RAIRO-Oper. Res. 49 (2015) 383–391
DOI: 10.1051/ro/2014051

RAIRO Operations Research
www.rairo-ro.org

$\mathbf{O(\log m)}$-APPROXIMATION FOR THE ROUTING OPEN SHOP PROBLEM *

Alexander Kononov[1,2]

Abstract. We consider the routing open shop problem which is a generalization of the open shop and the metric travelling salesman problems. The jobs are located in some transportation network, and the machines travel on the network to execute the jobs in the open shop environment. The machines are initially located at the same node (depot) and must return to the depot after completing all jobs. The goal is to find a non-preemptive schedule with the minimum makespan. We present a new polynomial-time approximation algorithm with worst-case performance guarantee $O(\log m)$, where m is the number of machines.

Keywords. Open shop, routing, approximation algorithms.

Mathematics Subject Classification. 90B06, 90B35, 68W25.

1. Introduction

We consider the routing open shop problem which is a generalization of the open shop and the metric travelling salesman problems. Both problems are strongly NP-hard (see [20] and [8], respectively).

Open shop problem [9] ($O||C_{\max}$)
Given a set of n jobs $\mathcal{J} = \{J_1, \ldots, J_n\}$ and a set of m machines $\mathcal{M} = \{M_1, \ldots, M_m\}$. Each job J_j has to be processed by each machine M_i, and each

Received February 15, 2012. Accepted September 26, 2014.

* *The author was partially supported by Russian Foundation for the Humanities (grant 13-22-10002)*

[1] Sobolev Institute of Mathematics, Russia. alvenko@math.nsc.ru

[2] Novosibirsk State University, Russia

operation O_{ji} takes $p_{ji} \in \mathbb{Z}^+$ time units. The operations of each job can be processed in an arbitrary order. Preemption is not allowed. Different machines cannot work on the same job simultaneously, and a machine cannot work on more than one job at a time. The goal is to minimize the makespan. (For this problem the makespan coincides with the maximum job completion time.)

Metric traveling salesman problem (metric TSP)
We have an undirected edge-weighted complete graph $G = \langle V, E\rangle$, the weight τ_{ij} of edge $e_{ij} = [v_i, v_j]$ is a nonnegative integer which represents a distance between nodes v_i and v_j. Distances satisfy the triangle inequality. The goal is to find a Hamiltonian tour R in G of minimum weight $|R| \doteq \sum_{e_{ij}\in R} \tau_{ij}$.

Routing open shop problem
The input of this problem combines the inputs of the two problems mentioned above. The jobs are located at the nodes of G. The machines have to travel between the jobs (with unit speed). Thus not only the processing times of the operations, but also the travel times between jobs have to be taken into account.

It is assumed that all machines are initially located at the same node (depot). They have to process the operations of all jobs and return to the depot after the completion of all jobs. Any number of machines can travel through the same edge or node simultaneously in any direction. We assume that the machines use the shortest paths while travelling between the nodes.

Specifying *a schedule* for this problem assumes the assignment of dates to the events such as starting an operation of a job or starting machine's movement from one node to another. *The makespan* of a feasible schedule is the interval between the date when the machines start working or moving and the date at which the last machine returns to the depot after finishing all its operations. The objective is to minimize the makespan $C_{\max}$.

Similarly to the standard three-field notation for the open shop problem ($O||C_{\max}$, see Lawler *et al.* [11]) the routing open shop problem is denoted as $RO||C_{\max}$ (or $ROm||C_{\max}$ for a fixed number of machines).

The routing open shop problem is introduced by Averbakh *et al.* in [2,3]. Examples of applications where machines have to travel between the jobs include situations where parts are too big or heavy to be moved between machines (*e.g.*, engine casings of ships), or scheduling of robots that perform daily maintenance operations on immovable machines located in different places of a workshop [1]. Another interesting application is related to the routing and scheduling of museum visitors traveling as homogeneous groups [18]. The model is embedded in a prototype wireless context-aware museum tour guide system developed for the National Palace Museum of Taiwan; one of the top five museums in the world.

The routing open shop problem is strongly NP-hard even for the single machine case as it contains the metric TSP as a special case. Moreover, the routing open shop problem is NP-hard even on a 2-node network with only two machines [3].

