Iranian Journal of Numerical Analysis and Optimization
Vol. 10, No. 1, (2020), pp 139–155
DOI:10.22067/ijnao.v10i1.73001

**Research Article**

# Moving to higher directional scheduling schemes

H.R. Yousefzadeh* and M. Nasehi

**Abstract**

A general overview of the scheduling's literature of some researches shows that among various factors, the priority rules and also the structure of projects are two main factors that can be affected on the performance of multidirectional scheduling schemes. In addition, a variation on the number of directors in scheduling schemes (e.g., single directional, bi-directional, and tri-directional scheduling scheme) produces different makespans. However, the question of when to move from the single direction to the multidirectional scheduling schemes remained open. In this paper, based on analytical and also empirical results, we show that when availability and distributions of resource measures such as the number of peaks and number of overflows or the average amount of overflows are increased, higher directional scheduling schemes can be produced smaller makespans. Hence, in the light of these resource measures, the multidirectional scheduling schemes can be reduced the dependency of the solution's quality on the priority rules used.

**AMS(2010):** Primary 90B35; Secondary 68M20.

**Keywords:** Scheduling; Priority rule; Heuristic algorithms; Multi-directional scheduling schemes; Resource measure.

## 1 Background

The resource-constrained project scheduling problem that is called by RCPSP is known as an NP-hard[1] optimization problem, that is, there is no algorithm

*Corresponding author
Received 1 October 2019; revised 12 February 2020; accepted 14 February 2020
Hamid Reza Yousefzadeh
Department of Mathematics, Payame Noor University (PNU), Iran. e-mail: usefzadeh.math@pnu.ac.ir

Maryam Nasehi
Department of Mathematics, Payame Noor University (PNU), Iran. e-mail: maryam5787@gmail.com

[1] Non-deterministic polynomial-time hard

for solving the RCPSP to optimally in polynomial time; see [20]. Various versions of the RCPSP can be studied by defining different types of precedence constraints between activities such as minimum and/or maximum time's lags, allowing or not the preemption between activities, and time-varying the resource availability.

Exact methods and heuristic (meta-heuristic) methods are generally two approaches to solve the RCPSPs. Mathematical modeling (e.g., [11]), dynamic programming (e.g., [23]), and branch and bound approach (e.g., [5]) are the base of exact methods. These methods solve the instances up to 120 activities (see, for example, [30, 31] for more details).

Among the heuristic (meta-heuristic) methods for solving real-world instances of the RCPCPs, we can refer to artificial immune system (e.g., [1]), simulated annealing (e.g., [2, 21, 26]), genetic algorithms (e.g., [10, 9, 24]), tabu search (e.g., [25]) and miscellaneous approaches (e.g., [15, 30, 10, 13, 27]). These kinds of methods do not guarantee to obtain the optimal feasible scheduling. Scheduling projects correspond to assign a starting and finishing time to each activity with satisfying both precedence and resource constraints (feasible solution). The serial scheduling scheme (SSS) and the parallel scheduling scheme (PSS), which both generate feasible solutions were proposed by Kolisch [15]. It is worth noting that the priority rules can be considered as the main part of heuristics/meta-heuristic methods and they have considerable effects on the quality of solutions (less makespan) that are generated by scheduling schemes (see, for example, [15, 4]).

In light of improving the performance of scheduling schemes, in 2000, Klein [13] suggested the bi-directional scheduling scheme, namely bidss, to generate feasible schedules better than those solutions that are generated by single directional scheduling schemes (i.e., the forward and the backward scheduling scheme). Depending on the priority rules used and the problem's structure, Yoosefzadeh, Tareghian, and Farahi [29] proposed the tri-directional scheduling scheme (trdss) to produce feasible solutions with fewer makespans.

Resource utilization for activities such as {RF[2], RS[3]}, the ratio of average slack per activity to the length of the critical path, and also project's complexity ($C$ as used in [3]), which are reported in [8], are considered as three crucial characteristics in the RCPSP's structure.

## 2 Discussion

In this paper, the RCPSP is described as follows: The project is depicted as an activity on node network with $N$ activities (the start node 1 and end

[2] Resource Factor-RF

[3] Resource Strength-RS

node $N$ of the project are dummy nodes). Duration (processing time) for activity $j$ is shown by notation $d_j$. There is no preemption between activities and finish to start logical control is defined for the precedence constraints between activities. Activity $j$ needs $u_{jr}$ $(j = 1, \ldots, N;\ r = 1, \ldots, |R|)$ units of resource $r$, where $u_{1r} = u_{Nr} = 0$ $(r = 1, \ldots, |R|)$. The renewable resource's availability of type $r$ is denoted by $a_r$, and it is constant per unit of time. Various objective functions have been considered in the literature as the scheduling objective(s) (to study more, we refer the reader, e.g., to [7, 22, 32]). In this paper, minimizing the project's makespan is considered as the main objective function.

Unfortunately, in the context of project scheduling, as the number and/or availability of resources, the number of activities, or other characteristics of scheduling projects vary, some misleading results can be obtained [15]. Many types of research are focused on the performance of priority rules. Their results showed that the priority rules depend on eligible activities, which are chosen in each step of the scheduling scheme (see [19, 18]). In general, some time-based and resource-based measures are affected by the performance of successful priority rules (see [8]). In other words, there exists a significant relationship between the priority rules and the number of resource's types; see [12]. In addition, in [28], Ulusoy and Özdamar reported that the percentage of critical activities, resource measures, network complexity, utilization factor, and obstruction value are the crucial factors in finding upon best priority rules.

