Scientia Iranica, Vol. 15, No. 5, pp 652–660

SCIENTIA IRANICA

Research Note

Applying Circular Coloring to Open Shop Scheduling

M. Modarres[1,*] and M. Ghandehari[1]

IN this paper, a new approach to formulate a class of scheduling problems is introduced, which can be applied to many other discrete problems with complicated structures. The concept of graph circular coloring is applied to develop a model for the special case of an open shop scheduling problem. In this problem, there are some independent jobs to be processed in a shop with dedicated renewable resources. Each job consists of several tasks with no precedence restriction. Each task is processed without preemption. The processing time of the tasks is given. Processing each task requires using some multiple specified types of resource, while no more than one task can use each resource, simultaneously. Some tasks can be shared by more than one job and the process may be repeated more than once. The objective is to develop a schedule which yields the minimal makespan length of all jobs, as well as the number of cycles. The model is first developed for cases when the processing time of each task is one unit and, then, it is generalized by relaxing this restriction. In both cases, a circular coloring formulation is shown in comparison with traditional formulation (single process execution) results in an improved makespan and also the required information regarding the optimum number of cycles to repeat the process.

INTRODUCTION

The objective of this paper is to introduce the special case of an open shop scheduling problem, in which the process can be repeated more than once. An approach is also developed by applying the concept of circular coloring for modeling a class of discrete optimization problems, such as scheduling and sequencing. In fact, a multi processor open shop scheduling is formulated that can be handled in more than one cycle. It is shown that the makespan obtained by this approach can be less than the makespan of the same problem, when calculated through a single period problem. This problem is also similar to a cyclic scheduling one, except that the number of cycles does not need to be infinite. It is shown that the makespan of the schedule can be decreased significantly, even for an example with the execution of two processing cycles. Circular coloring, an extension of graph coloring, has been studied in recent years and is more powerful in representing some discrete optimization problems.

Although the subject of scheduling and sequencing has been investigated during the last five decades, not many researchers have applied circular coloring for its representation. Due to the complexity of these problems, many researchers are still trying to develop more efficient methods to obtain good solutions.

In shop scheduling problems, there are a set of jobs to be processed on a number of machines. A job consists of several tasks, each to be performed on a specific machine for a given amount of time. Shop scheduling problems can be categorized as flow shops, job shops or open shops, depending on the ordering restrictions of job operations. In a flow shop, each job can have, at most, one operation on each machine and the tasks of all jobs are processed by the same order, i.e. each job passes the machines in the same order. In a job shop, the operations of each job must be processed in a given order, which is specific to that job, although different jobs have different tasks. In an open shop, there is no restriction in the order of job operations. For example, consider an automotive garage with specialized shops. A car may require the following work: Replacement of exhaust pipes and muffler; alignment of wheels and tune up. These three tasks may be carried out in any order, but it is not possible to perform two tasks simultaneously. The

1. *Department of Industrial Engineering, Sharif University of Technology, Tehran, Iran.*
*. *To whom correspondence should be addressed. E-mail: modarres@sharif.edu*

reason is that the exhaust system, alignment and tune up are carried out in different shops (see [1]). One can find many situations where the tasks of a job can be performed in any order, even though it is not possible to execute more than one task simultaneously, due to the nature of the job or the restriction of resources.

This paper addresses the problem of scheduling a generalized version of open shops to minimize the makespan, classified by Graham et al. [2], as a class of $O||C\max$. This scheduling problem is formulated by applying a graph modeling approach, or more specifically, as a circular coloring problem. In this generalized case of an open shop, some tasks need more than one type of resource, simultaneously. For example, in the automotive garage mentioned before, a task needs more than one resource, such as equipment, tools and mechanics, simultaneously. It can be shown that obtaining an optimal schedule is the same as determining an optimal circular coloring of the resulting graph. The solutions, obtained by applying interval versus circular coloring formulation, are also compared through several examples.

In the literature, scheduling problems revolve around a series of techniques, such as mathematical programming, dispatching rules, expert systems, neural networks, genetic algorithms, fuzzy logic and inductive learning (see [3]). In this paper, some literature regarding only open shop scheduling is reviewed briefly. The open shop scheduling model has received considerable research attention, because it occurs in many real world environments (see [4] or [5]).

The problem of minimizing the makespan of an open shop with n jobs and m machines, where every job consists of several tasks and each task must be processed on a different machine for a given amount of time, is considered by many researchers (see [1,5-25]). They developed suitable algorithms to solve the problem for a variety of cases, such as for a different number of machines or for different processing times. However, they considered the problem with only one execution and each task needs only one resource for execution. Taillard [20] used a tabu search to solve many open shop problems for open shop scheduling problems, which are used as a benchmark in the literature.

