No-Wait & No-Idle Open Shop Minimum Makespan Scheduling with Bioperational Jobs

Michał MAŁAFIEJSKI[1], Krzysztof OCETKIEWICZ[2], Krzysztof PASTUSZAK[3]

[1]*Gdansk University of Technology, Faculty of Electronics, Telecommunication and Informatics, G.Narutowicza 11/12, Gdańsk, Poland, e-mail: michal@animima.org*

[2]*Gdansk University of Technology, Faculty of Electronics, Telecommunication and Informatics, G.Narutowicza 11/12, Gdańsk, Poland, e-mail: krzysztof.ocetkiewicz@eti.pg.gda.pl*

[3]*Gdansk University of Technology, Faculty of Electronics, Telecommunication and Informatics, G.Narutowicza 11/12, Gdańsk, Poland, e-mail: krzysztof_pastuszak@yahoo.com*

Abstract: In the open shop scheduling with bioperational jobs each job consists of two unit operations with a delay between the end of the first operation and the beginning of the second one. No-wait requirement enforces that the delay between operations is equal to 0. No-idle means that there is no idle time on any machine. We model this problem by the interval incidentor (1, 1)-coloring (IIR(1, 1)-coloring) of a graph with the minimum number of colors which was introduced and researched extensively by Pyatkin and Vizing. An incidentor is a pair (v, e), where v is a vertex and e is an edge incident to v. In the incidentor coloring of a graph the colors of incidentors at the same vertex must differ. The interval incidentor (1, 1)-coloring is a restriction of the incidentor coloring by two additional requirements:colors at any vertex form an interval of integers and the colors of incidentors of the same edge differ exactly by one. In the paper we proposed the polynomial time algorithm solving the problem of IIR(1, 1)-coloring for graphs with degree bounded by 4, i.e., we solved the problem of minimum makespan open shop scheduling of bioperational jobs with no-wait & no-idle requirements with the restriction that each machine handles at most 4 job.

Keywords: graph theory, interval coloring, consecutive coloring, one-sided interval, incidentor, shop scheduling

1. OPEN SHOP SCHEDULING WITH BIOPERATIONAL JOBS

The open shop scheduling problem is given by a set of machines and jobs and their restrictions. Let $M = \{M_1, M_2... M_m\}$ be a set of machines, and $J = \{J_1, J_2... J_n\}$ be a set of jobs (also called tasks). Each task consists of distinct operations $J_j = \{O_{1j}, O_{2j}, ...\}$, each of them being assigned to a distinct machine. Each machine may process at most one operation at any given time. At most one operation of each task may be processed at any given time. Operations within a task may be processed in any order. Each operation has a certain processing time assigned, denoted by p_{ij} for O_{ij} operation of task J_j, executed by machine M_i. The problem of open shop scheduling is one of the classic scheduling theory problems, first introduced by Gonzalez and Sahni in 1976 [Gonzalez 1976]. In the three-field notation, first introduced in 1982 by Graham et al.[Graham 1982], open shop is denoted by O in the α field.

Additional constraints are often imposed on the open shop scheduling problems [Giaro 2003], i.e., restricted delays between execution of subsequent operations within a job, availability of resources, and restrictions on the space of considered instances, i.e., limited number of machines [Giaro 2003] or UET only operations. In general, open shop scheduling is NP-hard, even when restricted to UET only operations [Giaro 2003]. Introducing restrictions on the space of the instances may allow us to construct polynomial time exact algorithms for certain subclasses of the open shop scheduling problem. In the open shop with bioperational tasks each job consists of exactly two operations. An example could be a scenario with read and write mode operations, which cannot be executed concurrently, e.g., in databases. We will denote this constraint on the number of operations within a job J_j by $op_j = 2$ in the β field. Assuming UET only operations, we will denote restriction on the maximum load per machine, i.e., the number of operations executed by machine, by load $\leq k$, for load no greater than k. An instance of open shop problem with bioperational tasks may be modeled by a graph, where machines are represented by vertices and bioperational jobs by edges.

