A Theorem Proofs

Proof of Theorem 1. Let operation i scheduled on machine $\alpha(i)$. Operation i may have multiple job predecessors and successors that are included in the sets PJ_i and SJ_i, respectively. When the operation i is relocated between operations v and w in π_k, the paths $\rho(v,i)$ and $\rho(i,w)$ are created in the solution graph G. Therefore, a cycle is introduced in the solution graph only if paths $\rho(i,v)$ and/or $\rho(w,i)$ already exist in the graph.

First, we focus on the path $\rho(i,v)$. Let $\{sm_i \cup SJ_i\}$ denote the set of the immediate successor operations of i in G, or in other words the set $\bigcup\{e : \exists(i,e) \in G, \forall e \in \Omega\}$. If $L(i,v) > 0$, then $\rho(i,v)$ should pass either through sm_i or one of the immediate job successors $e \in SJ_i$ of i. When the move is applied the arc (i, sm_i) is removed from G and $\rho(i,v)$ should pass through one of the immediate job successors of i. If a job successor $e \in SJ_i$ exists such that $L(e,v) > 0$, then $r_v \geq r_e + p_{e,\alpha(e)}$, which contradicts the first statement of the theorem, i.e. $r_v < r_e + p_{e,\alpha(e)}, \forall e \in SJ_i$.

Following the same rationale for path $\rho(w,i)$, if an immediate job predecessor $e \in PJ_i$ exists, such that $L(w,e) > 0$, then $r_e \geq r_w + p_{w,k}$, which contradicts the second statement of the theorem, i.e., $r_w + p_{w,k} > r_e, \forall e \in PJ_i$. □

Proof of Theorem 2. Let s be the current solution, and $\acute{s}(\acute{\alpha}, \acute{\pi})$ be the solution after the relocation move has been applied and $G, \acute{G}$ their solution graphs respectively. The new makespan $C_{max}^{\acute{s}}$ cannot be shorter than the length of the longest path that passes through operation i:

$$\acute{r}_i + p_{i,k} + \acute{q}_i \leq C_{max}^{\acute{s}} \tag{27}$$

where $\acute{r}_i$ and $\acute{q}_i$ are the head and tail times of operation i in $\acute{s}$:

$$\acute{r}_i = \max\left(\max_{\forall e \in PJ_i}\left(\acute{r}_e + p_{e,\acute{\alpha}(e)}\right), \acute{r}_v + p_{v,\acute{\alpha}(v)}\right) \tag{28}$$

$$\acute{q}_i = \max\left(\max_{\forall e \in SJ_i}\left(\acute{q}_e + p_{e,\acute{\alpha}(e)}\right), \acute{q}_w + p_{w,\acute{\alpha}(w)}\right) \tag{29}$$

Note that for the sake of notation simplicity, in this proof, we drop the subscript and we replace sm_i and pm_i with sm and pm, respectively.

Considering solution $\acute{s}$, operation i cannot belong to any path between 0 and $e \in PJ_i$, since this would introduce cycles in the graph $\acute{G}$. Similarly, operation i cannot belong to any path between $e \in SJ_i$ and $*$. Therefore, in any feasible relocation of i we have that $\acute{r}_e = r_e$ and

$\acute{q}_e = q_e$.

Then, if we replace $\acute{r}_e$ with r_e and $\acute{q}_e$ with q_e in Equations 28 and 29 we have:

$$\acute{r}_i = \max\left(\max_{\forall e \in PJ_i}\left(r_e + p_{e,\alpha(e)}\right), \acute{r}_v + p_{v,\alpha(v)}\right) \tag{30}$$

$$\acute{q}_i = \max\left(\max_{\forall e \in SJ_i}\left(q_e + p_{e,\alpha(e)}\right), \acute{q}_w + p_{w,\alpha(w)}\right) \tag{31}$$

Note that the assignment information of all operations, expect for i, will remain the same in $\acute{s}$, i.e., $\alpha(e) = \acute{\alpha}(e), \forall e \in \Omega \setminus \{i\}$.

Given Equations (30) and (31), inequality (27) becomes:

$$\max\left(\max_{\forall e \in PJ_i}\left(r_e + p_{e,\alpha(e)}\right), \acute{r}_v + p_{v,\alpha(v)}\right) + p_{i,k} + \max\left(\max_{\forall e \in SJ_i}\left(q_e + p_{e,\alpha(e)}\right), \acute{q}_w + p_{w,\alpha(w)}\right) \leq C_{max}^{\acute{s}} \tag{32}$$

If we compare Equation (24) with (32), we end up with the conclusion that the proof of Theorem (2) is equivalent to proving that $\acute{r}_v \geq \hat{r}_v$ and $\acute{q}_w \geq \hat{q}_w$.

We consider the four cases below:

- Case 1: $i \notin \mathcal{P}_v$ in s
- Case 2: $i \in \mathcal{P}_v$ in s
- Case 3: $i \notin \mathcal{S}_w$ in s
- Case 4: $i \in \mathcal{S}_w$ in s

Case 1

If $i \notin \mathcal{P}_v$ in s, then operation i does not belong to any path leading to v, and therefore, the longest path from 0 to v is not affected by the relocation of i, thus $\acute{r}_v = r_v = \hat{r}_v$.

Case 2

If $i \in \mathcal{P}_v$, then any path connecting i and v should pass through sm, because otherwise, such path would pass through a job successor of i and the relocation move would be infeasible due to cycles that would be created in $\acute{G}$. Therefore, sm must be one of the predecessors of v, or v itself, i.e., $sm \in \{v\} \cup \mathcal{P}_v$. There exist two cases for the machine successor sm; a) $sm \in \rho(0, v)$ and b) $sm \notin \rho(0, v)$, which we discuss in the following separately.

Case 2a

If sm belongs to $\rho(0, v)$ in s, the following stands for the length of $\rho(0, v)$:

$$L(0, v) = L(0, sm) + L(sm, v) \tag{33}$$

However, in $\acute{s}$ sm may or may not belong to the longest path, which means that:

$$\acute{L}(0, v) \geq \acute{L}(0, sm) + \acute{L}(sm, v) \tag{34}$$

where $\acute{L}$ represents the length of the longest paths in $\acute{s}$.

Also, given that the path $\rho(sm, v)$ is not affected by the relocation of i:

$$\acute{L}(sm, v) = L(sm, v) \tag{35}$$

If we use Equations (33) and (35) on inequality (34) we derive:

$$\begin{aligned}
&\acute{L}(0, v) \geq \acute{L}(0, sm) + L(0, v) - L(0, sm) \overset{Def.(6)}{\Longrightarrow} \\
&\quad \acute{r}_v \geq r_v - r_{sm} + \acute{r}_{sm} \overset{Eq.(4)}{\Longrightarrow} \\
&\quad \acute{r}_v \geq r_v - r_{sm} + \max\left(\max_{\forall j \in PJ_{sm}} (\acute{r}_j + p_{j,\alpha(j)}), \acute{r}_{pm} + p_{pm,\alpha(i)}\right)
\end{aligned} \tag{36}$$

Any path $\rho(0, j)$ $\forall j \in PJ_{sm}$ is not affected by the relocation of operation i, because otherwise at least one path $\rho(i, j)$ would exist in G. Also, the path $\rho(i, j)$ would also pass through a job successor $e \in SJ_i$ of i. If there exist such path $\rho(i, j)$, then the following equation would stand:

$$r_e + p_{e,\alpha(e)} \leq r_j < r_{sm} \leq r_v, \forall e \in SJ_i \tag{37}$$

According to Theorem (1) the Equation (37) is not valid and therefore:

$$\acute{r}_j = r_j, \forall j \in PJ_{sm} \tag{38}$$

Also, the path $\rho(0, pm)$ is not affected by the relocation of i, because otherwise there would exist a path $\rho(i, pm)$ in G, which is not true, since this implies that G already contains cycles prior to the application of the move. Therefore:

$$\acute{r}_{pm} = r_{pm} \tag{39}$$

Note that Equations (38) and (39) are valid also when $sm \notin \rho(0, v)$.

Using Equations (38) and (39) on Equation (36):

$$\acute{r}_v \geq r_v - r_{sm} + \max(\max_{\forall j \in PJ_{sm}} (r_j + p_{j,\alpha(j)}), r_{pm} + p_{pm,\alpha(i)}) \overset{Eq.(25)}{\Longrightarrow}$$
$$\acute{r}_v \geq \hat{r}_v \tag{40}$$

Case 2b

If sm does not belong to the path $\rho(0, v)$ in s, then i does not belong to the same path as well; otherwise, $i \in \rho(0, v)$ and also $\rho(0, v)$ should pass through the job successors of i, which would create a cycle in G. Therefore, the relocation of i does not affect the new head time of v:

$$\acute{r}_v = r_v. \tag{41}$$

Also, the head time of sm can only decrease or remain the same after the relocation of operation i, thus:

$$r_{sm} \geq \acute{r}_{sm} \tag{42}$$

If we sum Equation (41) and (42) we obtain:

$$r_{sm} + \acute{r}_v \geq r_v + \acute{r}_{sm} \Longrightarrow$$
$$\acute{r}_v \geq r_v - r_{sm} + \acute{r}_{sm} \overset{Eq.(4)}{\Longrightarrow}$$
$$\acute{r}_v \geq r_v - r_{sm} + \max(\max_{\forall j \in PJ_{sm}} (\acute{r}_j + p_{j,\alpha(j)}), \acute{r}_{pm} + p_{pm,\alpha(i)}) \overset{Eq.(38),(39)}{\Longrightarrow}$$
$$\acute{r}_v \geq r_v - r_{sm} + \max(\max_{\forall j \in PJ_{sm}} (r_j + p_{j,\alpha(j)}), r_{pm} + p_{pm,\alpha(i)}) \overset{Eq.(25)}{\Longrightarrow}$$
$$\acute{r}_v \geq \hat{r}_v \tag{43}$$

Case 3

If $i \notin \mathcal{S}_w$ in s, then the operation i does not belong to any path from w to $*$, and therefore, the length of the path is not affected by the relocation of i and thus $\acute{q}_w = q_w = \hat{q}_w$.

Case 4

If $i \in \mathcal{S}_w$ then working similarly to Case 2 we can derive that:

$$\acute{q}_w \geq q_w - q_{pm} + \max\left(\max_{\forall j \in SJ_{pm}} (q_j + p_{j,\alpha(j)}), q_{sm} + p_{sm,\alpha(i)}\right) \xrightarrow{Eq.(26)}$$

$$\acute{q}_w \geq \hat{q}_w$$

□

B Detailed results for the FJSSP and Comparative Performance Analysis

This section provides detailed results for well-known FJSSP benchmark data sets, namely Brandimarte (1993) (BRData), Dauzère-Pérès and Paulli (1997) (DPData), Barnes and Chambers (1996) (BCData) and Hurink et al. (1994) (HUData). Note that the HUData set contains three groups of problems, namely *edata*, *vdata* and *rdata*. More specifically, Tables B.1–B.6 present the results for each data set. The first four columns contain the name, the size ($l \times m$), the flexibility (fx) and the best available lower bounds, respectively. The lower bounds are calculated using the maximum values between the calculated lower bounds from the MIP and CP models, as well as the reported lower bounds in the works of Mastrolilli and Gambardella (2000) and Behnke and Geiger (2012). The remaining three columns provide details regarding the solutions produced by the CP, the MIP and the EA, respectively. In particular, each table reports the C_{max}, the Gap(%) and the Time in seconds either to fully close the gap (CP and MIP), or to obtain the best feasible heuristic solution (EA) among the ten evaluated runs. The Gap(%) refers to the optimality gap and is calculated as $\frac{C_{max}-LB}{LB}\%$ for every problem instance. The last rows report the average gap (%) out of all problem instances when, as well as the number of new best solutions ($\#NB$) obtained, compared to the best known solutions of the literature. Finally, Table B.7 provides a more compact form of the results for the HUData problem set. More specifically, the problem instances are organized into nine groups according to their size. The first column contains the different problem group names, while the next columns show the average gap (%) for the corresponding instances on the available problem subsets (*edata*, *rdata* and *vdata*).

