**Table S1**

Positioning this study within the context of previous research

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Studies** | **Domain** | **Worker Type** | | **Worker Skills** | | **Objective Specifications** | | | **Solution Approaches** |
| **HT** | **SS** | | **MS** | **SO** | **MO** | **OF** |
| Li and Womer (2009) | - | ü |  | | ü | ü |  | TLC | MILP/CP-BDA |
| Heimerl and Kolisch (2010a) | - | ü |  | | ü | ü |  | TC | IPOPT |
| Heimerl and Kolisch (2010b) | - | ü |  | | ü | ü |  | TLC | MILP |
| Wongwai and Malaikrisanachalee (2011) | - | - | ü | | ü | ü |  |  | HA |
| Firat and Hurkens (2012) | TI | - |  | | ü | ü |  |  | MIP |
| Yannibelli and Amandi (2013) | - | ü |  | | ü |  | ü | EHR and | EA |
| Barz and Kolisch (2014) | TI | ü |  | | ü | ü |  | PC and WC | MDP |
| Montoya et al. (2014) | - | - |  | | ü | ü |  |  | B&P |
| Attia et al. (2014) | - | ü |  | | ü | ü |  | TC | GA |
| Correia and Da-Gama (2014) | - | ü |  | | ü | ü |  | TC | MPM |
| Myszkowski et al. (2015) | - | ü |  | | ü | ü |  | TC and | HACO |
| Kia et al. (2016) | - | ü |  | | ü |  | ü | , IC and PC | CPLEX |
| Maghsoudlou et al. (2016) | - | ü |  | | ü |  | ü | TC, and TQ | MOIWO |
| Almeida et al. (2016) | - | - |  | | ü | ü |  |  | PBH |
| Zheng et al. (2017) | - | ü |  | | ü | ü |  |  | TLBOA |
| Javanmard et al. (2017) | IP | ü |  | | ü |  |  | TC | GAMS solver |
| Maghsoudlou et al. (2017) | - | ü |  | | ü |  | ü | TC and RRA | MM |
| Chen et al. (2017) | IT | ü |  | | ü |  | ü | TE, DP and TES | NSGA-II |
| Han et al. (2017) | SI | - | - | | - | ü |  | OSPS | 3L-PCS |
| Myszkowski et al. (2018) | - | ü |  | | ü | ü |  |  | DEGR |
| Dai et al. (2018) | - | ü |  | | ü | ü |  | TC and | ITS |
| Wang and Zheng (2018) | - | ü |  | | ü |  | ü | TC and | MOFFOA |
| Zhu et al. (2019) | - | ü |  | | ü | ü |  |  | DOMVO |
| Najafzad et al. (2019) | ET | ü |  | | ü |  | ü | TC and | DMT |
| Zhu et al. (2019) | - |  | ü | | ü |  | ü | TC and | NSGA-II |
| Joshi et al. (2019) | - | ü |  | | ü | ü |  |  | TLBOA |
| Felberbauer et al. (2019) | - | ü |  | | ü | ü |  | EEC | MM and SSA |
| Zabihi et al. (2019) | - | ü |  | | ü |  | ü | , TE and CS | TLBOA |
| Ahmadpour and Ghezavati (2019) | - | ü |  | | ü | ü |  |  | CPLEX |
