IOP Conf. Series: Materials Science and Engineering **452** (2018) 042106 doi:10.1088/1757-899X/452/4/042106

5/3-approximation algorithm for scheduling three-crane integrated scheduling problem

Xie Xie[1, a], Li Zhou[2, b], Yongyue Zheng[3, c]

[1]Key Laboratory of Manufacturing Industrial and Integrated Automation, Shenyang 3
[2]China National Institute of standardization, Beijing, 100191, China
[3]Liaoning Institute of standardization, Shenyang, 110004, China

[a]xiexie_8118@163.com, [b]zhouli@cnis.gov.cn, [c]zhengyongyue@163.com

Abstract. In this paper, we deal with a three-crane integrated scheduling problem arising in a finery shop of an iron and steel making enterprise. It is commonly occurred in every large iron and steel company. The main objective of this process is to determine the sequence of loading operations so that the *makespan* of all required refined melted steel in ladles, that is, the latest finery completion time among all ladles of the melted steel, is minimized. By exploring the problem structure, an effective heuristic algorithm is designed to solve the problem. We prove that the worst case performance of the heuristic algorithm is 5/3. The results show that the proposed heuristic algorithm is capable of generating good quality solutions.

1. Introduction

Iron and steel production is a complicated multistage process that mainly consists of iron making, steel making, and refining stages, where refining processing is the most basically used production mode. A detailed description of various production processes in integrated steel production can be found in Tang et al. (2001). Our problems arise in refining process which is an important subsystem and refining process plays a very important role in modern steel plant. In refining process, the melted steel in each ladle is transported firstly by trolleys and then by parallel cranes mounted on a shared tracks for loading (or unloading) ladles onto (or from) refining furnaces (shorted as RFs) (see Fig. 1). The whole process is considered as a loaded move for a crane. After the crane loads a ladle onto a RF, the crane moves empty ladle to trolley to perform the next loaded move. Once the melted steel in the ladle has been refined completed, a crane unloads the ladle from the RF to a trolley. The crane moves along the track over the area while its pickup device (hoist) can move along the crane bridge. In this way the hoist of the crane can reach any position in the area. The cranes work synchronously and each crane performs at most one ladle at any time. It is strictly forbidden for any two cranes to cross each other during steel refining process.

In practice, cranes are scarce resource in any iron and steel factory, usually the area in the refining shop is served by two or three bridge cranes as illustrated in Figure 1. As shown in Fig.1, there are five RFs as an example in line for finery melted steel. More precisely, when one crane is performing an operation with a RF, the other crane to the left side of the current crane cannot pass the RF location. That is, cranes scheduling is subject to non-interference constraint. In particular, the scheduling of three cranes significantly influences the completion time of steelmaking process. The objective of this research is to determine the sequence of loading operations so that the *makespan* of all required refined melted steel in ladles, that is, the latest finery completion time among all ladles of the melted steel, is *minimized*.