Table B.1: Detailed results for the BRData set

Instances				CP			MIP			EA(10800)		
Name	$l \times m$	fx	LB	C_{max}	Gap(%)	Time	C_{max}	Gap(%)	Time	C_{max}	Gap(%)	Time
mk1	10x6	2.09	40	**40***	0.00	<1	**40***	0.00	10800	**40***	0.00	1
mk2	10x6	4.10	26	**26***	0.00	1900	28	7.69	10800	**26***	0.00	1
mk3	15x8	3.00	204	**204***	0.00	1	**204***	0.00	10800	**204***	0.00	2
mk4	15x8	1.91	55	**60**	9.09	10800	**60**	9.09	10800	**60**	9.09	1
mk5	15x4	1.7	168	**172**	2.38	10800	182	8.33	10800	**172**	2.38	2
mk6	10x10	3.26	39	58	48.72	10800	78	100.00	10800	**57**	46.15	62
mk7	20x5	2.83	133	140	5.26	10800	145	9.02	10800	**139**	4.51	19
mk8	20x10	1.43	523	**523***	0.00	<1	539	3.06	10800	**523***	0.00	3
mk9	20x10	2.52	307	**307***	0.00	4	355	15.64	10800	**307***	0.00	4
mk10	20x15	2.98	183	197	7.65	10800	305	66.67	10800	**193**	5.46	519
mk11	30x5	1.5	594	612	3.03	10800	761	28.11	10800	**609**	2.53	85
mk12	30x10	1.49	508	**508***	0.00	<1	540	6.30	10800	**508***	0.00	3
mk13	30x10	3.37	353	395	11.90	10800	596	68.84	10800	**386**	9.35	704
mk14	30x15	1.56	694	**694***	0.00	<1	870	25.36	10800	**694***	0.00	6
mk15	30x15	3.03	333	**333***	0.00	4102	-	-	10800	**333***	0.00	8
Average Gap (%)				5.87			-			5.30[1]		
# NB				0			0			1		

[1] The reported average gap is calculated by considering the best run among the ten runs per instance. The average gap when considering all ten runs per instance is 5.49% and the average gap when considering only the worst run per instance is 5.72%.

Table B.2: Detailed results for the BCData set

Instances				CP			MIP			EA(10800)		
Name	$l \times m$	fx	LB	C_{max}	Gap(%)	Time	C_{max}	Gap(%)	Time	C_{max}	Gap(%)	Time
mt10c1	10x11	1.10	927	**927***	0.00	6	**927***	0.00	808	**927***	0.00	1
mt10cc	10x12	1.20	908	**908***	0.00	5	**908***	0.00	236	**908***	0.00	3
mt10x	10x11	1.10	918	**918***	0.00	4	**918***	0.00	572	**918***	0.00	1
mt10xx	10x12	1.20	918	**918***	0.00	4	**918***	0.00	2236	**918***	0.00	1
mt10xxx	10x13	1.30	918	**918***	0.00	4	**918***	0.00	1082	**918***	0.00	1
mt10xy	10x12	1.20	905	**905***	0.00	4	**905***	0.00	480	**905***	0.00	1
mt10xyz	10x13	1.30	847	**847***	0.00	3	**847***	0.00	115	**847***	0.00	1
setb4c9	15x11	1.10	914	**914***	0.00	8	**914***	0.00	1292	**914***	0.00	2
setb4cc	15x12	1.20	907	**907***	0.00	6	**907***	0.00	10803	**907***	0.00	2
setb4x	15x11	1.10	925	**925***	0.00	14	**925***	0.00	3955	**925***	0.00	2
setb4xx	15x12	1.20	925	**925***	0.00	14	**925***	0.00	7993	**925***	0.00	1
setb4xxx	15x13	1.30	925	**925***	0.00	14	**925***	0.00	5539	**925***	0.00	2
setb4xy	15x12	1.20	910	**910***	0.00	6	**910***	0.00	1216	**910***	0.00	1
setb4xyz	15x13	1.30	902	**902***	0.00	5	**902***	0.00	599	905	0.33	2
seti5c12	15x16	1.07	1169	**1169***	0.00	39	1174	0.43	10800	**1169***	0.00	743
seti5cc	15x17	1.13	1135	**1135***	0.00	56	1136	0.09	10800	**1135***	0.00	146
seti5x	15x16	1.07	1198	**1198***	0.00	15	1207	0.75	10800	**1198***	0.00	21
seti5xx	15x17	1.13	1194	**1194***	0.00	24	1206	1.01	10800	**1194***	0.00	250
seti5xxx	15x18	1.20	1194	**1194***	0.00	17	1205	0.92	10800	**1194***	0.00	4
seti5xy	15x17	1.13	1135	**1135***	0.00	90	1138	0.26	10800	**1135***	0.00	90
seti5xyz	15x18	1.20	1125	**1125***	0.00	143	1133	0.71	10800	**1125***	0.00	6
Average Gap (%)				0.00			0.20			0.02[1]		
# NB				3			0			2		

[1] The reported average gap is calculated by considering the best run among the ten runs per instance. The average gap when considering all ten runs per instance is 0.05% and the average gap when considering only the worst run per instance is 0.06%.

Table B.3: Detailed results for the DPData set

Instances				CP			MIP			EA(10800)		
Name	$l \times m$	fx	LB	C_{max}	Gap(%)	Time	C_{max}	Gap(%)	Time	C_{max}	Gap(%)	Time
01a	10x5	1.13	2505	**2505***	0.00	26	2725	8.78	10800	**2505***	0.00	12
02a	10x5	1.69	2228	2235	0.31	10800	2550	14.45	10800	**2228***	0.00	472
03a	10x5	2.56	2228	**2228***	0.00	10800	2820	26.57	10800	**2228***	0.00	14
04a	10x5	1.13	2503	**2503***	0.00	59	2744	9.63	10800	**2503***	0.00	12
05a	10x5	1.69	2189	2214	1.14	10800	2593	18.46	10800	**2204**	0.69	600
06a	10x5	2.56	2162	2196	1.57	10800	-	-	10800	**2177**	0.69	1826
07a	15x8	1.24	2206	2280	3.35	10800	3066	38.98	10800	**2255**	2.22	2313
08a	15x8	2.42	2061	2067	0.29	10800	2649	28.53	10800	**2065**	0.19	390
09a	15x8	4.03	2061	2063	0.10	10800	-	-	10800	**2062**	0.05	629
10a	15x8	1.24	2197	2297	4.55	10800	2947	34.14	10800	**2246**	2.23	2740
11a	15x8	2.42	2017	2071	2.68	10800	-	-	10800	**2042**	1.24	2191
12a	15x8	4.03	1969	2031	3.15	10800	-	-	10800	**2005**	1.83	3943
13a	20x10	1.34	2161	2245	3.89	10800	3214	48.73	10800	**2234**	3.38	1875
14a	20x10	2.99	2161	2168	0.32	10800	-	-	10800	**2163**	0.09	4858
15a	20x10	5.02	2161	**2163**	0.09	10800	-	-	10800	**2163**	0.09	3234
16a	20x10	1.34	2148	2264	5.40	10800	3113	44.93	10800	**2230**	3.82	2574
17a	20x10	2.99	2088	2153	3.11	10800	-	-	10800	**2119**	1.48	2929
18a	20x10	5.02	2057	2134	3.74	10800	-	-	10800	**2104**	2.28	4752
Average Gap (%)				1.87			-			1.13[1]		
# NB				1			0			11		

[1] The reported average gap is calculated by considering the best run among the ten runs per instance. The average gap when considering all ten runs per instance is 1.37% and the average gap when considering only the worst run per instance is 1.59%.

C Comparative analysis for the FJSSP

This section compares the performance of the proposed EA with the current state-of-the-art heuristic solution methods of the FJSSP literature. For this purpose, we have considered in our analysis the Tabu Search (TS) of Mastrolilli and Gambardella (2000), the hybrid Genetic Algorithm (hGA) of Gao et al. (2008), the parallel Tabu Search based Metaheuristic Algorithm (TSBM) of Bozejko et al. (2010), the Climbing Depth Bounded Discrepancy Search (CDDS) Ben Hmida et al. (2010), the Hybrid Harmony Search and Large Neighborhood Search (HHS) of Yuan et al. (2013), the Scatter Search with Path Relinking (SSPR) of González et al. (2015), the Memetic Algorithm (MA) of Yi et al. (2016), and the Elitist Quantum inspired Evolutionary Algorithm (EQUA) of Wu and Wu (2017).

Tables C.1 – C.3 present the results for the BRData, the DPData and the BCData sets, respectively. The first column provides the name of every problem instance. Next, the second column LB provides the proven optimal solution (if any) or the tightest known lower bound. These values are taken from the work of Mastrolilli and Gambardella (2000) as well as from the results of the CP and MIP presented earlier in Section 4.3. At first, we present results of the proposed EA and CP considering different time limits. These limits have been selected in a way to suit the difficulty of each data set as well as to allow us to assess the progress observed over time. The numbers in the parenthesis next to the abbreviations EA and CP refer to the time limits imposed in seconds. Next, the remaining columns contain the results as originally

