



**HAL**  
open science

## A finite capacity production scheduling procedure for a Belgian steel company

Mario Vanhoucke, Dieter Debels

► **To cite this version:**

Mario Vanhoucke, Dieter Debels. A finite capacity production scheduling procedure for a Belgian steel company. *International Journal of Production Research*, 2008, 47 (03), pp.561-584. 10.1080/00207540701441970 . hal-00512989

**HAL Id: hal-00512989**

**<https://hal.science/hal-00512989>**

Submitted on 1 Sep 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



**A finite capacity production scheduling procedure for a Belgian steel company**

Journal:	<i>International Journal of Production Research</i>
Manuscript ID:	TPRS-2006-IJPR-0944.R1
Manuscript Type:	Original Manuscript
Date Submitted by the Author:	10-Apr-2007
Complete List of Authors:	Vanhoucke, Mario; Ghent University, Dept of Mgmt Info, Ops Mgmt and Tech Policy Debels, Dieter; Ghent University
Keywords:	PRODUCTION PLANNING, SCHEDULING, PRODUCTION CONTROL, SEQUENCING
Keywords (user):	



1  
2  
3  
4 **A finite capacity production scheduling**  
5  
6 **procedure for a Belgian steel company**  
7  
8  
9

10  
11  
12 **Mario Vanhoucke**

13  
14 Corresponding author

15  
16  
17  
18 *Faculty of Economics and Business Administration, Ghent University, Hoveniersberg 24, Ghent, Belgium*

19  
20 *Operations & Technology Management Centre, Vlerick Leuven Gent Management School, Reep 1, Ghent, Belgium*

21  
22 [mario.vanhoucke@ugent.be](mailto:mario.vanhoucke@ugent.be)

23  
24  
25  
26  
27 **Dieter Debels**

28  
29  
30 *Faculty of Economics and Business Administration, Ghent University, Ghent, Belgium*

31  
32 [dieter.debels@ugent.be](mailto:dieter.debels@ugent.be)

# A finite capacity production scheduling procedure for a Belgian steel company

**Abstract.** We present a multi-objective finite capacity production scheduling algorithm for an integrated steel company located in Belgium. The two-stage optimization model takes various company-specific constraints into account and optimizes various often conflicting weighted objectives. A first machine assignment stage determines the routing of an individual order through the network while a second scheduling stage makes a detailed timetable for each operation for all orders. The procedure has been tested on randomly generated data instances sampled from real-life data of the steel company. We report promising computational results and illustrate the flexibility of the optimization model with respect to the various weights in the multi-objective function.

**Keywords.** Master production scheduling; manufacturing planning and control; scheduling/sequencing.

## 1 Introduction

Due to the often complex nature of production planning, various Hierarchical Production Planning (HPP) approaches have been presented in literature to cope with interactions between product demand, production capacity, real-time data, etc... Each hierarchical level imposes various constraints and objectives to the lower level while the lower level returns feedback about the production progress. There exist different relations of aggregation and disaggregation between the hierarchical levels, based on product unit (product parts into products into product archetypes), production unit (machine into machine group into production step into factory unit), scope (short term, medium term, long term), time unit (minutes, days, weeks, months) and decision level (shop-floor workers, production management, top management). The literature on HPP is infinitely rich and a literature overview is outside the scope of this manuscript. We refer the reader to pioneering work of Hax and Meal (1975) who originally proposed the HPP approach, to overview papers written by McKay et al. (1995) and Okuda (2001) or to the articles of Kistner and Steven (1991) and Bitran and Tirupati (1993) published in Operations Management handbooks that discuss the wide variety of hierarchical levels in production planning.

In this paper, we present a multi-objective finite capacity production scheduling algorithm at a middle-term planning horizon level (between a long-term aggregate planning and a short-term machine scheduling level) for the integrated steel company Arcelor Ghent (formerly known as Sidmar), located on the Ghent-Terneuzen Canal, around 20 km from the centre of the city of Ghent (Belgium). Arcelor Ghent yearly produces 5 million tonnes of flat steel strip for the automotive industry and for all kinds of high-quality applications such as domestic appliances, sanitary, heating, construction and furniture and handles every step of the production process, from the supply of raw materials to the coating of steel and the production of laser-welded blanks. With 5,500 employees, they are one of the largest employers in the region. They are

1  
2  
3 part of Arcelor, one of the world's largest steel companies. For more information, we refer to  
4 www.sidmar.be.  
5  
6

7  
8 Planning and scheduling problems in iron and steel production have not drawn as wide an attention of the  
9 operations management researchers as many other industries. However, the iron and steel industry is both  
10 capital and energy intensive, which makes the importance of effective planning and scheduling in this  
11 industry by no means less than that in other industries (Tang et al., 2001). Moreover, Lee et al. (1996) argue  
12 that tools are available to develop efficient algorithms for the extremely difficult scheduling environment of  
13 steel and research in this area should be stimulated since the return on investment for software to support  
14 improved steel making productivity is very high.  
15  
16  
17  
18  
19

20  
21 The literature on steel production planning and scheduling is focused on a wide variety of planning and  
22 scheduling systems and we do not attempt to cover it all here in detail. Most studies focus on sub-parts of the  
23 production process (Harjunoski and Grossmann (2001), Cowling et al. (2004)) and ignore interactions  
24 between these sub-parts, although the objective of steel companies to reduce the lateness and lead-time of the  
25 orders increases the need for an integral approach of the problem. We cite Tang et al. (2002) who give a  
26 comparative analysis of the different steel production processes and review their planning and scheduling  
27 systems. Most research is dominated by static scheduling methods, as given by, amongst others, the multiple  
28 arc network model of Sasidhar and Achary (1991), the greedy constructive method of Petersen et al. (1992),  
29 the simulated annealing approach of Portmann and Rohr (1995), the implicit enumeration based algorithm of  
30 Assaf et al. (1997), the linear programming model of Chen and Wang (1997), the goal programming  
31 approach of Jacobs et al. (1998), the tabu search approach of Lopez et al. (1998), the mathematical  
32 programming model of Tang et al. (2000a), the genetic algorithm of Tang et al. (2000b), the combined  
33 mathematical programming/heuristic search approach of Cowling and Rezig (2000), the decomposition  
34 mathematical programming strategy of Harjunoski and Grossmann (2001), the Lagrangian relaxation  
35 approach of Tang et al. (2002), the genetic algorithm of Chen and Wu (2002), the integrated finishing line  
36 scheduling approach of Okano et al. (2004), the linear and non-linear programming method combination of  
37 Neureuther et al. (2004) and the combinatorial auction-based approach of Kumar et al. (2007). Recently, the  
38 focus has shifted towards robust dynamic scheduling which has been investigated by the tabu search based  
39 flexible decision support system of Crowling (2003), the multi-agents approaches of Ouelhadj et al. (2004)  
40 and Crowling et al. (2003, 2004) and the expert process knowledge analysis technique of Roy et al. (2004).  
41 We refer to Dutta and Fourer (2001) who present a survey of various mathematical programming  
42 applications for integrated steel plants.  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59

60 The outline of the paper is as follows. Section 2 briefly summarizes the scope and limitations of the current  
research project. Moreover, a problem formulation is given in which all company-specific constraints and

1  
2  
3 problem objectives are defined and explained into detail. Section 3 presents our **two-stage heuristic** solution  
4 procedure to solve the problem under study. In section 4 we present extensive computational results and  
5 conclusions are given in section 5.  
6  
7

## 8 9 **2 Problem formulation**

### 10 11 12 **2.1 Scope of the research project**

13  
14  
15  
16 This research is the result of a co-operation between researchers at the “Operations Research and  
17 Scheduling” group<sup>1</sup> from the Ghent University and the people from the “Systems and Models” department of  
18 Arcelor Ghent. The specific and challenging problem involves the development of a scheduling heuristic as a  
19 part of a three-tier hierarchical production planning process containing an aggregate planning, a disaggregate  
20 planning and a sequencing phase (see figure 1).  
21  
22  
23

24  
25  
26 **< insert figure 1 here >**  
27

28  
29 In the remainder of this sub-section, we briefly explain the well-defined scope and limitations of the research  
30 project since these clearly affect the specific approach and underlying assumptions of the developed solution  
31 approach in the current manuscript. After various meeting and interesting discussions with the people of the  
32 “Systems and Models” department, the specific scope of the project could be summarized as follows:  
33  
34

- 35 - The scope of the research project completely lies in and is strictly limited to the second tier, which  
36 aims at the construction of a production schedule taking various constraints and objectives into  
37 account. Upon request of the company, the solution approach must be restricted to a daily bucket  
38 system where orders are assigned to machines on a daily basis, taking the relevant information from  
39 the first tier as a given input, and providing aggregate planning information to the third tier, as  
40 follows:  
41  
42
  - 43 ○ Tier 1: Specific information resulting from the aggregate planning of the first tier is input for  
44 the second disaggregate planning tier and cannot be changed under any condition. This input  
45 information mainly involves the presence of production campaigns and production flows  
46 which are pre-described aggregate production volumes of order clusters defined in the first  
47 tier which will be discussed later.
  - 48 ○ Tier 3: The specific production sequencing is out of the research project scope as the  
49 company relies on in-house dedicated algorithms. Consequently, the second disaggregate  
50 planning tier only assigns orders to machines on a daily basis without determining the exact  
51 production sequence of the orders on a particular day. This exact sequence of orders *within a*  
52  
53  
54  
55  
56  
57  
58  
59  
60

---

<sup>1</sup> For more information, see [www.projectmanagement.ugent.be](http://www.projectmanagement.ugent.be)

1  
2  
3 daily bucket (i.e. the third sequencing tier) needs to be determined by shop floor decisions or  
4 process-specific algorithms and is outside the scope of this project.

