Extracting Structural Knowledge from Precedence-Induced Betweenness Patterns in the Job Shop Scheduling Problem

Isabel Catalá[1], **Christian Pérez**[1], **Miguel A. Salido**[1]

[1]Universitat Politècnica de València
{ icatfor, cripeber, misagre }@upv.es

Abstract

Betweenness-based descriptors have recently emerged as informative structural signals in feature-based analyses of the Job Shop Scheduling Problem (JSP), yet the precedence patterns that generate these signals remain poorly understood. Prior work shows that graph-derived descriptors can correlate with hardness, algorithm selection, or scheduling efficiency, but it typically relies on sampled benchmark sets and therefore does not reveal which concrete precedence motifs produce the observed centrality patterns. This paper addresses that gap through an exhaustive, fully controlled microspace study. We enumerate complete JSP precedence spaces in the classical 3×3 and 4×4 settings, assigning one of the $n!$ machine-order permutations to each of the n jobs, which yields spaces of size $6^3 = 216$ and $24^4 = 331{,}776$, respectively. All non-structural values are fixed so that only the precedence structure varies. For each profile, we build the induced graph, compute node betweenness, and summarize it in an $n \times n$ betweenness matrix, along with its mean, range, symmetry, and repetition signature. The analysis reveals strong structural compression and stable cross-size trends: fully aligned profiles maximize average mediation, while fully diverse profiles flatten the map and minimize it. In the richer 4×4 space, partially disrupted profiles, especially of type $3{+}1$, emerge as a distinct regime with the strongest heterogeneity, lower symmetry, and more distinctive solver-side behavior. These findings turn betweenness from a merely predictive feature into an interpretable object of knowledge extraction, with implications for benchmark design, motif-aware feature engineering, and structural diagnostics in scheduling.

Introduction

Nominal size is a weak proxy for structural behavior in scheduling. Even within the classical Job Shop Scheduling Problem (JSP), instances with the same number of jobs and machines can induce different precedence bottlenecks, different conflict structures, and different levels of algorithmic effort (Błażewicz, Domschke, and Pesch 2000; Pinedo 2016). This observation has motivated a broad literature on feature-based instance characterization, hardness estimation, and algorithm selection (Smith-Miles and van Hemert 2012; Smith-Miles and Lopes 2014), where the central idea is that structural descriptors can explain performance differences that remain invisible to (n, m) alone (Strassl and Musliu 2022; Mirshekarian and Sormaz 2016).

Graph-based descriptors are particularly appealing in JSP because the disjunctive-graph representation separates technological precedence from resource conflicts and exposes a compact structural backbone (Błażewicz, Domschke, and Pesch 2000). Recent work on instance-space analysis and explainable performance modeling suggests that summaries derived from graph structure can be highly informative. However, feature importance alone is not an explanation (Abdelmaguid 2009). Knowing that betweenness is useful for predicting or ranking instances is not the same as understanding which precedence arrangements produce high or low betweenness, or which recurring motifs control structural heterogeneity inside the graph (Strassl and Musliu 2022).

This paper focuses precisely on that missing structural layer. Rather than analyzing a large, heterogeneous corpus, we study complete-precedence micro-spaces in which all non-structural factors are fixed. Concretely, we consider exhaustive 3×3 and 4×4 JSP settings, where each job may follow any machine-order permutation. This yields two fully enumerable profile spaces, of sizes $6^3 = 216$ and $24^4 = 331{,}776$, respectively. Because all non-structural values are fixed within each space, every observed difference in the resulting betweenness patterns can be attributed to the precedence profile itself. This provides a controlled environment for moving from "betweenness is important" to "these precedence motifs create these betweenness signatures."

The paper's contribution is methodological rather than algorithmic. We do not propose a new solver, predictor, or benchmark generator. Instead, we use exhaustive enumeration to extract reusable structural knowledge from precedence spaces. More specifically, the paper makes four contributions. First, it defines and exhaustively traverses two complete precedence spaces for classical JSP, namely the 3×3 and 4×4 cases. Second, it introduces a betweenness-based representation of each configuration via a betweenness map, along with compact scalar summaries and repetition signatures. Third, it shows that the raw precedence spaces contain strong hidden redundancy, so that many distinct permutation profiles collapse to a much smaller set of structural classes. Fourth, it identifies a consistent pattern across both settings:

full alignment maximizes average mediation, full diversity minimizes it, and, in the richer 4×4 case, partial disruption emerges as a distinct regime with the strongest structural heterogeneity and propagation activity.

The remainder of the paper is organized as follows. The Related Work section positions the present study with respect to disjunctive-graph analyses, instance-space studies, explainable scheduling, and generator-based benchmark design. The Problem Formulation and Graph Definition section introduces the JSP formulation, the graph representation, and the exhaustive instance-space setup. The Results section reports the empirical findings for the 3×3 and 4×4 spaces and highlights the main structural regularities. Finally, the Conclusions section concludes the paper and outlines future research directions.