Table B.4: Detailed results for the HUData($vdata$) set

Instances				CP			MIP			EA(10800)		
Name	$l \times m$	fx	LB	C_{max}	Gap(%)	Time	C_{max}	Gap(%)	Time	C_{max}	Gap(%)	Time
m06	6x6	2.86	47	**47***	0.00	<1	**47***	0.00	3	**47***	0.00	58
m10	10x10	4.48	655	**655***	0.00	<1	**655***	0.00	749	**655***	0.00	329
m20	20x5	2.62	1022	**1022***	0.00	10800	1034	1.17	10800	**1022***	0.00	493
la1	10x5	2.84	570	**570***	0.00	10800	575	0.88	10800	**570***	0.00	202
la2	10x5	2.68	529	**529***	0.00	10800	532	0.57	10800	**529***	0.00	209
la3	10x5	2.56	477	**477***	0.00	10800	478	0.21	10800	**477***	0.00	209
la4	10x5	2.38	502	**502***	0.00	10800	503	0.20	10800	**502***	0.00	128
la5	10x5	2.38	457	**457***	0.00	10800	460	0.66	10800	**457***	0.00	158
la6	15x5	2.43	799	**799***	0.00	10800	808	1.13	10800	**799***	0.00	240
la7	15x5	2.43	749	**749***	0.00	10800	752	0.40	10800	**749***	0.00	280
la8	15x5	2.59	765	**765***	0.00	10800	778	1.70	10800	**765***	0.00	351
la9	15x5	2.53	853	**853***	0.00	10800	859	0.70	10800	**853***	0.00	198
la10	15x5	2.67	804	**804***	0.00	10800	807	0.37	10800	**804***	0.00	345
la11	20x5	2.53	1071	**1071***	0.00	10800	1094	2.15	10800	**1071***	0.00	513
la12	20x5	2.48	936	**936***	0.00	10800	945	0.96	10800	**936***	0.00	602
la13	20x5	2.45	1038	**1038***	0.00	10800	1053	1.45	10800	**1038***	0.00	611
la14	20x5	2.44	1070	**1070***	0.00	10800	1081	1.03	10800	**1070***	0.00	358
la15	20x5	2.63	1089	**1089***	0.00	10800	1106	1.56	10800	**1089***	0.00	610
la16	10x10	4.70	717	**717***	0.00	<1	**717***	0.00	389	**717***	0.00	332
la17	10x10	4.79	646	**646***	0.00	<1	**646***	0.00	408	**646***	0.00	317
la18	10x10	4.79	663	**663***	0.00	<1	**663***	0.00	706	**663***	0.00	321
la19	10x10	4.80	617	**617***	0.00	3	**617***	0.00	573	**617***	0.00	568
la20	10x10	4.87	756	**756***	0.00	<1	**756***	0.00	434	**756***	0.00	208
la21	15x10	4.77	800	**802**	0.25	10800	935	16.88	10800	804	0.50	1298
la22	15x10	4.51	733	735	0.27	10800	933	27.29	10800	735	0.27	1113
la23	15x10	4.53	809	812	0.37	10800	964	19.16	10800	812	0.37	1668
la24	15x10	4.85	773	**775**	0.26	10800	872	12.81	10800	**775**	0.26	1850
la25	15x10	4.83	751	754	0.40	10800	834	11.05	10800	**753**	0.27	1748
la26	20x10	4.58	1052	**1053**	0.10	10800	1388	31.94	10800	**1053**	0.10	1460
la27	20x10	4.58	1084	1085	0.09	10800	1369	26.29	10800	1085	0.09	2200
la28	20x10	4.49	1069	1070	0.09	10800	1312	22.73	10800	**1069***	0.00	2168
la29	20x10	4.41	993	**994**	0.10	10800	1394	40.38	10800	**994**	0.10	2497
la30	20x10	4.65	1068	**1069**	0.09	10800	1325	24.06	10800	**1069**	0.09	1903
la31	30x10	4.60	1520	**1520***	0.00	10800	2064	35.79	10800	**1520***	0.00	3309
la32	30x10	4.54	1657	**1658**	0.06	10800	2392	44.36	10800	**1658**	0.06	3460
la33	30x10	4.51	1497	1499	0.13	10800	-	-	10800	1498	0.07	3386
la34	30x10	4.62	1535	**1535***	0.00	10800	2385	55.37	10800	**1535***	0.00	3470
la35	30x10	4.65	1549	1550	0.06	10800	2152	38.93	10800	**1549***	0.00	2929
la36	15x15	6.70	948	**948***	0.00	1	1387	46.31	10800	**948***	0.00	2004
la37	15x15	6.63	986	**986***	0.00	1	1297	31.54	10800	**986***	0.00	1704
la38	15x15	6.57	943	**943***	0.00	1	1265	34.15	10800	**943***	0.00	1792
la39	15x15	6.43	922	**922***	0.00	5	1304	41.43	10800	**922***	0.00	2118
la40	15x15	6.48	955	**955***	0.00	1	1207	26.39	10800	**955***	0.00	1989
abz5	10x10	4.67	859	**859***	0.00	2	892	3.84	10800	**859***	0.00	491
abz6	10x10	4.29	742	**742***	0.00	<1	**742***	0.00	1448	**742***	0.00	229
abz7	20x15	6.50	492	**495**	0.61	10800	860	74.80	10800	498	1.22	4266
abz8	20x15	6.58	506	**510**	0.79	10800	887	75.30	10800	514	1.58	3838
abz9	20x15	6.65	497	**499**	0.40	10800	886	78.27	10800	504	1.41	4141
car1	11x5	2.75	5005	5007	0.04	10800	5030	0.50	10800	**5005***	0.00	260
car2	13x4	2.29	5929	5930	0.02	10800	5939	0.17	10800	**5929***	0.00	157
car3	12x5	2.60	5597	5599	0.04	10800	5649	0.93	10800	**5598**	0.02	227
car4	14x4	2.09	6514	6515	0.02	10800	6518	0.06	10800	**6514***	0.00	171
car5	10x6	2.83	4909	4923	0.29	10800	4977	1.39	10800	**4913**	0.08	264
car6	8x9	4.08	5486	**5486***	0.00	<1	**5486***	0.00	44	**5486***	0.00	171
car7	7x7	3.45	4281	**4281***	0.00	1	**4281***	0.00	13	**4281***	0.00	80
car8	8x8	3.97	4613	**4613***	0.00	1	**4613***	0.00	188	**4613***	0.00	246
orb1	10x10	4.46	695	**695***	0.00	<1	**695***	0.00	422	**695***	0.00	226
orb2	10x10	4.40	620	**620***	0.00	2	**620***	0.00	6848	**620***	0.00	321
orb3	10x10	4.67	648	**648***	0.00	1	**648***	0.00	3485	**648***	0.00	400
orb4	10x10	4.71	753	**753***	0.00	<1	**753***	0.00	467	**753***	0.00	268
orb5	10x10	4.77	584	**584***	0.00	1	**584***	0.00	1964	**584***	0.00	476
orb6	10x10	4.78	715	**715***	0.00	<1	**715***	0.00	228	**715***	0.00	230
orb7	10x10	4.55	275	**275***	0.00	<1	**275***	0.00	2115	**275***	0.00	628
orb8	10x10	4.55	573	**573***	0.00	<1	**573***	0.00	149	**573***	0.00	264
orb9	10x10	4.47	659	**659***	0.00	<1	**659***	0.00	180	**659***	0.00	287
orb10	10x10	4.51	681	**681***	0.00	1	**681***	0.00	207	**681***	0.00	348
Average Gap (%)				0.07			-			0.10[1]		
# NB				0			0			0		

[1] The reported average gap is calculated by considering the best run among the ten runs per instance. The average gap when considering all ten runs per instance is 0.14% and the average gap when considering only the worst run per instance is 0.18%.

Table B.5: Detailed results for the HUData(*edata*) set

Instances				CP			MIP			EA(10800)		
Name	$l \times m$	fx	LB	C_{max}	Gap(%)	Time	C_{max}	Gap(%)	Time	C_{max}	Gap(%)	Time
m06	6x6	1.17	55	**55***	0.00	<1	**55***	0.00	1	**55***	0.00	49
m10	10x10	1.13	871	**871***	0.00	3	**871***	0.00	274	**871***	0.00	250
m20	20x5	1.13	1088	**1088***	0.00	7	1101	1.19	10800	**1088***	0.00	251
la1	10x5	1.2	609	**609***	0.00	<1	**609***	0.00	21	**609***	0.00	118
la2	10x5	1.2	655	**655***	0.00	<1	**655***	0.00	26	**655***	0.00	150
la3	10x5	1.18	550	**550***	0.00	1	**550***	0.00	80	**550***	0.00	113
la4	10x5	1.18	568	**568***	0.00	1	**568***	0.00	23	**568***	0.00	103
la5	10x5	1.2	503	**503***	0.00	<1	**503***	0.00	23	**503***	0.00	143
la6	15x5	1.15	833	**833***	0.00	<1	**833***	0.00	10800	**833***	0.00	110
la7	15x5	1.15	762	**762***	0.00	2	765	0.39	10800	**762***	0.00	162
la8	15x5	1.15	845	**845***	0.00	1	**845***	0.00	10800	**845***	0.00	159
la9	15x5	1.15	878	**878***	0.00	3	**878***	0.00	10800	**878***	0.00	181
la10	15x5	1.16	866	**866***	0.00	<1	**866***	0.00	10800	**866***	0.00	99
la11	20x5	1.13	1103	**1103***	0.00	16	1129	2.36	10800	**1103***	0.00	243
la12	20x5	1.14	960	**960***	0.00	<1	**960***	0.00	10800	**960***	0.00	248
la13	20x5	1.14	1053	**1053***	0.00	<1	**1053***	0.00	10800	**1053***	0.00	237
la14	20x5	1.13	1123	**1123***	0.00	<1	**1123***	0.00	10800	**1123***	0.00	258
la15	20x5	1.13	1111	**1111***	0.00	2	1117	0.54	10800	**1111***	0.00	292
la16	10x10	1.13	892	**892***	0.00	2	**892***	0.00	13	**892***	0.00	150
la17	10x10	1.13	707	**707***	0.00	1	**707***	0.00	12	**707***	0.00	206
la18	10x10	1.13	842	**842***	0.00	3	**842***	0.00	9	**842***	0.00	208
la19	10x10	1.13	796	**796***	0.00	2	**796***	0.00	17	**796***	0.00	247
la20	10x10	1.13	857	**857***	0.00	1	**857***	0.00	9	**857***	0.00	247
la21	15x10	1.15	1009	**1009***	0.00	107	1041	3.17	10800	**1009***	0.00	368
la22	15x10	1.15	880	**880***	0.00	10	**880***	0.00	7610	**880***	0.00	365
la23	15x10	1.14	950	**950***	0.00	24	962	1.26	10800	**950***	0.00	363
la24	15x10	1.16	908	**908***	0.00	33	912	0.44	10800	**908***	0.00	333
la25	15x10	1.16	936	**936***	0.00	15	**936***	0.00	5405	**936***	0.00	480
la26	20x10	1.14	1106	1118	1.08	10800	1242	12.30	10800	**1106***	0.00	878
la27	20x10	1.14	1181	**1181***	0.00	377	1277	8.13	10800	**1181***	0.00	743
la28	20x10	1.13	1142	**1142***	0.00	8868	1214	6.30	10800	**1142***	0.00	744
la29	20x10	1.14	1107	**1107***	0.00	2016	1221	10.30	10800	1111	0.36	678
la30	20x10	1.14	1148	**1193**	3.92	10800	1304	13.59	10800	**1193**	3.92	613
la31	30x10	1.14	1523	1541	1.18	10800	1931	26.79	10800	**1532**	0.59	1692
la32	30x10	1.14	1698	**1698***	0.00	40	2137	25.85	10800	**1698***	0.00	1666
la33	30x10	1.13	1547	**1547***	0.00	437	1926	24.50	10800	**1547***	0.00	1637
la34	30x10	1.13	1599	**1599***	0.00	118	1875	17.26	10800	**1599***	0.00	1513
la35	30x10	1.13	1736	**1736***	0.00	4	1944	11.98	10800	**1736***	0.00	1758
la36	15x15	1.15	1160	**1160***	0.00	85	**1160***	0.00	6406	**1160***	0.00	843
la37	15x15	1.15	1397	**1397***	0.00	7	**1397***	0.00	10800	**1397***	0.00	885
la38	15x15	1.14	1141	**1141***	0.00	196	1144	0.26	10800	**1141***	0.00	659
la39	15x15	1.14	1184	**1184***	0.00	40	**1184***	0.00	10800	**1184***	0.00	943
la40	15x15	1.15	1144	**1144***	0.00	263	1178	2.97	10800	**1144***	0.00	731
abz5	10x10	1.13	1167	**1167***	0.00	3	**1167***	0.00	28	**1167***	0.00	263
abz6	10x10	1.14	925	**925***	0.00	2	**925***	0.00	11	**925***	0.00	188
abz7	20x15	1.13	564	639	13.30	10800	702	24.47	10800	**610**	8.16	944
abz8	20x15	1.13	586	655	11.77	10800	716	22.18	10800	**637**	8.70	1017
abz9	20x15	1.13	644	**644***	0.00	8269	692	7.45	10800	646	0.31	1667
car1	11x5	1.2	6176	**6176***	0.00	2	**6176***	0.00	230	**6176***	0.00	126
car2	13x4	1.21	6327	**6327***	0.00	4	**6327***	0.00	10800	**6327***	0.00	108
car3	12x5	1.18	6856	**6856***	0.00	2	**6856***	0.00	2489	**6856***	0.00	120
car4	14x4	1.2	7789	**7789***	0.00	<1	**7789***	0.00	10800	**7789***	0.00	134
car5	10x6	1.17	7229	**7229***	0.00	1	**7229***	0.00	32	**7229***	0.00	125
car6	8x9	1.17	7990	**7990***	0.00	3	**7990***	0.00	47	**7990***	0.00	189
car7	7x7	1.18	6123	**6123***	0.00	1	**6123***	0.00	1	**6123***	0.00	89
car8	8x8	1.17	7689	**7689***	0.00	1	**7689***	0.00	22	**7689***	0.00	164
orb1	10x10	1.14	977	**977***	0.00	13	**977***	0.00	318	**977***	0.00	285
orb2	10x10	1.13	865	**865***	0.00	6	**865***	0.00	56	**865***	0.00	172
orb3	10x10	1.14	951	**951***	0.00	14	**951***	0.00	466	**951***	0.00	208
orb4	10x10	1.14	984	**984***	0.00	8	**984***	0.00	143	**984***	0.00	163
orb5	10x10	1.14	842	**842***	0.00	2	**842***	0.00	45	**842***	0.00	243
orb6	10x10	1.14	958	**958***	0.00	10	**958***	0.00	390	**958***	0.00	214
orb7	10x10	1.13	387	**387***	0.00	6	**387***	0.00	48	**387***	0.00	221
orb8	10x10	1.14	894	**894***	0.00	2	**894***	0.00	104	**894***	0.00	185
orb9	10x10	1.14	933	**933***	0.00	4	**933***	0.00	319	**933***	0.00	208
orb10	10x10	1.14	933	**933***	0.00	5	**933***	0.00	51	**933***	0.00	227
Average Gap (%)				0.47			3.39			0.33[1]		
# NB				5			0			6		

[1] The reported average gap is calculated by considering the best run among the ten runs per instance. The average gap when considering all ten runs per instance is 0.37% and the average gap when considering only the worst run per instance is 0.41%.

