Scheduling Continuous Aluminium Casting Lines

Ekrem Duman
Dogus University, Acibadem, Istanbul, Turkey

Mehmet Bayram Yildirim
Wichita State University, bayram.yildirim@wichita.edu

Ali Fuat Alkaya
Marmara University, Goztepe, Istanbul, Turkey

Scheduling Continuous Aluminum Casting Lines

Ekrem Duman,* Mehmet Bayram Yildirim,[+] and Ali Fuat Alkaya*****

* Industrial Engineering Department, Dogus University, Acibadem, Istanbul, Turkey
** Department of Industrial and Manufacturing Engineering, Wichita State University, 1845 Fairmount, Wichita, KS 67260-0035
*** Computer Engineering Department, Marmara University, Goztepe, Istanbul, Turkey

+Corresponding author, email: bayram.yildirim@wichita.edu

Scheduling Continuous Aluminum Casting Lines

Abstract:
This study considers the problem of scheduling casting lines of an aluminum casting and processing plant. In aluminum processing plants, continuous casting lines are the bottleneck resources, i.e., factory throughput is limited by the amount of aluminum that can be cast. The throughput of a casting line might be increased by minimizing total setup time between jobs. The objective is to minimize setup time on production lines for a given time period while balancing workload between production lines to accommodate potential new orders.. A mathematical formulation for scheduling jobs to minimize the total setup time while achieving workload balance between the production lines is presented. Since the casting scheduling problem is an NP-hard problem, even with only one casting line, a four step algorithm to find efficient solutions in a reasonable amount of time is proposed. In this process, a set of asymmetric traveling salesman problems is followed by a pairwise exchange heuristic. The proposed procedure is applied to a case study from a real casting data.

Keywords: Aluminum cast scheduling, asymmetric traveling salesman problem, heuristics

1. Introduction

This study is inspired by a production scheduling problem in a large aluminum casting and processing factory in Istanbul, Turkey. This company casts aluminum coils of different alloys in various widths and thicknesses. According to customer specifications, these coils are then rolled to the desired dimensions and properties through cold milling, annealing, surface processing, and cutting operations. Rolled aluminum coils are used widely in manufacturing, including the production of aircraft, satellites, space laboratory structures, tankers and freight wagons, buses, truck bodies, tankers, radiators, traffic signs and lighting columns, chemical process plants, chemical carriers, food handling and processing equipment, packaging, cans, bottle caps, wrapping, packs, and containers.

The first operation in the manufacturing process of rolled aluminum is the casting operation, whereby ingots or slabs of pure aluminum are melted in a furnace. Then, additive elements such as magnesium, vanadium, etc., which determine the alloy of the aluminum, are added to the furnace in specific amounts. The melted metal in the furnace flows through a shaper and a rolling mill (see figure 1). The entire process takes place on a "casting line" (CL). In this type of process, continuous production is required to avoid reheating a furnace after it cools down, which is long and costly. After the casting operation, to obtain the desired width and thickness, the metal is fed in coil form through a series of cold rolling mills, which successively reduce the metal thickness and recoil it after each rolling pass, thus preparing it for the next pass, until the required thickness is obtained. Annealing may be required between passes, depending on the final temper required. Then, surface processing and cutting operations follow.

Casting is also a determinant of factory throughput. Production is limited by the amount that can be cast. In this case study, although the subsequent cold milling resources sometimes resulted in a bottleneck (in the case where the majority of orders consists of thin products), usually the casting lines were the actual bottleneck resources. Therefore, maximally utilizing casting lines (CLs) is a goal that should be considered.

[Insert Figure 1 around here]

On a casting line, several operations can be viewed as setups, most of which are sequence-dependent since the order of jobs on a production line affects the length of setup. These setups occur when different alloys are cast and/or when the width of the next job to be processed is different than the width of the current job. The objective of scheduling is to maximize the utilization of CLs and, at the same time, balance their loads in anticipation of future orders.

In some ways, the steel manufacturing industry is similar to the aluminum industry. Tang et al. (2001) review various manufacturing technologies used in the steel industry, and provide information on the planning and scheduling methods. Another paper by the same authors (Tang et al. 2000) undertakes the scheduling of hot rolling (similar to cold milling in the aluminum factory) in a steelmaking factory. They formulate the problem as a multiple traveling salesman problem and find a solution using genetic algorithms. Harjunkoski and Grossmann (2001) discuss the scheduling of pre-casting and casting operations in a steel plant. Instead of trying to solve the aggregate problem, they dissect the problem and solve it in four phases. Lopez et al.

(1998) formulate the hot strip mill production scheduling as a generalization of the prize-collecting TSP and solve it using the tabu search metaheuristic procedure. Gravel et al. (2002) considers the scheduling of a single aluminum casting line, where the scheduling objective is similar to ours. They formulate the problem as a traveling salesman problem (TSP) and suggest a solution methodology based on ant colony optimization metaheuristics.

Our problem is somewhat similar to parallel-machine scheduling problems with setups (in our case, only some of the machines are identical). Chen and Powell (2003) provide an exact algorithm for this problem, which is based on a column generation technique in the branch-and-bound method.

The following section describes a casting scheduling problem in more detail and then provides a mathematical formulation. In section three, a four-phase solution methodology is developed to determine near-optimal solutions to the mathematical program. In section four, the proposed solution methodology is illustrated on manufacturing data obtained from an aluminum processing plant, and the results are analyzed in detail. Finally, section five provides a short summary of the study and proposes possible future research directions.