Common constraints on feasible solution space include no-wait and no-idle restrictions [Giaro 2003]. No-wait requirement enforces that the delay between operations within a job is equal to 0. No-idle means that there is no idle time on any machine, once the machine started working. We will denote no-wait and no-idle restriction by NWI in the β field in three-field notation.

Open shop with bioperational tasks and UET operations, and with no-wait and no-idle restrictions is considered in this paper. The problem may be solved by computing an incidentor coloring of a graph that models the instance of a problem.

2. INCIDENTOR COLORING

The notion of incidentors was first introduced by A.A. Zykov in 60s [Pyatkin 1997, Pyatkin 2002]. The incidentor coloring model however remained unknown until the 90s, when A.V. Pyatkin and V.G. Vizing focused their research on the model and its applications in the network transmission and scheduling theory [Pyatkin 2002, Pyatkin 2006, Vizing 2009, Vizing 2012, Vizing 2007, Vizing 2014, Pyatkin 2015]. The incidentor coloring model arises from scheduling data transmission in the communication networks [Pyatkin 1997, Pyatkin 2002, Pyatkin 1999] and was first [Pyatkin 2002] introduced in 1997 paper [Pyatkin 1997] by Pyatkin and later described in greater detail in his Ph.D. Dissertation [Pyatkin 1999].

Let G = (V, E) be a graph. Incidentor is a pair (v, e), where $e \in E(G)$ is an edge incident to $v \in V(G)$. Each edge $e = \{v_i, v_j\}$, where $v_i, v_j \in V(G)$, may be represented by a pair of incidentors: (v_i, e) and (v_j, e). Two incidentors (v_i, e) and (v_j, f) are called mated if e = f, where $e, f \in E(G)$, and are called adjacent if $v_i = v_j$ [Vizing 2000].

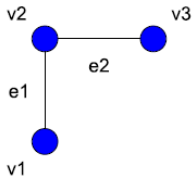

Figure 1: There are four incidentors in the graph: (v_1, e_1), (v_2, e_1), (v_2, e_2), (v_3, e_2). Incidentors (v_1, e_1) and (v_2, e_1) are mated. Incidentors (v_2, e_1) and (v_2, e_2) are adjacent.

The problem of incidentor coloring of graph is to assign the color to each incidentor, such that for each pair of adjacent incidentors their assigned colors differ. Additional constraints on the colors of mated incidentors may be introduced [Pyatkin 2013, Pyatkin 2004].

These constraints provide utility, that may be used to model the restrictions on time delays between the subsequently executed operations within a job. For any edge $e \in E(G)$ we may specify lower and upper bounds on the difference between the numbers of colors used. In general, any edge may have different bounds assigned, however most of the research focused on the graphs with the same constraints applied to each edge. The coloring c may be called IR(k, l)-coloring, if for $k \le l$, $k, l \in \mathbb{Z}$, for any pair, (u, e) and (v, e), where $v, u \in V(G)$, $e = \{u, v\} \in E(G)$, the following holds: $k \le |c(u,e) - c(v,e)| \le l$.

It may be also called IR(k, l)-coloring. The minimum number of colors sufficient for IR(k, l)-coloring of a graph G may be denoted by $\chi^{IR}_{(k,l)}(G)$. In particular, IR(0, 0)-coloring is equivalent to the edge coloring. In the IR(1, 1)-coloring the colors of mated incidentors form intervals of integers, hence it may be used to solve the no-wait open shop problem – O | UET, $op_j = 2$, NW | C_{max}. The IR(1, 1)-coloring always exists. If the degree of a graph G is even $(2|\Delta(G))$, then $\Delta(G)$ colors are always sufficient [Pyatkin 2004, Hanson 1998]. If the degree of a graph G is odd, then the problem of deciding, whether $\Delta(G)$ colors are sufficient, remains open [Pyatkin 2004]. For any graph, $\Delta(G) + 1$ colors are sufficient to construct IR(1, 1)-coloring. For any $k \ge \Delta(G) - 1$, $\chi^{IR}_{(k,k)}(G) = \chi^{IR}_{(k,\infty)}(G)$ [Vizing 2003, Małafiejska 2016].