We try, in this paper, to answer this question, "under what resource characteristics of the RCPSPs the multidirectional scheduling schemes outperform (generates a schedule with less makespan) the other multidirectional scheduling schemes with lower direction?"

In general, the structure of the paper is organized as follows: the outline of the bidss and the trdss are described in brief in Section 3. The experimental results are described in Section 4. Moreover, in this section, some of the conditions under which multidirectional scheduling schemes with higher direction can generate better solutions in quality are cited, and the corresponding results about the effect of the characteristics of the RCPCPs on the quality of the solution are described. Finally, some brief conclusions are given in Section 5.

## 3 Multidirectional scheduling schemes

In this section, the outline of the multidirectional scheduling scheme such as the trdss and the bidss is briefly explained. It is worth noting that, the parallel scheduling scheme [15] that is denoted by the PSS is implemented in both of the bidss and the trdss. Moreover, in our investigation, we have called all of the priority rules in [29], which they used.

### 3.1 Bidirectional scheduling scheme

In [13], Klein tried to improve the solutions' quality in comparison to the forward and backward scheduling schemes (we called them by single-direction scheduling schemes-sidss) by proposing the bidss. Two main phases of the bidss can be stated as follows: the first is the initialization phase and the second one is an interlinking phase. All of the activities are scheduled in the forward and backward directions in the initialization phase. In other words, at any decision time, the activities that all their predecessors are scheduled and completed in the left direction are eligible to schedule as the forward direction. On the other hand, the activities that all their successors are scheduled in the right direction are eligible to schedule as the backward direction. A tie-breaking rule is called for all activities that are eligible to schedule both directions. In the second phase (i.e., interlinking phase), the scheduled activities in the backward direction are shifted to the left to interlink the activities in both forward and backward directions. Hence a complete feasible schedule is obtained. Scheduled activities in the backward direction are shifted to left by considering both precedent and resource constraints (we refer the reader to [13] for more details).

### 3.2 Tri-directional scheduling scheme

Main phases of the trdss are the same as the bidss, but in the initialization phase all activities are planned and scheduled in three directions, namely, forward, midway, and backward directions (see [29]). Like the bidss, some of the activities are scheduled in the forward direction, and some of them are scheduled in the backward direction. However, activities that are eligible to schedule in both directions are now eligible to schedule in the midway direction (see [29] for more details).

## 4 Analysis of results

In this section, the experimental results due to the performance of multi-directional scheduling schemes are discussed. Moreover, some drawbacks of well-known resource measures, that is, the resource strength (RS), are presented, and the necessity for introducing a new measure is demonstrated. All correlated experiments were coded by *MATLAB* software. Our investigations are based on the well-known benchmark instances that are due to Kolisch's benchmark (see [16, 17]).

## 4.1 Quality of solutions

Initially, the performance of the single and multidirectional scheduling schemes (like in [29]), that is, the sidss (namely, forward scheduling scheme), the bidss, and the trdss, is examined on benchmark problems with more activities, that is, the $J120$ sets (each project contains 120 activities) of the Kolisch's benchmark problems. The comparison measures as in [29] are defined in (1) and (2) by the number of instances solved to optimality and the number of best instances that are obtained, respectively. In Table 1, the related results are presented.

The columns 5 and 9 of this table display the percentage of relative improvement of the trdss in comparison to the bidss, and they are computed as the following formulations:

$$\left(\frac{\mathrm{opt}_{\mathrm{trdss}} - \mathrm{opt}_{\mathrm{bidss}}}{\mathrm{OPT}_j}\right) \times 100; \quad j \in \{J90, J120\}, \tag{1}$$

$$\left(\frac{\mathrm{Best}_{\mathrm{trdss}} - \mathrm{Best}_{\mathrm{bidss}}}{\mathrm{Best}_{\mathrm{bidss}}}\right) \times 100, \tag{2}$$

where $\mathrm{opt}_{\mathrm{xdss}}$ and $\mathrm{OPT}_j$ indicate the number of optimal solutions that are found by $x$-dss and the number of optimal solution that exits for $j \in \{J90, J120\}$, respectively. And similarly, $\mathrm{Best}_{\mathrm{bidss}}$ is the number of best solutions that are obtained by the bidss regarding the other scheduling schemes.

