Communications in Mathematics and Applications
Vol. 10, No. 1, pp. 123–130, 2019
ISSN 0975-8607 (online); 0976-5905 (print)
Published by RGN Publications

http://www.rgnpublications.com

DOI: 10.26713/cma.v10i1.1229

Research Article

User Preferences With Urgent-Time Intervals in a Wireless-Network

Hani Alquhayz

Department of Computer Science and Information, College of Science at Zulfi, Majmaah University, Majmaah 11952, Saudi Arabia
h.alquhayz@mu.edu.sa

Abstract. This paper focuses on a scheduler component in a wireless network, with constraints based on user preferences. These constraints reflect the fact that users wish to privatize networks for fixed periods (intervals) in order to send urgent data. The user sets in advance certain data constraints and time periods. Each time period is presented as a frozen interval for the router and other users. Only the administrator can send data during these specifics periods and all others users will be frozen. During this interval, the intelligencer can send confidential and emergency data without any concurrency with other data through this interval, which is called an "emergency and secret interval". Each time, a scheduling problem due to unavailable machines and jobs will be solved. This problem is NP-hard. In this research, we develop two algorithms for transmitting data from the sender to the receiver.

Keywords. Data transmission; Wireless network; Packets; Heuristics; Scheduling into networks

MSC. 68M10

Received: January 7, 2019 **Accepted:** February 7, 2019

1. Introduction

Wireless networks are very important in our lives with regard to communication and sharing data [9]. Life becomes easier through using networks. The big data transmitted into wireless networks oblige us to seek a more secure process in order to protect users. The protection of wireless networks is becoming increasingly vital ([6, 8]). In a network that requires higher security, it is necessary to control the growing amount of data flowing across the network.

Besides warding intruders away from the network, it is also important to follow well in periods to ensure efficient data transmission. Scheduling theory is utilized more in networks to guarantee efficient transmission. In this work, we focused on the application of scheduling in a very specific network used by the intelligencer. The scheduling problem used in this work is based essentially on the unavailable machines and jobs. In [4], a branch bound problem is developed to solve the problem. A single machine with unavailability intervals and a setup time was studied in [1]. Using a neighborhood search approach and an effective heuristic *ERD-LPT*, a solution was developed [2]. The optimization of the problem which can have as its objective the minimization of the sum of the maximum tardiness and promptness of jobs has also been developed in several researches [10]. A randomized algorithms is used in [3] to develop heuristics for parallel machines with minimizing the makespan. The case of only one unavailability interval per machine is studied in [7] to prove that there exist an absolute bounds of the studied problem. In [5], authors gives an approximation for four problems. One of the four studied problem is a common deadline given for all tasks and the second one have as objective to minimize the makespan and maximum lateness.

This paper is structured as follows. Section 2 focuses on a description of the studied problem. Section 3 presents two heuristics as approximate solutions for the studied problem. The first heuristic is based on the repeating of a resolution of the subset-sum problems. The second is based on the randomized iteratively heuristic. Section 4 presents the experimental results to show the performance of heuristics in the network case.

2. Problem Description

In this section, we present a network problem and a scheduling problem. Our study concerns a special network used by the intelligencers, where time is critical and the diffusion of information has unpredictable results. The network problem is presented as follows. At fixed periods, important and confidential data must be transmitted on time. These emergency data must block all other data in the network, so that they can pass safely. Indeed, the blockage of other data becomes necessary. The periods of time will be predetermined in advance and called Urgent-Time Intervals (*UTI*). When the information is extremely confidential, it is vital to set a divulgence time for the sent data. The intelligencer is the person responsible for the divulgence of the data by fixing the "data time divulgence (*DTD*)". The main problem is how to assign data on the router while respecting all of the *UTI* on each router. The resolution will be performed based on the scheduling problem. In this work, we focus on the one router problem, so let there be only one router and we fix several *UTI*s. Each *UTI* will be represented by intervals as $[a_k - b_k]$ with $k \in \{1,\dots,n_I\}$ and n_I is the number of total urgent-time intervals.