| Tian et al. (2020) | - | ü |  | | ü |  | ü | TC and | MOFFOA |
| Lin et al. (2020) | - | ü |  | | ü | ü |  |  | GP-HH |
| Chen et al. (2020) | IT | ü |  | | ü |  | ü | EL, DC and QDP | ACA |
| Mao et al. (2020) | SI | - | - | | - | ü |  | OTAC | ICA |
| Li et al. (2020) | - | ü |  | | ü |  | ü | PE and SD | OS |
| Jeunet and Bou Orm (2020) | - | - | ü | |  |  | ü | TC, and QL | MILP |
| Almatroushi et al. (2020) | - | - | ü | |  | ü |  | TC | OP |
| Zhu et al. (2021) | - | ü |  | | ü |  | ü | TC and | MOGP-HH/D |
| Maghsoudlou et al. (2021) | ET | ü |  | | ü | ü |  | TC | EMA and GA |
| Ghamginzadeh et al. (2021) | - | - |  | | ü |  | ü | and TLC | MOICA |
| Snauwaert and Vanhoucke (2021) | - | ü | ü | | ü | ü |  |  | Hybrid GA |
| Barghi and Sikari (2022) | - | ü |  | | ü | ü |  |  | GA and SA |
| Javanmard et al. (2022) | EE | ü |  | | ü |  | ü | TC and PPA | GAMS solver |
| Nikaeen and Najafi (2022) | - | ü |  | | ü | ü |  |  | CP |
| Akbar et al. (2022) | SP | ü |  | | ü | ü |  |  | PA |
| Wang et al. (2022) | - | ü |  | | ü | ü |  | TC | GA-PR |
| Polo-Mejía et al. (2023) | - | ü | ü | | ü | ü |  |  | GRASP and LNS |
| Ma et al. (2023) | - | ü | ü | | ü | ü |  | SR | TS |
| Li et al. (2023) | DTP | ü |  | | ü | ü |  |  | EIAIS |
| (Ghasemi et al., 2023) | - | ü |  | | ü | ü |  |  | TFCP |
| Goudarzi et al. (2023) | - | ü |  | | ü |  | ü | and TDRU | KM, FCM and BSOS |
| Han et al. (2024) | PBU | ü |  | | ü | ü |  | TC | HWOA |
| Chen et al. (2024) | R&D | ü |  | | ü |  | ü | SE, TD and RPE | NSGA-II |
| Torba et al. (2024) | HMF | ü |  | | ü |  | ü | TT and | MA and SGA |
| Jeunet (2024) | - | ü | ü | |  | ü |  | TLC | MILP |
| You et al. (2024) | - | ü |  | | ü | ü |  | TTC | BA |
| Luo et al. (2024) | - | ü |  | | ü |  | ü | TC and | FFOA |
| Goudarzi et al., 2024) | EC | ü |  | | ü |  | ü | and TEC | VDO |
| (Akbar et al. (2024) | - | ü |  | | ü | ü |  |  | GPSA |
| Zarei et al. (2024) | RB | ü |  | | ü |  | ü | , PC and QPT | IVF-ABS |
| Mozhdehi et al. (2024) | - | ü |  | | ü | ü |  |  | LSM, TS and MBBO |
| Polancos and Seva (2024) | - | ü |  | | ü | ü |  | APR | OM and B&C |
| **Current study** | **SI-SSD** | **ü** | **ü** | | **ü** |  | **ü** | **TLC+WIPC, and WI** | **NSGA-II-based algorithms** |