5  
6 - Although the constraints and objectives were clearly defined at the start of the project (see sections  
7 2.3), their relative importance needs to be flexible to allow sensitivity analyses and comparisons of  
8 alternative plans for different input values for the multi-objective unit weights. Hence, exact values  
9 for the different unit weight parameters (section 2.2) were not determined, but instead, were left  
10 open as a tool to allow flexibility for the user.

11  
12 - The new procedure needs to be tested on different artificial test instances which reflect a wide  
13 variety of possible real-life scenarios. In section 4, we displays various experimental results and  
14 show the flexibility/sensitivity of the solution approach for the various input unit weight parameters.

15  
16  
17  
18  
19  
20  
21 The master production schedule is more easily developed for product archetypes, i.e. the first disaggregate  
22 planning tier than for individual product orders. Therefore, Arcelor Ghent clusters production orders on the  
23 basis of two criteria. These clusters are scheduled at the first aggregate planning tier and are used as input  
24 information for the scheduling heuristic of the second disaggregate planning tier. The two clusters can be  
25 described as follows:

26  
27  
28  
29  
30  
31 Production campaigns have been described in literature (see e.g. (Okano et al., 2004)) as various orders with  
32 similar product characteristics that are grouped in a production run with specific start and end times in which  
33 coils of a particular type are processed continuously on a process line. Hence, product characteristics of the  
34 orders determine whether they belong to a particular group, and the time window of each production  
35 campaign (defined by their start and finishing time) determines whether an order is allowed to be scheduled  
36 in that time window. A typical order characteristic example in Arcelor Ghent is the thickness of the coils. It  
37 is beneficial to start a campaign of thin coils at the cold rolling mill when new rollers are installed, since the  
38 thicker coils can be rolled later in time when rollers start to wear out. In doing so, the production campaigns  
39 implicitly incorporate sequence-dependent setup costs at the aggregate level by forcing to produce the thin  
40 coils before the thicker ones. The presence of production campaigns is simply incorporated in our scheduling  
41 heuristic as a binary variable which allows/forbids the scheduling of a particular order at a particular time  
42 period (see production order campaign constraints at section 2.3).

43  
44  
45  
46  
47  
48  
49  
50  
51  
52 Production flows are clusters of orders that are commonly scheduled in the first aggregate planning tier  
53 based on various (often hidden or unspoken) product characteristics. In contrast to the production campaigns,  
54 which simply allow or forbid groups of order to be scheduled in a particular time window in the second  
55 aggregate planning tier, production flows prescribe an 'ideal' volume of production (in tons) for a group of  
56 orders in a certain time window. A typical example at Arcelor Ghent is the presence of secondary resources.  
57 The production of steel requires primary resources (i.e. machines) as well as secondary resources, such as  
58 colourings or chemical additives. An efficient stock management of these secondary resources leads to  
59  
60

1  
2  
3 substantial cost reductions. Hence, the steel company clusters orders that use the same secondary resources at  
4 particular production steps in order groups or *production flows*. This allows the determination of a pre-  
5 specified production volume (in tons) for each production flow order group for a group of machines during a  
6 pre-defined time window (e.g. a week) at the aggregate planning level. The disaggregate scheduling heuristic  
7 takes these prescribed production flow volumes into account by penalizing deviations (either below or  
8 above) between the scheduled order volumes and the pre-specified flow volumes (see section 2.3). Thanks to  
9 these production flow volume quantities, the steel company can order the corresponding secondary resources  
10 on a just in time basis, avoiding excessive safety stocks. Unlike the presence of the production campaigns  
11 which can be easily incorporated as a binary decision in the scheduling heuristic, the presence of production  
12 flows increases the scheduling complexity dramatically, as will be described in section 3.1.

## 21 2.2 Problem parameters and decision variables

22  
23  
24 A steel shop environment is closely related to a flexible flow shop as it consists of several serial production  
25 steps, each consisting of several identical machines in parallel (Lee et al., 1996). The parallel machines may  
26 be clustered into machine groups (e.g. they are located at the same place and make use of the same storage  
27 facilities). A set of accepted production orders needs to be scheduled. The ordered coils can slightly differ  
28 from each other in terms of width, thickness, length, quality, etc. (Okano et al., 2004), resulting in a wide  
29 variety of product types. Figure 2(a) displays a **fictive** steel shop network with 4 production steps, 7 machine  
30 groups and 11 machines. Dummy machine 0 and dummy machine 12 (or in general  $nrj + 1$ ) are used to  
31 represent the start and finishing time of an order. Figure 2(b) displays the *order routing network* of an  
32 example order for which the routing is limited to production steps 1, 2 and 4. **Each order follows a route of**  
33 **operation steps which consists of a unique path in the order routing network from the dummy start machine 0**  
34 **to the dummy end machine 12.** The production steps of a coil of steel consist of casting, hot rolling, pickling,  
35 and cold rolling. In order to give specific properties to the coil, extra operations such as annealing, skin  
36 passing, galvanizing, coating, recoiling or cutting can be performed. Each production step has many typical  
37 production constraints and solving a steel shop scheduling problem involves handling a large number of  
38 complicated chemistry-, geometrical- and scheduling rules.

39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51 << insert figure 2 here >>

52  
53 Table 1 summarises the various parameters needed to formulate the production scheduling problem. These  
54 parameters will be used throughout the remainder of the manuscript.

55  
56  
57  
58  
59 << insert table 1 here >>



The disaggregate planning tier constructs a schedule which assigns all operations for all orders taking various constraints and multiple objectives into account (see section 2.3). Hence, every operation of order  $i$  needs to be executed on a machine  $j$  on a particular day  $k$ , resulting in the decision variables as follows:  $x_{ijk} = 1$ , if an operation of order  $i$  is assigned to machine  $j$  on day  $k$ , and  $x_{ijk} = 0$  otherwise. Note that we construct a production schedule where the daily machine capacity and processing times of the orders are expressed in minutes whereas the intermediate time between machines is expressed in days. For the production of steel in Arcelor Ghent, a typical order has less than one day processing time on a machine which lies far beyond the time needed to support intermediate manipulations such as transporting, heating or cooling down (typically several days). In the remainder of this paper, the multi-objective unit weights of table 1 will not have an explicit value since they do not always denote a real cost. Instead, they represent a relative weight to express the relative importance of the various objective function factors which can be changed to perform a sensitivity analysis. This is the subject of section 4.3. For more information about accurately estimating the cost elements of an order and the explanation how a finite-capacity environment might influence an order's cost, we refer the reader to Feldman and Shtub (2007).

### 2.3 The problem formulation

Previous research studies reveal that the multiple objectives that are used in the steel making industry are often very company-specific. Likewise, in our study, both the multiple objective function and the various constraints have been set up in dialogue with the people of the "Systems and Models" department of the company. More precisely, our disaggregate production scheduler optimizes a multiple objective function by minimizing the total cost as a sum of five different cost functions: assignment cost  $C^A$ , lateness cost  $C^L$ , earliness cost  $C^E$ , utilisation cost  $C^U$  and production flow cost  $C^F$ . Moreover, the problem is the subject to four different types of technical restrictions (capacity, assignment, precedence and setup constraints). A conceptual problem formulation can be given as follows:

**Minimize** (i) Assignment cost  $C^A$

$$\sum_{i=1}^{nri} \sum_{j=1}^{nrj} \sum_{j'=j+1}^{nrj} a_{ijj'} \sum_{k=1}^{nrk} x_{ijk} \sum_{k'=k+1}^{nrk} x_{ij'k'} \quad [1]$$

(ii) Lateness cost  $C^L$

$$+ \sum_{i=1}^{nri} l_i \max(0, \sum_{k=1}^{nrk} kx_{inrj+1k} - t_i) \quad [2]$$

(iii) Earliness cost  $C^E$

$$+ \sum_{i=1}^{nri} e_i \max(0, LST_{i0} - \sum_{k=1}^{nrk} kx_{i0k}) \quad [3]$$

(iv) Utilisation cost  $C^U$

$$+ \sum_{j=1}^{nrj} \sum_{k=1}^{nrk} u_j \left( c_{jk} + \Delta c_{jk-1} - \sum_{i=1}^{nri} p_{ij} x_{ijk} \right) \quad [4]$$

$$(v) \text{ Production flow deviation cost } C^F + \sum_{l=1}^{nrl} \sum_{n=1}^{nrm} \sum_{m=1}^{nrm} \begin{cases} \Delta f_{lmm} b_{ln}^- & \text{if } \Delta f_{lmm} \geq 0 \\ -\Delta f_{lmm} b_{ln}^+ & \text{if } \Delta f_{lmm} < 0 \end{cases} \quad [5]$$

**Subject to**

(vi) Capacity constraints

$$\sum_{i=1}^{nri} pt_{ij} x_{ijk} \leq c_{jk} + \Delta c_{jk-1} \quad j = 1, \dots, nrj \text{ and } k = 1, \dots, nrk \quad [6a]$$

$$\Delta c_{j0} = 0 \quad j = 1, \dots, nrj \quad [6b]$$

$$\Delta c_{jk} = \min \left( \tau, c_{jk} + \Delta c_{jk-1} - \sum_{i=1}^{nri} pt_{ij} x_{ijk} \right) \quad j = 1, \dots, nrj \text{ and } k = 1, \dots, nrk \quad [6c]$$