For the latter case a 6/5-approximation polynomial time algorithm was presented in [2]. Recently, Kononov [10] presented a FPTAS and closed the open question about the complexity of the two-machines two-nodes routing open shop problem posed in [3]. A 7/4-approximation algorithm for the general 2-machine case and a simple $(m+4)/2$-approximation algorithm for the m-machine case were given in [3]. Chernykh *et al.* [5,6] presented a 13/8-approximation algorithm for $RO2||C_{\max}$. Moreover, they devised an $O(\sqrt{m})$-approximation algorithm for $RO||C_{\max}$ using a job-aggregation idea and the greedy algorithm for the classical open shop. Yu and Zhang [19] improved the latter result and presented $O(\log m(\log\log m)^{1+\epsilon})$-approximation algorithm based on the reduction of the original problem to the classical flow shop problem.

2. Main result

In this note we present a new approximation polynomial-time algorithm with worst-case performance guarantee $O(\log m)$. The algorithm has asymptotically better approximation ratio than all known algorithms.

Theorem 2.1. *There exists an $O(\log m)$-approximation algorithm for $RO||C_{\max}$.*

For convenience, without loss of generality, we associate node v_j with job J_j for $j = 1,\dots,n$ and a special node v_0 with the depot. Thus we have a complete graph $G = \langle V, E\rangle$ with the set of nodes $V = \{v_0, v_1,\dots,v_n\}$ and the set of edges E, where all distances satisfy the triangle inequality.

Denote by $s_{ji}(\sigma)$ the starting time of operation O_{ji} in schedule σ. We define the *length* d_j of job $J_j \in \mathcal{J}$ as the total processing time of its operations, $d_j = \sum_{i=1}^{m} p_{ji}$, and denote $d_{\max} = \max_{J_j\in\mathcal{J}} d_j$. The total processing time of the operations on machine M_i is denoted by ℓ_i and it is called the *load* of machine M_i; $\ell_{\max} = \max_i \ell_i$ is the *maximum machine load* and $C_{\max}(\sigma)$ is the *makespan of schedule* σ; T^* stands for the length of the optimal tour in G. Let $\bar{C} = \max\{l_{\max}, d_{\max}\}$.

2.1. Our techniques

In this section we briefly sketch the ideas of our polynomial-time $O(\log m)$-approximation algorithm for $RO||C_{\max}$.

First, using any polynomial time constant approximation algorithm for metric TSP, *e.g.* [7] or [16], we find an approximate tour R in graph G.

Next, we replace the given set of jobs $\mathcal{J}$ by at most $\min\{2m, n\}$ new *aggregated* jobs. Each aggregated job combines several original jobs, called *component* jobs, consecutively located on a segment of tour R. The processing time of a new operation of an aggregated job on a machine is equal to the total processing time of the component job operations on that machine. The operations of the aggregated jobs are called A-operations. Now, instead of an instance I of $RO||C_{\max}$ we consider an instance I' of $O||C_{\max}$ with the same machines but $n' \le 2m$ jobs, such that $p'_{ij} \le d'_j \le \bar{C}$, where p'_{ij} denotes the processing time of an A-operation O'_{ij} and d'_j denotes the length of an aggregated job J'_j. Notice, in the constructed open shop instance the machine travel times are disregarded.

Let $p'_{\max}$ be the processing time of the largest operation in I'. Using standard scaling techniques [17], we transform the instance I' such that $p'_{\max} \leq 2m^2$ and all processing times of operations to be integral. Next, we round up the processing times of the A-operations to the nearest power-of-two numbers. A new instance I'' has at most $\lceil \log_2 p'_{\max} \rceil$ distinct processing times for the operations. Now we partition the instance I'' into at most $\lceil \log_2 p'_{\max} \rceil$ instances such that all A-operations have the same length in each instance.

Let I_k, $1 \leq k \leq \lceil \log_2 p'_{\max} \rceil$ be one of these instances. In the same way as in [19] we exchange the role of machines and jobs, and we find the processing order of the operations that belong to each new job. This way we obtain an instance I'_k of the acyclic job shop with unit operation lengths. For the latter problem, we run the known constant-factor approximation algorithm [13] that finds a schedule σ' with makespan at most constant time of $\bar{C}$.

