HAL
open science

Synchronization of Scarce Resources for a Parallel Machine Lotsizing Problem

Christian Almeder, Bernardo Almada-Lobo

To cite this version:

Christian Almeder, Bernardo Almada-Lobo. Synchronization of Scarce Resources for a Parallel Machine Lotsizing Problem. International Journal of Production Research, 2011, 10.1080/00207543.2010.535570 . hal-00711439

HAL Id: hal-00711439

https://hal.science/hal-00711439v1

Submitted on 25 Jun 2012

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.

Synchronization of Scarce Resources for a Parallel Machine Lotsizing Problem

Journal:	*International Journal of Production Research*
Manuscript ID:	TPRS-2010-IJPR-0503.R1
Manuscript Type:	Original Manuscript
Date Submitted by the Author:	08-Sep-2010
Complete List of Authors:	Almeder, Christian; University of Vienna, Faculty of Economics, Business and Statistics Almada-Lobo, Bernardo; Faculty of Engineering of University of Porto, Department of Mechanical Engineering and Industrial Management
Keywords:	LOT SIZING, PRODUCTION PLANNING, SCHEDULING, MIXED INTEGER LINEAR PROGRAMMING
Keywords (user):	Scarce Resources

SCHOLARONE™
Manuscripts

Synchronization of Scarce Resources for a Parallel Machine Lotsizing Problem

Christian Almeder[a], Bernardo Almada-Lobo[b*]

September 8, 2010

[a]Department of Information Systems and Operations, Vienna University of Economics and Business, Nordbergstr. 15, 1090 Vienna, Austria, christian.almeder@wu.ac.at
[b]Faculdade de Engenharia da Universidade do Porto, Rua Dr. Roberto Frias s/n, Porto 4200-465, Portugal, almada.lobo@fe.up.pt

Abstract

In this paper we present a novel approach to tackle the synchronization of a secondary resource in lotsizing and scheduling problems. This kind of problem occurs in different manufacturing processes (e.g. wafer testing in the semiconductor industry, production and bottling of soft drinks). We consider a scenario of parallel unrelated machines which have to be equipped with a tool or need a special kind of resource for processing. Our approach allows tracing the assignment of these secondary resources across different machines and synchronizing their usage independently of time periods. We present extensions of the general lotsizing and scheduling problem and of the capacitated lotsizing problem. We prove that the latter model is a special case of the first one, but it performs computationally much better.

Keywords: *Lotsizing and Scheduling, Scarce resources, Synchronization, Mixed-integer programming*

*corresponding author

For Peer Review Only

Introduction

Today production processes and systems are getting more complex, increasing the challenges of managing large shop floors. Lotsizing and scheduling models are tailored for short-term planning considering the trade-off between time-consuming and costly setup and configuration processes and high inventory costs. Many formulations (in particular big-bucket models) do not give very detailed information on how the production lots should be scheduled. Realistic settings are often neglected and in complex environments it might be sometimes impossible to implement the results of such models, simply because there is no feasible schedule possible.

In this paper we consider a lotsizing and scheduling problem in a parallel machine environment with another resource necessary for the production process. The usage of these secondary resource must be synchronized across the parallel production process on the shop floor. We develop a general modeling concept for such a situation. In particular, we will use the wafer test process, a part of the production process of semiconductors, as a template to illustrate how such a synchronization problem can be tackled. In that case the parallel machines are automatic test stations and the secondary resource is a test head that is used to perform the according test operations on the wafers in the test stations (cf. Ellis et al., 2004). Other examples of similar situations are the front-end wafer production process (cf. Mönch, 2004) or the example of an automobile supplier as reported in Tempelmeier and Buschkühl (2008). The latter one is a lotsizing problem enhanced by considering a setup operator who has to perform the setup tasks on different parallel machines. The setup operator can be modeled as such a secondary resource necessary to start production. In order to generate a feasible schedule, the timing of the setup operations have to be modeled explicitly. Another example for necessary synchronization is the multi-stage production process of soft drink plants (cf. Ferreira et al., 2009; Toledo et al., 2009). The two production stages of preparing the syrup in tanks and bottling it on multiple lines have to be synchronized. The syrup in the tanks can be seen as a common resource, which is only temporarily available, but can be used on several bottling machines in parallel.

The main contribution to the research on lotsizing and scheduling in this paper is twofold:

- We analyze the synchronization problem of a secondary scarce resource occurring in lotsizing and scheduling considering a parallel unrelated machine environment.
- We develop two different model formulations, a model based on the capacitated lotsizing problem (CLSP) and another one based on the general lotsizing and scheduling model (GLSP). We show that the GLSP-type approach allows a very accurate modeling of the situation but is computationally very hard to solve, whereas the CLSP-type model has some drawbacks for special cases but is much easier to solve. We compare the strength of both formulations and improve them by adding valid inequalities.