The only paper in the literature addressing open shops by application of graph coloring is one by Kubale and Nadoski [26]. However, they considered a cyclic open shop, in which the process is repeated infinitely and each task needs only one resource. They also showed that the problem of cyclic scheduling in open shop scheduling is NP-hard for a 3-processor system and polynomial for a 2-processor one. They proved that if a given open shop can be scheduled in 3 time units, then it is polynomial solvable. They also proved that in a compact open shop, the problem of minimizing $C_{\max}$ for a 2-processor to determine the existence of a legal schedule is also NP-hard.

This research is distinguished from previous ones from the point of view of problem definition, as well as formulation. In the authors' model, each task may need multiple resources. Furthermore, it is the special case of a cyclic open shop, in which the number of cycles can be limited and not necessarily infinite. To solve this model, an approach has been developed, based on circular coloring formulation.

This paper is organized as follows: First, the problem is defined, then, the definitions are reviewed, as well as some properties of graph and circular coloring. After that, the graphical model is developed, which represents multiprocessor open shop scheduling, and finally, some illustrative examples are presented.

PROBLEM DEFINITION

A multi processor open shop scheduling model with dedicated resources is considered, which can be executed in more than one cycle. The number of cycles is a positive integer. In each cycle, there are n independent jobs to be processed and each job consists of several tasks. Due to the nature of open shops, no more than one task can be executed at any time and no precedence relation exists between the tasks of a job. A task may belong to more than one job (such as the operation of a furnace, which can handle more than one task). Each task needs a set of resources to be executed and its processing takes some specified time. During the process of a task, no interruption is allowed.

There are r types of dedicated renewable resources. Each type contains a number of non-identical resources available to process the jobs. For example, one type of resource can be the set of machines in a shop. However, the machines are not the same and are used for different purposes. To process a task, some of these resources are needed simultaneously. However, at any time, more than one task cannot use the same resource.

The following notation is considered:

J:	Set of jobs,
S:	Set of tasks,
$\Re$:	Set of dedicated renewable resources,
$S(T) \subset S$:	Set of tasks of job $T \in J$,
$S(R) \subset S$:	Set of tasks that requires resource $R \in \Re$,
$R(s) \in \Re$:	Set of resources needed to execute task $s \in S$
t_s:	Initial point of the time interval of task $s \in S$ (It is the decision variable and determined by the model.)

As mentioned above, some tasks may be shared by more than one job. In other words, it is possible that $S(T_i) \cap S(T_j) \neq \varnothing$.

The objective is to determine the minimum number of cycles and a proper schedule of tasks and, consequently, the schedule of jobs to be executed, in order to minimize the makespan (the total length of processing) for all jobs. The problem of open shop scheduling is known to be NP-hard (see [27]). It is obvious that this problem is a generalized version of open shop scheduling.

DEFINITION AND BASIC PROPERTIES OF CIRCULAR CHROMATIC NUMBER

Since the authors' approach to formulate multi processor-multi period open shop scheduling problem is by applying circular coloring, in this section, the definition, concept and some of its basic properties are briefly reviewed. However, since it is a generalization of graph coloring, this classical topic is also defined and the relation between them is then discussed.

Graph Coloring

Definition 1

Consider a graph $G(V, A)$ with a vertex set $V(G)$ and an edge set $E(G)$ then, k-coloring of this graph is a labeling, $f : V(G) \to S$ where $|S| = k$. The labels are the colors of vertices, such that adjacent vertices have different labels. The chromatic number, $\chi(G)$, is the least k, such that G is k-colorable.

In a similar definition for chromatic number $\chi(G)$, the vertices are assigned unit intervals, rather than colors. The definition of r-interval coloring is as follows.

Definition 2

An r-interval coloring of a graph G is a mapping, g, which assigns each vertex, $x \in V(G)$, a unit length open sub-interval of interval $[0, r]$, such that the sub-intervals of adjacent vertices are disjoint. The chromatic number of a graph, denoted by $\chi(G)$, is defined as:

$$\chi(G) = \text{Inf}\ \{r : G \text{ is } r\text{-interval colorable}\}.$$

Similarly, an r-interval coloring of G corresponds to a mapping, f, from $V(G)$ to $[0, r)$, such that $1 \leq |f(x) - f(y)| \leq r - 1$ for every edge, $(x, y) \in G$. Furthermore, $f(x) \leq r - 1$, for all $x \in V(G)$.

Circular Coloring

The circular chromatic number, $\chi_c(G)$, of graph G is a generalization of the chromatic number of a graph. It was introduced in 1988 by Vince [28] as "the star-chromatic number". The mathematical definition of circular coloring is as follows.