Note: HM: homogeneous / HT: heterogeneous / SS: single-skilled / MS: multi-skilled / SO: single-objective / MO: multi-objective / OF: objective function / PL: proficiency level / TC: total cost / : makespan / WI: workload imbalance / APR: average project risk / OM: optimization model / B&C: branch-and-cut / LSM: local search method / TS: tabu search / MBBO: Modified discrete variant of the Biogeography-Based Optimization / RB: railway bridge / IVF-ABS: interval valued-approximate best solution / DTP: digital transformation project / EIAIS: efficient immunoglobulin-based artificial immune system / EC: energy consumption / TEC: total energy consumption / VDO: vibration damping optimization / TFCP: type-2 FCP / TTC: total tardiness cost / BA: bat algorithm / TT: total tardiness / MA: memetic algorithm / SGA: simulated GA / HMF: heavy maintenance factory / RPE: R&D processes and efficiency / KM: K-means / FCM: fuzzy C-means / BSOS: bi-objective symbiosis organisms search / TDRU: total deviation of resource utilization / GA-PR: genetic algorithm with priority rules / PA: proposed algorithm / GRASP: greedy randomized adaptive search procedure / LNS: large neighborhood search / PPA: priority-progress alignment / SA: simulated annealing / TLC: total labor cost / ET: energy tariff / EMA: electromagnetic like algorithm / OP: optimization package / PE: project effectiveness / OS: optimization solver / IT: information technology / EL: extent of learning / DC: development cycle / QDP: quality of the developed product / ACA: ant colony algorithm / SI: shipbuilding industry / GP-HH: genetic programming hyper- heuristic / MOFFOA: multi-objective FFOA / TE: total efficiency / CS: combined salaries / EEC: expected external cost / MM: metaheuristic method / SSA: sample average approximation / DMT: decision making techniques / TQ: total quality / MOIWO: multi-objective IWO / IC: idle cost / PC: penalty cost / MPM: mathematical programming model / TI: telecommunications industry / WC: waiting cost / MIP: mixed integer programming / SR: schedule robustness / QL: quality losses / HA: heuristic approach / IPOPT: interior point optimizer / OSPS: optimize shipbuilding project scheduling / OTAC: optimizing task allocation and collaboration processes / WIPC: work-in-progress cost