Table 1: Number of the optimal solutions and the number of best solutions for $J120$

| Priority | No. of instances solved to optimality | | | | No. of instances with best solutions | | | |
|---|---|---|---|---|---|---|---|---|
| Rules | sidss | bidss | trdss | %± | sidss | bidss | trdss | %± |
| LPT | 4 | 5 | 5 | **0** | 234 | 357 | 382 | **+5.8** |
| SPT | 3 | 5 | 5 | **0** | 219 | 359 | 361 | **+0.5** |
| MIS | 4 | 11 | 14 | +2.2 | 112 | 208 | 278 | +16.1 |
| MTS | 9 | 29 | 30 | **+0.74** | 87 | 218 | 291 | **+16.8** |
| GRPW | 4 | 20 | 21 | **+0.74** | 56 | 193 | 342 | **+34.3** |
| GRPW* | 11 | 34 | 34 | **0** | 134 | 194 | 265 | **+16.4** |
| | | | | | | | | |
| EST | 4 | 11 | 15 | **+2.94** | 37 | 215 | 362 | **+33.9** |
| EFT | 3 | 15 | 17 | **+1.47** | 28 | 230 | 361 | **+30.2** |
| LST | 11 | 29 | 34 | **+3.68** | 189 | 155 | 262 | **+24.7** |
| LFT | 12 | 31 | 35 | **+2.94** | 168 | 173 | 272 | **+22.8** |
| MSLK | 10 | 33 | 37 | **+2.94** | 65 | 211 | 328 | **+27** |
| | | | | | | | | |
| GRD | 6 | 17 | 17 | **0** | 85 | 241 | 382 | **+32.5** |
| WRUP | 3 | 17 | 26 | **+6.62** | 42 | 175 | 374 | **+45.9** |

In Table 1, with the priority rule[4] such as the MSLK (minimum of slack time), the trdss can be solved about 3% more instances of $J120$ to optimality in comparison to the bidss. With the number of best instances (as comparison measure) a similar trend is the same. For example, under the EST[5] rule, the number of best solutions obtained by the trdss is 33.9% more than the bidss, and this improvement is increased by the WRUP[6] as the priority rule to 45.9%. Note that the number of total optimal solutions for the Kolisch's benchmark $J120$ is 136 instances.

The results indicate that the multidirectional scheduling schemes always outperform the sidss.

Based on the obtained results in Table 1, for solving (scheduling) the RCPSP's instances, is it possible to conclude that, moving to a higher directional scheduling scheme is always beneficial (i.e., leads to solutions with less makespan)? If the answer is negative, what is the (most) influential factors that may affect the performance of higher directional scheduling schemes? Based on the results, the rate of solutions' quality that is yielded by higher directional scheduling schemes, is varied when they compared to the sidss. Could this different influence be done by the factors that they have on the performance of a higher directional scheduling scheme?

To answer the mentioned questions, we need performing a comprehensive examination, which is described in the later sections.

## 4.2 Parameters of resource measures

The density of the coefficient matrix reflected by the availability of resources measured by RS is shown in some studies. It is the most important parameter that can evaluate the resource complexity of an instance of the RCPSP and hence affects the solution's quality (see, for example, [13, 14, 16, 17, 19, 25] for more details).

In this paper, we are looking for a measure that can guide us to decide under what circumstances, it is beneficial to move to higher directional scheduling schemes. Among available measures, we initially consider the RS as defined in (3), a measure of project complexity with regards to the availability of resources (for more details, see [6, 17]).

$$RS = \max \{ \frac{a_r - u_r^{\max}}{k_r^{\text{peak}} - u_r^{\max}} \quad | \quad r = 1, \ldots, |R| \}, \tag{3}$$

[4] A list of priority rule's abbreviations can be found in [13]

[5] Earliest Starting Time-EST

[6] Weighted Resource Utilization ratio and Precedence-WRUP

where $k_r^{peak} = \max \{\sum_{j \in A(t)} u_{jr} \mid r = 1, \ldots, |R|, t = 1, \ldots, |T|\}$, $u_r^{\max} = \max \{u_{rj} \mid j = 1, \ldots, N\}$ and the set $A(t)$ contains the activities that are active at time $t$.

Table 2: Performance of the sidss, the bidss, and the trdss with different RS

| Name | RS | sidss | bidss | trdss | Name | RS | sidss | bidss | trdss |
|---|---|---|---|---|---|---|---|---|---|
| $J901-4$ | 0.2 | 145 | 139 | 141 | $J902-10$ | 0.5 | 90 | 80 | 80 |
| $J905-8$ | 0.2 | 104 | 92 | 93 | $J9010-9$ | 0.5 | 99 | 93 | 92 |
| $J905-10$ | 0.2 | 130 | 134 | 127 | $J9010-10$ | 0.5 | 84 | 84 | 84 |
| $J9013-10$ | 0.2 | 130 | 120 | 117 | $J9014-9$ | 0.5 | 121 | 125 | 125 |
| $J9017-10$ | 0.2 | 103 | 105 | 99 | $J9042-7$ | 0.5 | 99 | 100 | 99 |
| $J9021-10$ | 0.2 | 135 | 138 | 144 | $J9014-2$ | 0.5 | 114 | 103 | 97 |
| $J1202-2$ | 0.2 | 84 | 86 | 86 | $J12049-8$ | 0.5 | 133 | 127 | 129 |
| $J1201-1$ | 0.2 | 132 | 129 | 129 | $J12054-6$ | 0.5 | 141 | 128 | 129 |
| $J1206-1$ | 0.2 | 202 | 177 | 179 | $J12059-8$ | 0.5 | 127 | 122 | 120 |
| $J1207-2$ | 0.2 | 161 | 159 | 158 | $J12060-8$ | 0.5 | 125 | 123 | 123 |
| $J1207-6$ | 0.2 | 168 | 180 | 179 | $J1201-5$ | 0.5 | 144 | 144 | 144 |