Table B.6: Detailed results for the HUData($rdata$) set

Instances				CP			MIP			EA(10800)		
Name	$l \times m$	fx	LB	C_{max}	Gap(%)	Time	C_{max}	Gap(%)	Time	C_{max}	Gap(%)	Time
m06	6x6	2.06	47	**47***	0.00	<1	**47***	0.00	<1	**47***	0.00	65
m10	10x10	1.96	686	**686***	0.00	2	**686***	0.00	2336	**686***	0.00	314
m20	20x5	1.94	1022	**1022***	0.00	10800	1033	1.08	10800	**1022***	0.00	499
la1	10x5	1.92	570	572	0.35	10800	572	0.35	10800	**571**	0.18	158
la2	10x5	1.88	529	**529***	0.00	245	532	0.57	10800	530	0.19	155
la3	10x5	1.98	477	**477***	0.00	4058	478	0.21	10800	**477***	0.00	111
la4	10x5	2.02	502	**502***	0.00	1187	504	0.40	10800	**502***	0.00	160
la5	10x5	2.06	457	**457***	0.00	3577	459	0.44	10800	**457***	0.00	169
la6	15x5	1.88	799	**799***	0.00	10800	805	0.75	10800	**799***	0.00	256
la7	15x5	1.96	749	**749***	0.00	10800	756	0.93	10800	**749***	0.00	242
la8	15x5	1.93	765	**765***	0.00	10800	773	1.05	10800	**765***	0.00	232
la9	15x5	1.92	853	**853***	0.00	10800	864	1.29	10800	**853***	0.00	279
la10	15x5	1.96	804	805	0.12	10800	810	0.75	10800	**804***	0.00	200
la11	20x5	2.03	1071	**1071***	0.00	10800	1080	0.84	10800	**1071***	0.00	508
la12	20x5	1.99	936	**936***	0.00	10800	944	0.85	10800	**936***	0.00	345
la13	20x5	1.98	1038	**1038***	0.00	10800	1065	2.60	10800	**1038***	0.00	392
la14	20x5	1.97	1070	**1070***	0.00	10800	1080	0.93	10800	**1070***	0.00	302
la15	20x5	1.98	1089	**1089***	0.00	10800	1125	3.31	10800	1090	0.09	521
la16	10x10	2.01	717	**717***	0.00	2	**717***	0.00	21	**717***	0.00	233
la17	10x10	1.93	646	**646***	0.00	1	**646***	0.00	5	**646***	0.00	129
la18	10x10	1.99	666	**666***	0.00	1	**666***	0.00	103	**666***	0.00	340
la19	10x10	1.96	700	**700***	0.00	2	**700***	0.00	419	**700***	0.00	328
la20	10x10	1.99	756	**756***	0.00	2	**756***	0.00	11	**756***	0.00	330
la21	15x10	2.01	808	857	6.06	10800	914	13.12	10800	**829**	2.60	737
la22	15x10	2.04	737	767	4.07	10800	807	9.50	10800	**753**	2.17	677
la23	15x10	2.04	816	844	3.43	10800	939	15.07	10800	**832**	1.96	725
la24	15x10	1.98	775	814	5.03	10800	845	9.03	10800	**801**	3.35	663
la25	15x10	2.01	752	802	6.65	10800	846	12.50	10800	**782**	3.99	772
la26	20x10	1.96	1056	1068	1.14	10800	1167	10.51	10800	**1059**	0.28	1408
la27	20x10	1.96	1085	1093	0.74	10800	1162	7.10	10800	**1087**	0.18	1053
la28	20x10	2.01	1075	1080	0.47	10800	1183	10.05	10800	**1077**	0.19	996
la29	20x10	2.00	993	1002	0.91	10800	1125	13.29	10800	**996**	0.30	1105
la30	20x10	1.96	1068	1085	1.59	10800	1212	13.48	10800	**1072**	0.37	1558
la31	30x10	1.92	1520	1524	0.26	10800	1969	29.54	10800	1521	0.07	1939
la32	30x10	1.95	1657	1661	0.24	10800	2085	25.83	10800	**1658**	0.06	2835
la33	30x10	1.94	1497	1501	0.27	10800	1862	24.38	10800	**1498**	0.07	2701
la34	30x10	1.95	1535	1537	0.13	10800	2080	35.50	10800	1536	0.07	2369
la35	30x10	1.97	1549	**1550**	0.06	10800	1989	28.41	10800	**1550**	0.06	2772
la36	15x15	1.95	1023	**1023***	0.00	38	1083	5.87	10800	**1023***	0.00	854
la37	15x15	1.94	1062	**1062***	0.00	1107	1140	7.34	10800	1066	0.38	1371
la38	15x15	1.97	954	**954***	0.00	45	1021	7.02	10800	955	0.10	1776
la39	15x15	1.94	1011	**1011***	0.00	241	1050	3.86	10800	1013	0.20	1002
la40	15x15	1.96	955	**955***	0.00	2871	1011	5.86	10800	959	0.42	1494
abz5	10x10	1.98	954	**954***	0.00	7	**954***	0.00	357	**954***	0.00	278
abz6	10x10	1.94	807	**807***	0.00	2	**807***	0.00	20	**807***	0.00	228
abz7	20x15	1.96	492	541	9.96	10800	-	-	10800	**527**	7.11	2272
abz8	20x15	1.93	506	550	8.70	10800	615	21.54	10800	**540**	6.72	2317
abz9	20x15	1.95	497	551	10.87	10800	608	22.33	10800	**539**	8.45	2390
car1	11x5	2.02	5005	5037	0.64	10800	5072	1.34	10800	**5035**	0.60	113
car2	13x4	1.92	5929	5987	0.98	10800	5996	1.13	10800	**5986**	0.96	104
car3	12x5	2.02	5597	5631	0.61	10800	5655	1.04	10800	**5623**	0.46	206
car4	14x4	1.96	6514	**6515**	0.02	10800	6536	0.34	10800	**6515**	0.02	122
car5	10x6	1.92	5615	**5615***	0.00	184	5693	1.39	10800	**5615***	0.00	155
car6	8x9	1.94	6147	**6147***	0.00	2	**6147***	0.00	16	**6147***	0.00	186
car7	7x7	2.08	4425	**4425***	0.00	1	**4425***	0.00	7	**4425***	0.00	122
car8	8x8	1.92	5692	**5692***	0.00	2	**5692***	0.00	42	**5692***	0.00	229
orb1	10x10	1.96	746	**746***	0.00	3	**746***	0.00	78	**746***	0.00	243
orb2	10x10	1.97	696	**696***	0.00	3	**696***	0.00	1004	**696***	0.00	368
orb3	10x10	1.95	712	**712***	0.00	7	**712***	0.00	703	**712***	0.00	418
orb4	10x10	1.94	753	**753***	0.00	3	**753***	0.00	147	**753***	0.00	342
orb5	10x10	1.98	639	**639***	0.00	3	**639***	0.00	139	**639***	0.00	421
orb6	10x10	1.96	754	**754***	0.00	4	**754***	0.00	255	**754***	0.00	221
orb7	10x10	1.98	302	**302***	0.00	5	**302***	0.00	934	**302***	0.00	285
orb8	10x10	1.95	639	**639***	0.00	3	**639***	0.00	1412	**639***	0.00	288
orb9	10x10	1.94	694	**694***	0.00	2	**694***	0.00	55	**694***	0.00	248
orb10	10x10	1.98	742	**742***	0.00	4	**742***	0.00	8515	**742***	0.00	210
Average Gap (%)				0.96			-			0.63[1]		
# NB				5			0			12		

[1] The reported average gap is calculated by considering the best run among the ten runs per instance. The average gap when considering all ten runs per instance is 0.73% and the average gap when considering only the worst run per instance is 0.83%.

Table B.7: Detailed results of the EA for the HUData

Name	edata	rdata	vdata
mt06,10,20	0.00	0.34	0.00
la01-la05	0.00	0.07	0.00
la06-la10	0.00	0.00	0.00
la11-la15	0.29	0.02	0.00
la16-la20	0.00	1.64	0.00
la21-la25	5.25	2.81	0.52
la26-la30	2.58	0.27	0.08
la31-la35	0.21	0.06	0.03
la36-la40	6.65	2.94	0.00
Average Gap (%)	1.99	0.94	0.05

Table C.1: Computational results for the BRData set

Name	LB	EA(800)	EA(50)	EA(100)	CP(1800)	hGA	TS	HHS	MA	SSPR	CDDS
mk1	40	**40***	**40***	**40***	**40***	**40***	**40***	**40***	**40***	**40***	**40***
mk2	26	**26***	**26***	**26***	**26***	**26***	**26***	**26***	**26***	**26***	**26***
mk3	204	**204***	**204***	**204***	**204***	**204***	**204***	**204***	**204***	**204***	**204***
mk4	55	**60**	**60**	**60**	**60**	**60**	**60**	**60**	**60**	**60**	**60**
mk5	168	173	173	173	173	**172**	173	**172**	173	**172**	173
mk6	39	**57**	58	**57**	61	58	58	59	58	**57**	58
mk7	133	**139**	**139**	**139**	140	**139**	144	**139**	**139**	**139**	**139**
mk8	523	**523***	**523***	**523***	**523***	**523***	**523***	**523***	**523***	**523***	**523***
mk9	307	**307***	**307***	**307***	**307***	**307***	**307***	**307***	**307***	**307***	**307***
mk10	183	**193**	198	196	257	197	198	202	201	196	197
Average Gap (%)		6.76	7.35	6.98	11.42	7.24	7.73	7.76	7.51	6.92	7.29
# NB		1	0	0	0	-	-	-	-	-	-
CI-CPU		824	415	561	9431	91	74	-	-	383	96

published by the corresponding authors. The proven optimal solutions are marked with the (∗) symbol, while bold face is used to indicate the best upper bounds. The last rows provide information for the average gap (%) w.r.t the lower bound as well as the sum of the average CI-CPU (computer independent CPU) times of each method across all the instances of each data set. In this case, we normalise the CPU times of our machine with the machine reported by Gao et al. (2008). The normalization coefficient used is: (1248/1192)=1.047. For more details regarding the calculation of CI-CPU times we refer the reader to Section 4.4.

Regarding the BRData set, the EA seems to outperform, in terms of effectiveness, all other solution methods and it presents the lowest average gap. The EA also managed to find a new best solution for the $mk10$ problem instance, which is one of the largest problem instances of this data set. The performance figures are similar for the DPData set. The EA produced the best average gap compared to other algorithms, while the results on lower time limits showcase the effectiveness of the proposed framework in relatively short computational times. Note that the majority of problem instances in this data set have quite high flexibility and contain more than 300 operations. It is worth highlighting that a significantly lower average gap of 1.13% was obtained for a time limit of 10800 seconds (see Table B.3). The observations are similar for the BCData set. The EA performed better compared to all other algorithms and it had the lowest average gap. Note that this set includes medium and large-scale problem instances with low

Table C.2: Computational results for the DPData set

Name	LB	EA(400)	EA(100)	EA(200)	CP(1800)	TS	hGA	EQUA	CDDS	HHS	SSPR
01a	2505	**2505***	**2505***	**2505***	**2505***	2518	2518	2518	2518	**2505***	**2505***
02a	2228	**2228***	2229	2229	2236	2231	2231	2231	2231	2230	2229
03a	2228	**2228***	**2228***	**2228***	2229	2229	2229	2229	2229	**2228***	**2228***
04a	2503	**2503***	**2503***	**2503***	**2503***	**2503***	2515	**2503***	**2503***	2506	**2503***
05a	2189	**2204**	2210	2207	2216	2216	2217	2216	2216	2212	2211
06a	2162	2181	2187	2182	2203	2203	2196	2196	2196	2187	2183
07a	2206	2272	2285	2280	2340	2283	2307	2305	2283	2288	2274
08a	2061	2065	2067	2067	2070	2069	2073	2069	2069	2067	**2064**
09a	2061	2063	2065	2065	2064	2066	2066	2066	2066	2069	**2062**
10a	2197	2263	2280	2276	2323	2291	2315	2291	2291	2297	2269
11a	2017	2052	2061	2056	2078	2063	2071	2065	2063	2061	2051
12a	1969	2015	2024	2022	2045	2034	2030	2031	2031	2027	2018
13a	2161	2242	2253	2250	2265	2260	2257	2257	2257	2263	2248
14a	2161	2165	2168	2167	2170	2167	2167	2167	2167	2164	**2163**
15a	2161	2164	2167	2165	2164	2167	2165	2165	2165	2163	**2162**
16a	2148	2233	2243	2239	2268	2255	2256	2255	2256	2259	2244
17a	2088	2133	2144	2140	2160	2141	2140	2140	2140	2137	2130
18a	2057	2120	2131	2130	2149	2137	2127	2127	2127	2121	2119
Average Gap (%)		1.40%	1.93%	1.60%	2.32%	1.91%	2.02%	1.90%	1.84%	1.78%	1.47%
# NB		8	2	3	0	-	-	-	-	-	-
CI-CPU		6320	1676	3247	30300	2467	6206	-	2890	6279	1934

flexibility. In terms of the computational time, the limit of 1800 seconds seems to be sufficient for the EA to exhibit its best performance and produce new best solutions. One may also observe, that even low time limits are sufficient for the proposed EA to match the performance of the best methods of the literature. Overall, it seems that a good balance between effectiveness and efficiency is provided using a single set of parameters across all data sets.