Given schedule σ', we transform it into a feasible subschedule σ of I, *i.e.*, a feasible schedule for instance I of $RO||C_{\max}$ on the corresponding subset of the original operations. Let an A-operation occupy some interval in schedule σ'. We put the corresponding component job operations into the same interval in order of their appearance in tour R. We repeat this procedure for all A-operations in σ'. Then, we shift the starting time of each operation by the length of the path in tour R from the depot to the node containing this operation. We obtain a feasible subschedule σ of I such that the makespan of σ is at most $\rho\bar{C} + |R|$, where ρ is a constant. Now, we construct a generic feasible schedule $\bar{\sigma}$ as a concatenation of the obtained subschedules. We have $\lceil \log_2 p'_{\max} \rceil \leq 2m^2$ and hence we derive a $O(\log m)$-approximation algorithm for $RO||C_{\max}$.

2.2. Approximation algorithm for $\mathsf{RO}||\mathsf{C}_{\max}$

In this section we present more formally our approximation algorithm for the routing open shop problem. Each step of the algorithm described below is followed by a brief discussion and implementation details, if needed.

Algorithm ROS

Step I: Find a near-optimal hamiltonian tour R with length $|R| \leq \frac{3}{2}T^*$ in graph G. Without loss of generality we assume that R walks through the nodes in the order $v_0, v_1, \ldots, v_n$.
The best known approximation algorithm for this problem is due to Christofides [7] and Serdyukov [16]. The algorithm has performance ratio $\frac{3}{2}$ and its running time is $O(n^3)$.

Step II: Partition the tour R into disjoint paths $P_1, \ldots, P_k$, where the number of paths k is specified at the completion of step II.
Put $i := 0;\ q := 0$.
While $q \leq n$ **do begin**
$i := i + 1;\ P_i := \emptyset;$

while
(a) $\sum_{v_j \in P_i} d_j \le \ell_{\max}$ **and**,
(b) $q \le n$.
do $\{P_i := P_i \oplus v_j;\ q := q+1\}$ **end** (while)

Step III: Define an instance I' of $O||C_{\max}$ with k jobs that have to be executed on the set of m machines. To that end, for each path P_j, $j = 1, \ldots, k$, define A-job J'_j. The processing time of A-job J'_j on machine M_i is set to $p'_{ji} \doteq \sum_{v_h \in P_j} p_{hi}$.

Let d'_j be the length of A-job J'_j. From the step II we have $d'_j \le l_{\max}$ and $d'_j + d'_{j+1} > l_{\max}$. Since the total load of all machines does not exceed $ml_{\max}$, the latter inequality implies that the number of A-jobs in the instance I' does not exceed $2m$. Let $p'_{\max} = \max_{ij} p'_{ji}$ and ω be the total number of A-operations in I'. We note that $\omega \le 2m^2$.

Step IV: Round down each p'_{ji} to the nearest multiple of $p'_{\max}/\omega$, and denote this value by p''_{ji}, $p''_{ji} := \max\{k \in \mathbb{Z} | \frac{kp'_{\max}}{\omega} \le p'_{ji}\}$.

Let $C^*_{\max}$ be the makespan of an optimal schedule of instance I'. Step IV ensures that the value p'_{ji} takes at most ω distinct values, which are all multiples of $p'_{\max}/\omega$. Therefore we can treat the p''_{ji} as integers in $\{0, \ldots, \omega\}$; a schedule for this problem can be trivially rescaled to a schedule for the original operations of length at most $C^*_{\max} + p'_{\max}$.

Step V: Round up each p''_{ji} to the nearest power-of-two numbers.

The new instance I'' has $L \le \lceil \log_2 \omega \rceil$ distinct processing times of the operations.

Step VI: For each $l = 1, \ldots, L$ define an instance I'_l. The processing time of operation O_{ji} is set to $p^{(l)}_{ji} = \lceil \frac{p''_{ji} p'_{\max}}{\omega} \rceil$ if $p''_{ji} = 2^l$ and $p^{(l)}_{ji} = 0$, otherwise. For each $i = 1, \ldots, m$ and $j = 1, \ldots, k-1$ operation O_{ji} must be completed before operation O_{j+1i} starts.

Thus, each instance I'_l is an instance of the flow shop problem with k machines and m jobs, in which all non-zero operations have the same processing times. Moreover, we have $p^{(l)}_{ji} \ge p'_{ji}$.