Related Work

The disjunctive graph is the standard structural representation of the classical JSP. It models each operation as a node, job precedence as conjunctive arcs, and machine conflicts as disjunctive edges (Błażewicz, Domschke, and Pesch 2000; Pinedo 2016). This representation is useful not only for exact and heuristic optimization but also for structural diagnosis, as it makes explicit how technological order and resource contention interact. Alternative formulations also show that precedence structure can be analysed independently of solver details. For example, Abdelmaguid (2009) proposed permutation-induced acyclic networks as an alternative network view of JSP structure. Earlier work by Chu, Proth, and Wang (1998) studied how critical arc reversals affect makespan, illustrating that local precedence changes can have interpretable structural effects on solution quality. These works motivate our focus on precedence patterns as first-class structural objects.

The closest methodological background comes from instance-space analysis (ISA) and feature-driven hardness modelling. For JSP, Strassl and Musliu (2022) showed that feature-based representations reveal meaningful regions of the instance space and can support algorithm selection. Broader feature-engineering studies also found that rich structural descriptors correlate with scheduling efficiency and can be used to predict coarse performance classes (Mirshekarian and Sormaz 2016). More generally, feature-based knowledge discovery and difficulty analysis have long argued that instance properties should be connected to algorithm behaviour through explicit descriptors rather than through nominal size alone (Smith-Miles and van Hemert 2012; Smith-Miles and Lopes 2014). Our work aligns with that agenda, but differs in scope: instead of learning over broad benchmark sets, we exhaustively enumerate a complete small precedence space to isolate deterministic structural regularities.

Several studies have used network descriptors to interpret scheduling or manufacturing systems. Centrality measures have been employed to identify bottlenecks or vital nodes in scheduling networks and job-shop manufacturing settings (Zhuang et al. 2020; Yin, Xu, and Liu 2022). These studies support the idea that centrality can encode meaningful structural stress, but they do not analyse the complete precedence space of classical JSP. On the explainability side, Čyras et al. (2019) showed that scheduling decisions can be given formal, human-readable explanations. In parallel, controlled instance generators seek to increase structural diversity or tailor benchmark characteristics (March, Pérez, and Salido 2024; Vela et al. 2021). Our paper sits at the intersection of these lines: it uses exhaustive enumeration as a knowledge-extraction instrument and turns graph centrality into an interpretable catalogue of recurring precedence motifs, which is also consistent with knowledge-engineering perspectives on planning and scheduling models (Silva, Silva, and Vaquero 2020).

The main gap is therefore clear. Prior work shows that features matter, that centrality can be meaningful, and that instance diversity matters for benchmarking. What remains largely missing is an exhaustive mapping from complete precedence spaces to interpretable betweenness signatures in classical JSP. That is the gap addressed here.

Problem Formulation and Graph Definition

We first recall the classical mixed-integer linear programming (MILP) formulation of the JSP, then define its graph representation, and finally specialize the framework to the exhaustive $n \times n$ settings studied in this paper. This ordering is useful because the exhaustive experiments should be read as controlled families of instances of the same underlying optimization problem, rather than as separate combinatorial objects.

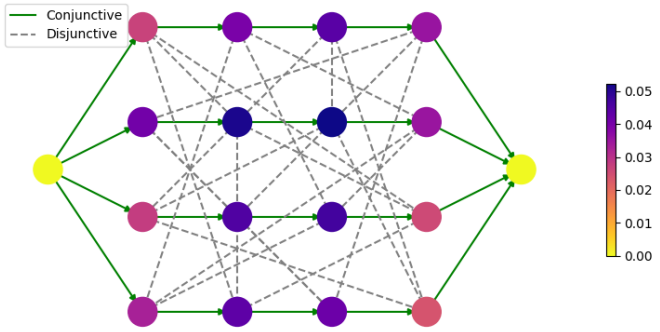

Figure 1: Graph representation of a 4×4 instance and its betweenness values for each operation node.

Sets and Parameters

We consider the classical $n \times n$ JSP, where the number of jobs and machines is the same. Let J be the set of jobs, let M be the set of machines, and let O be the set of operations. Each operation belongs to exactly one job and must be processed on exactly one machine. For each job $j \in J$, let $O_j \subseteq O$ denote the ordered sequence of its operations, and for each machine $m \in M$, let $O_m \subseteq O$ denote the set of operations processed on m. The precedence arcs inside jobs are collected in the set A, while machine-conflict pairs are collected in the set Q, as defined in Equation (1).

$$J = \{1, \dots, n\} \qquad M = \{1, \dots, n\} \qquad O = \bigcup_{j \in J} O_j$$
$$A \subseteq O \times O \qquad Q = \bigcup_{m \in M} \big\{\{u, v\} : u, v \in O_m,\ u < v\big\} \tag{1}$$

Equation (1) defines the basic structural ingredients of the model. The set A contains the technological precedences inside each job, while the set Q contains the unordered pairs of operations that compete for the same machine and therefore require a sequencing decision. For each operation $o \in O$, we use a processing-time parameter $p_o > 0$ and a sufficiently large constant H as a time horizon.