2. Problem Description and Formulation

The following subsections provide, first, a more detailed description of the casting scheduling problem and, then, a mathematical formulation.

2.1. Problem Description

The products ordered by customers may have various properties. These orders are scheduled on parallel production lines based on their size, width, thickness, and alloy. Important factors that must be considered when designing a casting schedule can be categorized as line specifications and job specifications:

Line Specifications:

i- *Width*

The width that each casting line can cast is limited. It is assumed that some production lines can cast up to three different widths: narrow, medium, or wide aluminum coils. In other words, production lines are not identical.

ii- *Speed*

It is assumed that the difference in the processing speed of production lines is negligible.

Job Specifications:

i- *Width*

If two consecutively scheduled jobs have different widths, then there will be a setup time after the first job. For two consecutive jobs, the setup time when the width of the second job is narrower than the first one is shorter than the setup time when the first job is narrower than the second. This is because both shaper and roller must be changed to process a wider job.

ii- *Alloy*
If an alloy change occurs between two consecutive jobs, then a significant setup could be required. Most of the time, no significant setup is required to process a more composite alloy after a purer alloy (i.e., some additive elements are added to the furnace and then production continues). However, in the reverse case, i.e., if a purer alloy is to be cast after a composite alloy, then the furnace must be cleaned thoroughly (i.e., hot cleaning), and this process usually takes significantly longer than the previous case.

iii. *Last Job of Previous Schedule*
An important consideration in minimizing setup time involves the properties of the last jobs cast on lines in the previous planning period. The current planning period's schedule should take this into account as the initial condition.

iv. *Changes in Thickness of Orders*
Other than the above specifications, jobs may have different casting thicknesses. However, any change in thickness can be handled in a very short amount of time and minimally impacts setup in a casting-scheduling problem.

The next subsection provides the mathematical programming model for the problem definition described above.

2.2. Mathematical Formulation

Let K be the set of casting lines. $|K|$ is the number of casting lines. Let $J=\{1,\ldots n,n+1,..n+|K|\}$ be the set of jobs where first n are the jobs to be produced in the current planning period, and the last $|K|$ are the last jobs produced on the corresponding casting line in the previous period. Furthermore, let K_j be the set of production lines on which order j can be processed, and let J_k be the set of jobs that can be processed on production line k. Let p_{ik} be the processing time of job i on casting line k, and let s_{ijk} be the setup time incurred if job i immediately precedes job j on casting line k. The parameter α, which takes values between zero and one, constrains the level of maximum imbalance. The variables that we have utilized in this formulation are C_{total} (the total processing and setup time of all jobs) C_k (the total processing and setup time on line k), and x_{ijk} and y_{ik}, where

$$y_{ik} = \begin{cases} 1 & \text{if job } i \text{ is assigned to casting line } k \\ 0 & \text{otherwise} \end{cases}$$

and $$x_{ijk} = \begin{cases} 1 & \text{if job is } i \text{ the immediate predecessor of job } j \text{ on casting line } k \\ 0 & \text{otherwise} \end{cases}$$

Then, the problem formulation becomes

$$
\begin{aligned}
\min \quad & C_{total} & & & \\
& C_{total} = \sum_{k \in K} C_k & & & (1)\\
& C_k = \sum_{i=1}^{n} y_{ik} p_{ik} + \sum_{i \in J_k} \sum_{j \in J_k} x_{ijk} s_{ijk} & & \forall k \in K & (2)\\
& C_k \le \frac{1}{m} C_{total}(1+\alpha) & & \forall k \in K & (3)\\
& C_k \ge \frac{1}{m} C_{total}(1-\alpha) & & \forall k \in K & (4)\\
& \sum_{k \in K_i} y_{ik} = 1 & & i = 1..n & (5)\\
& y_{ik} = 1 & & i = n+k \text{ and } k = 1..|K| & (6)\\
& x_{ijk} \le y_{ik} & & \forall i, j \in K_i, k \in K & (7)\\
& \sum_{i \in J_k} x_{ijk} = 1 & & \forall k \in K, j \in J_k & (8)\\
& \sum_{j \in J_k} x_{ijk} = 1 & & \forall k \in K, i \in J_k & (9)\\
& \sum_{i \in S} \sum_{j \in S, j \ne i} x_{ijk} \le |S_k| - 1 & & S_k \subseteq J_k & (10)
\end{aligned}
$$

In the above formulation, the objective is to minimize the total time needed to complete all jobs on all casting lines. Constraint (1) determines the total processing time of jobs on all lines, while constraint (2) determines the processing time of a particular CL. Constraints (3) and (4) are workload balancing constraints. Since a perfect balance may not (and most probably will not) be obtainable, a certain percentage (α) above or below the ideal average workload is acceptable. Typical values of α could be in the range of 0.05 to 0.30. Constraint (5) guarantees that each job is assigned to a casting line. Similarly, using constraint (6), $|K|$ additional jobs corresponding to the last jobs of the previous month are assigned to the corresponding casting lines. Constraint (7) ensures that a job cannot precede another job unless both jobs are assigned to the same line. Constraints (8) and (9) give a more detailed assignment of jobs to casting lines, i.e., information on job precedence relations is determined. The subtour elimination constraints are given by constraint (10). In this equation, S_k is a subst of jobs that can be processed on casting line k.

In the next section, the mathematical formulation is analyzed in detail, and a solution procedure is suggested.