(vii) Assignment constraints

$$\sum_{j \in S_q^{JO}} \sum_{k=1}^{nrk} x_{ijk} = 1 \quad i = 1, \dots, nri \text{ and } \forall q \in O_i^Q \quad [7]$$

(viii) Precedence constraints

$$\sum_{k=1}^{nrk} (kx_{ij_1k}) + d_{ij_1j_2} \leq \sum_{k=1}^{nrk} kx_{ij_2k} \quad i = 1, \dots, nri, j_1 \in S_{q_1}^{JO} \text{ and } j_2 \in S_{q_2}^{JO} \text{ and } q_1 \text{ and } q_2 \text{ are two subsequent production steps} \quad [8]$$

(ix) Production campaign constraints

$$\sum_{i \in C_{jk}} x_{ijk} = 0 \quad j = 1, \dots, nrj \text{ and } k = 1, \dots, nrk \quad [9]$$

(x) Production flow deviations

$$\Delta f_{lmm} = f_{lmm} + \Delta f_{lmm-1} - \left( \sum_{i \in F_l^O} v_i \sum_{j \in S_n^{JN}} \sum_{k \in W_m} x_{ijk} \right) \quad l = 1, \dots, nrl, n = 1, \dots, nrn \text{ and } m = 1, \dots, nrm \quad [10]$$

(xi) Binary and nonnegativity constraints

$$x_{ijk} = \text{binary (0 or 1)} \quad i = 1, \dots, nri, j = 1, \dots, nrj \text{ and } k = 1, \dots, nrk \quad [11]$$

The four different types of **technical restrictions** the Arcelor Ghent steel company incorporated in its production schedule can be described as follows:

**Capacity constraints:** Each machine  $j$  has a limited capacity  $c_{jk}$  expressed in minutes per day, which may not be exceeded by all assigned orders on that machine at day  $k$ . In order to avoid that this constraint leads to a structural under-use of the available machine capacity, Eq. [6a] adds the unused capacity  $\Delta c_{jk-1}$  of the previous day to the capacity  $c_{jk}$ . This capacity shift is initialized in Eq. [6b] and is limited to a threshold value  $\tau$  in Eq. [6c] such that a temporal shortage of eligible orders at day  $k - 1$  does not create unrealistic capacity at day  $k$ . In our production scheduling algorithm, we set  $\tau$  equal to 10 minutes.

Assignment constraints: In order to process all operations of an order, the algorithm must select a unique path from the order routing network consisting of a sequence of machines from the dummy start to the dummy end machine. Therefore, Eq. [7] implies that each operation of order  $i$  needs to be assigned to one machine  $j$  per production step  $q \in O_i^Q$ .

Precedence constraints: The precedence relations between operations of an order  $i$  are shown by the order routing network and define the relations between all machines  $j_1 \in S_{q_1}^{JQ}$  and  $j_2 \in S_{q_2}^{JQ}$  of two sub-sequent production steps  $q_1 \in O_i^Q$  and  $q_2 \in O_i^Q$  with a minimal time interval equal to the duration  $d_{i,j_1,j_2}$ . Hence, for each couple  $(q_1, q_2) \in O_i^Q$  of two subsequent production steps in the routing of an order  $i$ , the algorithm incorporates the precedence relations as given in Eq. [8].

Production campaign constraints: The production campaign constraints of Eq. [9] forbid orders to be scheduled at a certain machine group and on a certain day if this order does not belong to a production campaign that allows its orders to be scheduled at that current day.

The problem **objective function** consists of five cost factors that have to be minimized, as follows:

Assignment cost: The total assignment cost of Eq [1] is an immediate result of the selection of paths in the order routing networks and is weighted at a unit assignment weight  $a_{ij}$  for each arc  $(j, j')$  of the selected path of an order  $i$ .

Lateness cost: The lateness cost of Eq [2] penalises the production orders that finish later than the pre-negotiated due date  $t_i$ , and equals the unit lateness weight  $l_i$  times the number of days order  $i$  is late (or zero, if the order finishes earlier than  $t_i$ ). The algorithm determines the finishing day  $k$  of order  $i$  as the assignment of the order on the dummy end machine  $nrj + 1$ .

Earliness cost: The earliness cost is incorporated for various reasons, among which the lead-time and stock level minimization are the most important ones. Eq [3] states that orders should be started no earlier than necessary to finish within the pre-negotiated due date  $t_i$ . Therefore, the algorithm calculated the latest possible starting time  $LST_{i0}$  by means of simple backward calculations, starting from the end dummy machine  $nrj + 1$ . The earliness cost is equal to the unit earliness cost  $e_i$  times the number of days the order  $i$  starts earlier than its  $LST_{i0}$  (or zero, if the order starts after its  $LST_{i0}$ ).

Utilisation cost: The utilisation cost of Eq [4] penalises each time unit (minute) a machine is idle. Hence, the algorithm measures the deviation between the daily machine capacity  $c_{jk} + \Delta c_{jk-1}$  and the capacity use

$$\sum_{i=1}^{nri} p_{ij} x_{ijk} \text{ of the assigned orders.}$$

Production flow cost: The production flow cost of Eq. [5] takes the production flow constraints into account by penalizing deviations (weighted by unit weights per ton, either below  $(b_m^-)$  or above  $(b_m^+)$ ) between the scheduled order volumes and the pre-specified flow volumes. Note that the aggregate planning tier of

Arcelor Ghent uses weekly buckets to describe the pre-specified production flow volumes while the disaggregate planning tier schedules on a daily basis. Eq. [10] calculates the flow deviation  $\Delta F_{lmm}$  for production flow group  $l$  at machine group  $n$  during week  $m$  where  $W_m$  is used to denote the set of days belonging to week  $m$ .

### 3 The disaggregate planning solution approach

In this section, we describe our two-stage heuristic solution algorithm to solve the multi-objective production scheduling problem under study.

Step 1. Machine assignment problem: each order is assigned to one machine for each production step of its routing (assignment constraint), resulting in the total assignment cost  $C^A$ .

Step 2. Scheduling problem: all operations of each order need to be scheduled on a particular day (given the assigned machines of step 1), taking the three remaining constraints (capacity constraints, precedence constraints, setup constraints) and the multiple objectives (lateness costs, earliness costs, utilisation costs and production flow costs) into account.

<< insert figure 3 here >>

Figure 3 displays a fictive example with two orders. In step 1, each order is assigned to a specific machine on its routing (path 3 – 6 – 10 for order 1 and path 2 – 8 – 10 for order 2). These assigned machine paths will be used as an input to solve the scheduling problem of step 2. In figure 3, both orders start on day 2 and have been scheduled as soon as possible. In section 3.1, we discuss the scheduling problem (step 2) in detail. Section 3.2 elaborates on the machine assignment problem (step 1).

#### 3.1 Scheduling problem

In order to schedule all orders in time, the algorithm solves a knapsack problem for each machine and for each day of the scheduling horizon. Hence, our *schedule generation scheme* (SGS) iterates over all machines and all days, and can be shown in pseudo-code, as follows:

```

For  $k:=1$  to  $nrk$ 
    For  $j:=1$  to  $nrj$ 
        Knapsack Problem ( $j,k$ )

```

The knapsack problem determines for each machine and each day the set of *eligible orders*  $O_{jk}^E$  that are potential candidates to enter the knapsack (i.e. scheduled on machine  $j$  during day  $k$ ). An order  $i$  is eligible on machine  $j$  on day  $k$  if the following constraints are satisfied:

- Assignment constraints: an order can only be scheduled on a machine determined by the machine assignment problem of section 3.2,
- Precedence constraints: an order can only be scheduled if the previous operation of that order has been scheduled earlier, taking the intermediate duration into account,
- Setup constraints: an order can only be scheduled within the production campaign restrictions.

Hence, the knapsack problem boils down to the selection of orders to be scheduled on machine  $j$  of day  $k$ , satisfying the capacity constraints and optimizing the various costs factors of section 2.3.

$$\max TCR = \sum_{i \in O_{jk}^E} CR_{ijk}^L + CR_{ijk}^E + CR_{ijk}^U + CR_{ijk}^F \quad [12]$$

subject to

$$\sum_{i \in O_{jk}^E} P_{ij} x_{ijk} \leq c_{jk} + \Delta c_{jk-1} \quad [13]$$

The objective function maximizes the total cost reduction  $TCR$  when assigning order  $i$  to machine  $j$  on day  $k$ . Since the cost of scheduling that order depends on the schedule of all other orders, we need to estimate the total cost reduction  $TCR$  when assigning order  $i$  to machine  $j$  on day  $k$ , denoted by  $CR_{ijk}^L$  (estimated lateness cost reduction),  $CR_{ijk}^E$  (estimated earliness cost reduction),  $CR_{ijk}^U$  (estimated utilisation cost reduction) and  $CR_{ijk}^F$  (estimated production flow cost reduction). The determination of these estimates will be discussed in the remainder of this section. The constraint of equation [13] is equal to the capacity constraint of machine  $j$  on day  $k$  of equation [1]. The knapsack problem is proven to be an NP-complete problem (Hirschberg and Wong, 1976).