Decision Variables

The MILP uses a continuous start-time variable s_o for each operation $o \in O$, a continuous makespan variable $C_{\max}$, and a binary ordering variable x_{uv} for each machine-conflict pair $\{u, v\} \in Q$. These variables are defined in Equation (2).

$$\begin{aligned} &s_o \geq 0 && \forall o \in O \\ &C_{\max} \geq 0 \qquad x_{uv} \in \{0, 1\} && \forall \{u, v\} \in Q \end{aligned} \tag{2}$$

In Equation (2), the interpretation of x_{uv} is standard: $x_{uv} = 1$ means that operation u is scheduled before operation v, whereas $x_{uv} = 0$ means the reverse order.

Objective Function and Constraints

The JSP objective is to minimize the makespan while satisfying both technological precedence constraints and machine-capacity constraints. The complete MILP core used in this work is given by Equations (3)–(7).

$$\min C_{\max} \tag{3}$$
$$C_{\max} \geq s_o + p_o \qquad \forall o \in O \tag{4}$$
$$s_v \geq s_u + p_u \qquad \forall (u, v) \in A \tag{5}$$
$$s_v \geq s_u + p_u - \mathcal{H}(1 - x_{uv}) \qquad \forall \{u, v\} \in Q \tag{6}$$
$$s_u \geq s_v + p_v - \mathcal{H} x_{uv} \qquad \forall \{u, v\} \in Q \tag{7}$$

Equation (3) defines the makespan minimization objective. Equation (4) links the makespan variable to the completion time of every operation, ensuring that $C_{\max}$ is at least as large as the latest operation completion. Equation (5) enforces the technological order within each job: if $(u, v) \in A$, then operation v cannot start before operation u has finished.

Machine capacity is encoded through the standard big-M disjunctive formulation in Equations (6) and (7). For each conflicting pair $\{u, v\} \in Q$, exactly one ordering must hold. Equation (6) activates the ordering $u \prec v$ when $x_{uv} = 1$, whereas Equation (7) activates the reverse ordering $v \prec u$ when $x_{uv} = 0$. Together, Equations (3)–(7) define the classical JSP MILP that serves as the optimization backbone of the study. In the exhaustive experiments considered later, the non-structural values remain fixed, and the only varying component is the job-precedence structure induced by the machine-order permutations.

Conjunctive and Disjunctive Graphs

We represent each JSP instance by its standard graph structure. Let s and t denote a source and a sink node, respectively, and let $V = O \cup \{s, t\}$ be the node set. The graph contains two edge families: conjunctive arcs, which encode job precedences, and disjunctive arcs, which encode machine conflicts. The conjunctive graph is defined in Equation (8), while the disjunctive graph is defined in Equation (9).

$$\begin{aligned} G^c &= (V, E_c) \\ E_c &= \{(s, o_{j1}) : j \in J\} \\ &\quad \cup \{(o_{jk}, o_{j,k+1}) : j \in J,\ k = 1, \dots, n-1\} \\ &\quad \cup \{(o_{jn}, t) : j \in J\} \end{aligned} \tag{8}$$

Equation (8) defines the *conjunctive graph*, which captures the technological order inside each job from source to sink.

$$\begin{aligned} G^d &= (V, E_d) \\ E_d &= \bigcup_{m \in M} \{(u, v), (v, u) : u, v \in O_m,\ u < v\} \end{aligned} \tag{9}$$

Equation (9) defines the *disjunctive graph*. For every pair of operations competing for the same machine, both possible arc directions are represented. A feasible schedule can then be interpreted as a fitting orientation of these machine-conflict arcs.

Under the $n \times n$ setting considered here, graph size is invariant across all precedence profiles. The number of nodes and arcs is given in Equation (10).

$$\begin{aligned} &|V| = n^2 + 2 && |E| = n^3 + n \\ &|E_c| = n(n+1) && |E_d| = n^2(n-1) \end{aligned} \tag{10}$$

Equation (10) shows that, for fixed n, graph size remains constant and only the arrangement of precedence information changes.

Instance $n \times n$ Space Setup

We now specialize the previous formulation to exhaustive $n \times n$ instance spaces. Each job contains exactly one operation per machine, and the only source of structural variation is the order in which each job visits the machines. Let $\mathcal{P}_n$ denote the set of all machine-order permutations of size n. A precedence profile is defined in Equation (11), the full profile space is defined in Equation (12), and the two concrete exhaustive settings used in this paper are given in Equation (13).

$$\pi = (\pi_1, \dots, \pi_n) \qquad \pi_j \in \mathcal{P}_n \tag{11}$$
$$|\Pi_n| = |\mathcal{P}_n|^n = (n!)^n \tag{12}$$
$$|\Pi_3| = 6^3 = 216 \qquad |\Pi_4| = 24^4 = 331{,}776 \tag{13}$$

Equation (11) defines a precedence profile as an n-tuple of job permutations, one per job. Equation (12) then shows that

	m_0	m_1	m_2	m_3
j_0	0	1	2	3
j_1	0	1	2	3
j_2	0	1	2	3
j_3	0	1	2	3

(a) All jobs share the same machine-order permutation.

	m_0	m_1	m_2	m_3
j_0	0	1	2	3
j_1	0	1	2	3
j_2	0	1	2	3
j_3	0	1	3	2

(b) Only one job has a different machine-order permutation.

	m_0	m_1	m_2	m_3
j_0	0	1	2	3
j_1	0	1	3	2
j_2	0	2	1	3
j_3	0	3	1	2

(c) All jobs have different machine-order permutations.