3. Solution Methodology

The problem formulated in the previous section provides a means of minimizing the total setup time and attaining a balanced workload among the casting lines. In fact, the formulation given resembles the well-known vehicle routing problem (VRP), where the vehicle capacities are somewhat fuzzy. If the constraint sets (3) and (4) given for workload balancing are modified to say, for example, that "the workload assigned to any line cannot exceed 30 days," then our problem can be modeled exactly as a VRP with non-identical vehicle capacities. The VRP is one of the most difficult problems of combinatorial optimization and is NP-hard (Ghiani et al., 2003). Laporte et al. (2000) provides a survey of classical and modern heuristics for VRP. Some of the efficient solution procedures for the VRP can be found in Yoshiike and Takefuji (2002), Baker and Ayechew (2003), and Martinhon et al. (2004).

For a problem involving 20 jobs in a plant with five casting lines, running CPLEX 9.1 for three days (72 hours) on a 2.8Ghz Pentium IV computer with one gigabyte of memory did not provide proof of optimality. A typical problem in an aluminum processing plant is usually much larger. As a result, a four-step framework that would provide good solutions in a reasonable amount of time is proposed.

When the secondary load-balancing objective is disregarded (i.e., constraints (3) and (4) are relaxed), maximization of the resource utilization objective is equivalent to minimization of the total setup time on each production line (which can be modeled as a variant of the VRP). Additionally, if the job-line assignment decision is known, then the setup minimization problem turns out to be an asymmetric traveling salesman problem (ATSP) for each casting line. As a matter of fact, a decomposition algorithm may be suggested for the master problem.

When jobs are assigned to the casting lines in such a way that the balancing constraints are satisfied, then the rest of the problem reduces to minimizing the setup time on each individual CL. Technically, if the values of the y_{ik}-decision variables are known and the constraint sets (3), (4) and (5) are satisfied, then when the total completion time is calculated, $\sum_{i=1}^{n} y_{ik} p_{ik}$ will be a constant. Then, the minimization should be done over $\sum_{i \in J_k} \sum_{j \in J_k} x_{ijk} s_{ijk}$, i.e., the objective will be minimization of the total setup times on $|K|$ casting lines. In addition, the problem will decompose into $|K|$ subproblems, since when y_{ik} is fixed, the setup times on one casting line do not depend on the setup times on other casting lines. Thus, the objective becomes to minimize the setup time on each CL, and the objective function value will be the sum of these setup times plus the constant sum of processing times. The resulting formulation with constraint sets (8), (9), and (10) is the asymmetric traveling salesman problem. It is asymmetric in the sense that the setup time matrix (which represents the total setup time needed between jobs as a result of width changes and alloy type changes) is asymmetric.

At this point, by naming the originally formulated problem as the *master problem* and the resulting ATSPs after the decomposition as the *subproblems*, it is possible and reasonable to define a four-phase solution procedure to the master problem:

i. *Line Assignment*
Assign jobs to casting lines in a balanced manner.

ii. *Solution of Subproblems*
Solve the resulting ATSPs.

iii. *Improvement*
Try to improve the current solution of the master problem by considering changes in the line assignment made in phase 1.

iv. *Re-Solution of Subproblems*
Solve the ATSPs again to detect any possible improvements, since the original solutions may have deteriorated in the third phase.

A detailed description of these phases is as follows:

<u>Line Assignment</u>

Order the jobs in non-decreasing order of their widths. The casting line that can process narrower jobs has smaller indexes. The total processing time of all jobs that are planned to be scheduled during a planning period is calculated, and this amount is divided by the number of casting lines (m) to find the desired workload per CL (DWPCL) without considering the effect of setup times. Starting from the top of the list and the smaller index casting line, jobs are assigned to the CL if that assignment brings a decrease in the absolute value of the difference between the DWPCL and the total assigned workload. If not, the next job on the list is tried for assignment. The same procedure is used for all lines until all jobs are assigned.

Solution of Subproblems

Solve the resulting m ATSPs optimally for each line. However, before solving each ATSP, prepare the cost matrix (width-change setups plus alloy-change setups) by looking at the data and including the last job of the previous planning period as a city of the ATSP. Take the setup time between any regular job and the last job of the previous planning period as zero, and then arrange the directed ATSP route so that the last job of the previous planning period becomes the home city.

Improvement

Note that the quality of the solution procedure described thus far is primarily dependent on how well the line assignment of phase 1 is made. Changing the line assignment and then solving the ATSPs repeatedly is not easy and intractable in terms of the run time required. Therefore, a pairwise exchange on the solution obtained at the end of phase 2 is more reasonable and is described as follows:

Randomly determine any two jobs on all $|K|$ CLs (exclude the first jobs that correspond to the last jobs of the previous planning period), and check to see if they are exchangeable (i.e., width capacities of CLs are not violated and workloads of the corresponding lines are still within the tolerated range). If they are exchangeable, and if the total processing time is the same or would be improved, then perform the exchange of these two jobs. Continue this procedure until a predetermined number of exchange trials is made.

Note that, in this procedure, if the two randomly determined jobs happen to be on the same CL, then selection is still considered. Such a selection will hardly bring an improvement in the earlier phases since the sequence of each CL is determined optimally, but it may be the base for improvement in later iterations. First, it may provide a means of moving away from the current local optimum, and second, in later iterations, it may improve the sequence on a casting line since the original optimum ATSP solution may have deteriorated.

Re-Solution of Subproblems

Note that the improvement phase may have caused deterioration in the ATSPs that were solved in the second phase. Thus, it is advisable to solve them again to detect any potential improvement.