For any k, $\chi^{IR}_{(k,\Delta(G)-1)}(G) = \chi^{IR}_{(k,\infty)}(G) = \max\{\Delta(G), \lceil \frac{\Delta(G)}{2} \rceil + k\}$ [Vizing 2003]. If $k \ge \Delta(G)$ 2, $\chi^{IR}_{(k,k)}(G) = \lceil \frac{\Delta(G)}{2} \rceil + k$ [Vizing 2005]. For a graph G, $\Delta(G)$ colors are sufficient to find both IR(0, 1) and IR(0, ∞) colorings of G [Pyatkin 2004, Melynikov 2000]. In interval incidentor coloring, denoted later by IIR coloring, colors of adjacent incidentors must form interval of integers.

Chromatic scheduling is one of the approaches used in solving open shop scheduling problems [Giaro 2003]. The general idea behind the chromatic scheduling is first to create a graph model of the instance of a problem, then choose the adequate graph coloring model, construct a feasible coloring and construct the schedule from the coloring. Open shop with bioperational tasks may be solved by constructing a graph model, later called incidentor scheduling graph, and then computing incidentor coloring of the graph. Machines are modeled by vertices in the graph. Tasks are modeled by edges between vertices corresponding to the machines, on which constituent operations are executed. Operations are modeled by incidentors. Colors of incidentors correspond to time windows, in which relevant operations are executed. With an additional no-idle restriction, this problem may be modeled by interval incidentor (IIR) coloring. No-wait restriction is modeled by IR(1, 1)-coloring. No-wait and no-idle restriction is modeled by the IIR(1, 1)-coloring. Restrictions of the instance space of open shop scheduling problem, that limit the maximum degree of scheduling graph, may allow construction of polynomial time algorithms, even if the problem is NP-hard in general.

3. INTERVAL INCIDENTOR COLORING OF GRAPHS WITH Δ BOUNDED BY 4

In this section the linear time algorithm for construction of IIR(1, 1)-coloring of graphs with Δ bounded by 4, using 4 colors, is presented. This algorithm may be used to solve the no-wait and no-idle open shop scheduling problem with bioperational jobs and at most 4 operations per machine. The problem can be described in the three-field notation by: $O \mid UET, op_j = 2, load \leq 4, NWI \mid C_{max}$.

Let G = (V, E), with $\Delta(G) \leq 4$, be an incidentor scheduling graph. Consider an edge $e \in E(G)$ from vertex u to vertex v, where $u, v \in V(G)$. The following partial orientation of graph G:

- every vertex of degree 4 is adjacent to exactly two incoming arcs,
- every vertex of degree 3 is adjacent to exactly one incoming arc,
- every vertex of degree 3 or 4 is adjacent to exactly two outgoing arcs or undirected edges,

we call a *legal* partial orientation of graph G. An example of a legal partial orientation is shown in Figure 2.

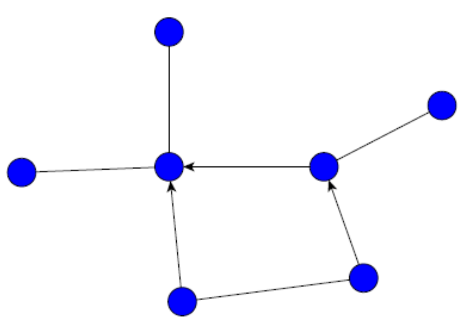

Figure 2: Example of partial orientation D.

Lemma 1. It is possible to construct a legal partial orientation D of the incidentor scheduling graph G, $\Delta(G) \leq 4$, in polynomial time.

Proof. We first create a partial orientation of a graph with some vertices split, then we join split vertices again and obtain a partial orientation D of original graph G. Only vertices of degree 3 or 4 will be split and only when one of the following conditions occurs:

- vertex of degree 3 is adjacent to exactly one incoming arc;
- vertex of degree 4 is adjacent to exactly two incoming arc.