The reminder of the paper is organized as follows: we start with a review of the relevant literature, followed by a detailed problem description. Furthermore, we develop GLSP- and CLSP-related models for the lotsizing and scheduling with synchronization problem. A comparison between models is carried out on a small example as well as the proof that shows that the CLSP-related model is a special case of the GLSP-related model. Valid inequalities are developed for both

For Peer Review Only

models. We report some computational experiments and conclude with final remarks and comments.

Literature review

Motivated by industrial practice, the research community has been trying to solve more realistic and comprehensive production planning problems (e.g. Almada-Lobo et al., 2008b). In fact, the need to be able to respond quickly to market changes requires refined production planning models better able to represent and exploit the flexibility of the production process (cf. Pochet and Wolsey, 2006). Naturally, the strong interrelation between lotsizing and scheduling decisions emphasize the importance of an integrated decision making.

The capacitated lotsizing problem (CLSP) is considered to be a big-bucket model as the planning horizon is partitioned into a small number of long time periods allowing for several products/setups to be produced/performed per bucket (cf. Billington et al., 1983). Standard CLSP does not sequence or schedule products within a period. Nevertheless, when setup times are considerable with respect to the length of a period and capacity is tight, it is necessary to consider which product a machine is ready to process in the beginning of each period, i.e., setups need to be carried over to the following periods. Here, lotsizing is linked to partial sequencing, as the products produced last and first in two consecutive periods are determined (e.g. Suerie and Stadtler, 2003). In the presence of significant sequence-dependent setup costs and/or times, it is necessary to schedule production in each time period. Haase (1996) introduces the capacitated lotsizing problem with sequence-dependent setup costs (CLSD) for a single-machine and Kang et al. (1999) extend this approach to parallel machines. For the case of positive sequence-dependent setup times and costs Almada-Lobo et al. (2008a) develop a model formulation. Quadt and Kuhn (2005) present a framework for applying a CLSP-like model to a flexible flow line problem. The reader is referred to Karimi et al. (2003) and Buschkühl et al. (2010) for a recent review on CLSP formulations and state-of-the-art solution methods and to Quadt and Kuhn (2008) for a review on the extensions of CLSP.

In contrast to the big-bucket approach, researchers developed also small-bucket models (e.g., DLSP - discrete lot-sizing problem, PLSP - proportional lot-sizing problem) where at most one setup might be performed in each period. Here, the planning horizon is divided into many short periods (such as days, shifts, or hours). Naturally, that kind of models delivers a more detailed production plan than the standard CLSP, because by default a complete production sequence is generated. Jans and Degraeve (2004) present an application of a small-bucket model to the production of tires where multiple resources have to be considered. But those approaches have significant drawbacks, as the period length is determined by the minimal lot size which might lead to a huge number of periods. Drexl and Kimms (1997) give a detailed overview on small-bucket and big-bucket models. One mixed approach, which tries to overcome the drawbacks of small- and big-bucket models is the general lotsizing and scheduling problem (GLSP) developed by Fleischmann and Meyr (1997) and extended later to parallel machines by Meyr (2002). Here, the consideration of sequence-dependent setups and multiple machines is straightforward.

Jans and Degraeve (2008) give an overview of recent developments in the field of modeling deterministic single-level dynamic lotsizing problems to cope with various industrial extensions.

For Peer Review Only

The authors point out interesting areas for future research, such as lotsizing and scheduling on parallel machines and an increasing attention to model specific characteristics of the production process, which is valuable in solving real-life planning problems. In many real-world cases the different production lots cannot be scheduled independently from each other. Either there might be precedence rules present, such as for the multi-level lotsizing problem or multiple resources or tools have to be shared among different production processes. When considering parallel machine environment, i.e. simultaneous production is possible, such interdependencies have to be respected, otherwise the computed production plan might be impossible to implement. Dastidar and Nagi (2005) develop a model for scheduling injection molding operations, where several different resources are necessary for the production. Using a small-bucket approach the authors are able to block the resources necessary at workcenters for production during the whole period, even if there is some idle time. Akturk and Onen (2002) describe a lotsizing problem for a single CNC-machine considering tooling decisions. The lotsizing decisions depend on the setups of the tooling process, but since it is a single-machine problem they do not consider sharing the tools with other parallel production processes.

Actually, our work is in line with these research directions. We intend to give new insights by addressing the synchronization problem and possible ways on how to deal with it.

Problem description

Let us consider a manufacturing environment with N products to be scheduled on M parallel machines with K secondary resources. External period (usually days or weeks) demand for products is known in advance. Both product switchover and production on a machine require an additional resource (hereafter referred as tool) that is continuously attached to the machine throughout those processes. The following requirements/assumptions are made for the considered lotsizing and scheduling problem:

- the shop floor is operated 24 hours a day;
- backlogging is allowed to overcome capacity-tight periods;
- two types of scarce resources: non-identical parallel machines and tools (only a few are available as they are usually very expensive);
- significant tool sequence-dependent setup times and costs are incurred for switchovers from one tool to another;
- setup times and costs for product switchovers on the same machine with the same tool are negligible, thus the sequence of batches produced with the same tool is not determined (this assumption might be dropped, if necessary, by the cost of an increased number of binary variables);
- the tool remains attached to the machine during the production process;
- both machine and tools are blocked during a tool changeover, and cannot be used at that time in any other production or setup process.