Table C.4 summarizes the average gap (%) reported from various solution methods for the HUData set (problem instances mt06, mt10, mt20, and la01 to la40). The first column lists the solution methods, while the other columns provide the average gap values for each problem subset. For this data set we provide results considering the time limits of 600, 1800 and 10800 seconds. The EA seems to perform best for the *edata* and *rdata* sets, while it remains very competitive for the *vdata* set. Note that the relative gaps for this particular data set are based exclusively on the lower bounds reported by Mastrolilli and Gambardella (2000) due to the fact that many authors provide only aggregate results for every group.

D Performance Assessment of the EA for the FJSSP

This section extends the experimentation towards a more detailed performance assessment of the EA regarding the evolution progress over time, the average deviation of the best solutions found over multiple runs, and effect of the internal local improvement method.

The first part of the experiments refer to the improvement observed over time on a subset of representative FJSSP instances. For this purpose, six instances with varying difficulty and

Table C.3: Computational results for the BCData set

Name	LB	EA(1800)	EA(20)	EA(50)	CP(1800)	TS	hGA	TSBM	CDDS	SSPR
mt10c1	927	**927***	**927***	**927***	**927***	928	**927***	**927***	928	**927***
mt10cc	908	**908***	**908***	**908***	**908***	910	910	**908***	910	**908***
mt10x	918	**918***	**918***	**918***	**918***	**918***	**918***	922	**918***	**918***
mt10xx	918	**918***	**918***	**918***	**918***	**918***	**918***	**918***	**918***	**918***
mt10xxx	918	**918***	**918***	**918***	**918***	**918***	**918***	**918***	**918***	**918***
mt10xy	905	**905***	**905***	**905***	**905***	906	**905***	**905***	906	**905***
mt10xyz	847	**847***	**847***	**847***	**847***	**847***	849	849	849	**847***
setb4c9	914	**914***	**914***	**914***	**914***	919	**914***	**914***	919	**914***
setb4cc	907	**907***	**907***	**907***	**907***	909	914	**907***	909	**907***
setb4x	925	**925***	**925***	**925***	**925***	**925***	**925***	**925***	**925***	**925***
setb4xx	925	**925***	**925***	**925***	**925***	**925***	**925***	**925***	**925***	**925***
setb4xxx	925	**925***	**925***	**925***	**925***	**925***	**925***	**925***	**925***	**925***
setb4xy	910	**910***	**910***	**910***	**910***	916	916	**910***	916	**910***
setb4xyz	902	905	905	905	**902***	905	905	903	905	905
seti5c12	1169	**1169***	1170	1170	**1169***	1174	1175	1174	1174	1170
seti5cc	1135	**1135***	1136	**1135***	**1135***	1136	1138	1136	1136	**1135***
seti5x	1198	**1198***	**1198***	**1198***	**1198***	1201	1204	**1198***	1201	**1198***
seti5xx	1194	**1194***	1197	1197	**1194***	1199	1202	1197	1199	1197
seti5xxx	1194	**1194***	1197	1197	**1194***	1197	1204	1197	1197	**1194***
seti5xy	1135	**1135***	1136	**1135***	**1135***	1136	1136	1136	1136	**1135***
seti5xyz	1125	**1125***	**1125***	**1125***	**1125***	**1125***	1126	1128	**1125***	**1125***
Average Gap (%)		0.02	0.04	0.03	0.00	0.18	0.25	0.10	0.19	0.03
# NB		2	0	0	3	-	-	-	-	-
CI-CPU		968	375	576	468	356	821	1068	253	248

Table C.4: Average gap (%) of various solutions methods for the HUData set (problem instances la01 to la40 and mt06/10/20)

Method	edata	rdata	vdata	# NB
TS	2.17	1.23	0.10	-
hGA	2.13	1.17	0.08	-
CDDS	2.32	1.34	0.11	-
HHS	2.11	1.18	0.11	-
SSPR	2.03	1.03	**0.04**	-
CP(1800)	2.11	1.37	0.08	7
EA(600)	2.01	1.02	0.06	11
EA(1800)	2.00	0.96	0.05	16
EA(10800)	**1.99**	**0.94**	0.05	18

sizes were selected from the BCData, BRData and DPData benchmark data sets. For each problem instance 10 runs were performed with a time limit of 1000 seconds. For each run, we recorded the makespan of the best solution s of the population. For each recorded solution cost C^s_{max}, we calculated the percentage difference w.r.t. the cost of the best known solution $s*$ of the literature, denoted as C^{s*}_{max}, i.e. $\frac{C^s_{max} - C^{s*}_{max}}{C^{s*}_{max}}$. These results are presented in Figure D. 1. The Y-axis depicts the average relative percentage deviation (ARPD) representing all 10 runs of each problem instance, while the X-axis depicts the time in a logarithmic scale. One can observe that during the first 100 seconds the improvement slope is steep. This indicates that high quality solutions can be found early in the search process. In this particular problem subset, we can see that the ARPD values after 200 seconds are less than 1%, while we can also notice improvements after 1000 seconds of runtime. This shows that the current best solutions are already very close to the best known solutions. Note that in cases of ARPD values less than 0%, the EA manages to obtain new best solutions.

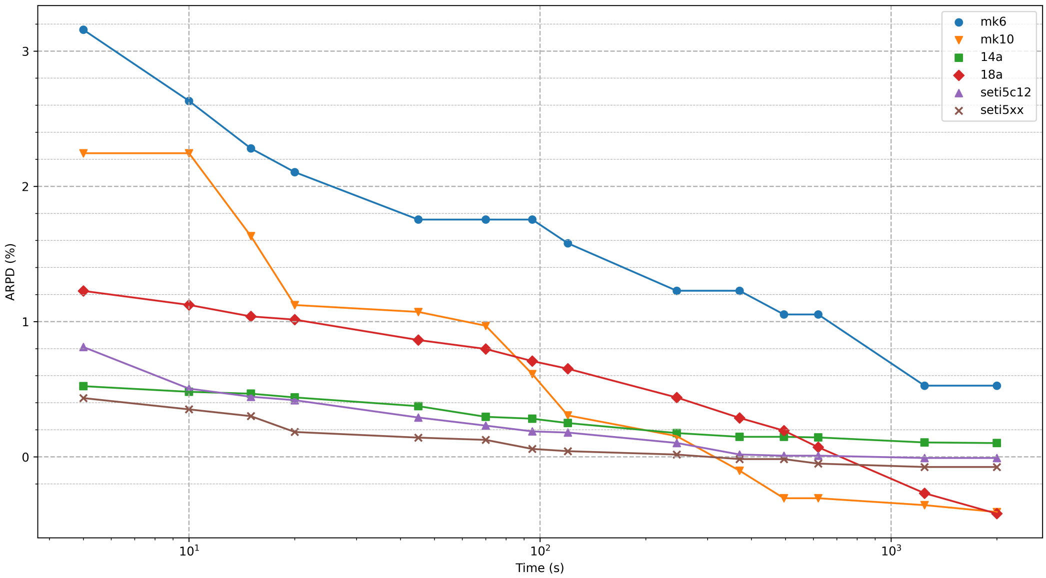

Figure D. 1: Performance of the EA for a selected set of representative FJSSP instances. The graph reports the average percentage difference relative to the best known solution of the literature

The second experiment considered a total of 178 FJSSP instances as described in Appendix B and examined the deviation of best found solutions over multiple runs. Figure D.2 summarizes the distribution of the ARPD values obtained per run for all problem instances grouped for each data set. These results were produced assuming a time limit of 10800 seconds. The top and the bottom edges of the boxes designate the Q_1 and Q_3 quartiles of the distribution of the values. On every box plot, the orange line denotes the median value, the whisker above the box designates the largest observed value smaller than $Q_3 + 1.5IQR$, while the whisker below the box designates the smallest observed value larger than $Q_1 - 1.5IQR$. IQR is the interquartile range, i.e. $IQR = Q_3 - Q_1$. Values past the whiskers lines are considered outliers and are denoted in the graph with small black circles. One can observe that on all data sets the median is very close to 0% and this strongly indicates the reliability and very small variability of the proposed EA over multiple runs. For the data sets HVData, HEData and BCData, almost no deviation is observed with very few outliers, while maximum and minimum observed deviations (not considering outliers) from the best found solutions is less than 0.2% and -0.4%, respectively. Lastly, across all problem instances approximately 88.76% of the best solutions found from each individual run are within 0.1% of the best known solution values.

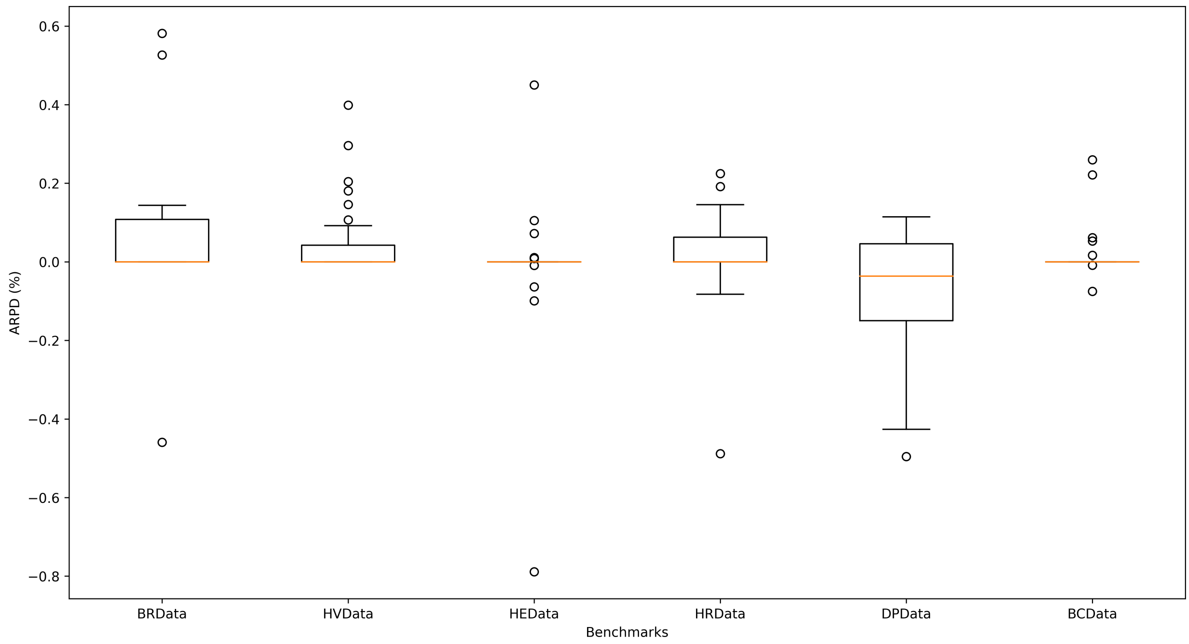

Figure D. 2: Distribution of the average percentage difference relative to the best known solutions of the literature for all FJSSP problem instances grouped by benchmark data set

Table D.1: Results obtained from the main components of the proposed EA on the DPData benchmark data set

Name	LB	MS-TS	EA w/o TS	EA
01a	2505	2505	2553	2505
02a	2228	2230	2237	2228
03a	2228	2228	2231	2228
04a	2503	2503	2517	2503
05a	2189	2216	2222	2204
06a	2162	2200	2199	2177
07a	2206	2340	2387	2255
08a	2061	2067	2085	2065
09a	2061	2065	2076	2062
10a	2197	2337	2345	2246
11a	2017	2061	2084	2042
12a	1969	2033	2047	2005
13a	2161	2255	2326	2234
14a	2161	2167	2189	2163
15a	2161	2164	2181	2163
16a	2148	2251	2353	2230
17a	2088	2142	2160	2119
18a	2057	2130	2164	2104
Average Gap (%)		2.06	3.24	1.13

The third set of experiments seeks to investigate the effect of the main components of the proposed EA to the overall performance. We were interested to examine the interplay between the evolutionary and the local search components. For this purpose, we considered a multi-start Tabu Search (MS-TS) scheme, where multiple initial solutions are generated as described in Section 3.1 and then improved by the local improvement method presented in Section 3.5 with the objective of minimizing the makespan. Alongside this, we also considered a simplified version of the proposed EA, where the local improvement method is disabled, namely EA w/o TS. To make the comparison fair with regards to the full EA scheme, for the multi-start TS *maxIterations* was set to 10000, while *maxGenerations* was set to 500. In all cases, a maximum time limit of 10800 seconds was considered. For these experiments we used the DPDATA benchmark set as it is the most demanding for the FJSSP.