No vertices of degree 3 in G may have more than one adjacent incoming arc, and no vertices of degree 4 in G may have more than two adjacent incoming arcs. We call vertices with these configurations legal. Only legal vertices may be split. Legal configurations of arcs and edges are shown in Fig. 3.

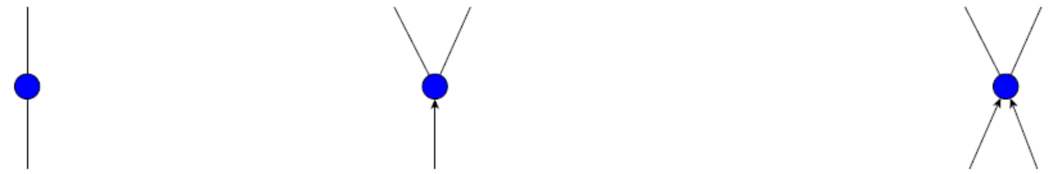

Figure 3: Straight lines represent edges or outgoing arcs. The "legal" combinations of the arcs and edges: left – vertices of degree 2, middle – vertices of degree 3, right – vertices of degree 4.

In the splitting process, vertex v is split into two vertices: one incident only to the incoming arcs (one, if in G, d(v) = 3, and two, if in G, d(v) = 4), and one incident only to two outgoing arcs or undirected edges. The splitting process is shown in Fig. 4.

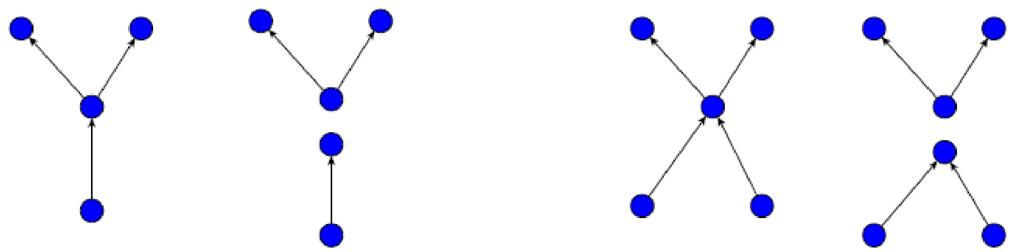

Figure 4: The splitting of vertices of degrees 3 (left) and 4 (right).

The following algorithm may be used to construct the partial orientation D:

In a loop:

1. Detect an undirected cycle C in a subgraph induced by vertices of degree 3 and 4 in the remaining graph.
2. Transform the cycle into a directed cycle.
3. Split vertices of degrees 3 and 4 in the remaining graph, if their configuration is already legal.

Once no more cycles can be found, in a loop:

4. Detect a path of maximal length in the remaining graph, so that a vertex of degree 1 in the remaining graph is the end of the path.
5. Transform the path into a directed path, starting in a vertex of degree 1.
6. Split vertices of degrees 3 and 4 in the remaining graph, if their configuration is already legal.

An example of contraction is shown in Fig. 5.

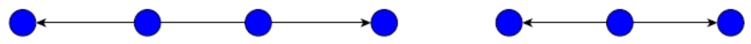

Figure 5: Contraction and reconstruction of the edges in undirected paths: left – original path, right – contracted path.

The algorithm returns a decomposition of G into paths and cycles. After joining split vertices again, we obtain a partial orientation D of original graph G.

Since Δ(G) is bounded by 4, it can be approximated by a constant. Both cycles and paths can be found in linear time, hence construction of D can be done in O(|E|) time.

Each vertex of degree 4 in G is an inner vertex in two paths or cycles. Each vertex of degree 3 is the end of one path, and the inner vertex in one path or cycle. As a result of the construction scheme, there are only the following structures in the decomposition :

- directed cycles, with two incoming arcs and two outgoing arcs, alternately,
- directed paths, with two incoming arcs and two outgoing arcs, alternately,
- undirected cycles,
- undirected paths.