Step VII: Find a near-optimal schedule σ'_l for each instance I'_l.

It is well known that the flow shop problem in which all non-zero operations have the same processing times can be considered as the corresponding packet routing problem with unit bandwidths and unit transit times [12]. Leighton *et al.* [12] proved the existence of a routing protocol whose length is linear in $C + D$, where C and D denote the trivial lower bounds congestion and dilation. In the flow shop problem the congestion C corresponds to the maximal machine load $l_{\max}$ and the dilation D corresponds to the maximal job length $d_{\max}$. In [13], Leighton et al. presented an algorithm that finds such a schedule in $O(\omega'_l(\log\log\omega'_l)\log\omega'_l)$ time, where ω' is the number of operations in I'_l. However, we note that the hidden constant in the schedule length is very large. Recently, Peis and Wiese [15] showed that there exists a routing protocol of length at most $23.4(C + D)$. The only non-constructive part of the proof in [15] is the using of Local Lovasz Lemma. Recently, Moser and Tardosh [14] gave a general algorithmic framework for the Local Lovasz Lemma and presented a randomized algorithm within this framework to construct the structures guaranteed by the Local Lovasz Lemma. Finally, Chandrasekaran *et al.* [4] developed the deterministic polynomial time algorithm that works in the general framework of Moser-Tardos.

Step VIII: For each schedule σ'_l eliminate all operations with zero length. For each non-zero A-operation repeat the following procedure. Let non-zero A-operation O'_{ji} of A-job J'_j occupy an interval $[\tau_0, \tau_1]$ in schedule σ'_l. Let $J_{h+1} \dots, J_{h+k}$ be the component jobs of J'_j.
Put $q := 1, \tau := \tau_0$.
While $q \le k$ **do**$\{s_{h+qi}(\sigma_l) := \tau;\ \tau := \tau + p_{h+qi};\ q := q+1\}$.

Let σ_l be a schedule obtained from the schedule σ'_l after Step VIII. By the construction of the algorithm we have that $\tau_1 - \tau_0 = p'_{ji} \ge \sum_{q=1}^{k} p_{h+qi}$. It follows that all operations of the corresponding component jobs are executed inside interval $[\tau_0, \tau_1]$ and σ_l is a feasible schedule.

Step IX: For each schedule σ'_l and each operation O_{ji} set $s_{ji}(\sigma_l) := s_{ji}(\sigma'_l) + \lambda_j$, where λ_j is the distance between the depot v_0 and node v_j in R.

Remind that in σ'_l each machine executes jobs in order their appearance in R. Thus after shifting of the starting time of each operation on the distance from the depot to the corresponding node, we obtain a feasible schedule σ_l with respect to routing of each machine. Moreover, all operations of the same job are shifted on the same distance and such shifting never produces overlap of any two operations of each job.

Step X: Return schedule $\bar{\sigma} = \sigma_1 \circ \sigma_2 \circ \cdots \circ \sigma_L$.

The running time of algorithm ROS depends on which algorithms we use on Step I and Step VII to solve the metric traveling salesman problem and the packet routing problem, correspondingly. As we mention in the comments on the algorithm, both problems can be solved in polynomial time. Next, we note that the makespan of

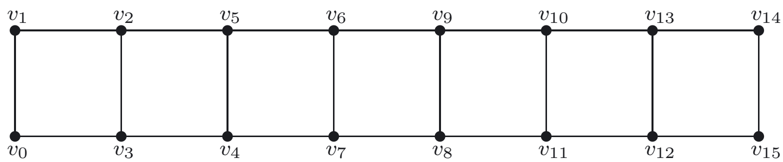

FIGURE A.1. A network for Example 1.

each schedule σ_l is within a constant factor of $\bar{C}$ and T^*. Taking into account that the $L \leq \lceil \log_2 \omega \rceil \leq 2 \log_2 2m$, we obtain that algorithm ROS is an $O(\log m)$-approximation algorithm for $RO||C_{\max}$ and Theorem 2.1 follows.

3. CONCLUSION

We have described a new approximation algorithm for the routing open shop problem. Our algorithm has a logarithmic worst-case performance ratio. It is the best known ratio for the problem at the moment. From the other hand, the routing open shop problem is a generalization of the open shop and the metric traveling salesman problems. The simple constant factor approximation algorithms are known for both problems. Thus it should be interesting to design a polynomial time approximation algorithm for the routing open shop problem with constant performance guarantee or prove a non-constant lower bound under $P \neq NP$.

