On a Polynomial Solvability of the Routing Open Shop with a Variable Depot

Antonina P. Khramova[1], Ilya Chernykh[1,2,3]

[1] Sobolev Intitute of Mathematics, Russia
akhramova@math.nsc.ru idchern@math.nsc.ru
[2] Novosibirsk State University, Russia
[3] Novosibirsk State Technical University, Russia

Keywords: Open shop, routing open shop, unrelated travel times, variable depot, computational complexity, exact algorithm, approximation algorithm.

1 Preliminaries

The open shop problem to minimize finish time (Gonzalez and Sahni 1976) is one of the classical multistage scheduling problems and can be described as follows. Let $\mathcal{M} = \{M_1, \dots, M_m\}$ be a set of m machines and $\mathcal{J} = \{J_1, \dots, J_n\}$ be a set of n jobs. Each job J_j consists of m operations $(O_{j1}, \dots, O_{jm})$. The operation O_{ji} takes p_{ji} time units and has to be processed on machine M_i, and no two operations of the same job can be processed at the same time, as well as no machine can process two jobs simultaneously. However, unlike flow shop model, the operations of a job can be processed in any order. We follow standard notation (Lawler *et. al.* 1993) to denote this problem as $Om||C_{\max}$.

It is known (Gonzalez and Sahni 1976) that if the number of the machines is at least three, $Om||C_{\max}$ is NP-hard. However, $O2||C_{\max}$ is polynomially solvable. Several algorithms are known to solve this problem in linear time. The first one was introduced by Gonzalez and Sahni (1976). Other algorithms for this problem were proposed by Pinedo and Schrage (1982), de Werra (1989), and Soper (2015).

The routing open shop problem is a certain generalization of the open shop problem and can be described as follows. Each job is assigned to a node of a transportation network given by an undirected edge-weighted graph G. The weight of an edge represents the time required by any machine to travel between the respective nodes. To process a job, a machine has to move to the node where the job is located. So, machines have to travel over the transportation network in order to process the jobs. It is assumed that any number of machines can travel over the same edge at the same time. All machines start at the same node, called the *depot*, and must return to the depot after completing all jobs. The goal is to minimize the makespan (*i.e.* the completion time of the last activity of a machine), denoted by $R_{\max}$. The notation $ROm||R_{\max}$ denotes the problem in case of m machines. We also use notation $ROm|G = X|R_{\max}$ in order to specify the structure X of the transportation network.

The routing open shop problem is a generalization of both the open shop problem (consider every edge of the graph to be of zero weight) and the metric traveling salesman problem (consider every operation to be of zero processing time), so it is obviously NP-hard in general case. The routing open shop problem was introduced and proved to be NP-hard even in the simplest case with two machines and $G = K_2$ in (Averbakh *et. al.* 2006). We further extend the problem statement with the following options introduced in (Chernykh 2016).

1. The depot in $ROm||R_{\max}$ may be either *fixed*, *i.e.* defined in the problem instance, or *variable*, *i.e.* it has to be chosen while composing a schedule. We write $ROm|variable-depot|R_{\max}$ to indicate the latter case.

2. The travel times between the nodes may differ for each machine. In particular, they can be *identical*, *uniform*, *i.e.* for any two machines M_{i_1} and M_{i_2} there is some $k > 0$ so that any edge for M_{i_1} is k times longer then it is for M_{i_2}, or *unrelated*. In the three-field notation, we write Qtt or Rtt in the last two cases respectively.

We write $easy - TSP$ in three-field notation if the structure of the transportation network G allows solving the underlying TSP in polynomial time. While the problem $RO2|variable - depot|R_{\max}$ is still NP-hard, being a generalization of the metric TSP, the algorithmic complexity of the problem $RO2|easy - TSP, variable - depot|R_{\max}$ was an open question. The special case $RO2|G = tree, Rtt, variable - depot|R_{\max}$ was proved to be polynomially solvable in (Chernykh 2016).

In this paper, we present a linear time algorithm for the $RO2|G = cycle, Rtt, variable - depot|R_{\max}$ problem, which also induces a new linear algorithm for classic open shop model. An important corollary of the result is the polynomial solvability of $RO2|Qtt, easy - TSP, variable - depot|R_{\max}$, which provides an answer to the open question mentioned above. As a by-product, we also provide an approximation result for the $RO2|Qtt, variable - depot|R_{\max}$ problem.