Definition 3

Let C be a circle (Euclidean) of length r. An r-circular coloring of graph G is a mapping, c, which assigns each vertex, $x \in V(G)$, an open unit length arc of C, say $c(x)$, such that, for every edge, $(x, y) \in G$, $c(x) \frown c(y) = \varnothing$. It is said that graph G is r-circular colorable, if there is an r-circular coloring of G. The circular chromatic number of a graph, denoted by $\chi_c(G)$, is defined as:

$$\chi_c(G) = \text{Inf}\ \{r : G \text{ is } r\text{-circular colorable}\}.$$

The alternate definition of circular coloring is presented by Vince [28], as follows.

Definition 4

For two integers, $1 \leq d \leq k$, $a(k, d)$, coloring of graph G is a coloring of the vertices of G with colors $\{0, 1, \cdots, k-1\}$, such that:

$$(x, y) \in E(G) \Rightarrow d \leq |c(x) - c(y)| \leq k - d.$$

The circular chromatic number is defined as:

$$\chi_c(G) = \text{Inf}\ \{k/d : G \text{ has } a(k, d) \text{ coloring}\}.$$

On the other hand, it can be proved that $\chi_c(G)$ is rational. To determine the circular chromatic number, $\chi_c(G)$, of a finite graph, G, it suffices to determine whether G is r-circular colorable or not, for each of those rational numbers, $r = p/q$, for which $q \leq \alpha(G)$ and $p \leq circumference(G) \leq |V(G)|$. $\alpha(G)$ is the maximum size of an independent set of vertices and circumference (G) is the length of the longest cycle in G. Consequently, a finite number of rational numbers must be examined to determine $\chi_c(G)$.

Relation Between the Chromatic Number of Graph and Circular Coloring

Lemma 1

For each graph, the following relation exists between chromatic and circular chromatic numbers:

$$\chi(G) - 1 < \chi_c(G) \leq \chi(G). \tag{1}$$

Proof. See [29].

Corollary 1

a) The chromatic number of a finite graph can be obtained, if its circular chromatic number is given. It is the minimum integer number, greater than, or equal to, its circular chromatic number;

b) Two finite graphs with the same chromatic number may have different circular chromatic numbers.

Case of Equal $\chi(G)$ and $\chi_c(G)$

The important question is under what conditions $\chi_c(G) = \chi(G)$ for a graph of G. This question was raised by Vince [28] and investigated in many other papers (see [29]). Guichard [30] proved that it is NP-hard to determine whether or not an arbitrary graph, G, satisfies $\chi_c(G) = \chi(G)$.

In open shop scheduling, if each task requires only one resource, like classical open shops, then it can be formulated as the edge coloring of a bipartite graph, G, with a two disjoint vertex set, V_1, V_2, such that each edge corresponds to one task and connects a vertex in V_1 (set of jobs) to a vertex in V_2 (set of resources). So, it is obvious that $\chi_c(G) = \chi(G) = \Delta$, where Δ is the maximum degree of vertices in $V(G)$.

Besides the chromatic number of a graph, there are many other graph parameters for which the relation with the circular chromatic number has been investigated. Among these parameters are: The fractional chromatic number, the clique number, the maximum degree, the girth and the connectivity. These relations are investigated by Zhu [29] and some other researchers.

Weighted Graph

Definition 5

A weighted graph, (G, w), is a graph, $G = (V, E)$, with a vertex set and an edge set, $E(G)$, and a weight function, $w : V \to [0, \infty)$.

Definition 6

An r interval coloring of weighted graph (G, w) is a mapping, Γ, of vertices of G to open sub-intervals of $[0, r]$, such that for each vertex, x, the length of $\Gamma(x)$ is equal to $w(x)$ and adjacent vertices are mapped to disjoint sub-intervals.

The interval chromatic number, $\chi(G, w)$, of weighted graph (G, w) is the minimum number, r, for which there is an r-interval coloring of G.

Definition 7

An r circular coloring of weighted graph G is a mapping, Γ, of the vertices of G to the open arc of a Euclidean circle, C, of length r, such that:

i) $\Gamma(x)$ and $\Gamma(y)$ are disjoint, if $(x, y) \in E(G)$;

ii) The length of $\Gamma(x)$is at least $w(x)$ for all vertices, $x \in V$.

The circular-chromatic number, $\chi_c(G, w)$, of weighted graph (G, w) is $\chi_c(G, w) = \text{Inf } \{r :$ there is an r circular coloring of $(G, w)\}$.

From this definition, it is obvious that whenever the weight function takes constant value 1, then, $\chi_c(G) = \chi_c(G, w)$ [31]. Dauber and Zhu [32] proved that $\chi_c(G, w)$ can always be attained.

FORMULATION OF THE SCHEDULING PROBLEM

In this section, the concept of circular coloring is applied to formulate the scheduling problem described previously. First, it is assumed that the processing time of each task is exactly one time unit. Then, this assumption is relaxed.