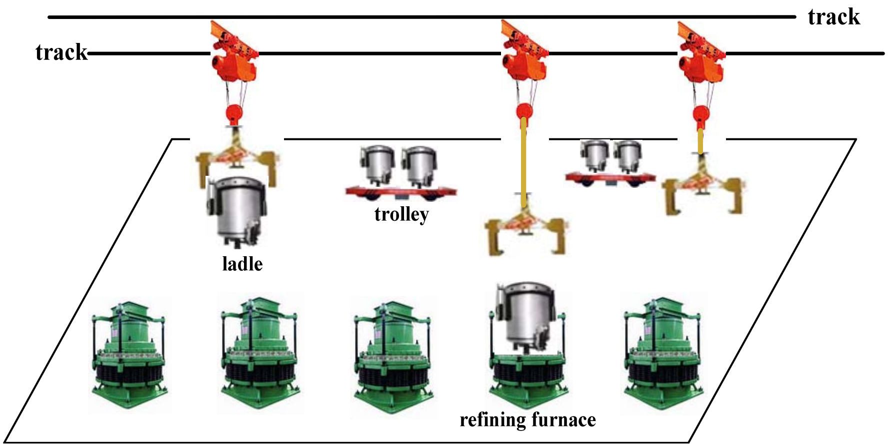

Figure 1. Layout with three cranes in a refining shop

Unlike general production scheduling in the machinery industry, our crane scheduling problem have to meet the special requirements of the steel production process. Most of the literature on crane scheduling received a great deal of attention focused on container terminal. This problem was first introduced by Daganzo (1989) and has been proved to be NP-complete even minimising the *makespan* of one single container vessel (Zhu and Lim, 2006; Lim, Rodrigues, and Xu (2007) Lee et al., 2008). Representative recent paper about some complexity results on crane scheduling see Liu et al. (2016). Multiple-crane scheduling problems studied in most existing papers mainly focus on two-crane case (Briskorn et al. (2016)). Few papers have considered three-crane scheduling problem. And, the non-crossing constraints were first incorporated by Kim and Park (2004), where cranes cannot cross over each other because they are on the same track. For an arbitrary number of cranes, several 2-approximation algorithms have been proposed in the literature (Lee et al., 2007; Lim et al., 2004b; 2007). Zhang et al. (2017) improved this bound and presented an approximation algorithm with a worst case ratio $2-2/m+1<2$ for any m cranes.

So far most of these papers only consider the time of loading and unloading, ignore the travel time of a crane. The problem we consider is scheduling three cranes without ignoring crane travelling time and with non-interference constraint case. Moreover, from the perspective of crane operators, it is required to adopt simple and easily implemented operation methods for practical steel production.

2. Problem description

Throughout the paper, the *makespan* of a given number of ladles is defined as the last refined completed ladle to be unloaded from its furnace to the trolley. We study the scheduling of cranes with non-crossing constraints to *minimize* the *makespan* of all required melted steel in ladles. There are L ladles waiting to be refined on a trolley. In the rest of the paper we may refer to a position of the trolley as the initial position of these ladles, whichever is more convenient. Suppose that there are F RFs from left to right according their position. Since the position of each furnace and ladle are known, the distance between any two positions can be calculated in advance. Only one furnace can handle one ladle till completion. For convenience of expression, the position of the trolley is also the final position of all the ladles.

Knowing that the studied problem is NP-hard we also know that it will probably not be possible to solve instances of realistic size by an exact procedure in-acceptable time. For this reason it is appropriate to use a heuristic, which will not necessarily find an optimal solution, but at least a reasonable one in acceptable time. Here we introduce a heuristic algorithm. In the following section, we propose a 5/3-approximation heuristic algorithm. The algorithm divided L ladles from the leftmost to the rightmost into F subsets, and again assigns these subsets into three pieces, that is, assign one piece to one crane in the corresponding position.

3. Heuristic algorithm and its worst case analysis

Step 1. Form a list of all ladles from left to right, and evenly divided the list into three pieces, cut out the first one third piece.

Step 2. The crossover RF (s), if any, belongs to either the first piece or the remaining list, according to which part occupies its majority. Ties are broken by assigning the hold to the remaining list. Assign the first piece to the most left crane.

Step 3. Divide the remaining list evenly into two pieces.

Step 4. The crossover RF (s), if any, belongs to the piece which occupies its majority, and ties are broken by assigning the hold to the last piece. Assign the second, i.e., the middle piece to the crane in the middle position, and the third piece to the most right crane.

In this algorithm, denote by c_1 the first crossover RF (s) as well as the number of ladles within it. We use a to indicate the total number of ladles in the first piece, excluding the crossover RF (s) c_1 if it belongs to the first piece. Let $\overline{a}$ be the number of RF (s) in the list excluding the first piece.

$$a + c_1 + \overline{a} = L \tag{1}$$

Otherwise if c_1 belongs to the second piece, then

$$a + \overline{a} = L \tag{2}$$

According to step 3, the remaining list of $\overline{a}$ RF(s) is divided into two pieces. Let c_2 denote the second crossover RF(s) as well as the number of ladles within it. Denote by b and d the total number of ladles in the second and the third pieces, respectively, excluding those in c_1 and c_2 crossover RF(s). According equations (1), if c_1 belongs to the first piece, then $b + c_2 + d = \overline{a}$. Otherwise if c_1 belongs to the second piece, then $c_1 + b + c_2 + d = \overline{a}$.

In either case,

$$a + c_1 + b + c_2 + d = L \tag{3}$$

For the case where c_1 belongs to the first piece, combining equations (1) and the fact $a + c_1 > H/3$, we have

$$\overline{a} < 2H/3 \tag{4}$$

For demonstrate the worst case of our heuristic algorithm, let $C_{\max}(\sigma^A)$ be the value of the *makespan* for a schedule σ^A generated by our algorithm A and $C_{\max}(\sigma^*)$ be the value of the *makespan* for an optimal schedule denoted by σ^*. Algorithm A is said to provide the worst case performance guarantee α if for any problem instance $C_{\max}(\sigma^A)/C_{\max}(\sigma^*) \le \alpha$.