4. Case Study in a Large-Sized Aluminum Processing Plant

In this section, the four-phase solution procedure, described above, is illustrated using a case study from a large-sized aluminum processing plant. It begins with a description of the real data, whereby the planning period is defined as a month (January to October 2004), and continues with the experimental runs and the economic interpretation of the improvement obtained.

The aluminum processing plant that is analyzed has five casting lines. Each CL can process orders up to a certain width. These limits are 1,400 mm for CL1, 1,700 mm for CL2, and 2,200 mm for CLs 3, 4, and 5. The processing speed of CLs varies with respect to the alloy. However, although there is a slight difference in processing time between the speed of CL1 and the others, the casting speed of any particular alloy is considered to be equal on different lines and, thus, will not influence the schedule. In other words, p_{ik}, the processing time of order i on line k, is assumed to be p_i.

On any casting line, when orders are scheduled, if the width of the next job is smaller, the shaper (which determines the desired width) must be changed. This operation (setup) takes 2.5 hours. On the other hand, if the width of the next job is larger, then both the shaper and the roller must be changed, and this takes 6 hours. To decrease setup times due to job widths, the plant is implementing some standard casting widths (mm): 1,220, 1,280, 1,320, 1,400, 1,450, 1,550, 1,600, 1,650, 1,700, 1,830, 2,080, 2,120, 2,140, and 2,200. The standard thicknesses are 3 mm and 5 mm for alloy 1050, and 4 mm and 6 mm for the other alloys.

When different types of alloys are scheduled consecutively, the casting lines may require a setup time that includes a hot cleaning operation that takes 10 hours on CL1 and 15 hours on CLs 2, 3, 4, and 5 (see table 1 for the detailed setup time for the alloys produced in this plant)

[insert table 1 about here]

4.1. Structure of Demand Data

In the aluminum processing plant, the schedule of the casting lines is prepared on a monthly basis. A typical-sized order is 3 to 10 tons. On the average, 600 to 1,800 orders are received each month, a number so high that it is almost impossible to solve a problem of this size in the current practical setting. Thus, orders are combined in such a way that they have sufficient common properties to be cast at the same time (e.g., orders having similar alloy types and widths). As a result of order consolidation, the number of jobs to be scheduled on a casting line decreases significantly, and the setup minimization objective for the group of consolidated jobs is achieved. As a result of consolidation, a casting job of 50 to 400 tons is formed, and the number of jobs to be scheduled on the casting line is reduced to 40 to 50 (8 to 10 jobs per line).

A week before the beginning of each month, production planners determine the schedule for that month. The desired schedule should make all casting lines fully loaded and perfectly balanced (i.e., loading all CLs until the midnight of the last day of the month). As expected, the casting jobs at hand do not allow for a schedule with such a perfect fit. In such cases, after preparing the schedule for each casting line, planners typically increase the amount of some popular alloy types (such as 1050 and 3003) so that the lines are loaded exactly until the end of the month. In doing so, they assume that the extra amount of cast alloy can be allocated to customer orders received during that month or the following month, at the latest. In this way, they obtain schedules that perfectly fit into the month.

A sample of consolidated data (for February) is given in table 2. This data provides information on the alloy, width, quantity, and processing time required to finish a cast job. For this demand data, figure 2 shows a Gantt chart that represents the implemented schedule. In the figure, the setup times are lightly shaded whereas the darkly shaded area on CL2 represents a periodic maintenance performed. In table 2, the first 39 jobs are those cast in February, whereas the last five jobs are those cast as the last jobs in January. The production planning department at this aluminum plant designed a schedule that takes 3,310.6 hours for overall processing and setup time of these orders.

[insert figure 2 about here]

[Insert Table 2 about here]

Normally, we would assume zero setup time and zero processing time for the last five jobs completed in the previous month. However, to handle the regular maintenance on each CL, the processing time of the last jobs of the previous month is assumed to be equal to the expected maintenance time on the respective CL.

4.2. Experimentation on Real Data

The implementation of phases 1, 3, and 4 of our solution procedure is rather straightforward. In phase 3 (the improvement phase), we need to decide on the number of pairwise exchange trials (i.e., the number of iterations). In our base setting, we took it to be n^3 (*n* being the number of jobs), but we also tested for n^2 pairwise exchanges.

On the other hand, the value of α, which indicates the level of tolerance in the workload imbalance of a casting line (constraint 2-3 in the problem formulation), is an important design parameter. We tested three different values of α (0.05, 0.10, and 0.30), where $\alpha = 0.05$ is an indication of an almost perfectly balanced schedule, and $\alpha = 0.30$ is an indication of a loosely balanced but still a reasonable schedule. However, regarding discussions on how the company manipulates the job amount, the choice of $\alpha = 0.30$ is suitable to represent the amount of benefit that the company can get from our solutions. The total improvement in overall processing time and setup time obtained for $\alpha = 0.30$ is provided in table 3. Here, the first row represents the company's schedule (CS) for months January through October. When the proposed framework is applied to the CS, after phase one (P1), an increase in the overall time occurs (recall that phase one includes an assignment using the DWCPL rule). In phase two (P2), the setup time on each line is minimized using the GAMS/CPLEX solver to work out the resulting asymmetric traveling salesman problems. These results are further improved by performing random exchanges in phase three (P3). Finally, in phase four (P4), the best available solution P3 is further improved by resolving the asymmetric traveling salesman problems optimally using GAMS. The overall procedure is relatively fast. On the average, it takes 3.90 CPU seconds on a Pentium IV Windows XP machine with 512 MB of RAM and 80 GB of hard disk space. As can be expected, the biggest improvement occurs in P2. Similar observations can be made on improvements in total setup time, which are presented in table 4. Table 4 also displays the number of pairwise exchange trials (# of iterations), the number of exchanges made (# of exchanges) and their ratio (# of exc. / # of iter.).