Although both interval coloring and circular coloring can be applied to model this problem, the latter may result in a better solution if the process is executed more than once. Thus, this scheduling problem is formulated as a circular coloring problem and it is shown that obtaining an optimal schedule is the same as determining an optimal circular coloring of the graph. The solutions obtained are also compared by applying interval versus circular coloring formulation through several examples.

Definition 8

A pair of tasks is called incompatible, if one of the following relations holds:

- They belong to the same job;
- They share the same resource.

Considering the above definition, the following notation is considered:

I(S): Set of tasks incompatible with $s \in S$.

Let $s \in S$ be a task belonging to job T, or by the authors' notation, $s \in S(t)$. If it needs some resources, $R(s) \in \Re$, then by the authors' notation, $s' \in S$ is incompatible with s, if either $s' \in S(T)$ or $R(s) \frown R(s') \neq \varnothing$.

Graph Model of the Problem

Let a graph of $G = (V, E)$, with a vertex set, $V(G)$, and an edge set, $E(G)$, represent the scheduling problem of the multi processor open shop, defined previously. Each vertex of this graph indicates one task. In this graph, incompatible tasks must be assigned disjoint intervals. In other words, they cannot be processed simultaneously. Therefore, if one considers each task as a vertex, $v \in V(G)$, in this graph, then it is connected to all of its incompatible tasks. In fact, there is an edge, $(x, y) \in E(G)$, between any pair of vertices, x, $y \in V(G)$, if and only if, $x \in I(y)$ or, similarly, $y \in I(x)$. This means two adjacent vertices of this graph are associated with a pair of incompatible tasks and are connected by an edge. In other words, if two vertices are not connected by an edge, it means they are compatible tasks and can be processed simultaneously.

It is obvious that any interval or circular coloring of this graph results in a feasible solution for the scheduling problem, due to the fact that a coloring solution prevents two adjacent vertices from having an overlapping time interval. In other words, it forces the processing periods of two incompatible tasks to be disjoint. Thus, it seems natural to solve this problem by determining the chromatic number of the corresponding graph, which is actually the required time to complete the jobs.

As mentioned before, the objective is not only finding an optimal schedule but also determining the minimum number of cycles in order to minimize the makespan.

In case the chromatic number of G, which represents the problem, is equal to its circular chromatic number, then scheduling for more than one cycle does not decrease the makespan. But, if they are not equal, then there exists a repeating schedule, such that its makespan is equal to circular chromatic. Since the inequality of $\chi_c(G) \leq \chi(G)$ can be strict, the chromatic number approach may not provide the optimal solution to this scheduling problem. It is also shown that the number of repeated cycles required to gain an improved makespan is limited.

Interval Coloring Formulation

There are various methods to obtain the interval or circular coloring solution of a graph. A mathematical model is formulated to derive the coloring solution of a graph.

An interval of length r is considered and a unit length subinterval is assigned to each vertex, by considering that adjacent vertices must be disjoint.

Let t_s denotes the decision variable for the coloring model representing the initial point of the subinterval assigned to vertex s. As a result, the following mathematical model represents the interval coloring formulation of the scheduling problem:

$$\min Z = r,$$

S.t.

$$1 \leq |t_s - t_{s'}| \leq r-1, \qquad s' \in I(s), \quad s \in S, \tag{2}$$

$$0 \leq t_s \leq r-1, \qquad s \in S. \tag{3}$$

This mixed integer programming has $(|V(G)|+1)$ variables and $2(|E(G)|+|V(G)|)$ constraints. The objective function minimizes the interval, in which G is r colorable. Starting task s at t_s, results in a feasible schedule with a makespan that is equal to r.

Circular Coloring Formulation

Now, a circle with length r is considered and a unit length arc is assigned to every vertex while considering the adjacency condition. The resulting mathematical model is the same as that of interval coloring, except that Relation 3 is replaced with the following relation:

$$0 \leq t_s \leq r, \qquad s \in S. \tag{4}$$

This mixed integer program is similar to the last one. In this case, the makespan is equal to r, if the schedule is executed repeatedly.

Interval Coloring of Weighted Graph

Now, the assumption that the processing time of each task is one unit is relaxed. In that case, vertex s of graph G has a label, named weight, which is equal to the processing time of task s. Thus, the problem is represented by a weighted graph. Obviously, in this graph, each feasible schedule is equivalent to an interval coloring of weighted graph G. Let w_j be the processing time of task s_j or the weight for vertex s_j. The following formulation of interval coloring can be converted into a mixed integer programming:

$$\min Z = r,$$

$$\begin{cases} r - w_i \geq t_{s_i}, & s_i \in S \\ t_{s_i} - t_{s_j} \geq w_i, & \text{if}: t_{s_j} \geq t_{s_j},\ s_i \in I(s_j) \\ t_{s_j} - t_{s_i} \geq w_j, & \text{if}: t_{s_i} \geq t_{s_i},\ s_i \in I(s_j) \end{cases} \tag{5}$$