**Estimated lateness cost reduction:** The lateness cost depends on the due date of order  $i$  and can only be determined when the last operation of order  $i$  has been scheduled. Hence, an estimate for the lateness cost reduction  $CR_{ijk}^L$  of assigning order  $i$  to machine  $j$  on day  $k$  is equal to the extra lateness cost we would obtain when postponing it to day  $k + 1$ . Hence, we need to rely on an estimate of the probability  $P_{ijk}$  as the chance that an order  $i$  will increase the lateness by one day if the order is not scheduled on day  $k$ . Consequently, the estimated lateness cost reduction is equal to  $CR_{ijk}^L = P_{ijk} L_i$  and is displayed in figure 4.

<< insert figure 4 here >>

The probability  $P_{ijk}$  is assumed to be 100% when the scheduling day  $k$  is larger then or equal to the latest start time  $LST_{ij}$ . This  $LST_{ij}$  can be easily determined by means of straightforward backward calculations starting from the order due date  $t_i$ . Moreover, we assume that this probability is a function of the remaining operations (denoted as the remaining production steps  $rps$ ) of the order and a slack-per-operation parameter  $\alpha$ . Therefore, we define a linear function such that the probability increases linearly from 0% to 100% between  $LST_{ij} - \alpha * rps \leq k \leq LST_{ij}$ . Outside the interval  $[LST_{ij} - \alpha * rps, LST_{ij}]$ , the probability equals 0% ( $k \leq LST_{ij} - \alpha * rps$ ) or 100% ( $k \geq LST_{ij}$ ). In our production scheduling algorithm, we set  $\alpha$  equal to 1.

**Estimated earliness cost reduction:** The earliness cost depends on the start of the first operation of order  $i$  and is measured as the deviation between the start of the first operation and the latest start time  $LST_{i0}$  of this operation. Hence, the estimated earliness cost reduction can be calculated as follows:

$$CR_{ijk}^E = \begin{cases} 0 & \text{if } k \geq LST_{i0} \text{ or if } j > 0 \\ -e_i, & \text{otherwise} \end{cases} \quad [14]$$

**Estimated utilisation cost reduction:** Each order  $i$  that enters the knapsack needs to be produced on machine  $j$  on day  $k$ , and hence, increases the utilisation by  $pt_{ij}$  minutes (and reduces the utilisation cost by  $u_j$  per minute). Hence, the estimated utilisation cost reduction can be calculated as follows:

$$CR_{ijk}^U = u_j pt_{ij} \quad [15]$$

**Estimated production flow cost reduction:** Estimating the reduction for the production flow cost for an order  $i$  for a particular machine  $j$  on a particular day  $k$  is the most complex calculation among all cost estimates. The reason is that each order  $i$  of production flow  $l$  ( $i \in F_l^O$ ) that enters the knapsack to be scheduled at machine  $j$  on a particular day  $k$  will affect the production flow deviation  $\Delta f_{lm}$  of machine group  $n$  (with  $j \in S_n^{JN}$ ) during the week  $m$  ( $k \in W_m$ ). Hence, the flow deviation calculations when scheduling an order  $i$  on a particular day are clearly affected by

- All orders that have been scheduled on previous days of the week. These orders are known at the start of scheduling day  $k$ .
- All orders that will probably be scheduled prior to order  $i$  at the same day  $k$ . These orders are not known at the start of scheduling day  $k$  and need to be estimated by a density measure.

Hence, the algorithm performs a three-step heuristic calculation procedure as follows:

**Step 1. Density measure calculation:** The algorithm calculates a *density measure* which heuristically estimates the relative scheduling priority of all eligible orders on day  $k$ . This density measure

$\delta_i = \frac{CR_{ijk}^L + CR_{ijk}^E + CR_{ijk}^U + b_{ln}^- v_i}{p_{ij}}$  measures the total estimated cost reduction per time unit (in minutes) if an

order  $i$  enters the knapsack at machine  $j$  on day  $k$  assuming that the entrance of an eligible order  $i$  reduces the production flow cost by  $b_{ln}^- v_i$  (there is still a flow ‘under-production’ on day  $k$ ). Since the knapsack problem

strives for a maximal knapsack density, it will give priority to orders with a high density measure value, and hence, this density measure serves as an estimate for the relative scheduling priority of each order  $i$  on day  $k$ .

**Step 2. Flow deviation estimate:** Some orders will enter the knapsack resulting in an increase of flow deviation (in case of a flow ‘over-production’). Therefore, the algorithm calculates an estimate for the flow deviation equations which predicts whether the entrance of order  $i$  on day  $k$  will increase or decrease the predefined production flow deviations of Eq. [10]. An estimate of these flow deviation equations for each order  $i$  (denoted by  $\Delta EF_i$ ) are calculated under the assumption that

- The orders  $i'$  of the same production flow  $l$  (i.e.  $i' \in F_l^O$ ) on machine  $j'$  that have been scheduled by the algorithm on previous days  $k' < k$  of the same week are already included.
- All eligible orders  $i''$  of the same production flow  $l$  (i.e.  $i$  and  $i'' \in F_l^O$ ) with a higher or equal density measure value of order  $i$  ( $\delta_{i''} \geq \delta_i$ ) will be scheduled (i.e. entering the knapsack) prior to scheduling order  $i$ , as

$$\Delta EF_i = \underbrace{f_{lm} + \Delta f_{lm-1}}_{(a)} - \underbrace{\sum_{i' \in F_l^O} v_{i'} \sum_{\substack{j' \in S_n^{JN} \\ k' < k}} x_{i' j' k'}}_{(b)} - \underbrace{\sum_{\substack{i'' \in F_l^O \\ \delta_{i''} \geq \delta_i}} v_{i''}}_{(c)} \quad [16]$$

with (a) the pre-specified production flow volumes for the current week, (b) the production volume of all scheduled orders  $i'$  and (c) the production volume of all orders  $i''$  that will probably be scheduled prior to scheduling order  $i$  plus the production volume of order  $i$ .

The value of  $\Delta EF_i$  reveals whether or not scheduling order  $i$  on machine  $j$  on day  $k$  will lead to overproduction of production flow  $l$ . If  $\Delta EF_i \geq 0$ , then scheduling order  $i$  will probably reduce the underproduction of flow  $l$ . If  $\Delta EF_i \leq -v_i$ , then scheduling order  $i$  will probably increase the overproduction of  $l$ . If  $-v_i < \Delta EF_i < 0$ , then scheduling order  $i$  will probably change the underproduction of  $l$  to an overproduction.



Step 3. The algorithm calculates the estimate of production flow reduction  $CR_{ijk}^F$  as follows:

$$CR_{ijk}^F = \begin{cases} b_{in}^- v_i & \text{if } \Delta EF_i \geq 0 \\ -b_{in}^+ v_i & \text{if } \Delta EF_i \leq -v_i \\ b_{in}^- (v_i + \Delta EF_i) + b_{in}^+ \Delta EF_i & \text{if } -v_i \leq \Delta EF_i < 0 \end{cases} \quad [17]$$

Figure 5 illustrates the production flow reduction estimate based on an example project of our production scheduler with 9 eligible orders on the third day of a particular week with 3 pre-specified production flows for which  $F_1^O = \{1, 6, 8\}$ ,  $F_2^O = \{2, 3, 7\}$  and  $F_3^O = \{4, 5, 9\}$ . The order volumes and density matrix of the orders are assumed to equal to (8, 7, 11, 12, 9, 9, 6, 13, 8) and (1.10, 0.93, 0.90, 0.85, 0.80, 0.76, 0.67, 0.63, 0.57), respectively.

<< insert figure 5 here >>

The black areas represent the production volume of all scheduled orders  $i'$  on previous days of the week for each flow. The density measure serves as a priority estimate for each order to be selected in the knapsack (see figure 5), as follows:

- Flow 1: 1 – 6 – 8: the entrance of all these orders leads to a decrease of the production flow deviation, and hence, the estimate of the flow deviation  $\Delta EF_i$  will be zero or positive.
- Flow 2: 2 – 3 – 7: the entrance of order 2 leads to a decrease of the production flow deviation ( $\Delta EF_i \geq 0$ ). The entrance of order 3 will change the underproduction of flow 2 to an overproduction ( $-v_i < \Delta EF_i < 0$ ) while the entrance of order 7 will increase the production flow deviation  $\Delta EF_i \leq -v_i$ .
- Flow 3: 4 – 5 – 9: the entrance of all these orders leads to an increase of the production flow deviation, and hence, all estimates  $\Delta EF_i \leq -v_i$ .

The above estimates of the flow deviation equations  $\Delta EF_i$  will be used to calculate the  $CR_{ijk}^F$  values (32, 14, -42, 120, -90, 36, -36, 52, -80) which will be used in the objective function of the knapsack problem.

### 3.2 Machine assignment problem

The basic machine assignment problem randomly selects for each order a single path from the order routing network consisting of a sequence of machines. However, the algorithm is able to control the machine assignment process by estimating cost factors in three various ways, as follows:



**Greedy assignment (based on assignment cost) (GA(A)):** Each order will be assigned to the path with the lowest assignment cost using a shortest path algorithm of Dijkstra (1959).

**Greedy assignment (based on assignment, utilisation and production flow cost) (GA(AUF)):** The orders will be assigned, one after another, to the path with the lowest total cost using the shortest path algorithm of Dijkstra (1959). The total cost is equal to the assignment cost (similar to the GA(A) approach) plus a penalty estimate for the utilisation and production flow cost. The penalty costs of each arc  $(j, j')$  are calculated based on the assignments of all previously assigned orders. To obtain utilisation and production flow estimations, we assume that these orders have been scheduled at their latest start time  $LST_{ij}$ .