**Table S2**

The *D1R* metric values of the proposed algorithms for all |L| values

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | **| L |=5** | | | | **| L |=7** | | | | **| L |=9** | | | |
| ***|I|*** | ***|S|*** | ***|W|*** | **A1** | **A2** | **A3** | **A4** | **A1** | **A2** | **A3** | **A4** | **A1** | **A2** | **A3** | **A4** |
| 12 | 5 | 30 | 0,37 | 0,10 | 0,20 | 0,17 | 0,32 | 0,24 | 0,28 | 0,22 | 0,31 | 0,10 | 0,15 | 0,45 |
| 12 | 5 | 40 | 0,33 | 0,12 | 0,12 | 0,10 | 0,48 | 0,42 | 0,25 | 0,74 | 0,68 | 0,06 | 0,08 | 0,10 |
| 12 | 5 | 50 | 0,32 | 0,17 | 0,11 | 0,19 | 0,68 | 0,32 | 0,33 | 0,46 | 0,45 | 0,21 | 0,21 | 0,29 |
| 12 | 10 | 30 | 0,44 | 0,09 | 0,14 | 0,06 | 0,37 | 0,10 | 0,29 | 0,17 | 0,96 | 0,06 | 0,35 | 0,10 |
| 12 | 10 | 40 | 0,35 | 0,12 | 0,19 | 0,10 | 0,40 | 0,11 | 0,13 | 0,11 | 0,38 | 0,15 | 0,21 | 0,24 |
| 12 | 10 | 50 | 0,43 | 0,18 | 0,21 | 0,21 | 1,00 | 0,02 | 0,10 | 0,02 | 0,88 | 0,16 | 0,20 | 0,11 |
| 12 | 15 | 30 | 0,33 | 0,19 | 0,34 | 0,24 | 0,79 | 0,08 | 0,09 | 0,04 | 1,16 | 0,10 | 0,38 | 0,07 |
| 12 | 15 | 40 | 0,44 | 0,15 | 0,27 | 0,26 | 0,33 | 0,22 | 0,47 | 0,44 | 1,61 | 0,25 | 0,66 | 0,08 |
| 12 | 15 | 50 | 1,24 | 0,12 | 0,22 | 0,05 | 0,44 | 0,18 | 0,30 | 0,25 | 0,39 | 0,10 | 0,13 | 0,14 |
| 16 | 5 | 30 | 0,49 | 0,17 | 0,18 | 0,28 | 0,41 | 0,13 | 0,47 | 0,28 | 0,46 | 0,21 | 0,40 | 0,37 |
| 16 | 5 | 40 | 0,37 | 0,11 | 0,18 | 0,16 | 0,54 | 0,16 | 0,22 | 0,36 | 0,39 | 0,13 | 0,19 | 0,14 |
| 16 | 5 | 50 | 0,27 | 0,09 | 0,12 | 0,15 | 0,45 | 0,09 | 0,13 | 0,16 | 0,37 | 0,20 | 0,22 | 0,15 |
| 16 | 10 | 30 | 0,42 | 0,23 | 0,12 | 0,40 | 0,60 | 0,29 | 0,15 | 0,22 | 0,46 | 0,11 | 0,33 | 0,10 |
| 16 | 10 | 40 | 0,82 | 0,16 | 0,17 | 0,06 | 0,31 | 0,05 | 0,11 | 0,08 | 0,46 | 0,14 | 0,23 | 0,17 |
| 16 | 10 | 50 | 0,27 | 0,07 | 0,23 | 0,22 | 0,38 | 0,14 | 0,17 | 0,33 | 0,28 | 0,07 | 0,24 | 0,10 |
| 16 | 15 | 30 | 0,37 | 0,07 | 0,22 | 0,26 | 0,35 | 0,15 | 0,19 | 0,15 | 0,28 | 0,10 | 0,49 | 0,41 |
| 16 | 15 | 40 | 0,52 | 0,06 | 0,11 | 0,11 | 0,45 | 0,32 | 0,43 | 0,54 | 0,30 | 0,13 | 0,44 | 0,36 |
| 16 | 15 | 50 | 0,37 | 0,07 | 0,20 | 0,28 | 0,46 | 0,26 | 0,21 | 0,24 | 0,49 | 0,23 | 0,35 | 0,37 |
| 20 | 5 | 30 | 0,25 | 0,05 | 0,13 | 0,14 | 0,89 | 0,12 | 0,16 | 0,14 | 0,37 | 0,13 | 0,24 | 0,22 |
| 20 | 5 | 40 | 0,43 | 0,41 | 0,22 | 0,32 | 0,34 | 0,14 | 0,46 | 0,35 | 0,28 | 0,07 | 0,20 | 0,21 |
| 20 | 5 | 50 | 0,56 | 0,49 | 0,37 | 0,63 | 0,57 | 0,06 | 0,12 | 0,08 | 0,69 | 0,23 | 0,28 | 0,44 |
| 20 | 10 | 30 | 0,35 | 0,08 | 0,20 | 0,24 | 0,36 | 0,08 | 0,19 | 0,12 | 0,72 | 0,10 | 0,05 | 0,02 |
| 20 | 10 | 40 | 0,40 | 0,13 | 0,13 | 0,17 | 0,72 | 0,16 | 0,23 | 0,39 | 0,43 | 0,10 | 0,17 | 0,19 |
| 20 | 10 | 50 | 0,55 | 0,20 | 0,42 | 0,58 | 0,56 | 0,03 | 0,05 | 0,02 | 0,76 | 0,05 | 0,12 | 0,08 |
| 20 | 15 | 30 | 0,24 | 0,15 | 0,19 | 0,20 | 0,45 | 0,24 | 0,27 | 0,30 | 0,34 | 0,25 | 0,29 | 0,50 |
| 20 | 15 | 40 | 0,39 | 0,10 | 0,28 | 0,19 | 0,95 | 0,11 | 0,08 | 0,01 | 1,19 | 0,33 | 0,33 | 0,04 |
| 20 | 15 | 50 | 0,58 | 0,08 | 0,11 | 0,17 | 0,44 | 0,07 | 0,31 | 0,14 | 0,56 | 0,22 | 0,27 | 0,39 |
| **Average** | | | **0,44** | **0,15** | **0,20** | **0,22** | **0,52** | **0,16** | **0,23** | **0,23** | **0,58** | **0,15** | **0,27** | **0,22** |