Empirical results show that the RS cannot serve the mentioned purpose when it is used solely. In Table 2, the makespan of some different benchmark instances (we choose them randomly) when they solved by three mentioned scheduling schemes, that is, the sidss, the bidss, and the trdss. When the value of RS is fixed, it is clear that no regular trend in the performance of scheduling schemes is observed. For example, when RS= 0.2, for $J9013-10$, moving to higher direction is beneficial and yield smaller makespans. However, for $J9021-10$, with RS= 0.2, the trend is reversed.
With regard to (3), the value of RS depends on $k_r^{\text{peak}}$ (among other parameters), and it does not consider the number of $k_r^{\text{peak}}$ during the horizontal time of the RCPSP's instance. Therefore, based on the obtained results in Table 3, when the value of RS is fixed, two instances do not necessarily have the same number of resource's peaks. Could the number of peaks (number of $k_r^{\text{peak}}$) have any effect on the solutions' quality, which is obtained by multidirectional scheduling schemes? Moreover if the answer is positive, could it serve the mentioned purpose?

To answer these questions, we gathered the number of peaks for each instance of Table 2 in Table 3. We can observe that each instance for each level of the RS has different numbers of peaks. Therefore it seems that the number of peaks is one of the main factors that have affected the performance of the scheduling schemes.
Furthermore, in Tables 4 and 5, we have classified $J90$ and $J120$ benchmark instances with RS= 0.2, based on their number of $k_r^{\text{peak}}$ when project activities are scheduled at their earliest possible time (early schedule). As it can be seen in Table 4, in the instances of $J90$ with RS= 0.2, there are at most five peaks, whereas in the instances of $J120$, the number of peaks reaches 8. In $J90$ with RS= 0.2, there are 69 instances with one peak, 31 instances with

Table 3: Number of peaks for each instance of Table 2

| Name | RS | Name | No. of Peaks | RS | No. of Peaks |
|---|---|---|---|---|---|
| $J901-4$ | 0.2 | 1 | $J902-10$ | 0.5 | 4 |
| $J905-8$ | 0.2 | 2 | $J9010-9$ | 0.5 | 4 |
| $J905-10$ | 0.2 | 2 | $J9010-10$ | 0.5 | 9 |
| $J9013-10$ | 0.2 | 1 | $J9014-9$ | 0.5 | 1 |
| $J9017-10$ | 0.2 | 1 | $J9042-7$ | 0.5 | 2 |
| $J9021-10$ | 0.2 | 1 | $J9014-2$ | 0.5 | 3 |
| $J1202-2$ | 0.2 | 3 | $J12049-8$ | 0.5 | 2 |
| $J1201-1$ | 0.2 | 2 | $J12054-6$ | 0.5 | 1 |
| $J1206-1$ | 0.2 | 1 | $J12059-8$ | 0.5 | 1 |
| $J1207-2$ | 0.2 | 1 | $J12060-8$ | 0.5 | 2 |
| $J1207-6$ | 0.2 | 1 | $J1201-5$ | 0.5 | 1 |

two peaks, and so on (see Table 4). In $J120$ with RS= 0.2, there are 177 instances with one peak, 53 instances with two peaks, and so on (see Table 5). Hence, for the fixed level of the RS, we can say that the number of peaks can affect the performance of scheduling schemes.

Table 4: Number of peaks in $J90$ with RS= 0.2

| Number of Peaks | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Number of Instances | 69 | 31 | 17 | 6 | 2 |

Table 5: Number of Peaks in $J120$ with RS= 0.2

| Number of Peaks | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Number of Instances | 177 | 53 | 16 | 9 | 3 | 1 | 0 | 1 |

In what follows, we investigate the concept of peaks and some of their related characteristics in more detail. We first give a formal definition for *resource overflow* (overflow for short).

**Definition 1.** Let $ES_j$ ($EF_j$) be earliest starting (finishing) time of activity $j$. For any resource $r \in R$, calculate $\omega_r(t) := \sum_{j \in A(t)} u_{jr} - a_r$ in which $A(t) = \{i \in J \,|ES_i \leq t < EF_i\}$. Therefore, if $\omega_r(t) > 0$, then we say that an overflow with respect to resource $r$ has occurred at time $t$ (see [30]).

It is clear to note that, in a resource-constrained project, if $\exists\ t;\ \omega_r(t) > 0\ \ (1 \leq t \leq T)$, where $T$ is early scheduling time in the project, then the set of project's overflows includes the set of peaks [30]. Especially speaking, each peak is an overflow if $\exists t;\ \omega_r(t) > 0\ \ (1 \leq t \leq T)$.

By using Definition 1, we investigate the effect of the number of overflows and also the effect of the average amount of overflows on the performance of the scheduling schemes.

