Available online at www.sciencedirect.com

SCIENCE DIRECT

Discrete Applied Mathematics 154 (2006) 604–607

DISCRETE APPLIED MATHEMATICS

www.elsevier.com/locate/dam

Research problems

These research problems have been edited for Discrete Applied Mathematics by

Gregory Gutin
Department of Computer Science, Royal Holloway, University of London, Egham, Surrey TW20 0EX, UK
E-mail address: gutin@cs.rhul.ac.uk

PROBLEM 1. Minimizing makespan in a two-machine reentrant flow shop

George Steiner,
Zhihui Xue
School of Business, McMaster University, Hamilton, Ont., Canada L8S 4M4
E-mail addresses: steiner@mcmaster.ca (G. Steiner), xzhm@yahoo.com (Z. Xue)

Received 16 March 2005; accepted 21 July 2005

Research on scheduling and sequencing in reentrant flow shops has received much attention in recent years. In a *reentrant flow shop*, jobs have to enter a certain machine or a set of machines for processing more than once. Thus a reentrant flow shop is an extension of the classical flow shop, but a special case of the job shop due to the fact that all jobs pass through the machines over the same route. Many real-life applications in semiconductor manufacturing and flexible machining systems can be modeled as reentrant flow shops, e.g., the assembly of printed circuit boards and wafer fabrication. Moreover, the reentrant flow shop has been examined in a client–server computer application by Błażewicz et al. [1].

There are various types of reentrant processing depending on the job-processing route. Here we consider a two-machine reentrant flow shop, where there are n jobs to be processed on two machines, M_1 and M_2. Each job $j \in \{1, 2, \ldots, n\}$ has a sequence of three operations in the order $O_{1j} \to O_{2j} \to O_{3j}$. We assume that two consecutive operations of a job are processed on different machines. Therefore, for each job j, O_{1j}, O_{2j} and O_{3j} have to be processed without preemption on M_1, M_2, and again on M_1, respectively. Each machine can only process one operation at a time, and there is unlimited input and output buffer space available for each machine. We focus on the makespan minimization problem, i.e., scheduling and sequencing the jobs so as to minimize the completion time of the last job. Following the standard three-field notation for machine scheduling problems, we denote the problem as $RF2|\ell=3|C_{\max}$, where $RF2$ indicates a two-machine reentrant flow shop, $\ell=3$ describes that each job has three operations, and $C_{\max}$ denotes the makespan objective.

This problem has been considered in various settings. Lev and Adiri [3] study the problem in a *V-shop* where the jobs follow the route $M_1 \to M_2 \to \cdots \to M_{m-1} \to M_m \to M_{m-1} \to \cdots \to M_2 \to M_1$. They show that the problem is $\mathscr{NP}$-hard for $m=2$. Wang et al. [6] analyze the problem in a *chain-reentrant shop* where the jobs follow the route $M_1 \to M_2 \to \cdots \to M_m \to M_1$. They present a branch-and-bound algorithm and an approximation algorithm with worst-case performance guarantee of $\frac{3}{2}$ for the $m=2$ case. Hall et al. [2] investigate the cycle time minimization

0166-218X/$ - see front matter
doi:10.1016/j.dam.2005.09.001

problem in a two-machine job shop, where each job consists of at most three operations. Steiner and Xue [5] show that the problem can be polynomially reduced to $RF2|\ell=3|C_{\max}$ and discuss some consequences; see also [7].

Although $RF2|\ell=3|C_{\max}$ has been studied extensively, it is a hard open question whether it is solvable in pseudo-polynomial time or not [4,6]. For a recent attempt to solve this problem, see [5,7]. Specifically, they show that if the answer is "Yes", then $RF2|\ell=3|C_{\max}$ admits a fully polynomial time approximation scheme(FPTAS) too—the best possible approximation algorithm for an $\mathcal{NP}$-hard problem.