**Table S3**

The *IGD* metric values of the proposed algorithms for all |L|

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | **| L |=5** | | | | **| L |=7** | | | | **| L |=9** | | | |
| ***|I|*** | ***|S|*** | ***|W|*** | **A1** | **A2** | **A3** | **A4** | **A1** | **A2** | **A3** | **A4** | **A1** | **A2** | **A3** | **A4** |
| 12 | 5 | 30 | 1024,87 | 259,01 | 478,89 | 272,02 | 1021,17 | 263,93 | 412,94 | 239,13 | 922,19 | 249,64 | 383,67 | 244,22 |
| 12 | 5 | 40 | 1379,14 | 350,19 | 616,41 | 334,57 | 1281,56 | 319,22 | 589,89 | 276,31 | 1349,53 | 324,53 | 542,52 | 362,06 |
| 12 | 5 | 50 | 1549,32 | 424,60 | 735,88 | 378,87 | 1626,78 | 422,92 | 758,20 | 403,53 | 1660,82 | 452,16 | 780,88 | 474,12 |
| 12 | 10 | 30 | 934,35 | 252,30 | 372,99 | 214,92 | 1000,22 | 270,53 | 405,55 | 236,93 | 995,37 | 285,24 | 370,60 | 268,97 |
| 12 | 10 | 40 | 1240,34 | 368,88 | 484,24 | 281,78 | 1219,76 | 347,89 | 480,61 | 321,09 | 1294,46 | 300,20 | 504,14 | 327,75 |
| 12 | 10 | 50 | 1491,28 | 442,56 | 615,92 | 358,68 | 1430,21 | 364,71 | 629,91 | 349,90 | 1439,03 | 377,55 | 697,68 | 391,44 |
| 12 | 15 | 30 | 931,90 | 271,67 | 325,25 | 229,16 | 891,07 | 312,89 | 327,94 | 223,21 | 935,51 | 282,62 | 332,92 | 271,95 |
| 12 | 15 | 40 | 1205,82 | 392,28 | 510,65 | 335,60 | 1258,76 | 382,03 | 523,55 | 329,76 | 1286,46 | 381,29 | 534,50 | 345,34 |
| 12 | 15 | 50 | 1507,62 | 447,39 | 631,67 | 402,80 | 1419,02 | 417,09 | 616,49 | 359,39 | 1492,16 | 453,69 | 665,87 | 395,76 |
| 16 | 5 | 30 | 1095,25 | 297,85 | 403,42 | 268,67 | 1057,86 | 265,72 | 432,19 | 270,31 | 1104,09 | 317,11 | 460,49 | 261,13 |
| 16 | 5 | 40 | 1338,58 | 364,76 | 575,61 | 335,57 | 1376,19 | 390,18 | 602,67 | 372,42 | 1379,02 | 362,79 | 624,74 | 338,34 |
| 16 | 5 | 50 | 1667,85 | 417,98 | 755,86 | 454,09 | 1613,34 | 420,99 | 702,81 | 394,81 | 1622,03 | 444,53 | 745,40 | 361,22 |
| 16 | 10 | 30 | 950,88 | 277,85 | 328,53 | 240,26 | 1125,33 | 353,52 | 420,73 | 285,71 | 1121,39 | 327,44 | 396,15 | 295,80 |