- The utilisation penalty cost for arc  $(j, j')$  of order  $i$  is equal to  $u_j$  when the cumulative available capacity of machine  $j$  before  $LST_{ij}$  has been reserved already completely by the previously assigned orders, and equals zero otherwise.
- The production flow penalty cost for arc  $(j, j')$  of order  $i$  equals  $b_m^+$  when the cumulative pre-specified production flow quantities remaining has been reserved already completely by the previously assigned orders, and equals zero, otherwise.

Consequently, positive penalty factors for an arc  $(j, j')$  give the shortest path algorithm an incentive to select another arc of another path in the order routing network for order  $i$ .

**Local search assignment (LSA):** The local search procedure embeds the SGS of section 3.1 in a local search procedure in order to find high quality schedules. The procedure starts with an initial schedule, constructed by the GA(A) approach and the SGS procedure, and searches for improvements by iteratively changing the machine assignments of a single order  $i$ . The pseudo-code of the local search procedure can be displayed as follows:

**Procedure LSA()**

Construct initial schedule

**For**  $k = 1$  **to**  $nrk$

**For**  $i = 1$  **to**  $nri$

**For**  $q = 1$  **to**  $nrq$

**For**  $j_1 =$  first machine **to** last machine of  $S_q^{JQ}$

**If**  $x_{ij_1k} = 1$

**For**  $j_2 =$  first machine **to** last machine of  $S_q^{JQ}$

                        change assignment of order  $i$  from  $j_1$  to  $j_2$

**If** “check constraint feasibility” **then**  $\Delta\text{costs} =$  “phase 1 cost estimate”

**If**  $\Delta\text{costs} < 0$  **then**  $\text{costs} = \text{SGS}()$

**If** costs < best found costs **then** replace best found schedule

**Else** change assignment of  $i$  again from  $j_2$  to  $j_1$ .

The local search procedure iteratively searches for each day  $k$  whether a re-assignment of an order would lead to improvements. Therefore, the algorithm considers all possible re-assignments of an order from machine  $j_1$  to machine  $j_2$  within a production step  $q$ , and checks the resulting constraint feasibility and the resulting cost changes, as follows:

- *Check constraint feasibility*: this sub-routine checks whether all constraints are satisfied when re-assigning order  $i$  from machine  $j_1$  to machine  $j_2$  of the same production step  $q$ .
- *New cost estimation*: The machine assignment change will lead to a new cost, that is estimated in two phases.
  - o *Phase 1. quick and rough estimate*: this cost estimate is a quick and rough estimate to evaluate whether the order re-assignment is a valuable alternative that needs further detailed analysis. The “phase 1 cost estimate” is equal to the change in the assignment cost plus the new utilisation and flow cost. For the latter two, the algorithm simply calculates the cost changes by ignoring the cost effect on the rest of the schedule. If the rough estimate shows a cost decrease, the algorithm calculates the more detailed cost estimate in phase 2 to decide whether the re-assignment will be executed.
  - o *Phase 2. estimate cost by SGS* (only if phase 1 gives an indication that a re-assignment would be beneficial): The schedule generation scheme schedules the order on the new machine  $j_2$  and calculates the resulting cost as described in section 3.1. When the resulting schedule cost is lower than the current schedule cost, the new schedule replaces the previous one and the algorithm continues its search.

Ideally, this process continues until all days have been considered. In order to limit the computational effort, the algorithm will be truncated after 100 generated schedules.

In the next section, these three machine assignment heuristics will be tested on an artificial dataset based on real-life data from Arcelor Ghent.

#### 4 Experimental results

In this section, we report detailed computational results of different versions of our solution procedure. All procedures have been programmed in Visual C++ 6.0 and tested on an Acer Travelmate 634LC with a Pentium IV 1.8 GHz processor. We rely on a self-generated test set of 50 problem instances explained in section 4.1. Section 4.2 reports detailed computational results for the various machine assignment procedures and the schedule generation scheme. In section 4.3, we illustrate the flexibility of the schedule generation

scheme and the use of the various penalty costs that can be modified to create a schedule that fulfils company-specific objectives.

#### 4.1 Generation of problem instances

In order to generate problem instances, we have developed an automatic problem generator taking the various problem parameters as pre-specified input values. The input values for the various parameters have been carefully chosen based on a detailed study of real-life data at Arcelor Ghent. We vary the number of orders ( $nri$ ) from 1,000; 2,000; 4,000; 8,000 to 16,000 and generate 10 problem instances per setting, resulting in 50 problem instances in total. The number of orders as well as all other parameters have been set based on the investigation of real-life data available at the company. All other parameters are fixed as follows:

Steel-shop characteristics:  $nrq = 8$ ,  $nrn = 12$ ,  $nrj = 20$ ,  $S_q^{NQ}$  has been created by random assignments of machine groups for each production step  $q$  and  $S_n^{JN}$  has been constructed by random assignments of machines for each machine group  $n$ .

Scheduling horizon:  $nrk = 21$  days and  $nrm = 3$  weeks

Order characteristics:  $v_i = \text{rand}[20, 40]$  (in tons),  $t_i = \text{rand}[3, 2*nrk]$  and  $O_i^Q$  has been created by random assignments of production steps to each order  $i$  (minimum 3 production steps per order). We distinguished  $nrl = 10$  order groups or production flows and also for the assignment of orders to order groups we relied on randomness. In order to generate realistic production flow quantities, the values for  $f_{im}$  (in tons) have been generated based on a simulated schedule for each problem instance.

Order routing network:

Durations:  $pt_{ij} = \text{rand}[0.9 v_i, 1.1 v_i]$  (in minutes),  $d_{ij} = \text{rand}[0, 7]$  (in days). Similar to the flow quantities, the daily machine capacities  $c_{jk}$  (in minutes) have been generated based on a simulated schedule for each problem instance.

Costs:  $a_{ijj} = \text{rand}[0, 99]$  (€),  $e_i = 4 v_i$  (€ per day),  $l_i = \text{rand}[0, 3] v_i$  (€ per day),  $u_j = 8$  (€) and  $b_m^- = b_m^+ = 8$  (€).

#### 4.2 Computational performance of our solution approach

Table 2 displays the performance of the schedule generation scheme (SGS) by solving the knapsack problem with an exact and a heuristic approach. More precisely, the exact branch-and-bound procedure of Kolesar (1967) is compared with a straightforward greedy search heuristic in which eligible orders are chosen in decreasing order of their density measure. The row with label "Avg. CPU" displays the average CPU time in seconds and the row with label "Avg. TC" displays the average total cost. The total costs consist of the individual cost factors of section 2.4 (Avg.  $C^A$ , Avg.  $C^L$ , Avg.  $C^E$  and Avg.  $C^F$ ), as displayed in the remaining

rows. Note that the machine assignment problem has been solved by randomly assigning each order to one machine for each production step of its routing. This machine assignment problem has been repeated ten times.

<< insert table 2 here >>

The table reveals that both the exact and the heuristic solution procedures for the knapsack problems are able to provide solutions within a reasonable time limit. The heuristic approach is able to generate high quality knapsack solutions (see the small Avg. *TC* deviations between the exact and heuristic approach) but the extra CPU time the exact approach needs is relatively small. Hence, in the remainder of this paper, we rely on the exact approach of Kolesar (1967) to solve the knapsack problems.

Table 3 compares the performance of the various machine assignment procedures of section 3.2. The table clearly reveals that more sophisticated assignments such as the greedy assignment GA(AUF) and the local search assignment LSA result in larger CPU times. But the resulting schedule quality of those assignment strategies outperforms the simple GA(AC) and the random assignment. The results can be summarised as follows:

**LSA versus GA(AC):** The LSA approach clearly outperforms the GA(AC) approach at the expense of a much larger CPU time. However, in relative terms, the improvement of LSA compared to the GA(AC) decreases from 8.48% for 1,000 orders to only 0.73% for 16,000 orders. Note that the LSA approach always has a higher assignment cost (the GA(AC) only takes the assignment cost into account) but a lower average total cost.

**GA(AUF) versus GA(AC):** The GA(AUF) approach clearly outperforms the GA(AC) approach, with a relative limited increase in CPU time. The relative improvement increases from 0.62% (1,000 orders) to 2.09% (16,000 orders). Improvements are most remarkable in the earliness cost, utilisation costs and production flow cost.

**LSA versus GA(AUF):** The LSA approach outperforms the GA(AUF) approach for problem instances with up to 4,000 orders, but performs worse for problem instances with 8,000 and 16,000 orders. Consequently, due to the heavy CPU-time burden, the LSA approach is not able to find high quality solutions (within the 100 generated schedules) that outperform the simple yet time efficient GA(AUF) approach.

<< insert table 3 here >>

#### 4.3 Flexibility and sensitivity of the objective function

In this section, we analyse the impact of all cost factors on the schedule quality and test the ability to modify the cost input parameters to obtain production schedules satisfying company specific objectives. In our

experiment, we carefully change the cost input factors  $e_i$ ,  $l_i$ ,  $u_j$ ,  $b_{ln}^-$  and  $b_{ln}^+$  and test their influence on the total quality of the schedule. More precisely, we multiply the original cost factor values (see section 4.1) by a factor 0.25, 0.5, 1, 2, 4 or 8 respectively, holding all other cost factors constant, and measure the resulting schedule quality by the following four performance measures:

**Average lateness:** measure the average lateness  $\bar{C}^L$  of all orders over the complete scheduling horizon (in days).

**Average earliness:** measures the average earliness  $\bar{C}^E$  of all orders over the complete scheduling horizon (in days).