Table 1: Representative precedence profiles in the exhaustive 4×4 space: full alignment, partial disruption, and full diversity.

the full profile space Π_n contains all possible combinations of these machine orders, and therefore grows as $(n!)^n$. Finally, Equation (13) gives the two fully enumerable spaces studied in this paper, namely the 3×3 and 4×4 cases. Table 1 collects three representative profiles from Π_4: full alignment, partial disruption, and full diversity, corresponding to Tables 1a, 1b, and 1c, respectively.

All non-structural values are fixed within each exhaustive space. This is deliberate: by keeping non-structural values constant, every observed change in the induced graph can be attributed solely to the precedence profile. The experiments are therefore designed to isolate the structural effect of precedence, allowing the subsequent analysis to focus on how machine-order configurations shape the graph structure and the resulting betweenness patterns.

Betweenness Maps and Structural Signatures

For each precedence profile π, we compute node betweenness centrality on the precedence projection of the JSP graph. Since each operation is uniquely identified by a job–machine pair, let $o_{jm} \in O$ denote the operation of job j processed on machine m. We then define the betweenness value associated with position (j, m) directly through the corresponding operation node. The resulting scalar descriptors and structural signatures are given in Equations (14)–(20).

$$\mathcal{B}_\pi(j,m) = \sum_{s,t \in O} \frac{\sigma_\pi(s,t \mid o_{jm})}{\sigma_\pi(s,t)} \qquad \forall j \in J \quad (14)$$

$$\mathbf{M}_\pi = \big(\mathcal{B}_\pi(j,m)\big)_{j \in J,\ m \in M} \quad (15)$$

$$\mu(\pi) = \frac{1}{n^2} \sum_{j=1}^{n} \sum_{m=1}^{n} \mathcal{B}_\pi(j,m) \quad (16)$$

$$R(\pi) = \max_{j,m} \mathcal{B}_\pi(j,m) - \min_{j,m} \mathcal{B}_\pi(j,m) \quad (17)$$

$$S(\pi) = 1.0 - \frac{\left\|\mathbf{M}_\pi - \mathbf{M}_\pi^T\right\|_F}{\|\mathbf{M}_\pi\|_F + \epsilon} \quad (18)$$

$$D(\pi) = \left|\{\pi_1, \ldots, \pi_n\}\right| \quad (19)$$

$$\Sigma(\pi) = \operatorname{sort}\left(\{|C| : C \in \mathcal{C}(\pi)\}\right) \quad (20)$$

Equation (14) defines the betweenness value of the operation located at position (j, m), where $\sigma_\pi(s, t)$ denotes the number of shortest paths between nodes s and t in the precedence projection induced by profile π, and $\sigma_\pi(s, t \mid o_{jm})$ denotes the number of such paths that pass through operation node o_{jm}. Equation (15) then defines the *betweenness map* of profile π, which preserves the spatial organization of mediation across job and machine positions. Equation (16) defines the mean $\mu(\pi)$, which measures the overall concentration of precedence paths, while Equation (17) defines the range $R(\pi)$, which measures how unevenly this mediation is distributed across positions. A large value of $\mu(\pi)$ indicates globally stronger mediation, whereas a large value of $R(\pi)$ indicates a more polarized map.

Equation (18) defines a numerical value, $S(\pi) \in [0, 1]$, measuring the symmetry level of the betweenness map associated with the precedence profile π. It is based on the normalized Frobenius distance between the map and its transpose: higher values indicate more balanced betweenness patterns, whereas lower values indicate stronger asymmetries and more polarized structures. Equation (19) defines the diversity $D(\pi)$ as the number of distinct job permutations appearing in the profile. Equation (20) defines the repetition signature $\Sigma(\pi)$ through the partition $\mathcal{C}(\pi)$ induced by permutation equality. This repetition signature compresses each profile into an interpretable motif family. In the exhaustive settings studied here, the possible signatures are 3, 2+1, and 1+1+1 for the 3×3 case, and 4, 3+1, 2+2, 2+1+1, and 1+1+1+1 for the 4×4 case.