Table D.1 presents the results obtained. The first column provides the name, the second column includes the best available known lower bound, while the remaining columns present the results of the MS-TS, the EA w/o TS and the EA for each instance, respectively. The bottom row of the table shows the average gap % that was obtained w.r.t the available lower bounds. One can notice that the EA w/o TS had overall the worst performance. The MS-TS performs better; however, the results of the full EA scheme are superior. The reader may conclude that there is an evident collaborative effect between the proposed local search and evolutionary components, while without any guidance a pure local search scheme has limited search capabilities.

E Detailed results for the FJSSP with arbitrary precedence graphs

This section contains computational experiments for the FJSSP with arbitrary precedence graphs.

Tables E.1 – E.3 contain the detailed results of the computational experiments conducted in Section 4.5. These tables share the following structure: the first three columns include the name, the size ($l \times m \times O_u$) and the density (δ) of the problem instance, while the remaining columns include the results for each flexibility value fx, namely the C_{max} and the Time(s) to solve each problem to optimality in seconds using the CP model.

Table E.4 presents the results obtained for the SCPC group (see Appendix F for the instance generation process). The first set of columns refers to the problem specifications, i.e., the name, the size, the slope γ, the flexibility fx and the best lower bound (LB) produced during our experiments. The remaining sets of columns provide the upper bound, the optimality gap % and the CPU time in seconds for the CP, the proposed MIP, the MIP of Birgin et al. (2014) and the EA, respectively.

Tables E.5 – E.8 provide the results obtained for the BCPC group (see Appendix F for the instance generation process). The structure of the tables remains the same; however, they are organized according to the density of the precedence graphs, i.e., low, medium and high values of γ. Given that BCPC instances are large scale, instead of reporting optimality gaps from lower bounds, we report % deviations from the best known solutions. To that end, the average % deviation values are provided in the bottom row of the tables. For the sake of readability, an extra column BKS is added to indicate the best known solutions (upper bounds). Note also that Table E.5 refers to the baseline FJSSP instances (BCPCN01–BCPCN27) with linear precedence graphs.

Table E.1: Computational results for instances generated with maximum two predecessors/successors

Instances			$fx = 1$		$fx = 2$		$fx = 3$	
Name	$l \times m \times O_u$	δ	C_{max}	Time	C_{max}	Time	C_{max}	Time
d1	2x4x30	0.060	721	1	428	1	397	2
d2	2x4x30	0.059	721	2	428	2	396	3
d3	2x4x30	0.058	721	2	428	1	396	2
d4	2x4x30	0.057	721	1	427	1	396	2
d5	2x4x30	0.056	721	2	427	1	396	2
d6	2x4x30	0.055	714	1	427	1	396	1
d7	2x4x30	0.054	714	2	427	2	396	1
d8	2x4x30	0.053	714	2	427	1	393	1
d9	2x4x30	0.052	714	2	424	2	391	1
d10	2x4x30	0.051	714	1	424	2	391	1
d11	2x4x30	0.050	707	1	422	1	384	3
d12	2x4x30	0.049	705	1	422	2	383	2
d13	2x4x30	0.048	703	1	422	3	383	4
d14	2x4x30	0.047	682	1	421	4	382	2
d15	2x4x30	0.046	682	1	421	2	382	3
d16	2x4x30	0.045	682	1	421	2	381	5
d17	2x4x30	0.044	682	1	421	2	381	3
d18	2x4x30	0.043	682	2	416	3	381	5
d19	2x4x30	0.042	682	1	412	2	381	5
d20	2x4x30	0.041	682	1	412	2	379	3
d21	2x4x30	0.040	671	2	406	6	376	8
d22	2x4x30	0.039	671	2	406	6	375	12
d23	2x4x30	0.038	667	3	406	3	370	13
d24	2x4x30	0.037	662	3	405	9	370	9
d25	2x4x30	0.036	656	4	403	6	370	17
d26	2x4x30	0.035	645	4	402	11	368	24
d27	2x4x30	0.034	623	5	399	6	368	15
d28	2x4x30	0.033	623	6	399	10	367	17
d29	2x4x30	0.032	615	2	399	10	367	35

Table E.2: Computational results for instances generated with maximum three predecessors/successors

Instances			$fx = 1$		$fx = 2$		$fx = 3$	
Name	$l \times m \times O_u$	δ	C_{max}	Time	C_{max}	Time	C_{max}	Time
t1	2x4x21	0.119	475	1	287	4	260	20
t2	2x4x21	0.117	473	1	287	3	260	42
t3	2x4x21	0.115	473	1	287	7	260	33
t4	2x4x21	0.113	473	1	287	3	259	40
t5	2x4x21	0.111	466	1	287	5	259	58
t6	2x4x21	0.109	466	1	287	3	257	49
t7	2x4x21	0.107	466	1	287	4	257	76
t8	2x4x21	0.105	466	1	287	4	256	57
t9	2x4x21	0.103	466	1	287	4	255	59
t10	2x4x21	0.101	466	1	287	3	255	54
t11	2x4x21	0.099	451	1	287	5	255	50
t12	2x4x21	0.097	451	1	284	5	252	55
t13	2x4x21	0.095	451	1	283	5	252	61
t14	2x4x21	0.093	450	1	283	6	252	41
t15	2x4x21	0.091	449	1	283	6	252	53
t16	2x4x21	0.089	449	<1	283	5	252	66
t17	2x4x21	0.087	449	<1	280	6	247	35
t18	2x4x21	0.085	449	<1	280	6	247	49
t19	2x4x21	0.083	443	<1	275	6	247	59
t20	2x4x21	0.081	441	1	275	5	245	102
t21	2x4x21	0.079	441	1	269	4	245	86
t22	2x4x21	0.077	441	1	268	4	243	137
t23	2x4x21	0.075	441	1	267	4	242	111
t24	2x4x21	0.073	441	1	266	3	238	100
t25	2x4x21	0.071	441	1	266	4	238	113
t26	2x4x21	0.069	441	2	266	12	238	194
t27	2x4x21	0.067	436	1	265	8	238	266
t28	2x4x21	0.065	436	1	265	9	238	164
t29	2x4x21	0.063	436	1	265	8	238	204
t30	2x4x21	0.061	436	1	265	9	238	250
t31	2x4x21	0.059	434	1	265	13	238	223
t32	2x4x21	0.057	416	<1	256	63	237	1736
t33	2x4x21	0.055	416	1	255	34	237	1540
t34	2x4x21	0.053	416	<1	253	39	237	3186
t35	2x4x21	0.051	416	<1	253	60	235	3307
t36	2x4x21	0.050	416	<1	253	55	235	5071

Table E.3: Computational results for instances generated with maximum four predecessors/successors

Instances			$fx = 1$		$fx = 2$		$fx = 3$	
Name	$l \times m \times O_u$	δ	C_{max}	Time	C_{max}	Time	C_{max}	Time
q1	2x4x20	0.156	442	1	248	9	226	222
q2	2x4x20	0.154	442	1	248	8	226	194
q3	2x4x20	0.152	441	1	247	12	226	227
q4	2x4x20	0.149	441	1	247	9	226	291
q5	2x4x20	0.147	441	1	245	7	224	199
q6	2x4x20	0.145	437	1	245	8	224	183
q7	2x4x20	0.143	437	1	245	10	223	210
q8	2x4x20	0.141	437	1	245	10	223	138
q9	2x4x20	0.139	432	<1	245	9	223	193
q10	2x4x20	0.136	432	1	245	8	223	150
q11	2x4x20	0.134	432	1	245	9	223	169
q12	2x4x20	0.132	432	1	245	10	223	219
q13	2x4x20	0.130	430	1	245	9	223	206
q14	2x4x20	0.128	430	1	245	8	220	237
q15	2x4x20	0.126	430	1	245	11	220	183
q16	2x4x20	0.123	429	1	245	13	220	267
q17	2x4x20	0.121	429	1	245	12	220	276
q18	2x4x20	0.119	429	1	245	13	220	276
q19	2x4x20	0.117	420	1	245	13	218	455
q20	2x4x20	0.115	413	1	244	18	218	389
q21	2x4x20	0.113	413	1	243	13	218	383
q22	2x4x20	0.110	413	1	243	17	218	462
q23	2x4x20	0.108	413	<1	243	12	218	322
q24	2x4x20	0.106	413	<1	243	18	218	719
q25	2x4x20	0.104	413	1	242	19	218	700
q26	2x4x20	0.102	397	1	242	24	218	513
q27	2x4x20	0.100	397	<1	242	23	218	429
q28	2x4x20	0.097	397	<1	242	25	217	461
q29	2x4x20	0.095	394	1	242	26	217	629
q30	2x4x20	0.093	394	1	242	31	217	661
q31	2x4x20	0.091	392	<1	239	19	216	575
q32	2x4x20	0.089	392	1	239	28	216	692
q33	2x4x20	0.087	392	1	239	24	216	660
q34	2x4x20	0.084	392	2	237	39	215	620
q35	2x4x20	0.082	392	<10	237	35	214	994
q36	2x4x20	0.080	392	1	237	50	214	980
q37	2x4x20	0.078	392	<1	235	50	214	3061
q38	2x4x20	0.076	392	1	235	55	214	1738
q39	2x4x20	0.074	392	<1	235	66	214	3491
q40	2x4x20	0.071	392	<1	235	68	214	3183
q41	2x4x20	0.069	392	<1	235	63	214	3546
q42	2x4x20	0.067	392	<1	234	57	214	3712
q43	2x4x20	0.065	392	<1	234	61	214	3279
q44	2x4x20	0.063	392	<1	231	73	214	4767
q45	2x4x20	0.061	391	<1	231	102	214	4786
q46	2x4x20	0.058	389	1	231	103	214	7688
q47	2x4x20	0.057	383	<1	231	114	214	6360
q48	2x4x20	0.056	380	<1	231	139	214	9093
q49	2x4x20	0.054	373	1	231	139	214	15845
q50	2x4x20	0.052	373	1	231	134	214	18875