Circular Coloring of Weighted Graph

Now, a circle with length r is considered and an arc is assigned to every vertex, s_i, with length w_i, considering the adjacency condition:

$$\min Z = r,$$

$$\begin{cases} r \geq t_{s_i}, & s_i \in S \\ \left.\begin{array}{l} t_{s_i} - t_{s_j} \geq w_i \\ t_{s_i} - t_{s_j} \leq r - w_j \end{array}\right\} & \text{if},\ t_{s_i} \geq t_{s_j},\ s_i \in I(s_j) \\ \left.\begin{array}{l} t_{s_j} - t_{s_i} \geq w_j \\ t_{s_j} - t_{s_i} \leq r - w_i \end{array}\right\} & \text{if},\ t_{s_j} \geq t_{s_i},\ s_i \in I(s_j) \end{cases} \tag{6}$$

ILLUSTRATIVE EXAMPLES

In this section, the proposed approach is illustrated through some examples.

Example 1

Consider an open shop scheduling problem with three jobs of $J = \{T_1, T_2, T_3\}$ and five tasks of $S = \{s_1, s_2, s_3, s_4, s_5\}$. The processing time of each task is equal to one unit. On the other hand, there are three different machines, $R_1 = \{A, B, C\}$ and operators, $R_2 = \{K, L, M\}$. Thus, the set of resources is $\Re = \{A, B, C, K, L, M\}$.

Job T_1 consists of tasks s_1 and s_2 or, by the authors' notation, $S(T_1) = \{s_1, s_2\}$. Similarly, $S(T_2) = \{s_3, s_4\}$ and $S(T_3) = \{s_5\}$. Task s_1 must be processed on machine A under supervising operator K or, by the authors' notation, $R(s_1) = \{A, K\}$. Similarly, $R(s_2) = \{B, L\}$, $R(s_3) = \{B, L\}$, $R(s_4) = \{C, M\}$ and $R(s_5) = \{C, K\}$.

There are no precedent constraints. Find a schedule of jobs to be executed such that the total scheduled length is minimized when the process is executed once or more.

The graph coloring concept is applied to solve this problem. The graph representing this problem is shown in Figure 1. Task s_j is represented by vertex s_j, for $j = 1, 2, \cdots, 5$ of graph G. Incompatible tasks are associated with adjacent vertices in this graph. For example, there is an edge between s_1 and s_2, because they belong to job T_1. There is also an edge between s_5 and s_1, because they need common resource K. Both interval and circular coloring concepts are applied to obtain the solution and then the answers are compared.

Interval coloring formulation determines the initial point of the subinterval of each vertex in graph G, or in fact, the starting time of processing the corresponding tasks in the scheduling problem. Table 1 and Figure 2 represent the schedule of all tasks, if interval coloring formulation is applied.

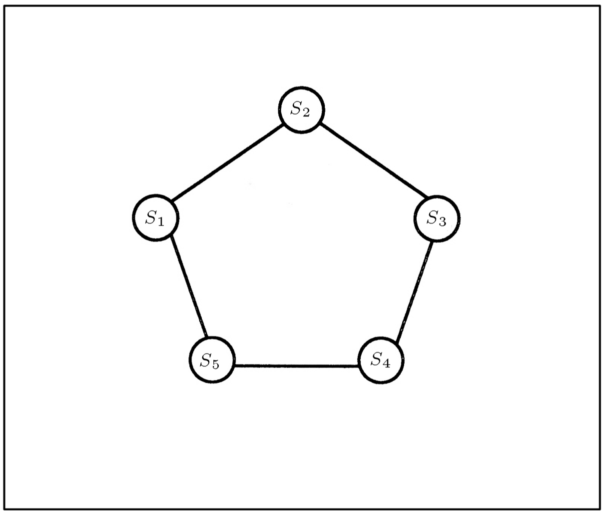

Figure 1. Graphical representation of Example 1.

Table 1. Starting time of tasks of Example 1 by interval coloring.

i	1	2	3	4	5
t_{s_i}	0	1	0	1	2

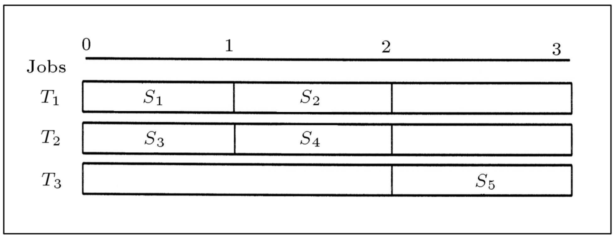

Figure 2. Schedule of tasks of Example 1 by interval coloring.

