. Prod. Res.*, vol. 53, no. 4, pp. 1065–1076, Feb. 2015.

CHEN-YANG CHENG received the Ph.D. degree in industrial and manufacturing engineering from Penn State University. He is currently a Professor with the Department of Industrial Engineering and Management, National Taipei University of Technology. His research interests include computer integrated manufacturing, human–computer interaction, distributed systems and control, and intelligent systems.

SHIH-WEI LIN received the bachelor's, master's, and Ph.D. degrees in industrial management from the National Taiwan University of Science and Technology, Taiwan, in 1996, 1998, and 2000, respectively. He is currently a Professor with the Department of Information Management, Chang Gung University, Taiwan. He is also with the Department of Neurology, Linkuo Chang Gung Memorial Hospital, Taoyuan City, Taiwan, and with the Department of Industrial Engineering and Management, Ming Chi University of Science and Technology, Taipei, Taiwan. His current research interests include meta-heuristics and data mining. His articles have appeared in *Computers & Operations Research*, the *European Journal of Operational Research*, the *Journal of the Operational Research Society*, the *European Journal of Industrial Engineering*, the *International Journal of Production Research*, the *International Journal of Advanced Manufacturing Technology*, *Knowledge and Information Systems*, *Applied Soft Computing*, *Applied Intelligence*, *Expert Systems with Applications*, and IEEE Access.

POURYA POURHEJAZY received the master's degree in industrial engineering from University Technology Malaysia, in 2014, and the Ph.D. degree in logistics engineering from INHA University, South Korea, in 2017. He is currently an Adjunct Assistant Professor and a Research Assistant Professor with the National Taipei University of Technology, Taiwan. His research interests include optimization and decision analysis in the supply chain, production, and transportation contexts.

KUO-CHING YING is currently a Distinguished Professor with the Department of Industrial Engineering and Management, National Taipei University of Technology. He is the (senior) Editor of ten international journals. His research interests include the operations scheduling and combinatorial optimization, in which he has published over 100 academic research articles in refereed international journals, such as *Applied Intelligence*, *Applied Soft Computing*, *Computers and Operations Research*, *Computers and Industrial Engineering*, the *European Journal of Industrial Engineering*, the *European Journal of Operational Research*, IEEE Access, the *International Journal of Production Economics*, the *International Journal of Advanced Manufacturing Technology*, the *International Journal of Innovative Computing*, *Information and Control*, the *International Journal of Production Research*, the *Journal of the Operational Research Society*, the *OMEGA-The International Journal of Management Sciences*, *Production Planning & Control*, and *Transportation Research Part E: Logistics and Transport Review*, among others.

SHU-FEN LI received the Ph.D. degree in industrial engineering and enterprise information from Tunhai University, Taiwan, in 2017. She is currently an Assistant Professor with the Department of Industrial Engineering and Management, National Chin-Yi University of Technology. Her research interests include operations scheduling, computational intelligence, data mining, and lean production.

YING-CHUN LIU received the master's degree from the Industrial Engineering and Management, National Taipei University of Technology, Taiwan. She is currently an Engineer with the Hsinchu Science Park Burau, Ministry of Science and Technology of Taiwan. Her research interests include operations research and production scheduling.