| 16 | 10 | 40 | 1314,99 | 347,47 | 524,96 | 319,20 | 1413,61 | 403,02 | 532,70 | 376,79 | 1365,43 | 374,59 | 486,65 | 320,65 |
| 16 | 10 | 50 | 1737,43 | 411,53 | 687,06 | 438,13 | 1535,64 | 415,45 | 613,25 | 416,72 | 1790,01 | 464,75 | 742,14 | 485,38 |
| 16 | 15 | 30 | 1039,20 | 306,93 | 376,43 | 307,61 | 1061,23 | 391,43 | 373,16 | 289,47 | 988,78 | 343,00 | 348,53 | 270,44 |
| 16 | 15 | 40 | 1368,38 | 430,60 | 495,36 | 367,78 | 1246,46 | 381,39 | 453,39 | 317,73 | 1265,92 | 348,71 | 480,82 | 336,59 |
| 16 | 15 | 50 | 1595,26 | 428,46 | 647,23 | 401,17 | 1659,35 | 525,48 | 627,40 | 424,04 | 1633,00 | 481,21 | 642,03 | 437,32 |
| 20 | 5 | 30 | 1210,24 | 343,55 | 425,25 | 297,08 | 1253,76 | 364,15 | 439,18 | 354,50 | 1160,06 | 348,23 | 431,38 | 325,39 |
| 20 | 5 | 40 | 1447,48 | 397,19 | 581,46 | 373,77 | 1490,15 | 378,99 | 594,12 | 386,10 | 1507,93 | 410,01 | 573,11 | 391,58 |
| 20 | 5 | 50 | 1846,67 | 509,99 | 734,01 | 523,43 | 1751,24 | 450,31 | 777,09 | 494,65 | 1741,23 | 443,95 | 778,89 | 445,95 |
| 20 | 10 | 30 | 1109,05 | 390,54 | 417,80 | 294,91 | 1171,84 | 383,85 | 435,20 | 289,11 | 1160,07 | 397,41 | 437,66 | 314,07 |
| 20 | 10 | 40 | 1464,23 | 410,74 | 570,66 | 385,49 | 1458,75 | 411,16 | 526,45 | 409,68 | 1538,17 | 450,57 | 562,15 | 415,72 |
| 20 | 10 | 50 | 1778,09 | 475,12 | 698,26 | 511,32 | 1770,42 | 512,91 | 744,28 | 470,62 | 1874,06 | 485,54 | 700,84 | 484,77 |
| 20 | 15 | 30 | 1130,66 | 445,82 | 412,23 | 294,16 | 1111,67 | 421,17 | 431,18 | 319,69 | 1097,26 | 421,66 | 419,46 | 295,38 |
| 20 | 15 | 40 | 1414,96 | 448,42 | 545,82 | 394,88 | 1445,46 | 498,90 | 552,37 | 408,41 | 1314,50 | 408,53 | 476,03 | 303,65 |
| 20 | 15 | 50 | 1655,10 | 460,71 | 654,69 | 471,42 | 1682,48 | 449,24 | 645,03 | 413,21 | 1713,64 | 505,97 | 628,27 | 470,66 |
| **Average** | | | **1349,22** | **384,24** | **540,98** | **351,38** | **1347,16** | **389,60** | **542,53** | **349,38** | **1361,19** | **386,78** | **546,20** | **356,88** |