Structures occurring in a decomposition are shown in Fig. 6. Boths ends of each path are vertices of degree 1 in the returned decomposition, of degree either 1, or 3 in G.

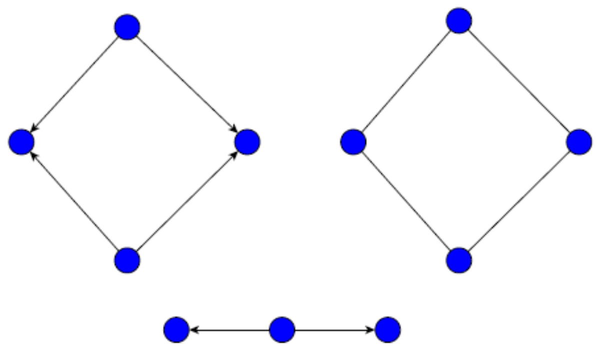

Figure 6: The structures obtained after the decomposition of the graph: left – directed cycle, right – undirected cycles, bottom – paths.

Theorem 1. For any graph G with Δ(G) ≤ 4 there is IR(1, 1)-coloring using at most 4 colors. For any graph G with Δ(G) ≤ 4 there is IIR(1, 1)-coloring using at most 4 colors.

Proof. According to Lemma 1, there always exists partial orientation of G. Incidentors of the incoming arcs may have only colors 1 or 4 assigned. Incidentors of undirected edges or outgoing arcs may assume only colors 2 or 3. Coloring of partial orientation D using 4 colors can be constructed using the following scheme:

1. Construct partial orientation of G. Do not join split vertices yet. Contract undirected paths into single vertices. An example of contraction is shown in Fig. 5.
2. Color undirected cycles using colors 2 and 3, alternately.
3. Color directed paths using colors 2, 1, 4, 3 (or 3, 4, 1, 2), repeatedly.
4. Restore contracted paths and color them, then join split vertices, thus obtaining colored partial orientation D of original graph G.

Transform arcs back into edges, while preserving the coloring from D, thus obtaining IR(1,1) coloring of G.

Lemma 2. IR(1, 1) coloring of G constructed by algorithm from Theorem 1, is also a feasible IIR(1, 1) coloring of G.

Proof. Let us remind, that in the decomposition only the structures shown in Fig. 6 occur, and that each vertex of degree 3 or 4 in G is split into two vertices, one with incoming arcs only, and one with outgoing arc or undirected edges only. Thus, for each pair of split vertices, after joining them again, a vertex with one or two incoming arcs and exactly two outgoing arcs or undirected edges forms. Since incidentors of incoming arcs are always colored with 1 or 4, and incidentors of undirected edges and outgoing arcs are always colored with 2 or 3 (and there are always two of them) colors of adjacent incidentors always form interval: {1, 2, 3} or {2, 3, 4} if the degree of the vertex was equal to 3, and {1, 2, 3, 4} if it was equal to 4. The original structure of G is restored after joining again each pair of split vertices and its incidentor coloring is a feasible IIR(1, 1) 4-coloring of G

Hence the theorem follows:

Theorem 2. Schedule of makespan 4 for the problem O|UET,NWI, $op_j = 2$, load $\leq 4|C_{max}$ always exists and it can be obtained in linear time.

4. SUMMARY

Presented linear time algorithm may be used for construction of schedule of makespan 4 for the problem O|UET,NWI, $op_j = 2$, load $\leq 4|C_{max}$. The problem of no-wait & no-idle open shop scheduling with bioperational jobs with at most k operations per machine remains open for $k \geq 5$. We conjecture, that for $k = 5$ the problem is polynomial, and for $k \geq 7$ is NP-hard. Another open problem is O|UET,NWI, $op_j \leq 3$, load $\leq 4|C_{max}$. Giaro proved both the sufficient and necessary conditions for the existence of schedule with makespan 4, however the problem of existence of schedule with makespan $C_{max} \in \{5,6\}$ remains open.