In a network with high security, an encrypted process is applied by the "Encrypter" component. The "Reveal data time", as shown in Figure 1, is fixed by the intelligencer. The process of the proposed "Intelligence network" is presented in Figure 1. For each selected data package J_j received by the intelligencer and collected in the "Variables collector", certain preferences $Pref_j$ will be chosen. The release time r_j is the time from which we can divulge the data J_j. This time is calculated as follows: $r_j = t_j + t_j^E$, with

- t_j is the time given by the component "$Timer : t$" and corresponding to the arrival time of

data J_j at the "Processing engine" .
- ∘ t_j^E is the encryption time of the data J_j.

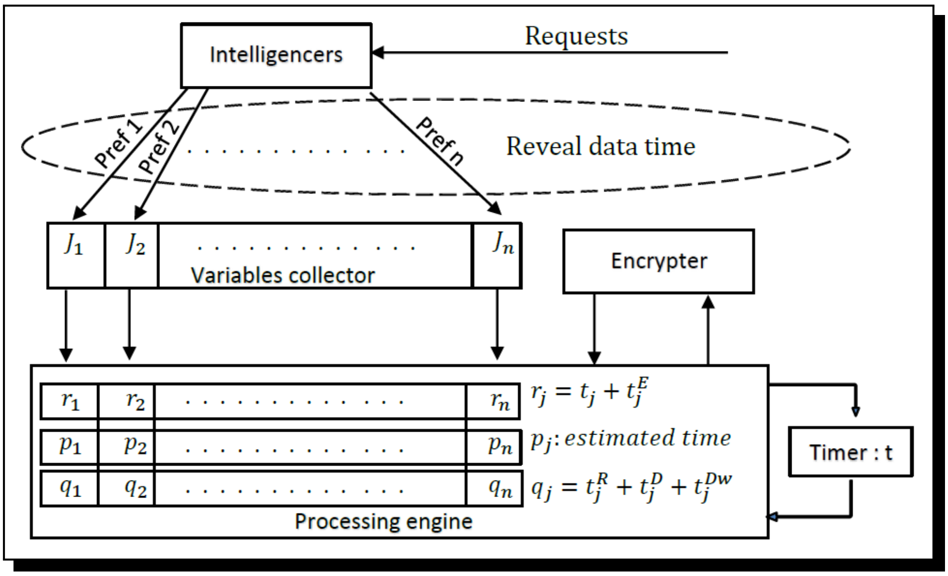

Figure 1. The intelligencer preferences intervention process

On the other hand, the time required q_j is the minimum time required by the receiver to collect data J_j. This time is calculated as follows: $q_j = t_j^R + t_j^D + t_j^{Dw}$, where

- t_j^R, the intelligencer is the responsible to fixe this time as preferences in advance for the data J_j.
- t_j^D is the decryption time of data J_j.
- t_j^{Dw} is the waiting time in the buffer to decrypt the data J_j.

The data processing (sending) time p_j is the time required to send the data J_j.

Now, the problem is how to send data which are characterized by r_j and q_j constraints through one router. We define the process as the "Intelligence Network Problem", denoted by *INP*.

3. Approximate Solutions

In this section we present two approximate solutions for *INP*.

Lemma 3.1. *INP is equivalent to* $1|r_j, q_j, h|C_{\max}$.

Proof. Each *UTI* in the network will correspond to machine unavailability. The router corresponds to one machine. The jobs correspond to data with r_j and q_j constraints, so there are jobs with unavailability. □

Theorem 3.2. *INP is NP-hard.*

Proof. Since *INP* is equivalent to $1|r_j, q_j, h|C_{\max}$. □

3.1 Subset-sum Repeating-Resolution based Heuristic (*SRR*)

The studied scheduling problem can be formulating as a subset-sum problem. The *SRR* heuristic can utilize the formulation in the subset-sum.

The heuristic is based on the following method. At time t, among the non-scheduled data we choose those that are available and test the following:

(a) data with the biggest q_j value

(b) $t + p_j \le a_{k+1}$.