If r is determined by Relations 2 and 3, then $r = 3$. On the other hand, if circular coloring (mathematical model consists of Relations 2 and 4) formulation is applied, then the makespan reduces to $r = 2.5$ and the minimum number of times that the process has to be repeated is 2. Actually, it is the minimum multiplier of the integer that makes it integer. Table 2 and Figure 3 represent the schedule of the tasks, if the circular coloring formulation is applied.

When the process is executed once, then the makespan is equal for both interval and circular coloring. Therefore, this method is helpful when the process is repeated more than once to achieve a better schedule with a lower makespan.

Following circular coloring formulation (when the process is executed repeatedly), the makespan reduces to 2.5.

If the process executes in a finite number of repeats, the data in Table 3 can be used to develop a schedule with a makespan of exactly 2.5. In the following schedule (see Figure 4), the order of tasks for

Table 2. Starting time of tasks of Example 1 by circular coloring.

i	1	2	3	4	5
t_{s_i}	0	1	2	0.5	1.5

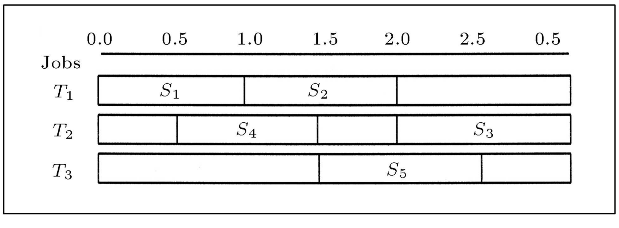

Figure 3. Schedule of tasks of Example 1 by circular coloring.

Table 3. Processing time of tasks of Example 2.

i	1	2	3	4	5	6	7
w_i	1	1	1	1	1.5	0.5	0.5

each process is the same as the previous one. Actually, in this schedule, the gap between the tasks is removed and compacted to improve the makespan.

In this schedule, the makespan is minimized and the total processing time is reduced to 2.5, provided the number of times this sequence is repeated be an even number.

In the following example, a situation is considered in which the processing times are different and not necessarily equal to one.

Example 2

Consider an open shop scheduling problem with four jobs of $J = \{T_1, T_2, T_3, T_4\}$ and seven tasks of $S = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7\}$. On the other hand, there are three different machines, $R_1 = \{A, B, C\}$ and three operators, $R_2 = \{K, L, M\}$. Thus, the set of resources are $R = \{A, B, C, K, L, M\}$.

Job T_1 consists of tasks s_1 and s_2, or by the authors' notation $S(T_1) = \{s_1, s_2\}$. Similarly, $S(T_2) = \{s_3, s_4\}$, $S(T_3) = \{s_5\}$ and $S(T_4) = \{s_6, s_7\}$. Task 1 must be processed on machine A under supervising operator K, or by the authors' notation.

Similarly, $R(s_2) = \{B, L\}$, $R(s_3) = \{B, L\}$, $R(s_4) = \{C, M\}$, $R(s_5) = \{C, K\}$, $R(s_6) = \{B, L\}$, $R(s_7) = \{A, M\}$ and $R(s_1) = \{A, K\}$.

In this example, the processing time of the tasks is not equal, as shown in Table 3.

Similar to Example 1, there is no precedence assumption. Again, the objective is to find a schedule of executing the tasks such that the total processing time is minimized, when the process is executed once or when it is repeated.

The graph which represents this problem is shown in Figure 5. Incompatible tasks are associated with adjacent vertices in this graph. Both interval and circular coloring are applied to obtain the solution and then the answers are compared.

By formulating the problem as an interval color-

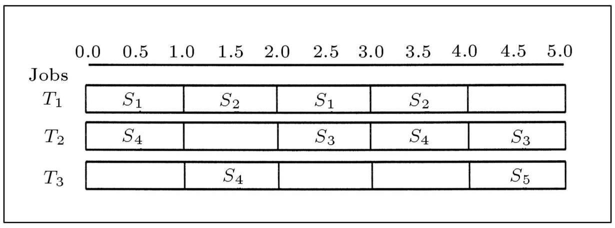

Figure 4. Schedule of tasks for two repeats of Example 1 by circular coloring.

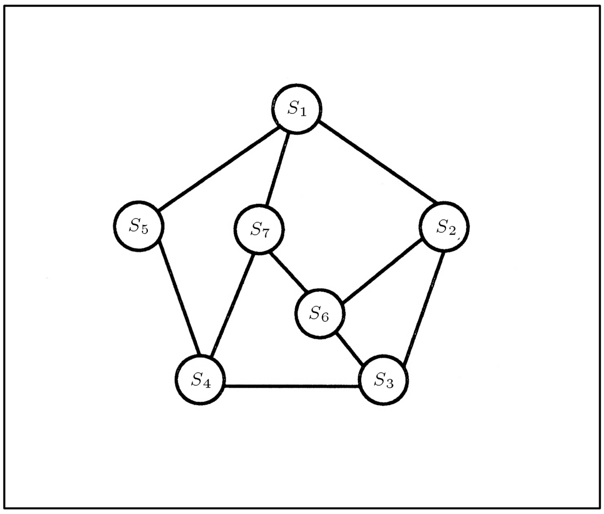

Figure 5. Graphical representation of Example 2.

ing, the tasks are executed within 3 units. The schedule of tasks is shown in Table 4 and Figure 6. However, by applying circular coloring, it is possible to decrease the makespan to 2.5.