[insert table 3 about here]
[insert table 4 about here]

The summary of results for overall processing time and setup time for different values of α and number of iterations in phase 2 is given in table 5 (table 2 shows the results of total setup time). As can be seen in table 5, the improvement, compared to the company schedule, is 2.59% for $\alpha = 0.05$, 3.03% for $\alpha = 0.10$, and 3.34% for $\alpha = 0.30$. As can be expected, improvement increases the more imbalance is tolerated. These improvements are obtained in the total processing time, and they are quite meaningful, as discussed in the following subsection. If we look at improvements in the total setup time, we see that these figures range from 33.07% to 45.31%. Regarding the unavoidable setup times due to width changes (at the least), we can conclude that our solutions are very close to the optima.

If we look at the improvements gained at each phase, we observe that phase 1 makes the company solution worse, which is natural. Phase 2 yields the most improvement, followed by phase 3. On the other hand, the improvement in phase 4 is very little. This indicates that pairwise exchanges perform fairly well, with little room for optimization after their completion.

In our base experimental setting, the number of iterations is taken to be n^3. In this case, the percentage of successfully implemented pairwise exchanges appears to be quite small (2.18%). For this reason, we also tested n^2 as the limit on the number of iterations. That is, we wanted to determine if too much computational time is lost by trying n^3 iterations. This time we see that improvement figures decrease (e.g., from 3.34% to 3.12%, when $\alpha = 0.30$), indicating that although successful exchanges are few, an increase in the number of iterations helps in finding a higher number of successful implementations.

[insert table 5 about here]
[insert table 6 about here]

Table 7 shows detailed CPU times required to execute the proposed framework for months June and September. Results indicate that phase 2 and phase 4 are the most time-consuming steps of the framework. The random exchange takes less than 1% of the time needed to solve the asymmetric traveling salesman problem when n^2 random exchanges are performed. For n^3 random exchanges, the CPU time for random exchanges increases drastically to a maximum of 29% of the CPU time to solve the asymmetric travelling salesman problem. Results also indicate that there is no direct relation between α values and CPU times.

[insert table 7 about here]

Finally, out of curiosity, we speculated on what would happen if we skipped phase 2, in which case first, pairwise exchanges would apply (phase 3) and then ATSPs would be solved optimally using GAMS (phase 4). The results are given in table 8. We see that the improvement gained in step 3 is reduced (from 3.28% to 2.91%). This is normal because when we started with a worse solution (phase 1 result), the pairwise exchange procedure failed to obtain its previous best result, even though more exchanges were implemented successfully (2.23% versus 2.18%). However, fortunately, the end result obtained after phase 4 was better than the previous case (3.45% versus 3.34%). This phenomenon can be explained as follows:

In the standard design, optimal solutions of the ATSPs are based on poor line assignment. But in the latter case, line assignment is improved initially by pairwise exchanges. Then, when the ATSPs are solved optimally, it was possible to arrive at better results in a shorter time. Thus, we modify our suggested solution procedure as one having phases 1, 3, and 4 and skipping phase 2.

[insert table 8 about here]

The economic benefit that the aluminum factory could gain if our solution procedure is used can be calculated as follows: In the base scenario ($\alpha = 0.30$), the improvement in the total processing time is 3.45%. This means an additional 124 hours of additional casting capacity in a 30-day month. The average amount that can be cast in 124 hours is 230 tons. The current average price per ton of the processed aluminum is approximately $2,200 USD. Assuming that the company can process and sell all aluminum that it can produce, this means $506,000 USD more revenue monthly, or $6,072,000 USD more revenue yearly.

5. Concluding Remarks

Lean manufacturing strategies aim for manufacturing without waste. Waste might take several forms, e.g., material, time, and inventory. In this paper, our goal is to minimize the total sequence dependent setup time involved in the scheduling of casting lines in a major aluminum casting and processing factory while achieving balance among production lines. Using the proposed four-phase solution procedure, on the average the total setup time required to process all incoming demand is reduced from 232.5 hours per month (roughly 6.41% of the overall production time) to 124.9 hours per month (3.56% of the overall production time after improvement). In other words, the total setup time is reduced to 53.7% of the setups incurred as a result of the current schedule. Thus, one can conclude that the results obtained when the solution procedure is applied to a set of real data are quite successful. Using our procedure, the factory can make $6,072,000 USD more revenue in a year.

Applying our procedure to an order combination procedure (the case of more casting jobs with smaller amounts) is an interesting idea requiring further study. Another problem that can be studied is the one where surface quality specifications is considered. Depending on the customer specification and/or the production characteristics of the customers, production scheduling department might decide on the minimum surface quality requirement for each casting job. When a new roller is installed, products with the best surface quality are obtained up to 400 hours. Then, for the next 400 hours of operation, the surface quality obtained is classified as medium. The same roller can be continually used for an additional 400 hours. After 1,200 hours, the roller is changed.

References

ASSAF, I. CHEN, M. and KATZBERG, J. 1997, Steel production schedule generation *International Journal of Production Research*, 35, 467-477.

BAKER, B. M., and AYECHEW, M. A., 2003, A genetic algorithm for the vehicle routing problem. *Computers and Operations Research*, **30**, 787-800.