References

[1] J. Błażewicz, P. Dell'Olmo, M. Drozdowski, Scheduling of client–server applications, Internat. Trans. Oper. Res. 6 (1999) 345–363.
[2] N.G. Hall, T.-E. Lee, M.E. Posner, The complexity of cyclic shop scheduling problems, J. Scheduling 5 (2002) 307–327.
[3] V. Lev, I. Adiri, V-shop scheduling, European J. Oper. Res. 18 (1984) 51–56.
[4] M. Middendorf, V.G. Timkovsky, On scheduling cycle shops: classification, complexity and approximation, J. Scheduling 5 (2002) 135–169.
[5] G. Steiner, Z. Xue, On minimizing cycle time in a two-machine job shop, Working Paper, School of Business, McMaster University, Canada, 2004.
[6] M.Y. Wang, S.P. Sethi, S.L. van de Velde, Minimizing makespan in a class of reentrant shops, Oper. Res. 45 (1997) 702–712.
[7] Z. Xue, Shop scheduling in manufacturing systems: algorithms and complexity, Ph.D. Thesis, McMaster University, Canada, 2004.

PROBLEM 2. $\mathscr{C}$-hypergraph perfection

Vitaly Voloshin
Troy University, Troy, AL, USA
E-mail address: vvoloshin@troy.edu

Received 20 March 2005; received in revised form 26 March 2005; accepted 21 July 2005

A $\mathscr{C}$*-hypergraph* is a pair $\mathscr{H}=(X,\mathscr{C})$, where X is the *vertex set* and $\mathscr{C}$ is a family of subsets of X, $\mathscr{C}$*-edges*. In every *proper coloring* of a $\mathscr{C}$-hypergraph, each $\mathscr{C}$-edge has at least two vertices with a $\mathscr{C}$ommon color. The maximum number of colors used in a proper coloring of $\mathscr{H}=(X,\mathscr{C})$ is called the *upper chromatic number* of $\mathscr{H}$ (denoted by $\bar{\chi}(\mathscr{H})$). The $\mathscr{C}$*-stability number* $\alpha_{\mathscr{C}}(\mathscr{H})$ is the maximum cardinality of a vertex subset containing no $\mathscr{C}$-edge. A $\mathscr{C}$-hypergraph $\mathscr{H}$ is *perfect* [3] if $\bar{\chi}(\mathscr{H}')=\alpha_{\mathscr{C}}(\mathscr{H}')$ for every induced subhypergraph $\mathscr{H}'$ of $\mathscr{H}$.

It was conjectured in [3], see also [4], that an r-uniform $\mathscr{C}$-hypergraph is perfect if and only if it has no induced monostar or cycloid of the form $\mathscr{C}^r_{2r-1}$, $r\geqslant 3$. These two natural non-perfect families served as an analog of Berge's Strong Perfect Graph Conjecture, which stated that a graph G is perfect if and only if no odd cycle of length at least 5 occurs as an induced subgraph of G or $\bar{G}$ (proved recently in [1]).

Very recently, Kral [2] has disproved the hypergraph perfection conjecture for each $r\geqslant 3$ by constructing a new family of minimal non-perfect $\mathscr{C}$-hypergraphs different from monostars and cycloids.

For $r=3$, we know the following 6 examples of minimal 3-uniform non-perfect $\mathscr{C}$-hypergraphs:

$$V_1=(\{1,2,3,4\},\{\{1,2,3\},\{1,3,4\},\{1,2,4\}\});\ \text{(monostar)},$$

$$V_2=(\{1,2,3,4,5\},\{\{1,2,3\},\{1,4,5\}\});\ \text{(monostar)},$$

$$V_3=(\{1,2,3,4,5\},\{\{1,2,3\},\{1,3,4\},\{1,4,5\}\});\ \text{(monostar)},$$

$$V_4=(\{1,2,3,4,5\},\{\{1,2,3\},\{1,3,4\},\{1,4,5\},\{1,2,5\}\});\ \text{(monostar)},$$

$$V_5=(\{1,2,3,4,5\},\{\{1,2,3\},\{2,3,4\},\{3,4,5\},\{4,5,1\},\{5,1,2\}\});\ \text{(cycloid } \mathscr{C}^3_5\text{)},$$

$$K_1=(\{1,2,3,4,5,6\},\{\{1,2,4\},\{2,3,5\},\{3,4,6\},\{4,5,1\},\{5,6,2\},\{6,1,3\},\\ \{1,3,5\},\{2,4,6\}\})\ \text{(Kral}'\text{s construction)}.$$