Let J_t be all data satisfying (a) and (b). Then we search for data that can be assigned on the router between $[t - a_{i+1}]$ using the following subset-sum:

$$SSP: \begin{cases} \max \sum_{j \in J_t} p_j x_j, \\ \text{subject to } \sum_{j \in J_t} p_j x_j \le a_{i+1} - t, \\ x_j \in \{0,1\}, \ \forall\ j \in J_t \end{cases} \tag{1}$$

Three cases are presented as follows:

- If $0 \le t < a_1$ then $i = 0$ and we apply *SSP*.
- If $b_k \le t < a_{k+1}$, $k \ge 1$ then $i = k$ and we apply *SSP*.
- If $t \ge b_{n_I}$, we do not have any intervals after the last one and we apply *Schrage* for $1|r_j, q_j|C_{\max}$.

3.2 Iterative-Randomized based Heuristic (*IR*)

In this heuristic, we order the data according to the non-increasing order. We choose one data package from among the two first data package with the greatest q_j with a respective probability α and $1 - \alpha$. If there are one or more data package that are not dispatched and which have the same q_j values, we select the data with the greatest value of p_j, with a respective probability of α and $1 - \alpha$. In an experimental study, $\alpha = 0.7$. This heuristic is repeated 500 times and the minimum value will be chosen.

4. Experimental Results

In this section, we present the results produced by the proposed algorithms. We used two classes of normal distribution. Each class generated 800 instances. The way of generating p_j, r_j, q_j and n_w characterizes every Class 1 and Class 2. In our study, we generate all instances for each class as follows:

Class 1: p_j is in $U[1-10]$.

r_j and q_j are generated in $U[0; n/n_I - 1]$.

For intervals: a_1 is in $U[1-30]$ and $b_1 = a_1 + st$, where st is an integer in $U[1-20]$.

$\forall\ k > 1$, $a_k = b_{k-1} + st_k$ with st_k is in $U[1-30]$ and $b_k = a_k + V_k$ with V_k is in $U[1-20]$. The number of intervals is $n_I \in \{1,3,5,10,15,50,100\}$. For each interval, we generate 2 varieties and for each variety we generate 10 instances. The number of data n is in $\{10,20,50,100,200,500,1000\}$. Thus, in totality, we have 800 instances for Class 1.

Class 2: p_j is in $U[1-10]$.
r_j and q_j are in $U[0-30]$ The intervals router is generated following the same method described for Class 1. The number $n_I \in \{1,3,5,10,15,50,100\}$. For each interval, we generate 2 varieties and for each variety we generate 10 instances.
The number of data n is in $\{10,20,50,100,200,500,1000\}$. Thus, we have 800 instances.

We denote by :

- *Min*: the number of instances where the studied heuristic is equal to the minimum heuristic value.
- *U*: the studied heuristic.
- *UB*: the minimum heuristic value.
- $Gap = \frac{U-UB}{UB} \times 100$; *GapM* is the max value of *Gap*.
- *Time*: the average time in seconds. We denote by '–' if the time is less than 0.001s.
- *Perc*: the percentage of the sum of the *Min* in the overall 800 instances.

Table 1 concerns Class 1. This table shows that the heuristic *IR* outperform *SRR* at 67.6%, with an average time of 0.982s. The gap value obtained for the *IR* heuristic for all instances for Class 1 is 10.8%. However, the gap value obtained for *SRR* is 24.8% for an average time of 0.010s.

Table 1. Over all instances for Class 1, global result

IR			*SRR*		
Perc	*GapM*	*Time*	*Perc*	*GapM*	*Time*
67.6%	10.8%	0.982	52.1%	24.8%	0.010

The results for Class 2 are presented in Table 2. Indeed, as shown in the table below, the maximum percentage of 74.9% is obtained for *IR* the heuristic with a gap equal to 3.1% in 0.894s. However, a minimum percentage of 46.5% is observed for *SRR*, with a corresponding gap equal to 32.9% in 0.010s.

Table 2. Over all instances for Class 2, global result

IR			*SRR*		
Perc	*GapM*	*Time*	*Perc*	*GapM*	*Time*
74.9%	3.1%	0.894	46.5%	32.9%	0.010

In Figures 2 and 3, we present the behavior of *Perc* according to n_I for respectively Class 1 and Class 2. Figure 2 shows the variation of *Perc* for Class 1. In this figure, the values of *Perc* increase as the value of n_I increases for the heuristic *SRR*. However, the *Perc* values decrease when the n_I increases for the heuristic *IR*.