**Table S4**

The *MID* metric values of the proposed algorithms for all |L|

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | **| L |=5** | | | | **| L |=7** | | | | **| L |=9** | | | |
| ***|I|*** | ***|S|*** | ***|W|*** | **A1** | **A2** | **A3** | **A4** | **A1** | **A2** | **A3** | **A4** | **A1** | **A2** | **A3** | **A4** |
| 12 | 5 | 30 | 10073 | 6251 | 7012 | 5600 | 10073 | 6091 | 6936 | 5567 | 9862 | 6318 | 7238 | 5545 |
| 12 | 5 | 40 | 12978 | 7496 | 9157 | 6624 | 12925 | 7795 | 9247 | 7019 | 13235 | 8265 | 9430 | 7221 |
| 12 | 5 | 50 | 15455 | 8494 | 11207 | 8296 | 16288 | 9665 | 11461 | 8887 | 16624 | 9875 | 11704 | 8719 |
| 12 | 10 | 30 | 9715 | 6432 | 7106 | 5576 | 10318 | 7215 | 7340 | 6168 | 10324 | 7026 | 6996 | 5948 |
| 12 | 10 | 40 | 12687 | 8146 | 8379 | 7295 | 12912 | 8318 | 9219 | 7493 | 13541 | 9105 | 9336 | 7804 |
| 12 | 10 | 50 | 15312 | 9281 | 11361 | 8912 | 15417 | 9914 | 11314 | 9074 | 16036 | 10421 | 12016 | 9628 |
| 12 | 15 | 30 | 10175 | 7225 | 7374 | 6040 | 10156 | 7715 | 7345 | 6319 | 10268 | 7659 | 7208 | 6169 |
| 12 | 15 | 40 | 13035 | 8984 | 8994 | 7401 | 13251 | 8978 | 9017 | 7407 | 13749 | 9458 | 9561 | 7957 |
| 12 | 15 | 50 | 15904 | 10379 | 11312 | 9229 | 15630 | 10411 | 11848 | 9848 | 16300 | 10829 | 11965 | 10077 |
| 16 | 5 | 30 | 10963 | 6959 | 7473 | 5924 | 10755 | 6829 | 7014 | 5802 | 11181 | 7372 | 7844 | 6022 |
| 16 | 5 | 40 | 13341 | 8075 | 9106 | 6990 | 13899 | 8830 | 9378 | 7206 | 13727 | 8401 | 9510 | 7537 |
| 16 | 5 | 50 | 16733 | 9781 | 11833 | 8663 | 16314 | 9473 | 11877 | 8805 | 16467 | 9810 | 12126 | 9042 |
| 16 | 10 | 30 | 10330 | 7177 | 7535 | 5673 | 11041 | 7443 | 7019 | 5668 | 11217 | 7946 | 7229 | 5824 |
| 16 | 10 | 40 | 13387 | 8683 | 9216 | 7639 | 14091 | 9439 | 9022 | 7634 | 14166 | 9300 | 9644 | 7815 |
| 16 | 10 | 50 | 17382 | 10788 | 11126 | 9195 | 15920 | 9863 | 11515 | 8756 | 17657 | 11323 | 11983 | 9581 |
| 16 | 15 | 30 | 10760 | 7862 | 7277 | 5575 | 11113 | 7922 | 7229 | 5834 | 10962 | 8217 | 7425 | 6220 |
| 16 | 15 | 40 | 14110 | 9568 | 8829 | 7411 | 13559 | 9316 | 8956 | 7606 | 13836 | 9898 | 9843 | 8066 |
| 16 | 15 | 50 | 16454 | 10585 | 10901 | 9034 | 17380 | 11462 | 11701 | 9505 | 17230 | 11595 | 11494 | 9515 |
| 20 | 5 | 30 | 11873 | 7644 | 7816 | 6041 | 11971 | 7712 | 7462 | 5407 | 11860 | 8018 | 7509 | 5984 |
| 20 | 5 | 40 | 14545 | 9012 | 9563 | 7334 | 15029 | 10101 | 9417 | 7774 | 14761 | 9008 | 9748 | 7430 |
| 20 | 5 | 50 | 18119 | 11037 | 11299 | 8602 | 17350 | 10565 | 11441 | 8384 | 18016 | 11056 | 12252 | 9513 |
| 20 | 10 | 30 | 11282 | 7989 | 6788 | 5406 | 11696 | 8012 | 6866 | 5920 | 11949 | 8334 | 7383 | 6012 |
| 20 | 10 | 40 | 14756 | 9732 | 9366 | 7478 | 14621 | 9894 | 9220 | 6951 | 15531 | 10689 | 9869 | 7732 |
| 20 | 10 | 50 | 18224 | 11737 | 11487 | 9151 | 17810 | 10980 | 11567 | 9288 | 18409 | 11858 | 11399 | 9263 |
| 20 | 15 | 30 | 11748 | 8416 | 6966 | 5676 | 11766 | 8705 | 6845 | 6046 | 11692 | 8766 | 6968 | 6293 |
| 20 | 15 | 40 | 14593 | 10233 | 9034 | 7189 | 15076 | 10265 | 9393 | 7748 | 14183 | 10201 | 9287 | 8272 |
| 20 | 15 | 50 | 17279 | 11287 | 11905 | 8984 | 17734 | 12188 | 11872 | 9501 | 17589 | 11340 | 11953 | 9024 |
| **Average** | | | **13749** | **8861** | **9238** | **7294** | **13855** | **9078** | **9316** | **7467** | **14088** | **9337** | **9590** | **7712** |