Results

This section reports the main empirical findings of the exhaustive 3×3 and 4×4 studies. The instances were generated using IGJSP, a reproducible instance generator for the Job Shop Scheduling Problem (Perez, March, and Salido 2025), under a controlled configuration in which the only varying component is the precedence profile. Our argument has three layers. First, repetition signatures provide a strong compression of the raw precedence spaces while retaining meaningful structural variation. Second, these signatures exhibit stable trends across both sizes, especially in terms of average betweenness and lower-bound behaviour. Third, the richer 4×4 space reveals an additional regime that is not yet fully expressed in 3×3: partial disruption, which combines lower symmetry with stronger structural heterogeneity and a more distinctive solver profile.

Structural Compression Through Repetition Signatures

The exhaustive spaces contain 216 raw precedence profiles in the 3×3 case and 331,776 in the 4×4 case. In both settings, however, the raw number of profiles overstates the effective structural diversity of the space. We therefore compress each profile using the repetition signature defined in

Equation (20), which records only how many jobs share the same machine-order permutation.

This descriptor provides an intermediate structural layer between raw permutation assignments and full betweenness maps. It is coarser than the exact profile, but still expressive enough to distinguish meaningful structural regimes. In the 3×3 setting, the possible signatures are 3, 2+1, and 1+1+1. In the 4×4 setting, they are 4, 3+1, 2+2, 2+1+1, and 1+1+1+1. Thus, the signature space grows with problem size, but remains very small compared with the raw permutation space.

$$N(\lambda) = \frac{(K)_r}{\prod_a m_a!} \frac{n!}{\prod_{i=1}^{r} \lambda_i!} \qquad K = n! \quad (21)$$

where $\lambda = (\lambda_1, \ldots, \lambda_r)$ is a repetition signature viewed as a partition of n, r is the number of distinct permutations used in the profile, $(K)_r = K(K-1)\cdots(K-r+1)$ is the falling factorial, and m_a is the multiplicity of the part value a inside the partition (O'Neill 2021). Equation (21) shows that the frequencies of the signatures are not arbitrary: they result from combining the choice of r distinct permutations among the $n!$ available ones with the multinomial allocation of their multiplicities across jobs, while correcting for repeated block sizes such as 2+2 or 1+1+1+1.

Configuration	Signature	$N(\lambda)$
3×3	3	6
	2+1	90
	1+1+1	120
4×4	4	24
	3+1	2,208
	2+2	1,656
	2+1+1	72,864
	1+1+1+1	255,024

Table 2: Repetition signatures and their frequencies in the exhaustive 3×3 and 4×4 spaces.

Table 2 reflects the combinatorial bias implied by Equation (21). Highly repetitive signatures are rare because they require several jobs to reuse the same permutation, whereas fully diverse signatures are much more numerous because they correspond to selecting different permutations for all jobs. This explains why 1+1+1 already dominates the 3×3 space, and why 1+1+1+1 dominates even more strongly in 4×4.

The same counts therefore have a structural interpretation: the exhaustive spaces are naturally biased toward diversity, while strong repetition occupies only a small fraction of the raw profile space. This makes the signature representation useful in two ways. First, it compresses the raw permutation space into a small set of interpretable motif families. Second, it reveals that these families are statistically very uneven, which is itself part of the geometry of the precedence space.

In 3×3, the full space of 216 raw profiles collapses to only 7 exact classes. In 4×4, the space of 331,776 profiles collapses to 922 distinct betweenness maps and 434 exact classes. Structural redundancy is therefore substantial in both settings, and the signature representation provides a compact way to retain the dominant regularities of the exhaustive spaces.

Cross-Size Trends, Symmetry, and Solver-Side Evidence

Once the exhaustive spaces are grouped by repetition signature, a consistent structural picture emerges across both configurations. The most stable cross-size pattern concerns the average mediation $\mu(\pi)$. As shown in Table 4, stronger repetition is associated with larger values of $\overline{\mu}$, whereas stronger diversity is associated with smaller ones. This monotone effect is visible in both exhaustive spaces: in 3×3, $\overline{\mu}$ increases from 0.08 for signature 1+1+1 to 0.12 for signature 3, and in 4×4 it increases from 0.06 for signature 1+1+1+1 to 0.09 for signature 4. This is the clearest shared trend across sizes: alignment concentrates precedence paths through recurrent positions, while diversity disperses mediation more evenly.

A second structural effect concerns heterogeneity. Here the two sizes no longer behave in exactly the same way. In 3×3, the average range $\overline{R}$ is still maximal for the fully aligned signature 3, so the dominant regime remains complete alignment. In 4×4, however, the picture becomes richer: while full alignment still maximizes $\overline{\mu}$, the partially disrupted signature 3+1 maximizes $\overline{R}$. This is the first sign that increasing the size of the precedence space does not simply reinforce the same pattern, but allows a new regime to emerge in which heterogeneity is driven not by complete repetition, but by a mostly aligned profile perturbed by a single outlier job.