Theorem 1. For three-crane scheduling problem, our heuristic algorithm is 5/3-approximation.

Proof. A straightforward conclusion we get is $C_{\max}(\sigma^*) \ge \max\{H/3,\ c_1,\ c_2\}$. If a special case where c_1 the first crossover RF(s) does not exist, our problem reduces to the two-crane problem. For another special case where c_2 does not exist, i. e, $c_2 = 0$, if c_1 is assigned to the first piece, then $C_{\max}(\sigma^A) = a + c_1$ and we have

$$\frac{C_{\max}(\sigma^A)}{C_{\max}(\sigma^*)} \le \frac{a+c_1}{\max\{H/3,\ c_1\}} = \frac{a+c_1/2+c_1/2}{\max\{H/3,\ c_1\}}$$
$$< \frac{H/3+c_1/2}{\max\{H/3,\ c_1\}} \le 3/2 < 5/3$$

We discuss the following two cases:

Case 1. If c_1 belongs to the first piece, and then $\overline{a} = b + c_2 + d$. According to step 2 of our algorithm, we have $a < H/3$ and $a + c_1 - H/3 < H/3 - a$. Hence

$$2a + c_1 < 2H/3 \tag{5}$$

Case 1.1 c_2 belongs to the second piece, according to step 4 of our algorithm, we have $b + c_2 - \overline{a}/2 < \overline{a}/2 - b$. Hence,

$$2b + c_2 < \overline{a} \tag{6}$$

and $C_{\max}(\sigma^A) = \max\{a + c_1, b + c_2\}$.

1) When $a + c_1 \ge b + c_2$, we have $C_{\max}(\sigma^A) = a + c_1$ and

$$\frac{C_{\max}(\sigma^A)}{C_{\max}(\sigma^*)} \le \frac{a+c_1}{\max\{H/3,\ c_1\}} = \frac{a+c_1/2+c_1/2}{\max\{H/3,\ c_1\}}$$
$$< \frac{H/3+c_1/2}{\max\{H/3,\ c_1\}} \le 3/2 < 5/3$$

2) When $a + c_1 < b + c_2$, we have $C_{\max}(\sigma^A) = b + c_2$. In this case, due to (4) and (6), we have $2b + c_2 < 2H/3$. In both cases of $c_2 > H/3$ and $c_2 \le H/3$, we have $C_{\max}(\sigma^A)/C_{\max}(\sigma^*) < 5/3$.

Case 1.2 c_2 belongs to the third piece, in this case, $C_{\max}(\sigma^A) = \max\{a + c_1, d + c_2\}$. According to to step 4 of our algorithm, we have $b + c_2 - \overline{a}/2 \ge \overline{a}/2 - b$. That is,

$$b \ge (\overline{a} - c_2)/2 \tag{7}$$

Combining equation (7) and the definition of $\overline{a}$, we have

$\overline{a} = b + c_2 + d \ge (\overline{a} - c_2)/2 + c_2 + d$, that is $2d + c_2 < \overline{a}$. Combing (4), we have $2d + c_2 < 2H/3$. Similarly to the case 1.1, and we still have $C_{\max}(\sigma^A)/C_{\max}(\sigma^*) < 5/3$.

Case 2. If c_1 belongs to the second piece, and then $\overline{a} = c_1 + b + c_2 + d$. According to step 2 of our algorithm, we have $a < H/3$ and $a + c_1 - H/3 < H/3 - a$. Hence,

$$a > H/3 - c_1/2 \tag{8}$$

Case 2.1 c_2 belongs to the second piece,

According to step 4 of our algorithm, we have $b + c_1 + c_2 - \overline{a}/2 < \overline{a}/2 - c_1 - b$. We have $2c_1 + 2b + c_2 < \overline{a}$, together with $\overline{a} = c_1 + b + c_2 + d$, we have

$$d > b + c_1 \tag{9}$$

If c_1 and c_2 are the same crossover furnace(s), we have $C_{\max}(\sigma^A) = C_{\max}(\sigma^*) = c_1$, otherwise, since $a < H/3$ and $d < c_1 + b + c_2$ and $C_{\max}(\sigma^A) = \bar{a} + c_1 + c_2 = L - a - d$.