5. REFERENCES

[1] Giaro, K. (2003). *Task scheduling by graph coloring (in Polish).* D.Sc. Dissertation. Gdansk University of Technology.

[2] Giaro, K. (1997). *The complexity of consecutive D-coloring of bipartite graphs: 4 is easy, 5 is hard.* Ars Combinatoria 47, pp. 287-300.

[3] Gonzalez, T., Sahni, S. (1976). *Open shop scheduling to minimize finish time*. Journal of the ACM, pp. 665–679.

[4] Graham, R., Lawler, E., Lenstra, J., Rinnooy Kan, A. (1979). *Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey.* Proceedings of the Advanced Research Institute on Discrete Optimization and Systems Applications of the Systems Science Panel of NATO and of the Discrete Optimization Symposium., pp. 287–326.

[5] Hanson, D., Loten, C. O. M., Toft, B. (1998). *On interval coloring of biregular bipartite graphs.* Ars ombinatoria, pp. 23–32.

[6] Małafiejska, A. (2016). *Wybrane problemy i modele końcówkowego kolorowania grafów (in Polish).* Raport Tech. WETI 3/2016, Politechnika Gdańska (2016), pp. 1–19.

[7] Melynikov, L., Pyatkin, A., Vizing, V. (2000). *On the (k, l)-coloring of incidentors (in Russian).* Diskretn. Anal. Issled. Oper. 1, pp. 29–37.

[8] Pyatkin, A. V. (1997a). *Some optimization problems of scheduling the transmission of messages in a local communication network.* Operations Research and Discrete Analysis, pp. 227–232, 1997.

[9] Pyatkin, A. (1997b). *The incidentor coloring problems and their applications, Ph.D. Dissertation (in Russian).* Institute of Mathematics SB RAS, Novosibirsk.

[10] Pyatkin, A. V. (2002). *The incidentor coloring of multigraphs and its applications.* Electronic Notes in Discrete Mathematics, pp. 103–104.

[11] Pyatkin, A. (2004). U*pper and lower bounds for an incidentor (k,l)-chromatic number (in Russian).* Diskretn. Anal. Issled. Oper., 1, pp. 93–102.

[12] Pyatkin, A., Vizing, V. (2006). *Incidentor coloring of weighted multigraphs.* Discrete Applied Mathematics, 120, pp. 209–217.

[13] Pyatkin, A. (2013). *Incidentor coloring: Methods and results.* Graph Theory and Interactions, Durham.

[14] Pyatkin, A. (2015). *On an Interval (1,1)-Coloring of Incidentors of Interval Colorable Graphs.* Journal of Applied and Industrial Mathematics, 9, pp. 271–278.

[15] Vizing, V. (2000). *On Incidentor Coloring in a Partially Directed Multigraph.* Fuzzy Sets and Systems, pp. 141–147.

[16] Vizing, V. (2003). *Interval colorings of the incidentors of an undirected multigraph (in Russian).* Diskretn. Anal. Issled. Oper. pp. 14–40.

[17] Vizing, V. (2005). *On the (p, q)-coloring of incidentors of an undirected multigraph (in Russian).* Diskretn. Anal. Issled. Oper. 1, pp. 23–39.

[18] Vizing, V. (2007). *On Bounds for the Incidentor Chromatic Number of a Directed Weighted Multigraph.* Journal of Applied and Industrial Mathematics, 1, pp. 504–508.

[19] Vizing, V. (2009). *On Incidentor Coloring in a Hypergraph.* Journal of Applied and Industrial Mathematics, 3, pp. 144–147.

[20] Vizing, V. (2012). *Multicoloring the Incidentors of a Weighted Undirected Multigraph.* Journal of Applied and Industrial Mathematics, 6, pp. 514–521.

[21] Vizing, V. (2014). *Multicoloring the Incidentors of a Weighted Directed Multigraph.* Journal of Applied and Industrial Mathematics, 8, pp. 604–608.