APPENDIX A. EXAMPLE

In this section we present an example that shows how Algorithm ROS works. Let us consider the following instance I of $RO||C_{\max}$. Given five machines and 15 jobs. The jobs and machines are located in the network shown in Figure A.1. Each job J_i is located at the vertex v_i and all machines are located in the vertex v_0. All edges shown in Figure A.1 have weight 1. Processing times of operations are presented in Figure A.2. For this instance, $l_{\max} = 120$ and $d_i = 40$ for all $i = 1, \ldots, 15$. We assume that Christofides' algorithm obtains the near optimal tour $v_0, v_1, v_2, \ldots, v_{14}, v_{15}, v_0$.

In step III we obtain the instance I' of $O||C_{\max}$ with five aggregated jobs. Each aggregated job combines three original jobs. The processing times of new operations are presented in Figure A.3. As seen from the table in Figure A.3 we have $p'_{\max} = 42$ and $w = 25$. Figure A.4 shows the processing times of operations after the rounding in steps IV and V.

Next we partition the instance I'' into four instances I'_1, I'_2, I'_3, and I'_4. It is easy to see that two first instances contain only two operations and the last instance contains four operations. It is trivial to find optimal solutions in these instances. For example, $s_{11}(\sigma'_1) = s_{52}(\sigma'_1) = 0$. Applying Step VIII and Step IX to σ'_1 we obtain $s_{11}(\sigma_1) = 1$, $s_{21}(\sigma_1) = 3$, $s_{31}(\sigma_1) = 6$, $s_{13,2}(\sigma_1) = 13$, $s_{14,2}(\sigma_1) = 17$, and

O_{ji}	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9	J_{10}	J_{11}	J_{12}	J_{13}	J_{14}	J_{15}
M_1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
M_2	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
M_3	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
M_4	6	7	8	9	10	6	7	8	9	10	6	7	8	9	10
M_5	10	9	8	7	6	10	9	8	7	6	10	9	8	7	6

FIGURE A.2. Processing times of operations in the instance I.

O_{ji}	J'_1	J'_2	J'_3	J'_4	J'_5
M_1	6	15	24	36	42
M_2	42	36	24	15	6
M_3	24	24	24	24	24
M_4	21	25	24	23	27
M_5	27	23	24	25	21

FIGURE A.3. Processing times of operations in the instance I'.

O_{ji}	J'_1	J'_2	J'_3	J'_4	J'_5
M_1	4	8	16	32	32
M_2	32	32	16	8	4
M_3	16	16	16	16	16
M_4	16	16	16	16	16
M_5	16	16	16	16	16

FIGURE A.4. Processing times of operations in the instance I''.

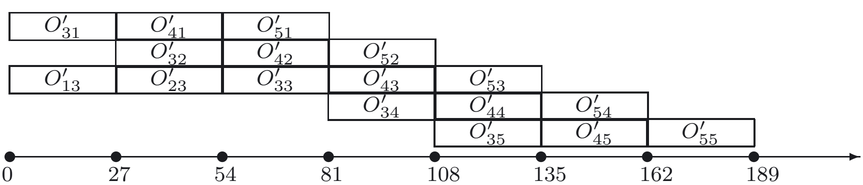

FIGURE A.5. The optimal schedule for the instance I'_3.

O_{ji}	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9	J_{10}	J_{11}	J_{12}	J_{13}	J_{14}	J_{15}
M_1	-	-	-	-	-	-	7	15	24	-	-	-	-	-	-
M_2	-	-	-	-	-	-	34	44	53	-	-	-	-	-	-
M_3	1	10	19	31	40	49	61	70	79	91	100	109	121	130	139
M_4	28	35	43	58	68	79	88	96	105	118	129	136	148	157	167
M_5	55	65	76	85	93	99	115	125	134	145	156	163	175	184	192

FIGURE A.6. Starting times of operations in the feasible schedule σ_3.