2 A linear time algorithm for $RO2|G = cycle, Rtt, variable - depot|R_{\max}$

Let G be a transportation network for an instance of general routing open shop problem. We use the lower bound $\overline{R}$ for the optimal makespan, defined by the formula $\overline{R} = \max\limits_i\{\ell_i + T_i, d_{\max}\}$, where $d_{\max} = \max\limits_j \sum\limits_{i=1}^{m} p_{ji}$ is the *maximum job length*, $\ell_i = \sum\limits_{j=1}^{n} p_{ji}$ is the *load* of the machine M_i, and T_i is the length of a minimal route over G for M_i.

For any list of jobs $\pi = (J_1, J_2, \ldots J_n)$, define $S(\pi)$ to be an early schedule such that:
(a) the machine M_1 performs operations in order $O_{21} \to O_{31} \to \ldots \to O_{n1} \to O_{11}$;
(b) the machine M_2 performs operations in order $O_{12} \to O_{22} \to O_{32} \ldots \to O_{n2}$;
(c) for any job but J_1, the order of operations is $O_{j1} \to O_{j2}$.
The notation π^{+k} is used for a shifted list $(J_k, J_{k+1}, \ldots, J_n, J_1, \ldots, J_{k-1})$.

For $RO2|G = cycle, Rtt, variable - depot|R_{\max}$, consider the following
ALGORITHM $\mathcal{A}$:
Input: An instance of the $RO2|G = cycle, Rtt, variable - depot|R_{\max}$ problem.

1. Let $\pi = (J_1, J_2, \ldots J_n)$ be a list of jobs such that in the list of respective nodes $(v_1, v_2, \ldots, v_n)$ we have either $v_i = v_{i+1}$ or v_i and v_{i+1} are adjacent in G for all $i \in \{1, \ldots, n-1\}$. Choose the node $v = v_1$ to be a depot.
2. If necessary, re-enumerate the machines so that $\ell_1 + T_1 \le \ell_2 + T_2$.
3. Compose a schedule $S(\pi)$.
4. If $R_{\max}(S(\pi)) = \overline{R}$, then **Output** $S(\pi)$.
   Else
   (a) Let J_k from a node u be the job that is processed after the last time the second machine idles in the schedule $S(\pi)$.
   (b) Taking u to be the depot, **Output** $S(\pi^{+k})$.

Theorem 1. ALGORITHM $\mathcal{A}$ *returns a schedule of length* $\overline{R}$ *in* $O(n)$ *time.*

Proof. Note that if $R_{\max}(S(\pi)) > \overline{R}$, then M_2 idles at some point. Indeed, if M_2 does not, then M_1 must. By definition of $S(\pi)$, the machine M_1 may only idle before starting O_{11}, which is only possible if O_{12} is processed in that idle interval. Then $R_{\max}(S_\pi) = \max\{d_1, l_2 + T_2\} \le \overline{R}$, a contradiction.

Let t be the completion moment of the last idle interval of M_2, which is also the starting time of O_{k2} for a certain $k \in \{1, \dots, n\}$. Consider the schedule $S'(\pi)$ that is obtained by shifting operations $O_{12}, \dots, O_{k-1,2}$ in $S(\pi)$ to the right so that M_2 only idles before processing of O_{21} starts as shown in Figure 1. The makespan of the schedule remains the same. Define the *blocks* (i.e. ordered sets of operations and travel times between the corresponding nodes) A_1, A_2, B_1, and B_2 as follows:

$A_1 = \boxed{\to O_{21} \to \dots \to O_{k1}}$; $A_2 = \boxed{\to O_{k+1,1} \to \dots \to O_{n1} \to O_{11}}$;

$B_1 = \boxed{O_{12} \to \dots \to O_{k-1,2} \to}$; $B_2 = \boxed{O_{k2} \to \dots \to O_{n2} \to}$.

The arrows denote the corresponding travel times.

Fig. 1. Example of schedule $S'(\pi)$

Let Δ be the moment the processing of B_1 starts, so that $R_{\max}(S'(\pi)) = \Delta + l_2 + T_2$. Note that the processing of A_2 ends at $l_1 + T_1$, and $l_1 + T_1 \leq l_2 + T_2$ implies

$$\Delta \leq R_{\max}(S'(\pi)) - (l_1 + T_1). \tag{1}$$

Consider an schedule obtained by placing the block A_2 before A_1, and the block B_2 in front of B_2 as shown in Figure 2.