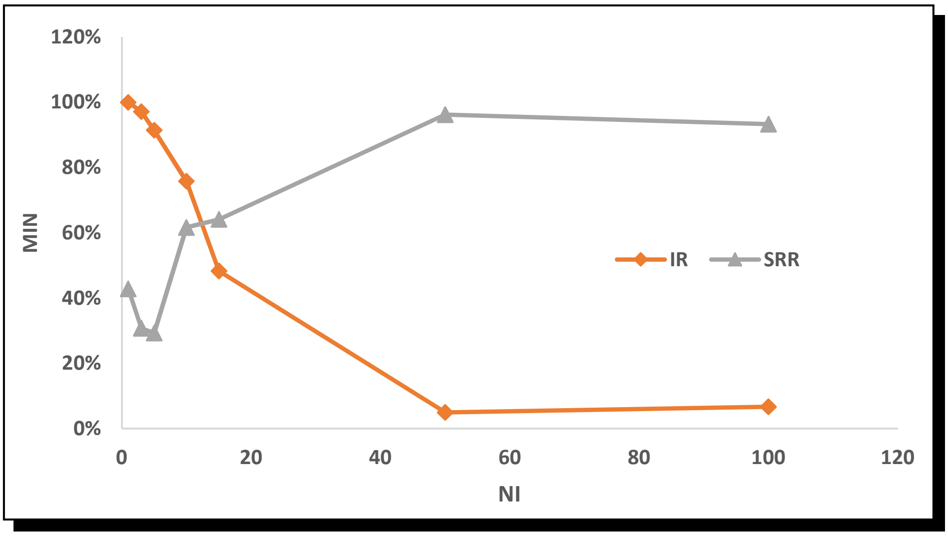

Figure 2. Behavior of *Perc* according to n_I for Class 1

Figure 2 shows that, there exist an intersection point between the *IR* curve and the *SRR* curve. This point represents the N_I value when the *Perc* has the same value for the two heuristics.

For Class 2, as shown in Figure 3, the behavior of *Perc* does not follow any particular order. It is clear that, for Class 2, the *IR* curve is always above the *SRR* curve.

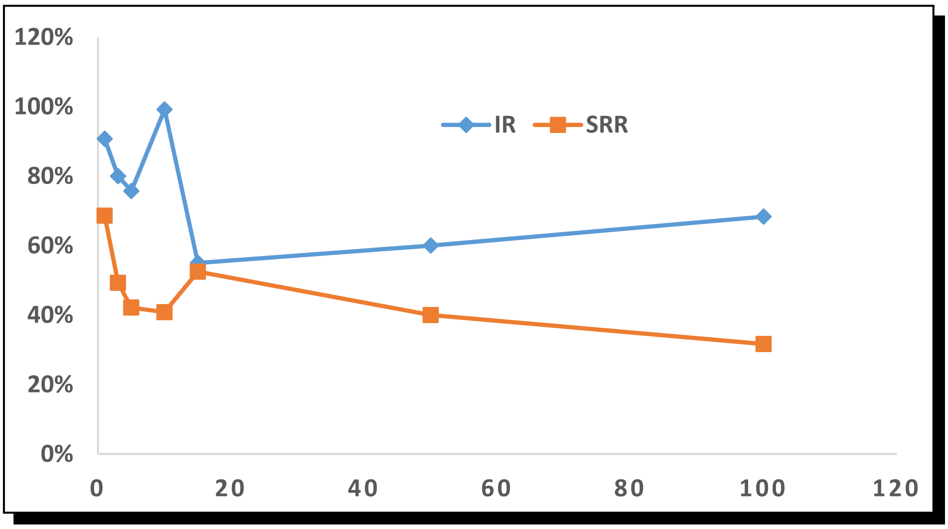

Figure 3. Behavior of *Perc* according to n_I for Class 2

Tables 3 and 4 present the performance of each heuristic according to n for respectively, Class 1 and Class 2. In Table 3, the minimum gap of 1.2% is obtained for $n = 1000$ for *IR*.

However, the maximum gap of 24.8% is obtained for $n = 10$ for SRR. The maximum time is 5.539s for IR heuristic when $n = 1000$.