### 4.2.1 Number of overflows

In order to categorize $J90$ and $J120$ instances according to the number of overflows that the early schedule of each instance has, we consider three levels for RS, that is, 0.2, 0.5 and 0.7 for $J90$ and two levels of RS, that is, 0.2 and 0.5 for $J120$. The number of overflows for each instance is calculated by (4)

$$\text{number of Overflow} = \sum_{t=0}^{T-1} \sum_{r \in R} \chi_r(t), \tag{4}$$

where

$$\chi_r(t) = \begin{cases} 1 & \omega_r(t) > 0, \\ 0 & \text{o.w.} \end{cases}$$

Table 6 shows the results. For $J90$, the number of overflows varies in the interval $[49, 287]$ for RS= 0.2; it varies in the interval $[17, 163]$ for RS= 0.5; and it varies in the interval $[6, 72]$ for RS= 0.7. Similarly, for $J120$, the number of overflows varies in the interval $[48, 369]$ for RS= 0.2; and finally, it varies in the interval $[23, 248]$ for RS= 0.5. The data show an inverse relationship between the level of RS and the number of overflows. As the level of RS decreases, the number of overflows increases. In other words, as expected, the more complex the problem is (RS$\rightarrow$ 0), the higher is the number of overflows.

Table 6: Number of overflows in $J90$ and $J120$ for different levels of RS

| | RS ($J90$) | | | RS ($J120$) | |
|---|---|---|---|---|---|
| | 0.2 | 0.5 | 0.7 | 0.2 | 0.5 |
| Minimum number of Overflow | 48 | 17 | 6 | 48 | 23 |
| Maximum number of Overflow | 287 | 163 | 72 | 369 | 248 |

The effect of the number of overflows on the solution quality (% instances that the scheduling schemes have been able to solve them with less makespan) of $J90$ and $J120$ instances when solved by the bidss (dashed lines) and the trdss (solid lines), is displayed in Figures 1 and 2, respectively. Note that the RS varies from 0.2 to 0.7. Instead of displaying the number of overflows individually, we divided the horizontal axes into four intervals such that the number of instances in each interval is approximately equal. By considering these figures, the following observations are noted.

- Irrespective of the level of RS and the number of overflows, the trdss is capable of obtaining better solutions for more instances than the bidss (this is in line with the conclusions of Yousefzadeh et al.). The performance curves of the trdss always stand above the bidss (solid lines stand above-dashed lines).

- As the number of overflows decreases, irrespective of the level of RS, no significant difference is observed between the performance of the bidss and the trdss (see the proposition). Note that as expected this is more evident with higher levels of RS.

**Proposition 1** For the instances where the number of overflows is zero, there is no difference between the performance of the bidss and the trdss.

*Proof.* It is clear that if the number of overflows is zero, then, for all $r$ and $t$, we have

$$\omega_r(t) = \sum_{j \in A(t)} u_{j\,r} - a_r \leq 0 \Rightarrow a_r \geq k_r^{\text{peak}}, \quad \text{for all } r,$$

that is, there is no resource limitation and both scheduling schemes will schedule the activities in their earliest starting time; therefore the scheduling schemes perform equally well. Furthermore, the resulting feasible makespan is equal to the lower bound, and hence the solution is optimum. □

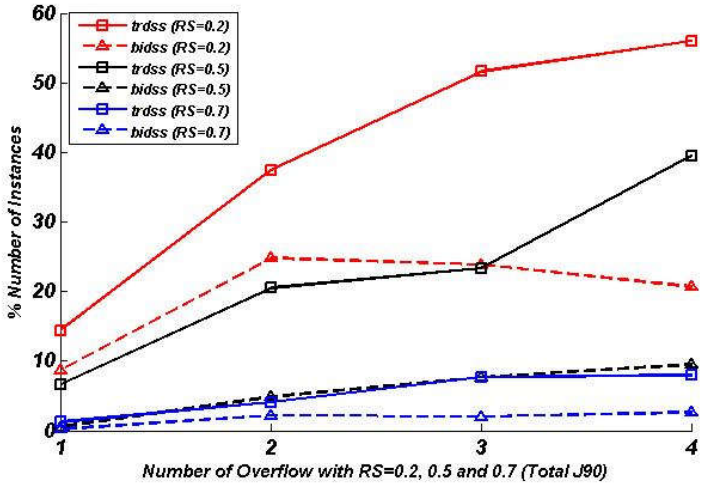


Figure 1: Performance of trdss v.s bidss regarding the number of overflows with different RS (J90)

- As mentioned before by decreasing the level of RS, the trdss begins to outperform the bidss. However, this is more evident in instances with a larger number of overflows.
- Irrespective of the level of RS and the number of activities of the instances, as the number of overflows increases, the trdss performs better than the bidss.

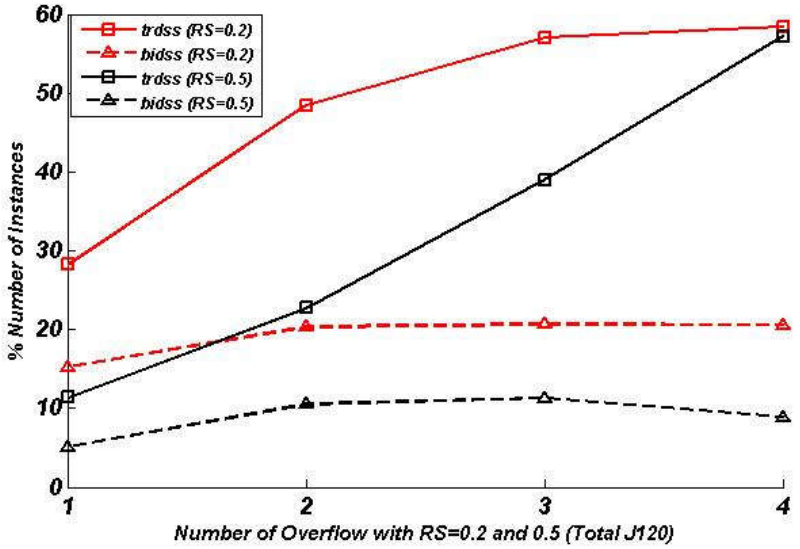


Figure 2: Performance of trdss v.s bidss regarding the number of overflows with different RS (J120)