Similarly, from the circular coloring formulation of Relation 6, the schedule is as in Table 5, as well as in Figure 7.

In this case, to find the minimum number of cycles, considering the multiple properties is not applicable anymore, because the processing times are not equal numbers.

By using circular coloring formulation, a schedule can be developed (given in Figure 8), in which the number of cycles is 2 and the makespan is 2.75.

According to these examples, a schedule in which the makespan is equal to the circular chromatic number of graph G can be determined. If the processing time of all tasks is the same, then the number of cycles is the least multiplier that makes that makespan an integer. The schedule is obtained from the mixed integer programming, which is developed by circular formulation first and which can then be compacted.

Table 4. Starting time of tasks of Example 2 by interval coloring.

i	1	2	3	4	5	6	7
t_{s_i}	1.5	0	1	2	0	2	1.5

Table 5. Starting time of tasks of Example 2 by circular coloring.

i	1	2	3	4	5	6	7
t_{s_i}	0	1	2	0.25	1.25	0.25	1.25

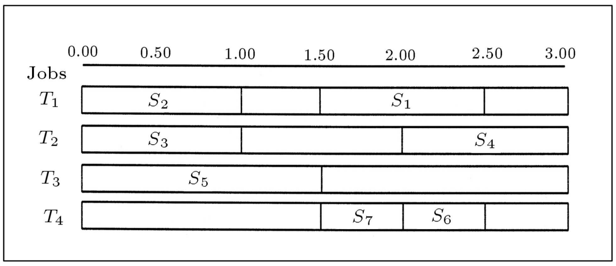

Figure 6. Schedule of tasks of Example 2 by interval coloring.

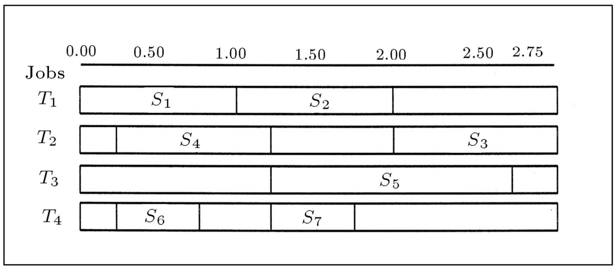

Figure 7. Schedule of tasks of Example 2 by circular coloring.

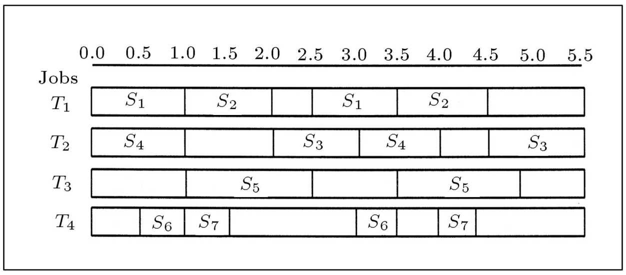

Figure 8. Improved schedule of tasks of Example 2 by circular coloring.

CONCLUSION

In this paper, a scheduling problem is introduced that can be considered a generalization of open shop scheduling. Also, a method based on circular coloring is developed to improve the makespan in situations when the process repeats. The objective is to minimize the total makespan. In this model, independent jobs are processed in an open shop. Each job consists of several tasks and each task needs a set of dedicated resources. Actually, one also can determine the minimum number of repetitions of jobs required, in order to minimize the makespan. This approach can be extended for many other problems in the real world. In fact, circular coloring models are capable of improving the results of these cases. For further research, some metaheuristic algorithms; such as Ant Colony Optimization (ACO), can be developed for problems of a larger size. It is also possible to extend this research by hybridization of the ACO algorithm with other analytical or efficient local search methods to improve its precision and speed.