**Average utilisation:** measures the average machine utilisation/capacity ratio as

$$\bar{C}^U = \frac{\sum_{i=1}^{nri} \sum_{j=1}^{nrj} \sum_{k=1}^{nrk} pt_{ij} x_{ijk}}{\sum_{jk} c_{jk} + \Delta c_{jk-1}}.$$

**Average production flow:** measures the ratio of all production flow deviations and all pre-specified flow quantities as  $\bar{C}^F = \frac{\sum_{l=1}^{nrl} \sum_{n=1}^{nrn} \sum_{m=1}^{nrm} |F_{lnm}|}{\sum_{l=1}^{nrl} \sum_{n=1}^{nrn} \sum_{m=1}^{nrm} F_{lnm}}$ .

Note that this experiment has also been set up to validate the quality of our cost reduction estimates  $CR_{ijk}^L$ ,  $CR_{ijk}^E$ ,  $CR_{ijk}^U$  and  $CR_{ijk}^F$ , since these estimates will influence the objective function of the knapsack problem and hence, the quality of the constructed schedule. Figure 6 displays the results for the problem instances with 8,000 orders and 10 different random machine assignments. All other problem instances or machine assignment procedures reveal similar results.

<< insert figure 6 here >>

Figure 6 clearly shows that the schedule quality, expressed in terms of the four performance measures, clearly depends on the input cost factors. All figures show an improved performance for the corresponding input factor (e.g. figure 6(a) shows an improved average lateness for increasing values for the unit lateness cost  $l_i$ , figure 6(b) shows an improved average earliness for increasing value for the unit earliness cost  $e_i$ , etc...). This result illustrates the quality of our cost reduction estimates (see section 3.1) and the ability of the users of the scheduling algorithm to influence and define the schedule quality (the importance of each part of the multiple objective) by modifying the input cost factors. Note that the four cost factors of the multiple objective function not always show a trade-off. Figures 5(a) and 5(c) show that increasing importance of order lateness costs has a beneficial effect on the order lateness as well as on the average machine utilisation, and vice versa. Hence, both objectives are correlated as they stimulate earliest start schedules.

## 5 Conclusions

In this paper, we presented a **two-stage** finite capacity production scheduling algorithm for an integrated steel company located in Belgium. The **disaggregate planning** algorithm takes various case-specific constraints into account and aims at the optimization of multiple objectives.

The algorithm consists of two solution steps. A machine assignment step assigns each order to a unique machine for each production step. We have tested three different machine assignment methods, each taking various cost factors into account. The second step constructs a schedule where each operation of all orders is assigned to a particular day, given the assigned machines of the previous step. To determine which orders should be selected for scheduling at each machine during each day, we construct knapsack problems that take capacity constraints, precedence constraints and set-up constraints as well as the multiple objectives (lateness costs, earliness costs, utilisation costs and flow costs) into account.

We have tested our algorithm on a randomly generated dataset and have shown that our algorithm is flexible towards the user in terms of input cost parameters. Moreover, we show that a local search machine assignment (step 1) combined with an optimal knapsack solver (step 2) leads to the best performing results.

## Acknowledgements

We would like to thank Krist Blomme, (member of SYMO (Systems and Models), specialist in line scheduling) and Alain Zegers (member of SYMO, specialist in line scheduling) of Arcelor Ghent for drawing our attention to the challenging nature of the production scheduling project and for giving us the permission to use the data of the project. We are also grateful to Frederik Fransoo (member of SYMO, specialist in production scheduling) for providing us information about the data and for the numerous conversations in constructing new fictive data that reflects the real-life characteristics of the production scheduling environment.

## References

- Assaf, I., Chen, M. and Katzberg, J., 1997. Steel production schedule generation, *International Journal of Production Research*, 35, 467-477.
- Bitran, G.R. and Tirupati, D., 1993. Hierarchical production planning. In: *Logistics of Production and Inventory*, edited by S.C. Graves, A.H.G. Rinnooy Kan and P.H. Zipkin, North Holland Publishing, Amsterdam, pp. 523-567.
- Chen, X. and Wu, Q., 2002, Formulating the steel scheduling problem as a TSPTW, *Intelligent Control and Automation*, 3, 1744-1748.

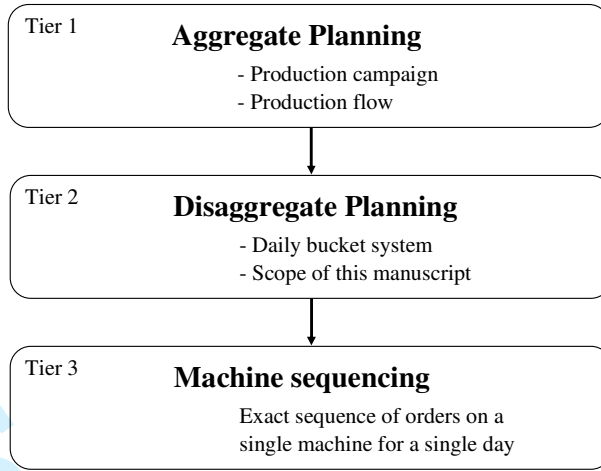


- 1  
2  
3 Chen, M. and Wang, W., 1997. A linear programming model for integrated steel production and distribution  
4 planning, *International Journal of Operations and Production Management*, 17, 592-610.  
5  
6 Cowling, P.I., 2003. A flexible decision support system for steel hot rolling mill scheduling, *Computers and*  
7  
8 *Industrial Engineering*, 45, 307-321.  
9  
10 Cowling, P.I., Ouelhadj, D. and Petrovic, S., 2003. A multi-agent architecture for dynamic scheduling of  
11  
12 steel hot rolling, *Journal of Intelligent Manufacturing*, 14, 457-470.  
13  
14 Cowling, P.I., Ouelhadj, D. and Petrovic, S., 2004. Dynamic scheduling of steel casting and milling using  
15  
16 multi-agents, *Production Planning and Control*, 15, 178-188.  
17  
18 Cowling, P.I. and Rezig, W., 2000. Integration of continuous caster and hot strip mill planning for steel  
19  
20 production, *Journal of Scheduling*, 3, 185-208.  
21  
22 Dijkstra, E.W., 1959. A note on two problems in connexion with graphs, *Numerische Mathematik.*, 1, 269-  
23  
24 271.  
25  
26 Feldman, P. and Shtub, A., 2007. Model for cost estimation in a finite-capacity environment, *International*  
27  
28 *Journal of Production Research*, 44:2, 305-327.  
29  
30 Dutta, G. and Fourer R., 2001. A survey of mathematical programming applications in integrated steel  
31  
32 plants, *Manufacturing and Service Operations Management*, 3, 387-400.  
33  
34 Harjunkski, I., Grossman, I.E., 2001. A decomposition approach for the scheduling of a steel plant  
35  
36 production, *Computers & Chemical Engineering*, 25, 1647-1660.  
37  
38 Hax, AC and Meal, HC.(1975) Hierarchical integration of production planning and scheduling. In *TIMS*  
39  
40 *Studies in Management Science.* . New York : Logistics North-Holland, American Elsevier.  
41  
42 Hirschberg, D.S., Wong, C.K., 1976. A polynomial-time algorithm for the knapsack problem with two  
43  
44 variables, *Journal of the ACM*, 23 (1), 147-154.  
45  
46 Jacobs, T.L., Wright, J.R. and Cobbs, A.E., 1998. Optimal inter-process steel production scheduling,  
47  
48 *Computers and Operations Research*, 15, 497-507.  
49  
50 Kistner, K.P. and Steven, M. 1991. Applications of operations research in hierarchical production planning.  
51  
52 In: *Modern Production Concept, Theory and Application*, edited by G. Fandel and G. Zapfel, Springer-  
53  
54 Verlag, Berlin.  
55  
56 Kolesar, J., 1967. A branch and bound algorithm for the knapsack problem. *Management Science*, 13 (9),  
57  
58 723-735.  
59  
60 Kumar, V., Kumar, S., Tiwari, M.K. and Chan, F.T.S., 2007. Auction-based approach to resolve the  
scheduling problem in the steel making process, *International Journal of Production Research*, 44, 1503-  
1522.  
Lee, H.S., Murthy, S.S., Haider, S.W., Morse, D.V., 1996. Primary production scheduling at steelmaking  
industries. *IBM Journal of Research Development*, 40 (2), 231-252.  
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, 317-335.