Table E.4: Computational results for the SCPC benchmark set

Instances					CP			MIP			MIP (Birgin et.al)			EA		
Name	$l \times m \times O_u$	γ	fx	LB	C_{max}	Gap(%)	Time	C_{max}	Gap(%)	Time	C_{max}	Gap(%)	Time	C_{max}	Gap(%)	Time
SCPCN01	4x4x15	-	1.62	71	**71***	0.00	1	**71***	0.00	12	**71***	0.00	12	**71***	0.00	1
SCPCN02	4x8x15	-	1.58	59	**59***	0.00	<1	**59***	0.00	1	**59***	0.00	1	**59***	0.00	<1
SCPCN03	8x4x15	-	1.62	86	**103**	19.77	10800	109	26.74	10800	113	31.40	10800	**103**	19.77	15
SCPCN04	8x8x15	-	1.62	73	**73***	0.00	5	**73***	0.00	106	**73***	0.00	106	**73***	0.00	1
SCPCN05	4x4x15	-	2.18	51	**51***	0.00	2	**51***	0.00	93	**51***	0.00	93	**51***	0.00	1
SCPCN06	4x8x15	-	2.2	47	**47***	0.00	1	**47***	0.00	5	**47***	0.00	5	**47***	0.00	1
SCPCN07	8x4x15	-	2.17	54	**81**	50.00	10800	92	70.37	10800	97	79.63	10800	**81**	50.00	68
SCPCN08	8x8x15	-	2.19	56	**56***	0.00	340	58	3.57	10800	58	3.57	10800	**56***	0.00	1
SCPC01	4x4x15	0.2	1.62	56	**56***	0.00	4	57	1.79	10800	57	1.79	10800	**56***	0.00	273
SCPC02	4x8x15	0.2	1.58	31	**31***	0.00	1	**31***	0.00	32	**31***	0.00	8	**31***	0.00	3
SCPC03	8x4x15	0.2	1.62	84	**101**	20.24	10800	111	32.14	10800	117	39.29	10800	**101**	20.24	73
SCPC04	8x8x15	0.2	1.62	54	**54***	0.00	394	58	7.41	10800	58	7.41	10800	**54***	0.00	393
SCPC05	4x4x15	0.5	1.62	56	**56***	0.00	3	57	1.79	11579	57	1.79	10800	**56***	0.00	317
SCPC06	4x8x15	0.5	1.58	32	**32***	0.00	1	**32***	0.00	8	**32***	0.00	21	**32***	0.00	550
SCPC07	8x4x15	0.5	1.62	84	**102**	21.43	10800	110	30.95	10800	121	44.05	10800	**102**	21.43	83
SCPC08	8x8x15	0.5	1.62	54	**54***	0.00	209	58	7.41	10800	58	7.41	10800	**54***	0.00	112
SCPC09	4x4x15	0.8	1.62	57	**57***	0.00	2	58	1.75	10800	58	1.75	10800	**57***	0.00	373
SCPC10	4x8x15	0.8	1.58	33	**33***	0.00	1	**33***	0.00	4	**33***	0.00	5	**33***	0.00	247
SCPC11	8x4x15	0.8	1.62	84	102	21.43	10800	110	30.95	10800	114	35.71	10800	**101**	20.24	292
SCPC12	8x8x15	0.8	1.62	54	**54***	0.00	272	58	7.41	10800	59	9.26	10800	**54***	0.00	800
SCPC13	4x4x15	0.2	2.18	26	**41**	57.69	10800	43	65.38	10800	46	76.92	10800	**41**	57.69	6
SCPC14	4x8x15	0.2	2.2	24	**24***	0.00	1	**24***	0.00	700	**24***	0.00	126	**24***	0.00	3
SCPC15	8x4x15	0.2	2.17	26	**80**	207.69	10800	97	273.08	10800	109	319.23	10800	**80**	207.69	83
SCPC16	8x8x15	0.2	2.19	24	**44**	83.33	10800	50	108.33	10800	54	125.00	10800	**44**	83.33	53
SCPC17	4x4x15	0.5	2.18	27	**41**	51.85	10800	43	59.26	10800	42	55.56	10800	**41**	51.85	3
SCPC18	4x8x15	0.5	2.2	26	**26***	0.00	2	**26***	0.00	560	**26***	0.00	141	**26***	0.00	558
SCPC19	8x4x15	0.5	2.17	31	**80**	158.06	10800	91	193.55	10800	112	261.29	10800	**80**	158.06	154
SCPC20	8x8x15	0.5	2.19	27	**44**	62.96	10800	47	74.07	10800	50	85.19	10800	**44**	62.96	256
SCPC21	4x4x15	0.8	2.18	41	**41***	0.00	8311	43	4.88	10800	43	4.88	10800	**41***	0.00	89
SCPC22	4x8x15	0.8	2.2	27	**27***	0.00	1	**27***	0.00	63	**27***	0.00	14	28	3.70	3
SCPC23	8x4x15	0.8	2.17	31	**80**	158.06	10800	100	222.58	10800	115	270.97	10800	**80**	158.06	299
SCPC24	8x8x15	0.8	2.19	27	**44**	62.96	10800	48	77.78	10800	53	96.30	10800	45	66.67	31
Average Gap (%)						30.48			40.66			48.70			30.68[1]	

[1] The reported average gap is calculated by considering the best run among the ten runs per instance. The average gap when considering all ten runs per instance is 30.68% and the average gap when considering only the worst run per instance is 30.68%.

Table E.5: Computational results for the baseline BCPCN01–BCPCN27 problem instances

Instances					CP			MIP			MIP (Birgin et al.)			EA(10800)		
Name	$l \times m \times O_u$	fx	LB	BKS	C_{max}	Dev(%)	Time	C_{max}	Dev(%)	Time	C_{max}	Dev(%)	Time	C_{max}	Dev(%)	Time
BCPCN01	10x5x20	3.52	69	118	119	0.85	10800	-	-	10800	-	-	10800	**118**	0.00	532
BCPCN02	10x5x30	3.51	99	182	186	2.20	10800	-	-	10800	-	-	10800	**182**	0.00	390
BCPCN03	10x5x40	3.50	139	247	253	2.43	10800	-	-	10800	-	-	10800	**247**	0.00	5348
BCPCN04	10x10x20	3.52	63	71	72	1.41	10800	79	11.27	10800	85	19.72	10800	**71**	0.00	124
BCPCN05	10x10x30	3.51	100	106	108	1.89	10800	-	-	10800	-	-	10800	**106**	0.00	207
BCPCN06	10x10x40	3.50	127	145	150	3.45	10800	-	-	10800	-	-	10800	**145**	0.00	1668
BCPCN07	10x15x20	3.49	71	71	**71***	0.00	1	**71***	0.00	1118	71*	0.00	410	**71***	0.00	2
BCPCN08	10x15x30	3.51	98	98	**98***	0.00	30	105	7.14	10800	117	19.39	10800	**98***	0.00	49
BCPCN09	10x15x40	3.48	128	128	**128***	0.00	271	-	-	10800	-	-	10800	129	0.78	291
BCPCN10	20x5x20	3.51	74	236	248	5.08	10800	-	-	10800	-	-	10800	**236**	0.00	5071
BCPCN11	20x5x30	3.52	105	366	370	1.09	10800	-	-	10800	-	-	10800	**366**	0.00	6588
BCPCN12	20x5x40	3.52	141	484	514	6.20	10800	-	-	10800	-	-	10800	**484**	0.00	5184
BCPCN13	20x10x20	3.52	73	114	117	2.63	10800	-	-	10800	-	-	10800	**114**	0.00	1673
BCPCN14	20x10x30	3.50	101	176	179	1.70	10800	-	-	10800	-	-	10800	**176**	0.00	5832
BCPCN15	20x10x40	3.50	133	243	244	0.41	10800	-	-	10800	-	-	10800	**243**	0.00	5940
BCPCN16	20x15x20	3.49	71	86	**86**	0.00	10800	-	-	10800	-	-	10800	**86**	0.00	1476
BCPCN17	20x15x30	3.52	103	132	138	4.55	10800	-	-	10800	-	-	10800	**132**	0.00	3191
BCPCN18	20x15x40	3.50	146	183	**183**	0.00	10800	-	-	10800	-	-	10800	185	1.09	3687
BCPCN19	30x5x20	3.54	75	357	376	5.32	10800	-	-	10800	-	-	10800	**357**	0.00	5292
BCPCN20	30x5x30	3.53	120	567	588	3.70	10800	-	-	10800	-	-	10800	**567**	0.00	5184
BCPCN21	30x5x40	3.54	141	733	**733**	0.00	10800	-	-	10800	-	-	10800	744	1.50	7236
BCPCN22	30x10x20	3.51	73	173	176	1.73	10800	-	-	10800	-	-	10800	**173**	0.00	4487
BCPCN23	30x10x30	3.52	114	279	282	1.08	10800	-	-	10800	-	-	10800	**279**	0.00	4644
BCPCN24	30x10x40	3.51	133	374	**374**	0.00	10800	-	-	10800	-	-	10800	378	1.07	7452
BCPCN25	30x15x20	3.52	73	122	123	0.82	10800	-	-	10800	-	-	10800	**122**	0.00	4139
BCPCN26	30x15x30	3.52	104	184	189	2.72	10800	-	-	10800	-	-	10800	**184**	0.00	6156
BCPCN27	30x15x40	3.52	146	260	**260**	0.00	10800	-	-	10800	-	-	10800	262	0.77	7128
Average Deviation (%)						1.82			-			-			0.17[1]	

[1] The reported average deviation is calculated by considering the best run among the ten runs per instance. The average deviation when considering all ten runs per instance is 0.34% and the average deviation when considering only the worst run per instance is 0.48%.

Table E.6: Computational results for the BCPC01–BCPC27 problem instances ($\gamma = 0.2$)

Instances					CP			MIP			MIP (Birgin et al.)			EA(10800)		
Name	$l \times m \times O_u$	fx	LB	BKS	C_{max}	Dev(%)	Time	C_{max}	Dev(%)	Time	C_{max}	Dev(%)	Time	C_{max}	Dev(%)	Time
BCPC01	10x5x20	3.52	33	116	**116**	0.00	10800	169	45.69	10800	-	-	10800	117	0.86	795
BCPC02	10x5x30	3.51	44	179	180	0.56	10800	283	58.10	10800	-	-	10800	**179**	0.00	7560
BCPC03	10x5x40	3.5	54	244	**244**	0.00	10800	-	-	10800	-	-	10800	245	0.41	2350
BCPC04	10x10x20	3.52	28	55	56	1.82	10800	-	-	10800	92	67.27	10800	**55**	0.00	1089
BCPC05	10x10x30	3.51	44	83	86	3.61	10800	122	46.99	10800	-	-	10800	**83**	0.00	1458
BCPC06	10x10x40	3.5	51	116	120	3.45	10800	196	68.97	10800	-	-	10800	**116**	0.00	1796
BCPC07	10x15x20	3.49	31	39	**39**	0.00	10800	49	25.64	10800	45	15.38	10800	**39**	0.00	124
BCPC08	10x15x30	3.51	40	57	58	1.75	10800	-	-	10800	-	-	10800	**57**	0.00	1910
BCPC09	10x15x40	3.48	47	78	79	1.28	10800	-	-	10800	-	-	10800	**78**	0.00	1838
BCPC10	20x5x20	3.51	36	237	238	0.42	10800	-	-	10800	-	-	10800	**237**	0.00	2001
BCPC11	20x5x30	3.52	58	356	360	1.12	10800	-	-	10800	-	-	10800	**356**	0.00	5400
BCPC12	20x5x40	3.52	63	487	494	1.44	10800	-	-	10800	-	-	10800	**487**	0.00	4644
BCPC13	20x10x20	3.52	32	113	115	1.77	10800	209	84.96	10800	-	-	10800	**113**	0.00	4736
BCPC14	20x10x30	3.5	46	176	**176**	0.00	10800	-	-	10800	-	-	10800	177	0.57	4968
BCPC15	20x10x40	3.5	55	234	**234**	0.00	10800	-	-	10800	-	-	10800	242	3.42	4752
BCPC16	20x15x20	3.49	34	77	79	2.60	10800	-	-	10800	-	-	10800	**77**	0.00	655
BCPC17	20x15x30	3.52	46	118	122	3.39	10800	-	-	10800	-	-	10800	**118**	0.00	2211
BCPC18	20x15x40	3.5	61	165	**165**	0.00	10800	-	-	10800	-	-	10800	**165**	0.00	6912
BCPC19	30x5x20	3.54	35	356	**356**	0.00	10800	-	-	10800	-	-	10800	358	0.56	7128
BCPC20	30x5x30	3.53	48	546	**546**	0.00	10800	-	-	10800	-	-	10800	549	0.55	6588
BCPC21	30x5x40	3.54	58	716	**716**	0.00	10800	-	-	10800	-	-	10800	720	0.56	4860
BCPC22	30x10x20	3.51	31	170	**170**	0.00	10800	-	-	10800	-	-	10800	**170**	0.00	4529
BCPC23	30x10x30	3.52	47	263	272	3.42	10800	-	-	10800	-	-	10800	**263**	0.00	6264
BCPC24	30x10x40	3.51	58	351	**351**	0.00	10800	-	-	10800	-	-	10800	363	3.42	4860
BCPC25	30x15x20	3.52	34	118	122	3.39	10800	-	-	10800	-	-	10800	**118**	0.00	1126
BCPC26	30x15x30	3.52	48	179	184	2.79	10800	-	-	10800	-	-	10800	**179**	0.00	4428
BCPC27	30x15x40	3.52	62	234	249	6.41	10800	-	-	10800	-	-	10800	**234**	0.00	5076
Average Deviation (%)						1.45			-			-			0.38[1]	

[1] The reported average deviation is calculated by considering the best run among the ten runs per instance. The average deviation when considering all ten runs per instance is 0.50% and the average deviation when considering only the worst run per instance is 0.63%.