- The performance gap between the two scheduling schemes widens as the number of overflows increases. We can also observe the same behavior as the number of activities increases. Tables 7 and 8 give the corresponding information. For instance, when RS= 0.2, the performance gap in the interval 1 for $J90$ is 5.7 and for $J120$ is 12.9. However, when RS= 0.2 the performance gap in the interval 4 for $J90$ is 35.3 and for $J120$ is 35.8.

### 4.2.2 Average amount of overflows

Apart from the number of overflows, the average amount of overflows can also affect the performance of the scheduling schemes. By considering that there

Table 7: Effect of increasing the number of activities on the trdss and the bidss (RS= 0.2)

| Intervals | $J90$ | | | $J120$ | | |
|---|---|---|---|---|---|---|
| | bidss | trdss | gap | bidss | trdss | gap |
| 1 | 8.7 | 14.4 | 5.7 | 15.3 | 28.2 | 12.9 |
| 2 | 24.8 | 37.5 | 12.7 | 20.3 | 48.5 | 28.2 |
| 3 | 23.8 | 51.6 | 27.8 | 20.7 | 57.0 | 36.3 |
| 4 | 20.7 | 56 | 35.3 | 21.6 | 57.4 | 35.8 |

Table 8: Effect of increasing the number of activities on the trdss and the bidss (RS= 0.5)

| Intervals | $J90$ | | | $J120$ | | |
|---|---|---|---|---|---|---|
| | bidss | trdss | gap | bidss | trdss | gap |
| 1 | 0.5 | 6.7 | 6.2 | 5.1 | 11.4 | 6.3 |
| 2 | 4.9 | 20.5 | 15.6 | 10.6 | 22.7 | 12.2 |
| 3 | 7.7 | 23.3 | 15.6 | 11.3 | 39 | 27.7 |
| 4 | 9.5 | 39.5 | 30 | 8.9 | 57.2 | 48.3 |

are $M$ overflows in an instance, its average amount of overflows is calculated as $\sum_{r\in R}\sum_{t\in OT}\omega_r(t)/M$, where OT is the set of overflows' time.
As the following, an inverse relation between the value of overflow and the value RS is proved:
Let $r \in R$; then from (3), we have

$$RS_r = \frac{a_r - u_r^{\max}}{k_r^{\text{peak}} - u_r^{\max}},$$

Thus

$$k_r^{\text{peak}} = u_r^{\max} + \frac{a_r - u_r^{\max}}{RS_r}.$$

By the definition of overflow, there is a coefficient like $c_r$ subject to:

Average amount of overflows for resource type

$$\begin{aligned} r &= \sum_{t\in OT}\omega_r(t)/M \\ &= k_r^{\text{peak}} \times c_r \\ &= \left(u_r^{\max} + \frac{a_r - u_r^{\max}}{RS_r}\right) \times c_r. \end{aligned}$$

If let $\varphi_1(r) = c_r u_r^{\max}$ and $\varphi_2(r) = c_r a_r$, then

$$\sum_{t\in OT} \omega_r(t)/M = \left(\varphi_1(r) + \frac{\varphi_2(r) - \varphi_1(r)}{RS_r}\right). \tag{5}$$

Hence,

$$M = \sum_{t\in OT} \omega_r(t) \Big/ \left(\varphi_1(r) + \frac{\varphi_2(r) - \varphi_1(r)}{RS_r}\right) \tag{6}$$

From (6), we can find an inverse relationship between the average amount of overflows and the RS, that is, as $\mathrm{RS}_r$ decreased (suppose that $\varphi_1(r)$ and $\varphi_2(r)$ are fixed), the average amount of overflows for the resource type $r$ increased. Table 9 confirms the mentioned statement. In Table 9, for example, in $J90$, the average amount of overflows for RS= 0.2, 0.5, and 0.7 vary in intervals [4.8.36.3], [4, 23.8], and [3, 14.5], respectively.

The relation between the RS and the average number of overflows is also mentioned in (6).

Table 9: Effect of RS's level on mean amount of overflow

| | $RS(J90)$ | | | $RS(J120)$ | |
|---|---|---|---|---|---|
| | 0.2 | 0.5 | 0.7 | 0.2 | 0.5 |
| Min average amount of Overflows | 4.8 | 4.0 | 3.0 | 4.5 | 4.1 |
| Max average amount of Overflows | 36.3 | 23.8 | 14.5 | 40.0 | 29.7 |

From (5) and (6), we can conclude that despite the inverse relation between overflow and the RS, this relation also depends on the topology of networks such as availability of the resource, resource demand, and precedence relation, and so on. Hence the effect of the RS and the overflow for different projects is not the same.