Problem 1. *Is it true that every 3-uniform $\mathscr{C}$-hypergraph is perfect if and only if it does not contain any of the $\mathscr{C}$-hypergraphs above as induced subhypergraphs?*

References

[1] M. Chudnovsky, N. Robertson, P. Seymour, R. Thomas, The strong perfect graph theorem, Ann. Math., to appear.
[2] D. Kral', A counter-example to Voloshin's hypergraph co-perfectness conjecture, Australasian J. Combin. 27 (2003) 25.
[3] V.I. Voloshin, On the upper chromatic number of a hypergraph, Australasian J. Combin. 11 (1995) 25–45.
[4] V.I. Voloshin, Coloring Mixed Hypergraphs: Theory, Algorithms and Applications, Fields Institute Monograph, American Mathematical Society, 2002.

PROBLEM 3. Is integrality gap in cutting stock problem less than 2?

Eugene J. Zak
TietoEnator Majiq Inc., 8343 - 154th Avenue NE, Redmond, WA 98052, USA
E-mail address: eugene.zak@tietoenator.com

Received 10 June 2005; accepted 21 July 2005

The standard integer programming formulation of a cutting stock problem (CSP) with a tight linear programming (LP) relaxation is the following [1,2]:

$$\text{Minimize} \quad \mathbf{1}^{\mathrm{T}}\mathbf{x} \tag{1}$$

$$\text{Subject to} \quad A\mathbf{x} \geqslant \mathbf{b}, \tag{2}$$

$$\mathbf{x} \in Z_{+}^{n}. \tag{3}$$

The objective function (1) minimizes the number of stock rolls used. Constraint (2) ensure that the customer demand—vector **b**—has been met, and Constraint (3) define non-negativity and integrality of the variables. Columns of matrix A represent all possible cutting patterns, and vector **x** is an unknown vector of column activities.

A full-scale matrix A is unknown due to enormous number of possible patterns, but its columns $\mathbf{a}_j$ are points of the following knapsack polyhedron: $K_{\mathrm{p}} = \{\mathbf{a} \in Z_{+}^{m} : \ \mathbf{w}^{\mathrm{T}}\mathbf{a} \leqslant d\}$, where **w** is an m-dimensional vector of item widths; d is a scalar knapsack capacity. In the CSP notation, a knapsack capacity is the width of a stock item that should be cut into smaller, customer specified, widths. Such formulation of CSP falls into a category of the column generation form where problem (1)–(3) is a master problem, and the knapsack (4)–(6) is an auxiliary problem that is also called a pricing problem.

$$\text{Maximize} \quad \mathbf{c}^{\mathrm{T}}\mathbf{a} \tag{4}$$

$$\text{Subject to} \quad \mathbf{w}^{\mathrm{T}}\mathbf{a} \leqslant d, \tag{5}$$

$$\mathbf{a} \in Z_{+}^{m}. \tag{6}$$

An LP relaxation of CSP is an LP problem with relaxed integrality constraints (3), so constraint (3) is replaced with the following constraint:

$$\mathbf{x} \in R_{+}^{n}. \tag{7}$$

Definition. The CSP integrality gap is a difference between the CSP optimal value and the rounded up optimal value of its LP relaxation.

For the vast majority of practical CSPs the integrality gap equals to 0. There are few known instances of CSP with the integrality gap of 1. There are not known instances with the integrality gap exceeding 1 [3,4].

Open problem. It has been conjectured that the integrality gap for any CSP is less than 2. A similar conjecture was proposed for the "dual" to CSP—Skiving Stock Problem—in [5].

References

[1] P.C. Gilmore, R.E. Gomory, A linear programming approach to the cutting-stock problem, Oper. Res. 9 (1961) 849–859.
[2] P.C. Gilmore, R.E. Gomory, A linear programming approach to the cutting-stock problem: Part 2, Oper. Res. 11 (1963) 863–888.
[3] O. Marcotte, The cutting stock problem and integer rounding, Math. Programming 33 (1985) 82–92.
[4] G. Scheithauer, J. Terno, The modified integer round-up property of the one-dimensional cutting stock problem, European J. Oper. Res. 84 (3) (1995) 562–571.
[5] E. Zak, The skiving stock problem as a counterpart of the cutting stock problem, Internat. Trans. Oper. Res. 10 (6) (2003) 637–650.