Figure 2 refines this interpretation for the richer 4×4 case. The top panel confirms the monotone increase of $\overline{\mu}$ with stronger repetition. The middle panel shows that the greatest heterogeneity does not occur under complete alignment, but rather under partially disrupted signatures, especially 3+1. The bottom panel adds a complementary perspective through the symmetry score $\overline{S}$. The general tendency is that more symmetric betweenness maps correspond to more balanced configurations, whereas lower symmetry is associated with more dispersed or polarized mediation patterns. In this sense, symmetry does not replace the signature taxonomy, but refines it: the signature describes how repetition is distributed across jobs, while $S(\pi)$ indicates how balanced the resulting betweenness map remains.

This refinement is particularly useful for distinguishing fully diverse from partially disrupted configurations. Fully diverse signatures tend to produce flatter and more balanced maps, while the partially disrupted signatures in 4×4 combine moderate average mediation with lower symmetry and higher heterogeneity. Structural complexity is therefore not only a matter of how much repetition exists, but also of how that repetition is broken.

A third common effect concerns objective-bound distributions. Table 3 shows that, in both configurations, stronger repetition shifts the mass toward larger objective-bound values. In 3×3, the fully aligned signature 3 is concentrated entirely at objective bound 14, whereas the fully diverse signature 1+1+1 is centered at lower values and is dominated

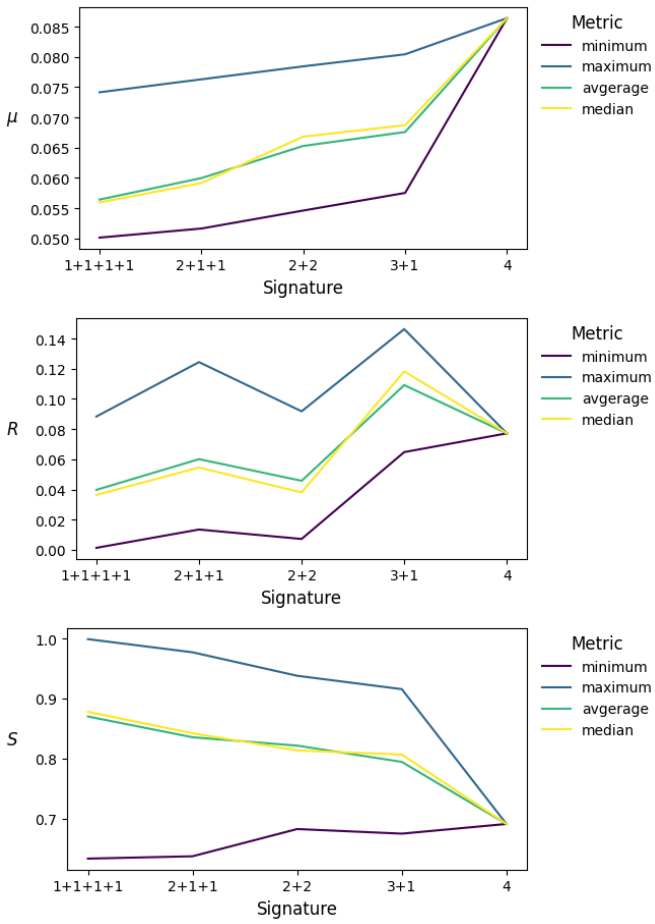

Figure 2: Linear representation of $\overline{\mu}$, $\overline{R}$ and $\overline{S}$ grouped by repetition pattern for the exhaustive 4×4 setting.

by bound 13. In 4×4, the same pattern appears at a larger scale: signature 4 is concentrated entirely at objective bound 23, whereas the fully diverse signature 1+1+1+1 is centered at 22 and still allocates non-negligible mass to 20 and 21. Thus, the same ordering already observed in $\overline{\mu}$ reappears in the lower-bound distributions: more repetition is associated with tighter precedence-induced bounds.

A fourth source of evidence comes from the solver-side statistics. Table 4 reports the average values of the main structural and solver metrics by repetition signature. In the table, $\overline{t}_{\text{flat}}$ is the average flattening time, $\overline{LB}$ is the average objective bound, $\overline{n}_{\text{bool}}$ is the average number of Boolean variables, $\overline{f}$ and $\overline{p}$ are the average numbers of failures and propagations, $\overline{t}_{\text{solve}}$ and $\overline{t}_{\text{wall}}$ denote average solve time and total wall-clock time, and $\overline{n}_{\text{sol}}$ is the average number of solutions.

Table 4 shows that the signature view is also reflected in the search profile. In both configurations, the average lower bound $\overline{LB}$ increases from the fully diverse regime to the fully aligned one, while the average number of solutions $\overline{n}_{\text{sol}}$ decreases in the same direction. The same monotone pattern also appears in $\overline{\mu}$, which increases with stronger repetition in

Configuration	Signature	Obj. Bound	Frequency	Percent (%)
3×3	1+1+1	12	12	10.00
		13	72	60.00
		14	36	30.00
	2+1	13	36	40.00
		14	54	60.00
	3	14	6	100.00
4×4	1+1+1+1	20	576	0.22
		21	63,216	24.79
		22	172,800	67.76
		23	18,432	7.23
	2+1+1	21	3,744	5.14
		22	44,928	61.66
		23	24,192	33.20
	2+2	21	72	4.35
		22	576	34.78
		23	936	56.52
		24	72	4.35
	3+1	22	672	30.43
		23	1,440	65.22
		24	96	4.35
	4	23	24	100.00

Table 3: Frequency table of objective-bound values by repetition signature for the exhaustive 3×3 and 4×4 settings.

both sizes. Together, these trends indicate that stronger repetition is associated with more concentrated mediation and a tighter, less permissive search space.