CHEN, Z. L., and POWELL, W. B., 2003, Exact algorithms for scheduling multiple families of jobs in parallel machines. *Naval Research Logistics*, **50**, 823-840.

GHIANI G., GUERRIERO, F., LAPORTE, G., and MUSMANNO, R., 2003, Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies. *European Journal of Operational Research*, **151**, 1-11.

GRAVEL, M., PRICE, W.L., and GAGNE, C., 2002, Scheduling continuous casting of aluminum using a multiple objective ant colony optimization metaheuristic. *European Journal of Operational Research*, **143**, 218-229.

HARJUNKOSKI, I., and GROSSMANN, I. E., 2001, A decomposition approach for the scheduling of a steel plant production. *Computers and Chemical Engineering,* **25**, 1647-1660.

HONG, L. and SHANG, J. 2001, Integrated model for production planning in a large iron and steel manufacturing environment, International Journal of Production Research, 39, 2037- 2062.

LAPORTE, G., GENDREAU, M., POTVIN, J., and SEMET, F., 2000, Classical and modern heuristics for the vehicle routing problem, *International Transactions in Operational Research*, **7**, 285-300.

LI, Y., SHEN, GHENNIWA, W. and WANG, C., 2005, A desired load distribution model for scheduling of unrelated parallel machines, International Journal of Production Research, 43, 5033–5046

LOPEZ, L., CARTER, M. W. and GENDREAU, M. 1998, The hot strip Mill production scheduling problem: A tabu search approach. European Journal of Operational Research, 106 (2-3), 317-335.

MARTINHON, C., LUCENA, A., and MACULAN, N., 2004, Stronger K-tree relaxations for the vehicle routing problem. *European Journal of Operational Research*, **158**, 56-71.

TANG, L., LIU, J., RONG, A., and YANG, Z., 2000, A multiple traveling salesman problem model for hot rolling scheduling in Shanghai Iron & Steel Complex. *European Journal of Operational Research*, **124**, 267-282.

TANG, L., LIU, J., RONG, A., and YANG, Z., 2001, A review of planning and scheduling systems and methods for integrated steel production. *European Journal of Operational Research*, **133**, 1-20.

TANG, L., LUH, P. B., LIU, J. and FANG, L. 2002 Steel-making process Scheduling using lagrangian relaxation. International Journal of Production Research, 40 (1), 55-70.

YOSHIIKE, N., and TAKEFUJI, Y., 2002, Solving vehicle routing problems by maximum neuron model. *Advanced Engineering Informatics*, **16**, 99-105.

Table 1. Alloy Change Setup Matrix for CLs 2, 3, 4, and 5 (for CL1, replace 15 hours with 10 hours).

	1050	**1100**	**1200**	**1235**	**8011**	**8079**	**8006**	**8111**	**3003**	**3005**	**3105**	**5005**	**5049**	**5052**	**5182**	**5754**
1050	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1235	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8011	15	15	0	0	0	0	0	0	15	15	15	15	0	15	15	15
8079	15	15	15	15	0	0	0	0	15	0	0	15	15	15	15	15
8006	15	15	15	15	15	15	0	15	15	15	15	15	15	15	15	15
8111	15	15	15	15	0	15	15	0	15	15	15	15	15	15	15	0
3003	15	15	15	15	15	15	15	15	0	0	15	15	15	15	15	15
3005	15	15	15	15	15	15	15	15	15	0	15	15	0	15	15	15
3105	15	15	15	15	15	15	15	15	15	0	0	0	0	0	0	0
5005	15	15	15	15	15	15	15	15	15	0	15	0	0	0	0	0
5049	15	15	15	15	15	15	15	15	15	15	15	15	0	15	15	15
5052	15	15	15	15	15	15	15	15	15	15	15	15	15	0	0	0
5182	15	15	15	15	15	15	15	15	15	15	15	15	15	15	0	15
5754	15	15	15	15	15	15	15	15	15	15	15	15	15	0	0	0

Table 2. Casting Data for February 2004.

Job No	Alloy	Width (mm)	Quantity (tons)	Proc. Time (hrs)
1	3105	1320	50	40,5
2	1050	1320	200	139,4
3	1100	1320	50	34,8
4	3105	1320	200	162,1
5	8006	1220	85	74,5
6	1050	1320	220	153,3
7	3003	1320	40	32,4
8	3105	1320	110	86,7
9	1050	1700	150?	66,2
10	5005	1700	273	167
11	8011	1660	150	85,8
12	8111	1660	130	94
13	8011	1460	50	32,5
14	1050	1700	220	97,1
15	3003	2120	150	67,9
16	8006	2120	240	107,5
17	1050	2120	150	53,1
18	1050	2120	150	53,1
19	8111	2120	160	90,6
20	3003	2120	150	67,9
21	3003	2120	50	22,6
22	3003	2120	150	67,9
23	3003	2120	230	104,2
24	8111	1660	150	108,4
25	1050	2200	25	8,5
26	5005	2200	200	94,5
27	3005	2200	260	141,8
28	5049	2120	100	68,9
29	1050	2200	200	78,4
30	5005	2200	100	47,3
31	1050	2120	70	24,1
32	3003	2200	40	17,5
33	1050	2200	400	156,8
34	1200	2200	50	18
35	8111	2120	140	79,2
36	8111	1660	70	50,6
37	1050	2200	300	102,3
38	1100	2200	100	35,9
39	3105	2200	330	156
40	3105	1320	0	0
41	3105	1320	0	0
42	3003	2120	0	0
43	8111	1660	0	46
44	3003	2200	0	0

Table 3. Improvements Obtained in Total Time and Setup Time (n^3 iterations, $\alpha = 0.30$) (total time = processing + setup hours).