1) When $c_1 > H/3$, according to (8) and (9), we have

$C_{\max}(\sigma^A) = L - a - d < L - (H/3 - c_1/2) - (c_1 + b) < 2H/3 - c_1/2$ And combining the equation $C_{\max}(\sigma^*) \ge c_1$ and $c_1 > H/3$, we have

$$\frac{C_{\max}(\sigma^A)}{C_{\max}(\sigma^*)} < \frac{2H/3 - c_1/2}{c_1} < 3/2 < 5/3 .$$

2) When $c_1 \le H/3$, in case of $c_2 > H/3$, due to (3) and (8), and together with (9) we have $c_1/2 + b + c_2 + (c_1 + b) \le 2H/3$, that is,

$c_1 + b < 4H/9 - 2c_2/3 - b/3$.

According to $C_{\max}(\sigma^*) \ge c_2$, $b \ge 0$, we have

$$\frac{C_{\max}(\sigma^A)}{C_{\max}(\sigma^*)} \le \frac{c_1 + b + c_2}{c_2} < \frac{4H/9 + c_2/3}{c_2} < 5/3 .$$

In case of $c_2 \le H/3$, if $c_1 \ge a$, together with (8), we have $c_1 \ge 2H/9$. According to (8) and (9),

$$\frac{C_{\max}(\sigma^A)}{C_{\max}(\sigma^*)} \le \frac{L - a - d}{L/3} < \frac{L - (L/3 - c_1/2) - (c_1 + b)}{L/3}$$

$$= \frac{2L/3 - c_1/2 - b}{L/3} \le 5/3$$

Otherwise if $c_1 < a$ and according to (8), we have $a \ge 2H/9$. Since c_2 belongs to the second piece and step 4 of our algorithm, that is, $d > (L - a - c_2)/2$. Hence,

$$\frac{C_{\max}(\sigma^A)}{C_{\max}(\sigma^*)} \le \frac{L - a - d}{L/3} < \frac{L - a - (L - a - c_2)/2}{L/3}$$

$$= \frac{L - a + c_2}{2L/3} \le 5/3$$

Case 2.2 c_2 belongs to the third piece,

If c_1 and c_2 are the same crossover furnace(s), that is, $d = 0$. Since c_2 must be assigned to the second but not the third piece according to step 4 of our algorithm, thus, we have $C_{\max}(\sigma^A) = C_{\max}(\sigma^*) = c_2 = L - a$. By step 4 of the algorithm, we have $c_1 + b + c_2 - \bar{a}/2 \ge \bar{a}/2 - c_1 - b$, together with $\bar{a} = c_1 + b + c_2 + d$,

$$d \le b + c_1 \tag{10}$$

1) When $c_1 > H/3$, according to (8), we have $a + c_1 > (H/3 - c_1/2) + c_1 > H/2$. Thus,

$C_{\max}(\sigma^A) = L - (c_1 + a) - d < H/2$, and

$$\frac{C_{\max}(\sigma^A)}{C_{\max}(\sigma^*)} < \frac{H/2}{H/3} < 5/3 .$$

2) When $c_1 \le H/3$, according to (3) and (8), we have $c_1/2 + b + c_2 + d \le 2H/3$. Together with (10), we obtain $d/2 + b/2 + c_2 + d \le 2H/3$, that is, $d \le 4H/9 - 2c_2/3$. According to $b \ge 0$, we have

$$\frac{C_{\max}(\sigma^A)}{C_{\max}(\sigma^*)} \le \frac{c_2 + d}{\max\{H/3, c_2\}} < \frac{4H/9 + c_2/3}{\max\{H/3, c_2\}} \le 5/3 .$$

4. Computational experiments

The algorithms described in previous section were programmed in C# 5.0. All tests were run on an x64 PC with an Intel Core i7-3770 3.4 GHz CPU and 8,192MBof RAM. The test problems are generated randomly by considering the following parameters: for each instance, the full loaded move time t followed a uniform discrete distribution in the range [30, 100]. The empty move time et followed a uniform discrete distribution in the range [10, 50]. The number (L) of ladles and the number (F) of furnaces that ranged in [6, 30] and [5, 10], respectively. Now a series of computational experiments are conducted to examine the average performance of this algorithm. The quality is measured by its relative deviation from the LB, $(C_{\max}(\sigma^A) - LB)/LB * 100\%$. The average error ratio (Avg.ER) and the maximum error ratio (Max.ER) measured over the derived lower bound of the *makespan* are used for the performance test.