- 1  
2  
3 Mckay, K.N. Safayeni, G.P. and Buzacott, J.A., 1995. A review of hierarchical production planning, and its  
4 applicability to modern manufacturing, *Production Planning and Control*, 6, 384-394.  
5  
6 Neureuther, B., Polak, G. and Sanders, N., 2004. A hierarchical production plan for a make-to-order steel  
7 fabrication plant, *Production Planning & Control*, 15 (3), 324-335.  
8  
9 Okuda, K., 2001. Hierarchical structure in manufacturing systems: a literature survey, *International Journal*  
10 *of Manufacturing Technology Management*, 3, 210-224.  
11  
12 Okano, H., Davenport, A.J., Trumbo, M., Reddy, C., Yoda, K., Amano, M., 2004. Finishing line scheduling  
13 in the steel industry, *IBM journal of research and development*, 48 (5), 811-830.  
14  
15 Ouelhadj, D., Petrovic, S., Cowling, P.I. and Meisels, A., 2004. Inter-agent cooperation and communication  
16 for agent-based robust dynamic scheduling in steel production, *Advanced Engineering Informatics*, 18,  
17 161-172.  
18  
19 Petersen, C.M., Sorensen, K.L. and Vidal, R.V.V., 1992. Inter-process synchronization in steel production,  
20 *International Journal of Production Research*, 30, 1415-1425.  
21  
22 Portmann, M.C. and Rohr, D., 1995. Hierarchical production management applied to an iron and steel  
23 industry, *Journal of Intelligent Manufacturing*, 6, 79-85.  
24  
25 Roy, R., Adesola, A. and Thornton, S., 2004. Development of a knowledge model for managing schedule  
26 disturbance in steel-making, *International Journal of Production Research*, 42, 3975-3994.  
27  
28 Sasidhar, B. and Achary, K.K., 1991. A multiple arc network model of production planning in a steel mill,  
29 *International Journal of Production Economics*, 22, 195-202.  
30  
31 Tang, L., Liu, J., Rong, A. and Yang, Z., 2000a. A mathematical programming model for scheduling  
32 steelmaking continuous casting production, *European Journal of Operational Research*, 120, 423-435.  
33  
34 Tang, L., Liu, J., Rong, A. and Yang, Z., 2000b. A multiple travelling salesman problem model for hot  
35 rolling scheduling in Shanghai Baoshan Iron and Steel Complex, *European Journal of Operational*  
36 *Research*, 124, 267-282.  
37  
38 Tang, L., Liu, J., Rong, A., Yang, Z., 2001. A review of planning and scheduling systems and methods for  
39 integrated steel production, *European Journal of Operational Research*, 133, 1-20.  
40  
41 Tang, L., Luh, P.B., Liu, J., Yang, Z., 2002. Steel-making process scheduling using Lagrangian relaxation,  
42 *International Journal of Production Research*, 40, 55-70.  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60



**Figure 1.** The three-tier hierarchical planning of Arcelor Ghent

For Peer Review Only

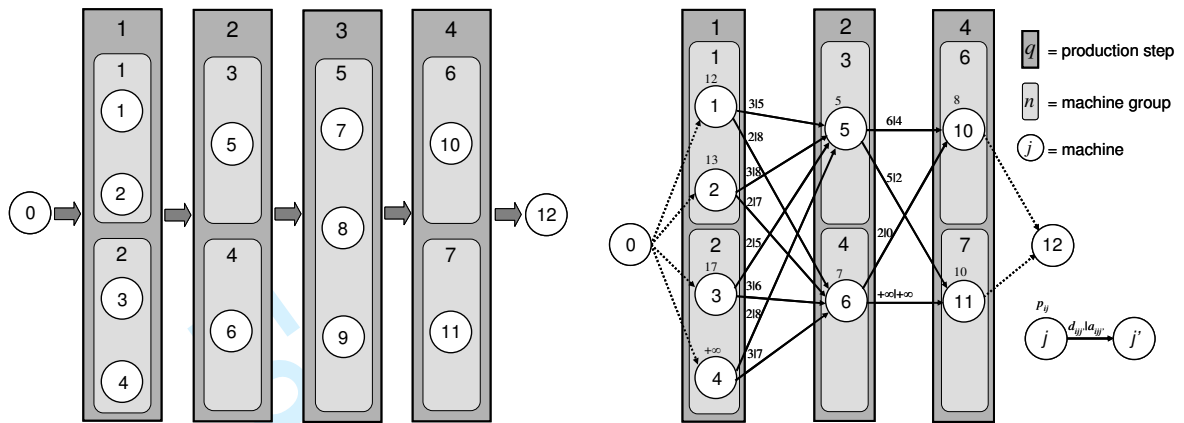


Figure 2. An example steel shop (left (a)) and an example order routing network (right (b))

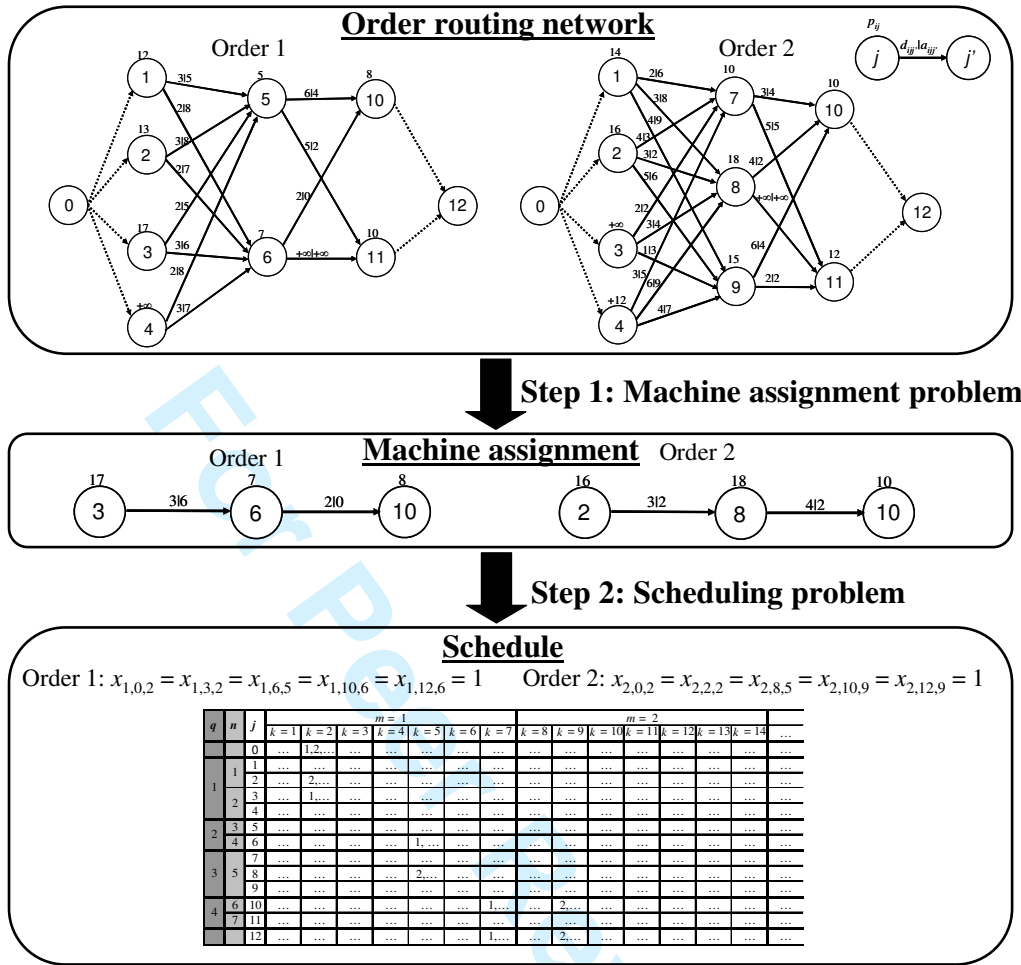
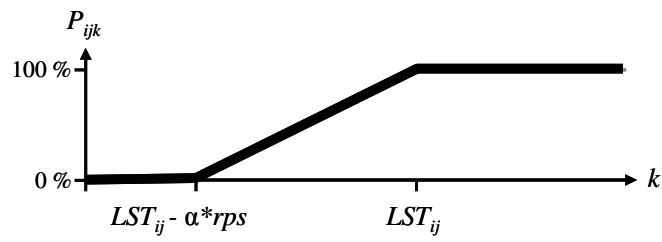


Figure 3. The disaggregate planning approach



**Figure 4.** An estimate for order lateness

For Peer Review Only

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

$l$	$f_{lmm} + \Delta f_{lmm-1}$	$b_{ln}^-$	$b_{ln}^+$	$\sum_{i' \in F_l^O} v_{i'} \sum_{j' \in S_n^{JN}} \sum_{\substack{k' \in W_m \\ k' < k}} x_{i'j'k'}$
1	40	4	8	10
2	30	2	6	20
3	50	8	10	52

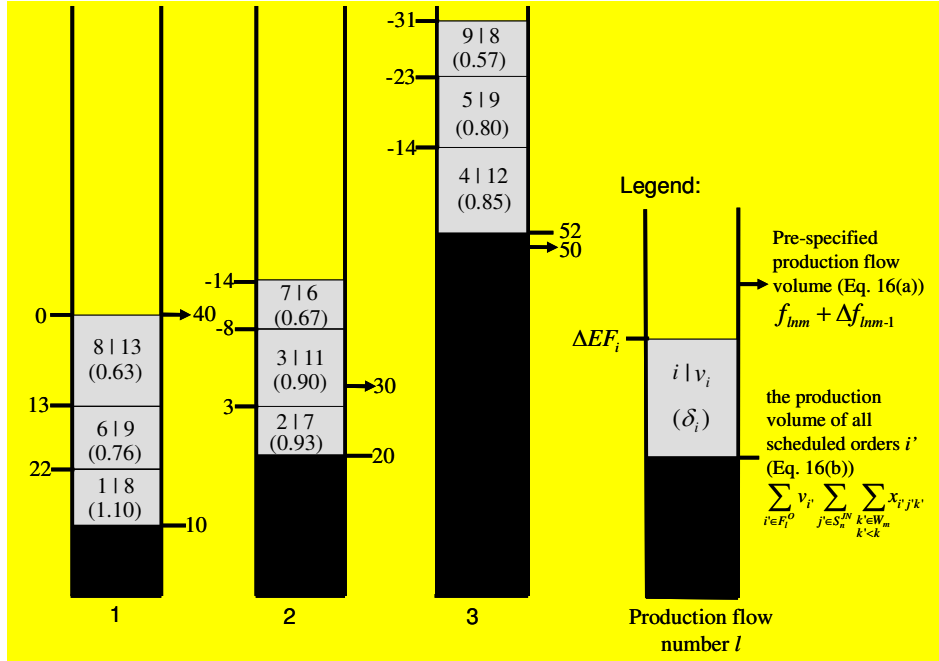


Figure 5. A fictive example to illustrate the calculation of the production flow cost reduction