|  |  |
| --- | --- |
| **Listing 1:** Decoding procedure for chromosomes | |
| **1** | **inputs:** initial population, number of workstations (||), number of workers (||), number of skills (||), population size (), precedence relations of the workstations, skill sets for workers, total skill number for workstations, skill number for workstations, skill matrix for workers, task time of each skill, unit cost of single and multi-skilled worker ( and ), work-in-progress cost for single and multi-skilled workers at workstation ( and ) |
| **2** | **output:** objective values of total cost, makespan and workload imbalance |
| **3** | **Begin** |
| **4** | **while** all fitness values remain to be calculated **do** |
| **5** | Assign workers to each chromosome |
| **6** | Evaluate the labor cost for each chromosome based on the assigned workers |
| **7** | Calculate the total cost |
| **8** | Calculate the makespan |
| **9** | Calculate the workload imbalance |
| **10** | **end while** |
| **11** | return all fitness values for population |
| **12** | **end** |

|  |  |
| --- | --- |
| **Listing 2:** Initialization process for NSGA-II-based algorithms | |
| **1** | **inputs:** number of workstations (||), population size (), precedence relations of the workstations, skill sets for workers, total skill number for workstations, skill matrix for workers |
| **2** | **output:** initial population |
| **3** | **Begin** |
| **4** | **while** population size is not reached **do** |
| **5** | Create encoding population as a random strategy |
| **6** | Assign workers to workstations according to their skill requirements |
| **7** | **end while** |
| **8** | return initial population |
| **9** | **end** |

|  |  |
| --- | --- |
| **Listing 3:** Pseudo-code for one-point crossover | |
| **1** | **inputs:** number of the workstations (||), population size (), set of the chromosome pairs, precedence and successor relations matrix, probability of crossover () |
| **2** | **output:** crossover population |
| **3** | **Begin** |
| **4** | **while** the number of offspring is less than the population size () **do** |
| **5** | **if** rand () <  **then** |
| **6** | Randomly select a crossover point () between 1 and || |
| **7** | Extract part-A from parent-A (tasks and workers before ) for offspring-A |
| **8** | Extract part-B from parent-B (tasks and workers after ) for offspring-A |
| **9** | Combine part-A and part-B to form offspring-A |
| **10** | Check for duplicate workstations in part-B of offspring-A |
| **11** | **if** part-A and part-B share common workstations **then** |
| **12** | Assign workers of common workstations in part-B to their counterparts in part-A |
| **13** | Set tasks in part-B's common workstations to zero |
| **14** | **end if** |
| **15** | Identify kk-A (workstations missing in offspring-A) |
| **16** | Assign a dummy kk-A as the first workstation in kk-A |
| **17** | Check predecessor and successor relations for the dummy workstation |
| **18** | Reassign the workstation based on its predecessor and successor sets |
| **19** | Repeat until all workstations in kk-A are reassigned |
| **20** | Apply the same steps in reverse for offspring-B |
| **21** | Validate workers in offspring-A and offspring-B |
| **22** | **end if** |
| **23** | **end while** |
| **24** | Transfer the generated crossover population to the next step |
| **25** | **end** |

|  |  |
| --- | --- |
| **Listing 4:** Pseudo-code for mutation operator | |
| **1** | **inputs:** probability of mutation (), population size (), crossover population, skill matrix for workers, total skill number for workstations |
| **2** | **output:** mutation population |
| **3** | **Begin** |
| **4** | **while** the number of mutant chromosomes is less than the population size () **do** |
| **5** | **if** rand () <  **then** |
| **6** | Select one chromosome randomly from non-elite chromosomes |
| **7** | **Repeat** |
| **8** | Generate two random indices, and , such that > |
| **9** | Assign the workstation at as x and the workstation at as y |
| **10** | **Until** workstations x and y share at least one common skill type |
| **11** | Identify skill types and the number of workers required for workstations x and y |
| **12** | Identify common skill types between x and y |
| **13** | **For** each common skill type **do** |
| **14** | Identify workers assigned to x and y for the skill |
| **15** | Randomly select one worker from x and one from y |
| **16** | **if** the selected worker from x is not assigned to y and the selected worker from y is not assigned to x **then** |
| **17** | Swap the positions of the selected workers between workstations x and y |
| **18** | **end if** |
| **19** | **end for** |
| **20** | **end if** |
| **21** | **end while** |
| **22** | Transfer the generated mutation population to the next step |
| **23** | **end** |