	Jan	**Feb**	**Mar**	**Apr**	**May**	**Jun**	**Jul**	**Aug**	**Sep**	**Oct**	**Average**
CS	3688.2	3310.6	3712.7	3590.6	3712.7	3565.6	3696	3707.7	3590	3701.6	3627.6
After P1	3771.4	3397.3	3894.2	3714.7	3726.1	3617.1	3736.3	3704.3	3626.6	3752.6	3694.1
After P2	3646.9	3270.8	3654.2	3520.2	3559.6	3473.6	3585.8	3531.3	3494.1	3644.6	3538.1
After P3	3619.4	3249.8	3613.9	3506	3546.4	3447.3	3553	3481.4	3445.7	3617.8	3508.0
After P4	3617.4	3248	3611.3	3503.2	3544.4	3446.1	3548.9	3479.8	3443.6	3616.5	3505.9
CPU Time (seconds)	3.68	3.51	4.08	4.44	4.38	4.56	3.51	3.72	3.21	3.55	3.90
Improvement											
(CS-P1)/CS	-2.26%	-2.62%	-4.89%	-3.46%	-0.36%	-1.44%	-1.09%	-0.09%	-1.02%	-1.38%	**-1.84%**
(CS-P2)/CS	1.12%	1.20%	1.58%	1.96%	4.12%	2.58%	2.98%	4.76%	2.67%	1.54%	**2.45%**
(CS-P3)/CS	1.87%	1.84%	2.66%	2.36%	4.48%	3.32%	3.87%	6.10%	4.02%	2.27%	**3.28%**
(CS-P4)/CS	1.92%	1.89%	2.73%	2.43%	4.53%	3.35%	3.98%	6.15%	4.08%	2.30%	**3.34%**

Table 4. Improvements Obtained in Total Time and Setup Time (n^3 iterations, $\alpha = 0.30$) (setup time in hours).

	Jan	**Feb**	**Mar**	**Apr**	**May**	**Jun**	**Jul**	**Aug**	**Sep**	**Oct**	**Average**
CS	193.5	175	221	239	316	262.5	255	227.5	247.5	187.5	232.5
After P1	276.5	262	403	360	329	314	310.5	338.5	284	253.5	313.1
After P2	152	135.5	163	165.5	162.5	170.5	160	165.5	151.5	145.5	157.2
After P3	124.45	114.5	122.7	151.25	149.25	144.2	127.15	115.55	103.1	118.65	127.1
After P4	122.45	112.7	120.05	148.5	147.25	143	123.1	113.95	100.95	117.35	124.9
# of exchanges	1947.3	1846.9	1822.5	4202.6	3214.5	4030.2	2159.8	2727.5	1200.4	3117.5	2626.9
# of iterations	110592	85184	125000	175616	175616	195112	91125	110592	64000	91125	122396.2
CPU Time (seconds)	3.68	3.51	4.08	4.44	4.38	4.56	3.51	3.72	3.21	3.55	3.90
Improvement											
(CS-P1)/CS	-42.89%	-49.71%	-82.35%	-50.63%	-4.11%	-19.62%	-21.76%	-48.79%	-14.75%	-35.20%	**-36.98%**
(CS-P2)/CS	21.45%	22.57%	26.24%	30.75%	48.58%	35.05%	37.25%	27.25%	38.79%	22.40%	**31.03%**
(CS-P3)/CS	35.68%	34.57%	44.48%	36.72%	52.77%	45.07%	50.14%	49.21%	58.34%	36.72%	**44.37%**
(CS-P4)/CS	36.72%	35.60%	45.68%	37.87%	53.40%	45.52%	51.73%	49.91%	59.21%	37.41%	**45.31%**
# of exc. / # of iter.	1.76%	2.17%	1.46%	2.39%	1.83%	2.07%	2.37%	2.47%	1.88%	3.42%	**2.18%**

Table 5. Improvement Figures Obtained for Different Settings (Total Time).

# of iter.	Phases	Total time		
		$\alpha = 0.05$	$\alpha = 0.10$	$\alpha = 0.30$
n^2	(CS-P1)/CS	-1.84%	-1.84%	-1.84%
	(CS-P2)/CS	2.45%	2.45%	2.45%
	(CS-P3)/CS	2.52%	2.69%	2.96%
	(CS-P4)/CS	2.55%	2.79%	3.12%
n^3	(CS-P1)/CS	-1.84%	-1.84%	-1.84%

	(CS-P2)/CS	2.45%	2.45%	2.45%
	(CS-P3)/CS	2.57%	2.96%	3.28%
	(CS-P4)/CS	2.59%	3.03%	3.34%

Table 6. Improvement Figures Obtained for Different Settings (Setup Time).