Table E.7: Computational results for the BCPC28–BCPC54 problem instances ($\gamma = 0.5$)

Instances					CP			MIP			MIP (Birgin et al.)			EA(10800)		
Name	$l \times m \times O_u$	fx	LB	BKS	C_{max}	Dev(%)	Time	C_{max}	Dev(%)	Time	C_{max}	Dev(%)	Time	C_{max}	Dev(%)	Time
BCPC28	10x5x20	3.52	36	117	**117**	0.00	10800	154	31.62	10800	-	-	10800	**117**	0.00	288
BCPC29	10x5x30	3.51	43	179	**179**	0.00	10800	-	-	10800	-	-	10800	180	0.56	2120
BCPC30	10x5x40	3.5	54	246	**246**	0.00	10800	383	55.69	10800	-	-	10800	**246**	0.00	6696
BCPC31	10x10x20	3.52	31	56	57	1.79	10800	69	23.21	10800	95	69.64	10800	**56**	0.00	133
BCPC32	10x10x30	3.51	44	83	84	1.20	10800	136	63.86	10800	-	-	10800	**83**	0.00	1951
BCPC33	10x10x40	3.5	55	115	118	2.61	10800	-	-	10800	-	-	10800	**115**	0.00	1865
BCPC34	10x15x20	3.49	33	40	**40**	0.00	10800	47	17.50	10800	52	30.00	10800	**40**	0.00	1101
BCPC35	10x15x30	3.51	43	58	59	1.72	10800	77	32.76	10800	112	93.10	10800	**58**	0.00	3104
BCPC36	10x15x40	3.48	49	77	79	2.60	10800	-	-	10800	-	-	10800	**77**	0.00	2119
BCPC37	20x5x20	3.51	36	235	**235**	0.00	10800	-	-	10800	-	-	10800	238	1.28	4392
BCPC38	20x5x30	3.52	58	359	**359**	0.00	10800	-	-	10800	-	-	10800	**359**	0.00	6372
BCPC39	20x5x40	3.52	65	486	**486**	0.00	10800	-	-	10800	-	-	10800	**486**	0.00	7560
BCPC40	20x10x20	3.52	33	113	114	0.88	10800	-	-	10800	-	-	10800	**113**	0.00	1758
BCPC41	20x10x30	3.5	45	170	177	4.12	10800	-	-	10800	-	-	10800	**170**	0.00	2393
BCPC42	20x10x40	3.5	55	234	237	1.28	10800	-	-	10800	-	-	10800	**234**	0.00	4320
BCPC43	20x15x20	3.49	34	78	79	1.28	10800	-	-	10800	-	-	10800	**78**	0.00	753
BCPC44	20x15x30	3.52	50	122	**122**	0.00	10800	-	-	10800	-	-	10800	**122**	0.00	5400
BCPC45	20x15x40	3.5	63	163	**163**	0.00	10800	-	-	10800	-	-	10800	**163**	0.00	7560
BCPC46	30x5x20	3.54	36	352	**352**	0.00	10800	-	-	10800	-	-	10800	355	0.85	4752
BCPC47	30x5x30	3.53	52	555	556	0.18	10800	-	-	10800	-	-	10800	**555**	0.00	5832
BCPC48	30x5x40	3.54	57	720	727	0.97	10800	-	-	10800	-	-	10800	**720**	0.00	6588
BCPC49	30x10x20	3.51	36	172	174	1.16	10800	-	-	10800	-	-	10800	**172**	0.00	7236
BCPC50	30x10x30	3.52	48	262	271	3.44	10800	-	-	10800	-	-	10800	**262**	0.00	4968
BCPC51	30x10x40	3.51	59	346	**346**	0.00	10800	-	-	10800	-	-	10800	364	5.20	4968
BCPC52	30x15x20	3.52	34	119	121	1.68	10800	-	-	10800	-	-	10800	**119**	0.00	1346
BCPC53	30x15x30	3.52	52	176	187	6.25	10800	-	-	10800	-	-	10800	**176**	0.00	4320
BCPC54	30x15x40	3.52	67	246	253	2.85	10800	-	-	10800	-	-	10800	**246**	0.00	5940
Average Deviation (%)						1.26			-			-			0.29[1]	

[1] The reported average deviation is calculated by considering the best run among the ten runs per instance. The average deviation when considering all ten runs per instance is 0.46% and the average deviation when considering only the worst run per instance is 0.63%.

Table E.8: Computational results for the BCPC55–BCPC81 problem instances ($\gamma = 0.8$)

Instances					CP			MIP			MIP (Birgin et al.)			EA(10800)		
Name	$l \times m \times O_u$	fx	LB	BKS	C_{max}	Dev(%)	Time	C_{max}	Dev(%)	Time	C_{max}	Dev(%)	Time	C_{max}	Dev(%)	Time
BCPC55	10x5x20	3.52	38	117	**117**	0.00	10800	181	54.70	10800	197	68.38	10800	**117**	0.00	1573
BCPC56	10x5x30	3.51	48	178	**178**	0.00	10800	-	-	10800	-	-	10800	179	0.56	3863
BCPC57	10x5x40	3.5	56	245	**245**	0.00	10800	-	-	10800	-	-	10800	246	0.41	2111
BCPC58	10x10x20	3.52	32	57	**57**	0.00	10800	70	22.81	10800	92	61.40	10800	**57**	0.00	225
BCPC59	10x10x30	3.51	47	83	86	3.61	10800	-	-	10800	-	-	10800	**83**	0.00	3418
BCPC60	10x10x40	3.5	55	117	118	0.85	10800	-	-	10800	-	-	10800	**117**	0.00	1841
BCPC61	10x15x20	3.49	34	41	**41**	0.00	10800	48	17.07	10800	55	34.15	10800	**41**	0.00	882
BCPC62	10x15x30	3.51	44	60	**60**	0.00	10800	-	-	10800	118	96.67	10800	**60**	0.00	688
BCPC63	10x15x40	3.48	53	79	**79**	0.00	10800	-	-	10800	-	-	10800	**79**	0.00	2517
BCPC64	20x5x20	3.51	38	235	**235**	0.00	10800	-	-	10800	-	-	10800	236	0.43	4805
BCPC65	20x5x30	3.52	59	351	**351**	0.00	10800	-	-	10800	-	-	10800	357	1.71	5616
BCPC66	20x5x40	3.52	67	482	**482**	0.00	10800	-	-	10800	-	-	10800	485	0.62	6048
BCPC67	20x10x20	3.52	35	113	114	0.88	10800	194	71.68	10800	-	-	10800	**113**	0.00	5292
BCPC68	20x10x30	3.5	49	173	179	3.47	10800	-	-	10800	-	-	10800	**173**	0.00	5128
BCPC69	20x10x40	3.5	60	232	240	3.45	10800	-	-	10800	-	-	10800	**232**	0.00	6372
BCPC70	20x15x20	3.49	36	77	79	2.60	10800	-	-	10800	-	-	10800	**77**	0.00	1228
BCPC71	20x15x30	3.52	50	119	121	1.68	10800	-	-	10800	-	-	10800	**119**	0.00	4860
BCPC72	20x15x40	3.5	65	166	**166**	0.00	10800	-	-	10800	-	-	10800	167	0.60	7452
BCPC73	30x5x20	3.54	38	351	**351**	0.00	10800	-	-	10800	-	-	10800	353	0.57	943
BCPC74	30x5x30	3.53	54	553	559	1.08	10800	-	-	10800	-	-	10800	**553**	0.00	7452
BCPC75	30x5x40	3.54	62	731	**731**	0.00	10800	-	-	10800	-	-	10800	742	1.50	5724
BCPC76	30x10x20	3.51	36	170	171	0.59	10800	-	-	10800	-	-	10800	**170**	0.00	7452
BCPC77	30x10x30	3.52	51	263	272	3.42	10800	-	-	10800	-	-	10800	**263**	0.00	4428
BCPC78	30x10x40	3.51	64	357	365	2.24	10800	-	-	10800	-	-	10800	**357**	0.00	5616
BCPC79	30x15x20	3.52	37	120	**120**	0.00	10800	-	-	10800	-	-	10800	**120**	0.00	4860
BCPC80	30x15x30	3.52	53	178	185	3.93	10800	-	-	10800	-	-	10800	**178**	0.00	7128
BCPC81	30x15x40	3.52	68	246	249	1.22	10800	-	-	10800	-	-	10800	**246**	0.00	6804
Average Deviation (%)						1.08			-			-			0.24[1]	

[1] The reported average deviation is calculated by considering the best run among the ten runs per instance. The average deviation when considering all ten runs per instance is 0.53% and the average deviation when considering only the worst run per instance is 0.82%.

F Generation of new FJSSP instances with arbitrary precedence graphs

This section describes the mechanism used for generating new problem instances for the FJSSP with arbitrary precedence graphs. The generated benchmark set is available on the following address: https://github.com/gkasapidis/CPG-FJSSP. The mechanism followed involves two steps. At the first step, a problem instance is created with no precedence relationships among operations of each job, while at the second step the successor and predecessor sets of every operation i are constructed and therefore the arbitrary precedence graphs per job are generated. At this step the precedence graphs are generated as follows: We start from an empty graph. A value for the upper bound θ on the maximum number of predecessors and successors per operation i is selected. Initially, the start node of the a job u, i_u°, is added into G_u^P. Next, a random number between one and θ is chosen, in order to create the successor arcs. Then, as many operations as the number of successors of i_u° are added into the graph, and the operations are connected with i_u° via the associated arcs. At the next iteration, a random number between one and θ is chosen for the number of successors of the recently added operations, and the procedure continues until all operations $i \in O_u$ are added into the graph. Note that every insertion of an operation i to the graph also defines the sets PJ_i and SJ_i. The goal is to reach a target density δ_{target}.

The target density δ_{target} can be determined assuming a linear function of the maximum density of the problem δ_{max}, i.e, $\delta_{target} = \gamma\delta_{max}$, where γ denotes the slope and it is used to control the density of the precedence graph per job.

Due to the heuristic nature of the above procedure, it is possible that the final graph density deviates from the target value. For this reason, an additional refining procedure follows up that tries to further add or remove arcs in order to bring the final density of the problem closer to the target value.

We considered three different density levels for every problem instance. In particular, we examined three values for γ, i.e., 0.2, 0.5 and 0.8. Our aim was to create instances that differ only in terms of the density of the precedence graphs. This allows us to examine in isolation the impact of the density of the problem. For this purpose, we initially derived the most dense version of each problem instance (i.e., $\gamma = 0.8$) following the procedure described above. On this basis, we progressively removed arcs from the job precedence graphs in order to reduce the density of the problem to the desired value, i.e., $0.5 \cdot \delta_{max}$ and $0.2 \cdot \delta_{max}$.

Overall, we generated two groups of large problem instances namely SCPC and BCPC. The

SCPC data set consists of 32 problem instances. Among them, there are eight baseline FJSSP instances with linear precedence relationships, namely SCPCN01–SCPCN08, with the following specifications: $m \in \{4, 8\}$, $|O_u| = 15$ for every $u \in J$, $l \in \{4, 8\}$, $|M_i| \in \{1, 2, 3\}$ for every $i \in \Omega$, $\theta = 1$, $p_{i,k} \in [1, 10]$ for every $i \in \Omega$ and for every $k \in M_i$. Next, in order to create the FJSSP instances with arbitrary precedence graphs we increased the number of maximum job successors and predecessors, setting $\theta = 3$, and we added the precedence graphs by using three settings of γ described above. The 24 problem instances generated using these settings are named as SCPC01–SCPC24.

The BCPC data set consists of 108 problem instances. Among them, there are 27 baseline FJSSP instances with linear precedence relationships, namely BCPCN01–BCPCN27, with the following specifications: $m \in \{5, 10, 15\}$, $|O_u| \in \{20, 30, 40\}$ for every $u \in J$, $l \in \{10, 20, 30\}$, $|M_i| = \{3, 4\}$ for every $i \in \Omega$, $\theta = 1$, $p_{i,k} \in [1, 10]$ for every $i \in \Omega$ and for every $k \in M_i$. In a manner similar to the generation of the SCPC problem group, a total of 81 problem instances, namely BCPC01–BCPC81, were derived for the different values of γ.