<u>Steel-shop characteristics</u>	
$nrq$	Number of production steps (index $q = 1, \dots, nrq$ )
$nrn$	Number of machine groups (index $n = 1, \dots, nrn$ )
$nrj$	Number of machines (index $j = 1, \dots, nrj$ )
$S^{NQ}$	Set of machine groups of production step $q$
$S^{JQ}$	Set of machines of production step $q$
$S_n^{JN}$	Set of machines of machine group $n$
<u>Scheduling horizon</u>	
$nrk$	Number of days (index $k = 1, \dots, nrk$ )
$nrn$	Number of weeks (index $m = 1, \dots, nrn$ )
<u>Order characteristics</u>	
$nri$	Number of orders (index $i = 1, \dots, nri$ )
Each order can be characterised by:	
$v_i$	Volume of order $i$ (in tons)
$t_i$	Due date or delivery date of order $i$
$O_i^o$	Set of production steps needed to produce order $i$ ( $O_i^o \subset \{1, \dots, nrq\}$ )
The orders can be aggregated in <u>production campaigns</u> , as follows:	
$C_{jk}$	Set of production orders $i$ that belong to a campaign for machine $j$ on day $k$ (defines whether orders are allowed or forbidden to be produced for each machine and each day)
The orders can be aggregated in <u>production flows</u> , as follows:	
$nrl$	Number of order groups (production flows) for the flow constraints (index $l = 1, \dots, nrl$ )
$F_l^o$	Set of production orders that belong to production flow $l$
$f_{lmm}$	Pre-specified flow quantity (in tons) for order group $l$ on machine group $n$ during week $m$
<u>Order routing network</u>	
<u>Durations:</u>	
$p_{ij}$	Processing time (in minutes) i.e. function of order $i$ , machine $j$ and volume $v$
$d_{jj'}$	Intermediate duration for order $i$ between machine $j$ and machine $j'$ (in days) i.e. transportation and minimal cooling down time
$c_{jk}$	Capacity of machine $j$ on day $k$ (in minutes)
<u>Unit weights:</u>	
$a_{ijj'}$	Assignment cost of order $i$ (in €) if assigned to machines $j$ and $j'$ i.e. sum of costs of transportation and cooling down
$e_i$	Earliness unit cost for order $i$ (in € per day)
$l_i$	Lateness unit cost for order $i$ (in € per day)
$u_j$	Utilization cost of machine $j$ (in € per minute deviation of 100% utilization)
$b_{ln}^-$	Lower flow penalty cost (scheduled production $< f_{lmm}$ ) (in € per ton)
$b_{ln}^+$	Upper flow penalty cost (scheduled production $> f_{lmm}$ ) (in € per ton)

Table 1. Problem formulation parameters

<i>nri</i>	1000		2000		4000		8000		16000	
	exact	heuristic	exact	heuristic	exact	heuristic	exact	heuristic	exact	heuristic
Avg. CPU	0.04s	0.04s	0.08s	0.07s	0.18s	0.16s	0.47s	0.38s	1.70s	0.84s
Avg. $TC$	448,739	453,800	802,156	809,077	1,510,731	1,519,521	2,928,450	2,937,001	5,721,389	5,730,693
Avg. $C^A$	126,557	126,557	253,846	253,846	506,713	506,713	1,009,435	1,009,435	2,024,264	2,024,264
Avg. $C^L$	23,127	22,829	30,735	30,751	53,344	53,897	90,705	90,806	166,188	166,402
Avg. $C^E$	29,339	28,828	64,943	64,218	132,080	131,066	277,023	275,966	565,306	564,322
Avg. $C^U$	129,451	135,275	212,094	219,668	374,385	383,357	701,847	711,758	1,368,577	1,379,292
Avg. $C^F$	140,264	140,309	240,536	240,593	444,208	444,487	849,440	849,035	1,597,052	1,596,411

**Table 2.** Comparison between exact and heuristic knapsack procedure

For Peer Review Only

<i>nri</i>		1000	2000	4000	8000	16000
Avg. CPU	Random	0.04s	0.08s	0.18s	0.47s	1.70s
	GA(AC)	0.07s	0.13s	0.30s	0.92s	2.46s
	LSA	5.54s	9.86s	25.39s	84.63s	229.38s
	GA(UAF)	0.17s	0.32s	0.88s	2.54s	6.60s
Avg. TC	Random	448,739	802,156	1,510,731	2,928,450	5,721,389
	GA(AC)	362,651	645,390	1,209,376	2,358,950	4,604,384
	LSA	331,868	605,474	1,174,837	2,330,723	4,570,717
	GA(UAF)	360,407	638,359	1,192,511	2,322,494	4,507,826
Avg. $C^A$	Random	126,557	253,846	506,713	1,009,435	2,024,264
	GA(AC)	43,110	86,780	173,116	345,530	691,843
	LSA	45,013	88,945	174,456	346,356	692,463
	GA(UAF)	53,739	103,660	205,559	410,145	820,593
Avg. $C^L$	Random	23,127	30,735	53,344	90,705	166,188
	GA(AC)	17,837	25,501	46,462	82,581	155,232
	LSA	15,440	22,276	43,245	81,326	152,738
	GA(UAF)	24,974	34,900	61,035	112,022	209,644
Avg. $C^E$	Random	29,339	64,943	132,080	277,023	565,306
	GA(AC)	27,164	60,428	119,746	255,733	514,755
	LSA	27,211	59,645	120,009	255,524	514,987
	GA(UAF)	23,192	49,362	99,788	198,063	399,730
Avg. $C^U$	Random	129,451	212,094	374,385	701,847	1,368,577
	GA(AC)	118,611	193,099	340,400	634,104	1,248,311
	LSA	113,521	189,348	337,780	634,050	1,249,648
	GA(UAF)	105,974	171,961	301,392	568,778	1,110,039
Avg. $C^F$	Random	140,264	240,536	444,208	849,440	1,597,052
	GA(AC)	155,930	279,582	529,652	1,041,002	1,994,244
	LSA	130,684	245,260	499,346	1,013,468	1,960,880
	GA(UAF)	152,527	278,477	525,736	1,033,487	1,967,819

Table 3. Comparison of different machine assignment approaches



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

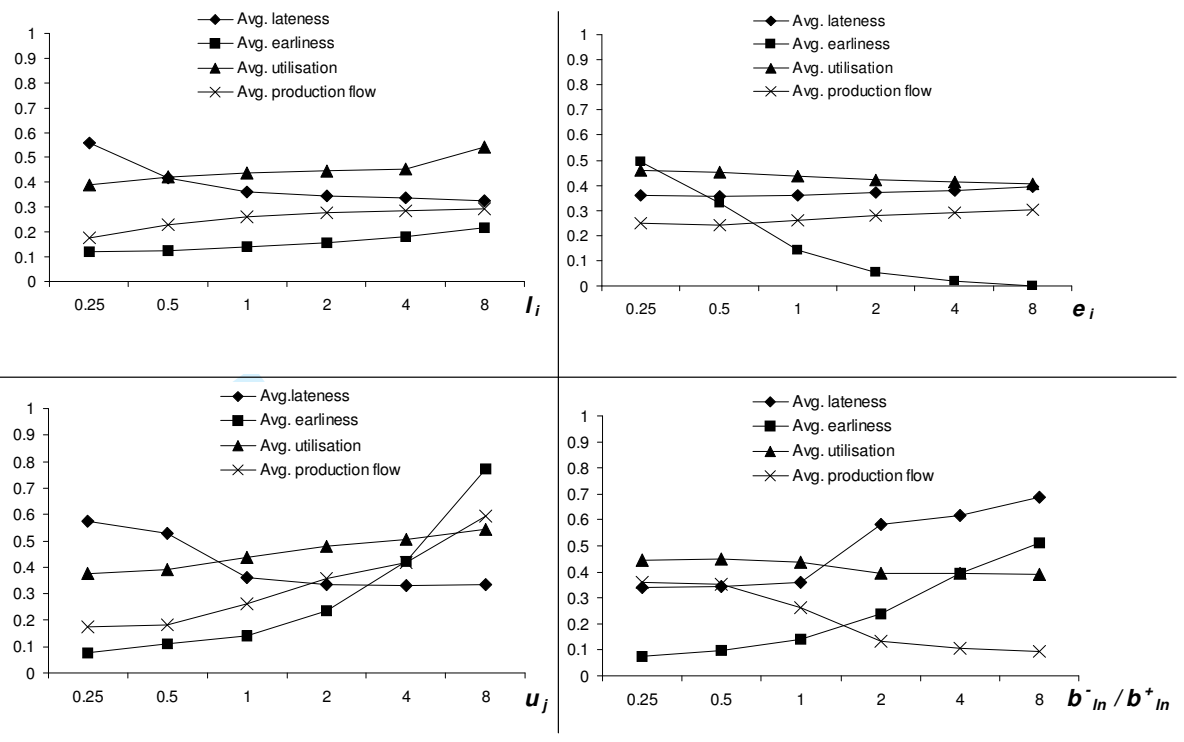


Figure 6. Influence of the input cost parameters on the performance measures