Fig. 2. Result of the block permutation

The schedule derived by the permutation is feasible due to the inequality (1), unless operations of J_k overlap, and in fact, it is exactly $S(\pi^{+k})$. In case O_{k2} does end later than O_{k1} starts, we obtain $S(\pi^{+k})$ by shifting O_{k1} to the right accordingly. By the construction of the schedule, the machine M_2 never idles, and M_1 may only idle before processing O_{k1}, so $R_{\max}(S(\pi^{+k}))$ is either the length of J_k, or $R_{\max}(S(\pi^{+k})) = \max\{l_1 + T_1, l_2 + T_2\} \leq \overline{R}$. Hence $R_{\max}(S(\pi^{+k})) = \overline{R}$, as wanted.

It is evident that an early schedule can be obtained in linear time. Thus, Algorithm $\mathcal{A}$ runs in linear time, too. □

3 Corollaries

Note that the problem $O2||C_{\max}$ is a special case of $RO2|variable-depot|R_{\max}$ when the travel time between any two nodes is zero. Thus, Algorithm $\mathcal{A}$ induces a linear algorithm for the classic two-machine open shop problem that differs qualitatively from the algorithms proposed before.

The main principle of ALGORITHM $\mathcal{A}$ is composing an early schedule such that the orders of operation processing for the two machines are identical up to cyclic permutation of jobs, with both machines following their optimal route at the same time. With that, we consider two subcases of $RO2|Rtt, variable - depot|R_{\max}$ that can be easily proved to be solvable with the use of ALGORITHM $\mathcal{A}$.

Corollary 1. *The problem $RO2|Qtt, easy - TSP, variable - depot|R_{\max}$ is solvable in time $O(n + t_{TSP})$, where t_{TSP} is the time required to solve TSP on G.*

Corollary 2. *The problem $RO2|Rtt, G = cactus, variable-depot|R_{\max}$ is solvable in $O(n)$.*

In case we have an approximate solution to TSP instead of an exact one, we can use ALGORITHM $\mathcal{A}$ to obtain the same approximation for $RO2|Qtt, variable - depot|R_{\max}$. In particular, by applying Christofides-Serdyukov algorithm (Christofides 1976, Serdyukov 1978), we derive the following

Corollary 3. *There exists a $\frac{3}{2}$-approximate algorithm for $RO2|Qtt, variable-depot|R_{max}$.*

Acknowledgements

This research was supported by the Russian Federation for Basic Research, project 20-01-00045.

References

Averbakh, I., Berman, O., Chernykh, I., 2006, "The routing open shop problem on a network: complexity and approximation", *Eur. J. Oper. Res.*, Vol. 173(2), p. 531-539.

Chernykh, I., 2016, "Routing open shop with unrelated travel times", In: DOOR 2016,Vladivostok, Russia, September 16-23, 2016, Proceedings, pp. 272-283.

Christofides, N., 1976, "Worst-case analysis of a new heuristic for the travelling salesman problem", Report 388, Graduate School of Industrial Administration, CMU.

Gonzalez T., Sahni S., 1976, "Open shop scheduling to minimize finish time", *J. ACM*, Vol. 23(4), pp. 665-21679.

Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A., Shmoys, D.B., 1993, Chapter 9: Sequencing and scheduling: Algorithms and complexity, In: Logistics of Production and Invertory, *Handbooks in Operations Research and Management Science*, Vol. 4, pp. 445-522. Elsevier.

Pinedo, M., Schrage, L., 1982, "Stochastic shop scheduling: a survey", In: Deterministic and Stochastic Scheduling, Dempster, M., Lenstra, J.K., Rinnooy Kan, A. (eds.) *NATO Advanced Study Insitute Series*, Vol. 84, pp. 181-196. Springer, Dordrecht.

Serdyukov, A., 1978, "On some extremal routes routes in graphs" (in Russian), *Upravlyaemye Sistemy*, Vol. 17, pp. 76-79.

Soper, A., 2015, "A cyclical search for the two machine flow shop and open shop to minimize finish time", *J. Sched*, Vol. 18, pp. 311-314.

de Werra, D., 1989, "Graph-theoretical models for preemptive scheduling", In: Advances in Project Scheduling, Slowinski, R., Weglarz, J. (eds.) pp. 171-185. Elsevier, Amsterdam.