Table 3. Heuristic performance according to n for Class 1

n	*IR*			*SRR*		
	Min	*GapM*	*Time*	*Min*	*GapM*	*Time*
10	57	2.2%	0.000	25	24.8%	0.000
20	81	10.8%	0.001	53	12.3%	0.000
50	74	3.3%	0.007	51	3.7%	0.000
100	80	9.0%	0.033	61	7.7%	0.001
200	86	5.8%	0.151	72	1.2%	0.002
500	87	2.7%	1.145	64	0.6%	0.012
1000	76	1.2%	5.539	91	0.3%	0.056

In Table 4, it is clear that, the maximum gap obtained is 32.9% for $n = 20$ for the SRR heuristic and Class 2. Zero gap values are obtained when $n = 10$ and $n = 20$.

Table 4. Heuristic performance according to n for Class 2

n	*IR*			*SRR*		
	Min	*GapM*	*Time*	*Min*	*GapM*	*Time*
10	60	0.0%	0.000	23	16.6%	0.000
20	100	0.0%	0.001	33	32.9%	0.000
50	91	2.1%	0.008	36	8.9%	0.000
100	85	3.1%	0.035	55	2.8%	0.001
200	101	1.9%	0.159	63	2.0%	0.002
500	78	0.8%	1.123	74	1.9%	0.012
1000	84	0.7%	4.934	88	1.4%	0.054

Acknowledgement

The author would like to thank the Deanship of Scientific Research at Majmaah University for supporting this work under Project Number No. 1440-65.

Competing Interests

The author declares that he has no competing interests.

Authors' Contributions

The author wrote, read and approved the final manuscript.

References

[1] F. Angel-Bello, A. Alvarez, J. Pacheco and I. Martinez, A single machine scheduling problem with availability constraints and sequence-dependent setup costs, *Applied Mathematical Modelling* **35**(4) (2011), 2041 – 2050, DOI: 10.1016/j.apm.2010.11.017.

[2] W.W. Cui and Z.Q. Lu, Minimizing the makespan on a single machine with flexible maintenances and jobs' release dates, *Computers and Operations Research* **80** (2017), 11 – 22, DOI: 10.1016/j.cor.2016.11.008.

[3] M. Haouari, A. Gharbi and M. Jemmali, Tight bounds for the identical parallel machine scheduling problem, *International Transactions in Operational Research* **13**(6) (2006), 529 – 548, DOI: 10.1111/j.1475-3995.2006.00562.x.

[4] W. Hfaiedh, C. Sadfi, I. Kacem and A. Hadj-Alouane, A branch-and-bound method for the single-machine scheduling problem under a non-availability constraint for maximum delivery time minimization, *Applied Mathematics and Computation* **252** (2015), 496 – 502, DOI: 10.1016/j.amc.2014.11.103.

[5] I. Kacem and H. Kellerer, Approximation schemes for minimizing the maximum lateness on a single machine with release times under non-availability or deadline constraints, *Algorithmica* **80** (12) (2018), 3825 – 3843 DOI: 10.1007/s00453-018-0417-6.

[6] D. Ma and G. Tsudik, Security and privacy in emerging wireless networks, *IEEE Wireless Communications* **17**(5) (2010), 12 – 21, DOI: 10.1109/MWC.2010.5601953.

[7] A. Mahjoub, J. Kaabi and Y. Harrath, Absolute bounds of list algorithms for parallel machines scheduling with unavailability periods, *International Transactions in Operational Research* (2018), 1 – 17, DOI: 10.1111/itor.12589.

[8] H.K.D. Sarma and A. Kar, Security threats in wireless sensor networks, Carnahan Conferences Security Technology, *Proceedings 2006 40th Annual IEEE International. IEEE*, 2006, DOI: 10.1109/CCST.2006.313457.

[9] S. Sharma, R. Mishra and K. Singh, A review on wireless network security, *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, Springer, Berlin, Heidelberg, 2013, DOI: 10.1007/978-3-642-37949-9_58.

[10] M. Yazdani, S.M. Khalili, M. Babagolzadeh and F. Jolai, A single-machine scheduling problem with multiple unavailability constraints: A mathematical model and an enhanced variable neighborhood search approach, *Journal of Computational Design and Engineering* **4**(1) (2017), 46 – 59, DOI: 10.1016/j.jcde.2016.08.001.