The effect of the average amount of overflows on the performance of the bidss and the trdss is displayed in Figures 3 and 4.
By considering these figures, the following observations are noted:

- In all instances and under different levels of the RS, in both $J90$ and $J120$, the trdss performs better than the bidss.
- As the average number of overflows increases, the trdss performs better than the bidss, and this becomes more evident when the number of activities increases.
- Both scheduling schemes perform well as the average amount of overflows is decreased.

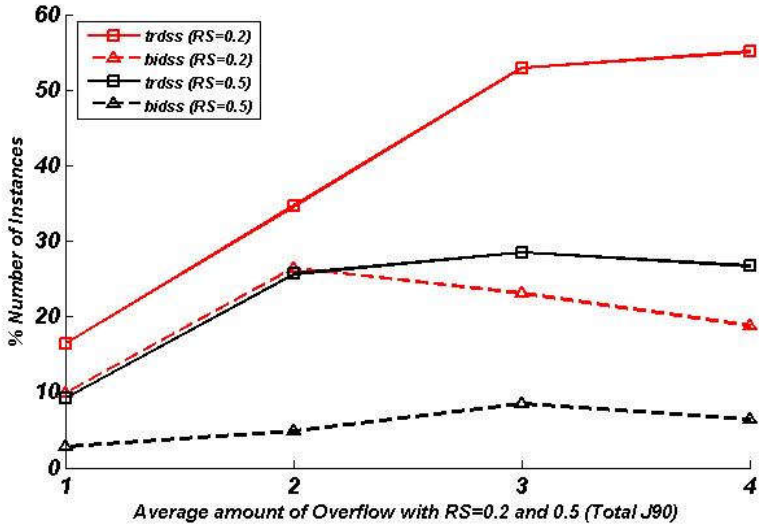


Figure 3: Effect of average amount of overflows on the trdss and the bidss in $J90$

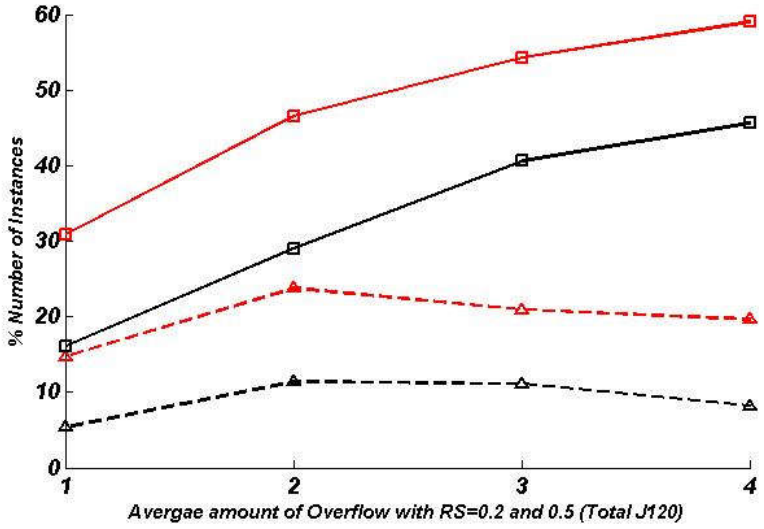


Figure 4: Effect of average amount of overflows on the trdss and the bidss in $J120$

## 5 Summary and conclusions

In our paper, we tried to address the question: "What are the conditions that are beneficial to move to higher directional scheduling schemes (obtain a feasible solution with less makespan)?" For doing so, we first, by numerical results, showed that the number of the scheduling activities that are being shifted to the left and the separation of the scheduling activities into some groups, are two main factors that reduce the makespan. Based on the experimental results, by considering the number of peaks and the average number of overflows as a new measure, we concluded that the multidirectional scheduling schemes could lead to a feasible solution with quality improvement if the number of directors is increased. As such, the dependence of the quality of the solution to the type of priority rules used can be dampened.

## References

1. Agarwal, R., Tiwari, M.K. and Mukherjee, S.K. *Artificial immune system based approach for solving resource constraint project scheduling problem*, Int. J. Adv. Manuf. Technol. 34(5-6) (2007), 584–593.

2. Bouleimen, K. and Lecocq, H. *A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version*, European J. Oper. Res. 149(2) (2003), 268–281.

3. Browning, T.R. and Yassine, A.A. *A random generator of resource-constrained multi-project network problems*, J. Sched. 13(2) (2010), 143–161.

4. Brucker, P., Drexl, A., Möhring, R., Neumann, K. and Pesch, E. *Resource-constrained project scheduling: Notation, classification, models, and methods,* European J. Oper. Res. 112(1) (1999), 3–41.

5. Brucker, P., Knust, S., Schoo, A. and Thiele, O. *A branch and bound algorithm for the resource-constrained project scheduling problem*, European J. Oper. Res. 107(2) (1998), 272–288.

6. Cooper, D.F. *Heuristics for Scheduling Resource-Constrained Projects: An Experimental Investigation,* Manag. Sci. 22(11) (1976), 1186–1194. doi:10.1287/mnsc.22.11.1186.