At the same time, the table makes the difference between 3×3 and 4×4 much clearer. In 3×3, both $\overline{\mu}$ and $\overline{R}$ are maximized by the fully aligned signature 3, so the dominant regime is still complete alignment. In 4×4, however, the pattern becomes richer: full alignment still maximizes $\overline{\mu}$, but the partially disrupted signature 3+1 maximizes both $\overline{R}$ and the number of propagations $\overline{p}$. This agrees with the structural picture suggested by Figure 2: full alignment increases global mediation and backtracking, whereas partial disruption produces the strongest polarization and the most intense propagation activity.

The solver-side metrics reinforce the same contrast. In 4×4, signature 4 yields the highest failure count $\overline{f}$, while signature 3+1 yields the highest propagation count $\overline{p}$. In 3×3, by comparison, the space is more compact and the solver-side distinctions are less clean: failures peak at 2+1 and propagations peak at 1+1+1, even though the aligned signature still dominates both $\overline{\mu}$ and $\overline{R}$. The basic repetition-versus-diversity trend is already present, but the more differentiated heterogeneity regime has not yet emerged as clearly.

This contrast is itself informative. The 3×3 configuration captures the basic structural tendency: alignment increases mediation, diversity reduces it, and fully diverse signatures remain flatter and more balanced. The 4×4 configuration, however, is the first one large enough to expose a distinct partial-disruption regime, where a mostly aligned structure perturbed by a single outlier job becomes more heterogeneous than a fully aligned one. This is precisely the role of signature 3+1: it shows that structural complexity is not governed only by how much repetition exists, but also by how that repetition is broken.

	3×3			4×4				
Metric	1+1+1	2+1	3	1+1+1+1	2+1+1	2+2	3+1	4
$\overline{\mu}$	0.08	0.10	**0.12**	0.06	0.06	0.07	0.07	**0.09**
$\overline{R}$	0.04	0.07	**0.09**	0.04	0.06	0.05	**0.11**	0.08
$\overline{t}_{\text{flat}}$	**71.63**	70.34	69.71	69.12	69.13	69.11	**69.19**	69.06
$\overline{LB}$	13.20	13.60	**14.00**	21.82	22.28	22.61	22.74	**23.00**
$\overline{n}_{\text{bool}}$	**14.95**	13.60	11.33	**29.05**	28.77	28.09	27.67	25.29
$\overline{f}$	0.59	**0.64**	0.00	0.43	1.00	1.48	2.54	**5.13**
$\overline{p}$	**211.37**	200.46	115.50	375.27	390.44	398.58	**440.34**	386.42
$\overline{t}_{\text{solve}}$	16.36	15.76	**16.43**	19.91	**19.94**	19.93	19.72	17.99
$\overline{t}_{\text{wall}}$	117.42	115.17	**208.33**	**119.65**	118.95	117.36	116.06	105.79
$\overline{n}_{\text{sol}}$	**2.66**	2.18	1.00	**5.59**	5.13	4.20	3.63	1.00

Table 4: Average structural and solver metrics by repetition signature for the exhaustive 3×3 and 4×4 settings. Bold indicates the maximum value within each configuration block and row; underlined values indicate the minimum.

Taken together, Figure 2, Table 3, and Table 4 support the same interpretation. Repetition signatures preserve stable trends across both sizes, especially for average mediation, lower-bound behaviour, and search-space tightness, while the richer 4×4 setting reveals an additional structural regime. Across both 3×3 and 4×4, full alignment maximizes average mediation and pushes the lower-bound distribution upward, whereas full diversity minimizes average mediation and yields flatter, more symmetric maps. The main difference is that, in 4×4, partial disruption emerges as a separate heterogeneity regime with lower symmetry, stronger heterogeneity, and more distinctive solver activity.

Implications for Larger JSP Instances

The results reported in this paper should not be read as an exact structural law for larger JSP instances. The exhaustive analysis is restricted to complete 3×3 and 4×4 precedence micro-spaces, and the identified repetition signatures are defined in that controlled setting. However, the extracted motifs still provide a useful structural vocabulary beyond these sizes. In particular, the contrast between full alignment, full diversity, and partial disruption suggests that larger instances may also contain localized or mesoscopic precedence regimes with different mediation profiles.

From this perspective, the main transferable contribution of the paper is not a direct predictor for larger JSPs, but a knowledge-extraction framework that identifies interpretable structural patterns and links them to observable betweenness behaviour. For a concrete 10×10 instance, for example, the proposed analysis can be used diagnostically: one may compute the induced betweenness map, inspect the degree of repetition and disruption in the job permutations, and test whether aligned-like, diverse-like, or partially disrupted substructures coincide with concentrated mediation or stronger internal heterogeneity.