REFERENCES

1. Gonzalez, T. and Sahni, S. "Open shop scheduling to minimize finish time", *Journal of the Association for Computing Machinery*, **23**(4), pp 665-679 (1976).
2. Graham, R.L., Lawler, E.L., Lenstra, J.K. and Rinnooy Kan, A.H.G. "Optimization and approximation in deterministic sequencing and scheduling: A survey", *Annals of Discrete Mathematics*, **5**, pp 287-326 (1979).
3. Jones, A. and Rabelo, L.C. "Survey of job shop scheduling techniques", *Tech. Rep., National Institute of Standards and Technology*, Gaithersburg, MD (1998).
4. Brucker, P., *Scheduling Algorithms*, Berlin Heidelberg, Springer (1995).
5. Pinedo, M., *Scheduling: Theory, Algorithms and Systems*, Englewood cliffs, NJ, Prentice Hall (1995).
6. Strusevich, V.A. "Two machine open shop scheduling problem with setup, processing and removal times separated", *Computers and Operations Research*, **20**(6), pp 597-611 (1993).
7. Brucker, P., Hurink, J., Jurish, B. and Wostmann, B. "A branch and bound algorithms for the open shop problem", *Discrete Applied Mathematics*, **76**, pp 43-59 (1997).
8. Gueret, C., Jussien, N. and Prince, C. "Using intelligent backtracking to improve branch-and-bound methods: An application to open shop problems", *European Journal of Operational Research*, **127**, pp 344-354 (2000).
9. Dorndorf, U., Pesch, E. and Phan Huy, T. "Solving the open shop scheduling problem", *Journal of Scheduling*, **4**, pp 157-174 (2001).
10. Fiala, T. "An algorithm for the open-shop problem", *Mathematics of Operations Research*, **8**(1), pp 100-109 (1983).
11. Adiri, I. and Aizikowitz, N. "Open shop scheduling problems with dominated machines", *Naval Research Logistics Quarterly*, **36**, pp 273-281 (1989).
12. Strusevich, V.A. "Dominating machines in deterministic systems, in applications of mathematical methods and computer engineering for solving problems of national economy", *Proceedings of Republican Meeting*, Minsk, pp 151-152 (1986).
13. Tanaev, V.S., Sotskov, Y.N. and Strusevich, V.A., *Scheduling Theory: Multi-Stage Systems*, Kluwer Academic Publishers, Dordrecht (1994).
14. Rock, H. and Schmidt, G. "Machine aggregation heuristics in shop-scheduling", *Methods of Operations Research*, **45**, pp 303-314 (1983).
15. Aksjonov, V.A. "A polynomial-time algorithm of approximate solution of a scheduling problem", *Upravlyaemye Sistemy*, **28**, pp 8-11 (1998).

16. Shmoys, D.B., Stein, C. and Wein, J. "Improved approximation algorithms for the shop scheduling problems", In *Proceedings of the Second ACM-SIAM Symposium on Discrete Algorithms*, pp 148-157 (1991).

17. Chen, B. and Strusevich, V.A. "Approximation algorithms for three-machine open shop scheduling", *ORSA Journal on Computing*, **5**, pp 321-328 (1993).

18. Brasel, H., Tautenhahn, T. and Werner, F. "Constructive heuristic algorithms for the open shop problem", *Computing*, **51**, pp 95-110 (1993).

19. Gueret, C. and Prins, C. "Classical and new heuristics for the open-shop problem: A computational evaluation", *Europian Journal of Operatioal Research*, **107**, pp 306-314 (1997).

20. Taillard, E. "Benchmarks for basic scheduling problems", *European Journal of Operation Research*, **6**, pp 278-283 (1993).

21. Liaw, C.F. "An iterative improvement approach for the non preemptive open shop scheduling problem", *European Journal of Operation Research*, **111**, pp 509-517 (1998).

22. Liaw, C.F. "An efficient tabu search approach for the two-machine preemptive open shop scheduling problem", *Computers and Operations Research*, **30**, pp 2081-2095 (2003).

23. Liaw, C.F. "A hybrid genetic algorithm for the open shop scheduling problem", *European Journal of Operation Research*, **124**, pp 28-42 (2000).

24. Schurman, P. and Woeginger, G.J. "Approximation algorithms for the multiprocessor open shop scheduling problem", *Memorandum COSOR*, Eindhoven University of Technology, pp 97-23 (1997).

25. Blum, C. "Beam-ACO- hybridizing ant colony optimization with beam search: An application to open shop scheduling", *Computers and Operations Research*, **32**(6), pp 1565-1591 (2003).

26. Kubale, M. and Nadolski, A. "Chromatic scheduleng in a cyclic open shop", *European Journal of Operation Research*, **164**(99), pp 585-591 (2005).

27. Garey, M.R. and Johnson, D.S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco (1979).

28. Vince, A. "Star chromatic number", *Journal of Graph Theory*, **12**, pp 551-559 (1988).

29. Zhu, X. "Circular chromatic index: A survey", *Discrete Mathematics*, **229**, pp 371-410 (2001).

30. Guichard, D.R. "A cyclic graph coloring and complexity of star chromatic number", *Journal of Graph Theory*, **17**, pp 129-134 (1993).

31. Zhu, X. "Star-chromatic numbers and products of graphs", *Journal of Graph Theory*, **16**, pp 557-569 (1992).

32. Deuber, W. and Zhu, X. "Circular coloring of weighted graphs", *Journal of Graph Theory*, **23**, pp 365-376 (1997).