7. Creemers, S. *Minimizing the expected makespan of a project with stochastic activity durations under resource constraints,* J. Sched. 18(3) (2015), 263–273.

8. Davis, E.W. and Patterson, J.H. *A Comparison of Heuristic and Optimum Solutions in Resource-Constrained Project Scheduling,* Manag. Sci. 21(8) (1975), 944–955.

9. Goncharov, E.N. and Leonov, V.V. *Genetic algorithm for the resource-constrained project scheduling problem,* (Russian); translated from Avtomat. i Telemekh. 2017, , no. 6, 173–189 Autom. Remote Control 78 (2017), no. 6, 1101–1114.

10. Hartmann, S. *A self-adapting genetic algorithm for project scheduling under resource constraints,* Naval Res. Logist. 49(5) (2002), 433–448. doi:10.1002/nav.10029.

11. Icmeli, O. and Rom, W.O. *Solving the resource constrained project scheduling problem with optimization subroutine library,* Comput. Oper. Res. 23(8) (1996), 801–817.

12. Kanit, R., Ozkan, O. and Gunduz, M. *Effects of project size and resource constraints on project duration through priority rule-base heuristics,* Artif. Intell. Rev. 32(1-4) (2009), 115–123. doi:10.1007/s10462-009-9138-1.

13. Klein, R. *Bidirectional planning: improving priority rule-based heuristics for scheduling resource-constrained projects,* European J. Oper. Res. 127(3) (2000), 619–638.

14. Klein, R. and Scholl, A. *PROGRESS: Optimally solving the generalized resource-constrained project scheduling problem,* Special issue on project scheduling. Math. Methods Oper. Res. 52(3) (2000), 467–488.

15. Kolisch, R. *Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation,* European J. Oper. Res. 90(2) (1996), 320–333.

16. Kolisch, R., Schwindt, C. and Sprecher, A. *Benchmark instances for project scheduling problems*, In: Weglarz J. (eds) Project Scheduling. International Series in Operations Research & Management Science, vol 14. Springer, Boston, MA. (1999).

17. Kolisch, R., Sprecher, A. and Drexl, A. *Characterization and Generation of a General Class of Resource-Constrained Project Scheduling Problems,* Manag. Sci. 41(10) (1995), 1693–1703.

18. Kurtulus, I. *Multi-project scheduling: Analysis of scheduling strategies under Unequal DealyPenalities,* J. Oper. Manag. 5(3) (1985), 161–172.

19. Kurtulus, I. and Davis, E.W. *Multi-Project Scheduling: Categorization of Heuristic Rules Performance,* Manag. Sci. 28(2) (1982), 161–172.

20. Lenstra, J.K. and Rinnooy, K.A.H.G. *Complexity of Scheduling under Precedence Constraints,* Operations Res. 26(1) (1978), 22–35.

21. Liang, Y., Cui, N., Wang, T. and Demeulemeester, E. *Robust resource-constrained max-NPV project scheduling with stochastic activity duration,* OR Spectrum. 41(1) (2019), 219–254.

22. Mortazavi Nejad, M., Tareghian, H.R. and Sari, Z. *Payment scheduling under project crashing based on project progress,* Iranian Journal of Numerical Analysis and Optimization, 7(2) (2017), 39–56.

23. Petrović, R. *Optimization of Resource Allocation in Project Planning,* Operations Res. 16(3) (1968), 559–568.

24. Raghavendra, B.V. *Scheduling in parallel machines environment using genetic algorithm,* J. Appl. Eng. Sci. 16(1) (2018), 36–42.

25. Servranckx, T. and Vanhoucke, M. *A tabu search procedure for the resource-constrained project scheduling problem with alternative subgraphs,* European J. Oper. Res. 273(3) (2019), 841–860.

26. Shukla, S.K., Son, Y.J. and Tiwari, M.K. *Fuzzy-based adaptive sample-sort simulated annealing for resource-constrained project scheduling,* Int. J. Adv. Manuf. Technol. 36(9-10) (2008), 982–995.

27. Sokolov, B., Tolpegin, O., Ipatov, Y. and Andrianov, Y. *Polymodel description and qualitative analysis of problems for measurement computer operations planning in cyber-physical systems,* J. Appl. Eng. Sci. 16(4) (2018), 577–582.

28. Ulusoy, G. and Özdamar, L. *Heuristic Performance and Network/Resource Characteristics in Resource-constrained Project Scheduling,* J. Oper. Res. Soc. 40(12) (1989), 1145–1152.

29. Yoosefzadeh, H.R., Tareghian, H.R. and Farahi, M.H. *Tri-directional Scheduling Scheme: Theory and Computation,* J. Math. Model. Algorithms. 9(4) (2010), 357–373.

30. Yoosefzadeh, H.R., Tareghian, H.R. and Farahi, M.H. *Multidirectional scheduling scheme in resource-constrained project scheduling problem,* Naval Res. Logist. 61(1) (2013), 44–55.

31. Zamani, M. *A high-performance exact method for the resource-constrained project scheduling problem,* Comput. Oper. Res. 28(14) (2001), 1387–1401.

32. Zhu, X., Ruiz, R., Li, S., and Li, X. *An effective heuristic for project scheduling with resource availability cost,* European J. Oper. Res. 257(3) (2017), 746–762.