This makes the extracted motifs relevant not as exact classes for larger instances, but as reusable hypotheses for structural diagnosis, feature construction, and future validation in broader JSP families.

Conclusions

This paper studied precedence-induced betweenness patterns in classical JSP through exhaustive 3×3 and 4×4 micro-spaces. Rather than treating betweenness as a useful but opaque feature, we used exhaustive enumeration to identify the precedence motifs that generate recurrent centrality regimes. The results show that repetition signatures provide a compact, interpretable compression of the raw precedence space while preserving meaningful structural variation. Across both exhaustive settings, stronger repetition is associated with larger average mediation and tighter lower-bound behavior, whereas stronger diversity produces flatter and more balanced maps.

The comparison between 3×3 and 4×4 is particularly informative. The smaller space already captures the basic repetition-versus-diversity trend, but the richer 4×4 space is the first one to expose a distinct partial-disruption regime. In that setting, the signature $3{+}1$ does not maximize average mediation, but it does maximize heterogeneity and propagation activity, while also exhibiting lower symmetry. This shows that structural complexity is not determined solely by the amount of repetition, but also by how that repetition is broken. In this sense, symmetry refines the signature-based view by distinguishing balanced configurations from polarized ones.

The contribution of the paper is therefore methodological rather than algorithmic. We do not claim that the identified motifs constitute exact structural laws for larger JSP instances. Instead, we show that exhaustive small-instance analysis can extract reusable structural knowledge and provide a vocabulary of interpretable motifs, full alignment, full diversity, and partial disruption that can later be tested in larger and more realistic settings. From this perspective, the proposed framework can support motif-aware feature engineering, benchmark design, and structural diagnostics for broader JSP families. A natural next step is to study whether analogous mesoscopic repetition and disruption patterns can be detected within larger instances and whether they preserve the same relationship with betweenness, symmetry, and solver effort.

References

Abdelmaguid, T. F. 2009. Permutation-Induced Acyclic Networks for the Job Shop Scheduling Problem. *Applied Mathematical Modelling*.

Błażewicz, J.; Domschke, W.; and Pesch, E. 2000. The Job Shop Scheduling Problem: Conventional and New Solution Techniques. *European Journal of Operational Research*, 123(1): 1–33.

Chu, C.; Proth, J.-M.; and Wang, C. 1998. Improving Job-Shop Schedules Through Critical Pairwise Exchanges. *International Journal of Production Research*.

Čyras, K.; Letsios, D.; Misener, R.; and Toni, F. 2019. Argumentation for Explainable Scheduling. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

March, C.; Pérez, C.; and Salido, M. Á. 2024. A Novel Instance Generator for Benchmarking the Job Shop Scheduling Problem. In *Lecture Notes in Computer Science*.

Mirshekarian, S.; and Sormaz, D. 2016. Correlation of Job-Shop Scheduling Problem Features with Scheduling Efficiency. *Expert Systems with Applications*.

O'Neill, B. 2021. The Classical Occupancy Distribution: Computation and Approximation. *The American Statistician*, 75(4): 364–375.

Perez, C.; March, C.; and Salido, M. A. 2025. IGJSP: A Reproducible Instance Generator for Job Shop Scheduling. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '25 Companion, 159–162. New York, NY, USA: Association for Computing Machinery. ISBN 9798400714641.

Pinedo, M. L. 2016. *Scheduling: Theory, Algorithms, and Systems*. Springer, 5 edition.

Silva, J. R.; Silva, J. M.; and Vaquero, T. S. 2020. *Formal Knowledge Engineering for Planning: Pre and Post-Design Analysis*, 47–65. Cham: Springer International Publishing. ISBN 978-3-030-38561-3.

Smith-Miles, K.; and Lopes, L. 2014. Measuring Instance Difficulty for Combinatorial Optimisation Problems. *ACM Computing Surveys*, 47(2): 1–34.

Smith-Miles, K.; and van Hemert, J. I. 2012. Discovering the Suitability of Optimisation Algorithms for a Given Problem. *Evolutionary Computation*, 20(1): 1–17.

Strassl, S.; and Musliu, N. 2022. Instance Space Analysis for the Job Shop Scheduling Problem. *European Journal of Operational Research*, 299(3): 1072–1087.

Vela, A.; Cruz-Duarte, J. M.; Ortiz-Bayliss, J. C.; and Amaya, I. 2021. Tailoring Job Shop Scheduling Problem Instances. *IEEE Access*.

Yin, Y.; Xu, J.; and Liu, X. 2022. A Complex Network Modeling for Job-Shop Manufacturing System. In *Proceedings of ICITBS 2022*.

Zhuang, Z.; Chen, Y.; Sun, Y.; and Qin, W. 2020. Complex Scheduling Network: Objective Performance Testing Platform. *The International Journal of Advanced Manufacturing Technology*.