From Table 1 we can make the following observations: as the problem size or number of furnace and ladle increase, the quality of the proposed heuristic solution is quite good, as the Avg. ER is below 21%. The proposed heuristic generates optimal solutions for the group of the smallest instances because the short move time is negligible relative to the processing time. The quality of the solution remains steady on average when the number of furnaces is not similar to the number of ladles. The average computational times for smaller-scale instances are obtained instantaneously.

Table 1. Average optimal gaps of the lower bounds with respect to our heuristic algorithm

		$t = 50$, $et = 20$			$t = 80$, $et = 30$		
		Max.ER	Avg.ER	Avg.CPU	Max.ER	Avg.ER	Avg. CPU
F=5	L=6	15.205	15.036	0.000	16.501	16.228	0.000
	L=10	18.012	17.925	0.000	19.066	18.859	0.000
	L=20	18.123	18.102	0.003	19.152	19.001	0.003
	L=30	18.308	18.220	0.127	19.375	19.203	0.212
F=8	L=6	16.312	17.010	0.000	19.032	18.699	0.000
	L=10	18.278	18.019	0.000	19.585	19.116	0.000
	L=20	18.436	18.205	0.005	20.039	20.003	0.008
	L=30	18.579	18.328	0.267	20.404	20.015	0.304
F=10	L=6	18.210	17.968	0.000	19.395	19.009	0.000
	L=10	18.306	18.025	0.005	20.310	20.105	0.001
	L=20	18.592	18.269	0.010	20.557	20.277	0.017
	L=30	18.784	18.562	0.561	20.636	20.302	0.810

5. Conclusion

In this paper, we studied the crane scheduling problem in the iron and steel enterprise. The objective function is to minimize the latest refined completion time among all ladles of the melted steel. We explored the property of the studied problem and proposed a heuristic algorithm. Further the worst case performance of the algorithm is demonstrated as 5/3. The result illustrated that the proposed heuristic algorithm can generate robust and acceptable solutions quickly. Future research could investigate an improvement of the computation times, which would allow to take into account a higher number of cranes.

Acknowledgments

This research is supported by National Natural Science Foundation of China (Grant No. 71672117 and 71201104), Liaoning Province Natural Science Foundation (Grant No. 201602526) and China Postdoctoral Science Foundation (Grant No. 202126).

References

[1] Tang, L.X., Liu, J.Y., Rong, A.Y., Yang, Z.H. A review of planning and scheduling systems and methods for integrated steel production. Eur. J. Oper. Res. (2001) 133 (1): 1 – 20.

[2] Daganzo, C.F. The crane scheduling problem. Transp. Res. Part B: Methodological (1989) 23, 159 – 175.

[3] Zhu, Y., Lim, A. Crane scheduling with non-crossing constraint. J. Oper. Res. Soc. (2006) 57, 1464 – 1471.
[4] Lee, D.H., Wang, H.Q., Miao, L. Quay crane scheduling with non-interference constraints in port container terminals. Transp. Res. Part E: Logis. Transp. Rev. (2008)44, 124 – 135.
[5] Kim, K.H., Park, Y. M. A crane scheduling method for port container terminals. Eur. J. Oper. Res., (2004)156 (3), 752 – 768.
[6] Lim, A., Rodrigues, B., Xu, Z. A m-parallel crane scheduling problem with a non-crossing constraint. Nav. Res. Log. (2007)54, 115 – 127.
[7] Liu, M., Wang, S.J., Chu, F., Xu, Y.F. Some complexity results and an efficient algorithm for quay crane scheduling problem. Discrete Mathematics, Algorithms and Applications Vol. 8, No. 4 (2016) 1650058 (19 pages).
[8] Briskorn, D., Emde, S., Boysen, N. Cooperative twin-crane scheduling. Discrete Appl. Math. (2016) 211, 40 – 57.
[9] Zhang, A., Zhang, W.S., Chen,Y., Chen, G.T., Chen, X.F. Approximate the scheduling of quay cranes with non-crossing constraints. Eur. J. Oper. Res. (2017) 258, 820 – 828.
[10] Lim, A., Rodrigues, B., Xu, Z. Approximation schemes for the crane scheduling problem. In T. Hagerup, & J. Katajainen (Eds.), Proceedings of the ninth Scandinavian workshop on algorithm theory, SWAT2004 . In LNCS: (2004b) 3111 (pp. 323–335).
[11] Lee, D.H., Wang, H.Q., Miao, L. An approximation algorithm for quay crane scheduling with noninterference constraints in port container terminals. In Proceedings of the sixth triennial symposium on transportation analysis, TRISTAN VI . Phuket. June (2007), 10 – 15.