$s_{15,2}(\sigma_1) = 20$, and $C_{\max}(\sigma_1) = 28$. Let us consider the instance I_3. Figure A.5 shows the possible optimal schedule σ'_3 for the instance I'_3. Step VIII and Step IX transform σ'_3 into a feasible subschedule σ_3 of the routing open shop problem. The starting times of operations in σ_3 are presented in Figure A.6, $C_{\max}(\sigma_1) = 205$.

References

[1] I. Averbakh and O. Berman, A simple heuristic for m-machine flow-shop and its applications in routing-scheduling problems. *Oper. Res.* **47** (1999) 165–170.

[2] I. Averbakh, O. Berman and I. Chernykh, A $\frac{6}{5}$-approximation algorithm for the two-machine routing open shop problem on a 2-node network. *Eur. J. Oper. Res.* **166** (2005) 3–24.

[3] I. Averbakh, O. Berman and I. Chernykh, The routing open-shop problem on a network: complexity and approximation. *Eur. J. Oper. Res.* **173** (2006) 521–539.

[4] K. Chandrasekaran, N. Goyal and B. Haeupler, Deterministic algorithms for the Lovasz local lemma, in the *Proc. of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms* (2010) 992–1004.

[5] I. Chernykh, N. Dryuck, A. Kononov and S. Sevastyanov, The routing open-shop problem: New approximation algorithms, in *the Proceedings of WAOA*, edited by E. Bampis, K. Jansen 2009, in Vol. 5893 of *Lect. Notes Comput. Sci.* (2010) 75–85.

[6] I. Chernykh, A. Kononov and S. Sevastyanov, Efficient approximation algorithms for the routing open shop problem. *Comput. Oper. Res.* **40** (2013) 841–847.

[7] N. Christofides, *Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem*, Report 388. Graduate School of Industrial Administration. Carnegie-Mellon University, Pittsburgh, PA (1976).

[8] M. Garey and D. Johnson, *Computers and Intractability, A Guide to the theory of NP-completeness*. W.H. Freemann and Company, San Francisco, CA (1979).

[9] T. Gonzalez and S. Sahni, Open Shop Scheduling to Minimize Finish Time. *J. ACM* **23** (1976) 665–679.

[10] A. Kononov, On the routing open shop problem with two machines on a two-vertex network. *J. Appl. Ind. Math.* **6** (2012) 318–331, Original Russian Text A.V. Kononov, published in Diskretnyi Analiz i Issledovanie Operatsii **19** (2012) 55–75.

[11] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys, Sequencing and scheduling: algorithms and complexity. Vol. 4 of *Handbooks Oper. Res. Management Science, Logistics of Production and Inventory*. North Holland, Amsterdam (1993) 445–522.

[12] F.T. Leighton, B.M. Maggs and S.B. Rao, Packet routing and job-scheduling in O(congestion + dilation) steps. *Combinatorica* **14** (1994) 167–186.

[13] F.T. Leighton, B.M. Maggs and A.W. Richa, Fast algorithms for finding O(Congestion + Dilation) packet routing schedules. *Combinatorica* **19** (1999) 375–401.

[14] R.A. Moser and G. Tardos, A constructive proof of the general Lovasz Local Lemma. *J. ACM* **57** (2010) 11:1–11:15.

[15] B. Peis and A. Wiese, Universal Packet Routing with Arbitrary Bandwidth and Transit Times, in *the Proc. of IPCO, 2011*, edited by O. Günlük and G. Woeginger. *Lect. Notes Comput. Sci.* **6655** (2011) 362–375.

[16] A. Serdyukov, On some extremal routes in graphs. *Upravlyaemye Sistemy* **17** (1978) 76–79 (in Russian).

[17] D. Shmoys, C. Stein and J. Wein, Improved approximation algorithms for shop scheduling problems. *SIAM J. Comput.* **23** (1994) 617–632.

[18] V. F. Yu, S.-W. Lin and S.-Y. Chou, The museum visitor routing problem. *Appl. Math. Comput.* **216** (2010) 719–729.

[19] W. Yu and G. Zhang, Improved approximation algorithms for routing shop scheduling, in *the Proc. of ISAAC 2011. Lect. Notes Comput. Sci.* **7074** (2011) 30–39.

[20] D.P. Williamson, L.A. Hall, J.A. Hoogeveen, C.A.J. Hurkens, J.K. Lenstra, S.V. Sevastianov and D.B. Shmoys, Short shop schedules. *Oper. Res.* **45** (1997) 288–294.