# of iter.	Phases	Setup time α = 0.05	Setup time α = 0.10	Setup time α = 0.30
n^2	(CS-P1)/CS	-36.98%	-36.98%	-36.98%
	(CS-P2)/CS	31.03%	31.03%	31.03%
	(CS-P3)/CS	32.06%	34.72%	39.25%
	(CS-P4)/CS	32.41%	36.31%	41.90%
	# of exc. / # of iter.	1.02%	2.32%	3.38%
n^3	(CS-P1)/CS	-36.98%	-36.98%	-36.98%
	(CS-P2)/CS	31.03%	31.03%	31.03%
	(CS-P3)/CS	32.80%	39.12%	44.37%
	(CS-P4)/CS	33.07%	40.28%	45.31%
	# of exc. / # of iter.	0.91%	1.77%	2.18%

Table 7. Detailed CPU times for June and September for n^2 and n^3 Iteration for Different α Values.

		n^2					n^3				
	α	Phase 1	Phase 2	Phase 3	Phase 4	**Total**	Phase 1	Phase 2	Phase 3	Phase 4	Total
June	**5**	0	1.63	0.01	1.80	3.44	0	1.62	0.65	2.24	4.51
	10	0	1.63	0.02	1.97	3.62	0	1.63	0.65	4.56	6.84
	30	0	1.65	0.01	2.03	3.69	0	1.66	0.65	2.25	4.56
September	**5**	0	1.56	0	1.64	3.20	0	1.54	0.15	1.56	3.25
	10	0	1.54	0	1.57	3.11	0	1.55	0.15	1.57	3.27
	30	0	1.53	0	1.61	3.14	0	1.53	0.15	1.53	3.21

Table 8. Improvements When Phase 2 is Skipped for Phases 1+3+4 (n^3 cube iterations, $\alpha = 0.30$).

	Total Time	Setup Time
(CS-P1)/CS	-1.84%	-36.98%
(CS-P3)/CS	2.91%	37.92%
(CS-P4)/CS	3.45%	46.92%

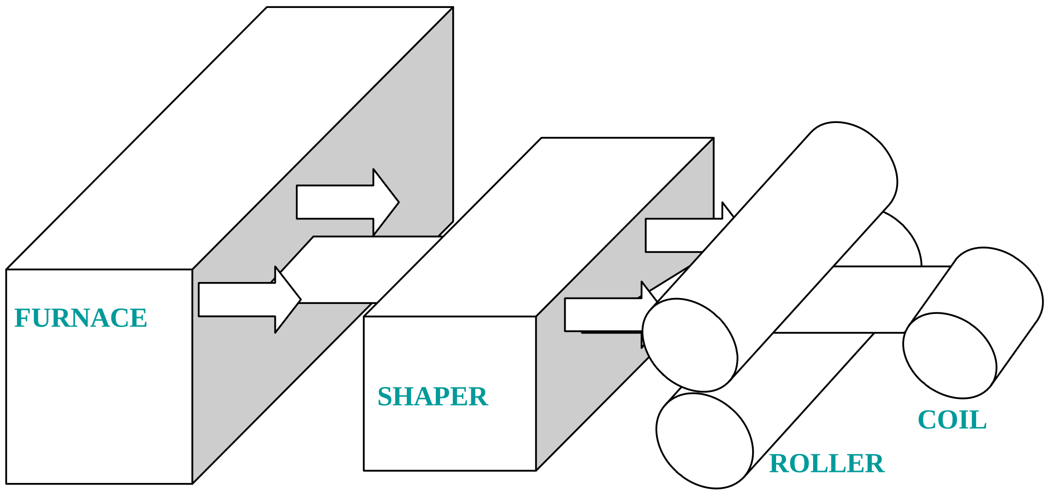

Figure 1. An aluminum casting line.

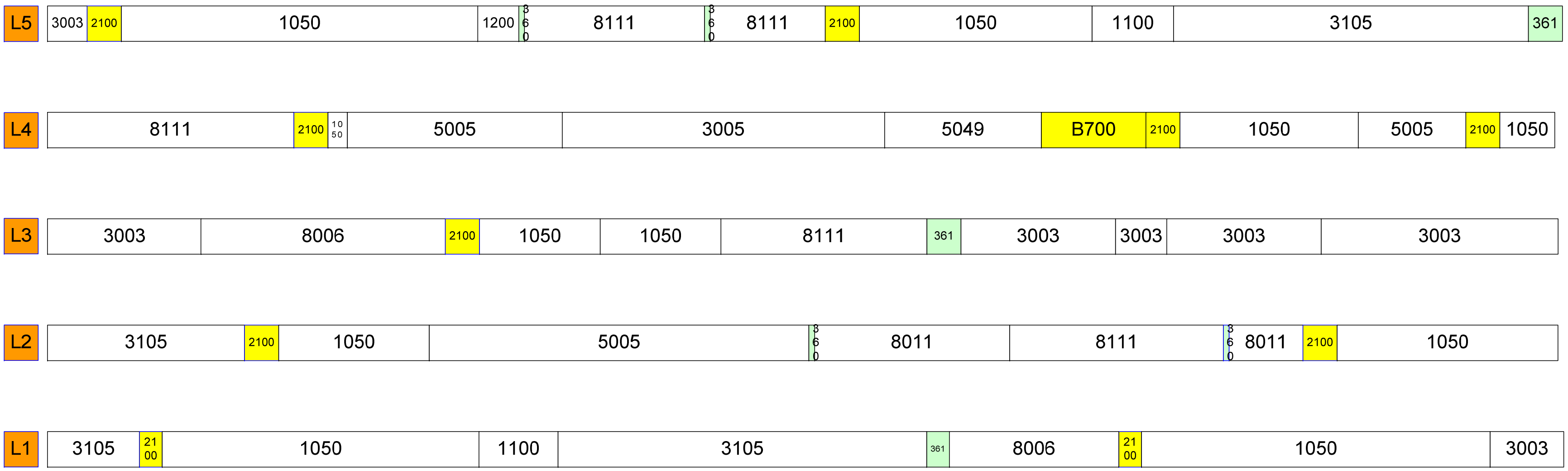

Figure 2. Gantt-chart for the company schedule for monthly data from table 2.