For Peer Review Only

- triangular inequality holds for tool setup costs and times. (Trianlular setups are the usual case. Non-triangular setups would require in addition minimum lot size constraints in order to avoid phantom setups without production. Furthermore, it would complicate the formulation of a big-bucket model. See also Menezes et al., 2010.)

For a better understanding Figure 1 illustrates the usage of tools on three parallel machines. In order to produce products 3, 1 and 7, tool k is attached to machine 1. This tool is blocked on machine 1 during the switching from l_1 to k, the production of those three batches, some idle time in between and the switching from k to l_2. The order of the three production batches is irrelevant and is not determined.

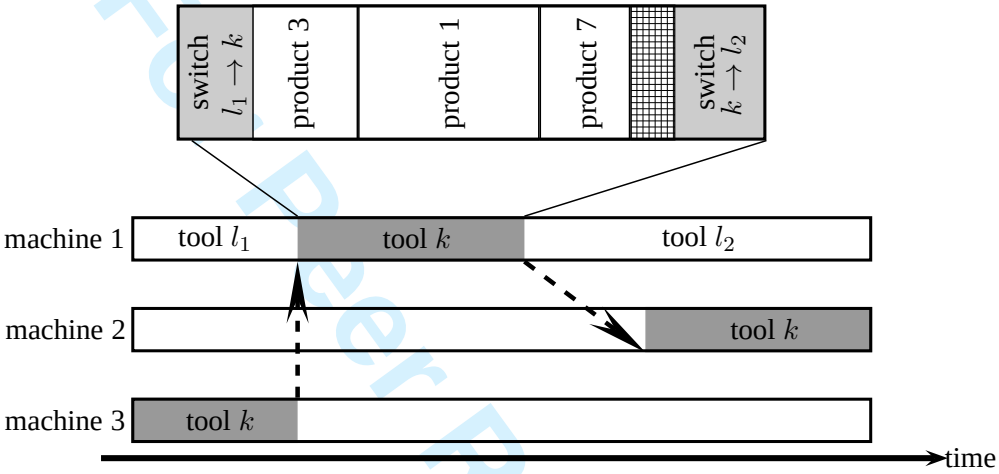

Figure 1: Synchronization of the usage of tools across different machines

The above assumptions are in line with the wafer test process. To consider the example of the automobile supplier, there would be only a single tool in the system representing the setup operator. That tool would be detached from the machine if the setup operation is finished. In the case of the soft drink production the tools would represent syrups prepared and stored in the tanks. Those tools would be available only for a certain time, but might be used simultaneously on different machines. The total production possible with a tool would be limited by the capacity of the according tanks.

Model formulation

GLSP-based model formulation

We start by developing a mixed-integer program (MIP) using the concept of the combined small-bucket–big-bucket approach on which the GLSP is based. (cf. Fleischmann and Meyr, 1997; Meyr, 2002). The main idea is to use a big-bucket model, where each of the "big" macro-periods is subdivided into "small" micro-periods. In order to gain more flexibility, the length of the micro-periods are variable and determined by the production quantities.

In our case a macro-period might be the usual day or week and inventory and backlog levels are determined based on these longer time periods. Each macro-period consists of several micro-periods which start with the beginning of a tool switch and end at the beginning of the next

For Peer Review Only

tool switch, except the first (last) micro-period which starts (finishes) at the beginning (end) of the macro-period. It is necessary to determine in advance the number of micro-periods in each macro-period. That automatically limits the number of tool switches during a macro-period. Note, that it is also possible to vary the number of micro-periods for each macro-period to gain more flexibility. But still these values are fixed a priori. For the sake of simplicity we use the same number of micro-periods for each macro-period.

In order to formulate the MIP, we first introduce the necessary notation (for simplicity we assume that the length of a macro period is normalized to one):

Dimension and indices:

N	number of products (index: i)
M	number of machines (index: m)
K	number of tools (index: k)
R	number of micro-periods in a macro-period (index: r)
T	number of macro-periods considered for production (index: t)

Parameters:

B_{i0}	initial backlog level of product i
b_{it}	penalty cost for the backlog of product i at the end of macro-period t
d_{it}	demand of product i in macro-period t
I_{i0}	initial inventory level of product i
h_{it}	holding cost for product i at the end of macro-period t
p_{imk}	time for producing one unit of product i on machine m using tool k
S_{im}	set of possible tools that can be used for product i on machine m
s_{mkl}	setup time for changeover from tool k to tool l on machine m
σ_{mkl}	setup cost for changeover from tool k to tool l on machine m
y_{mk0}	initial setup state (equals 1 if machine m is set up with tool k)

Decision variables:

B_{it}	backlog of product i at the end of macro-period t
I_{it}	inventory level of product i at the end of macro-period t
u^s_{trm}	start time of micro-period r in macro-period t on machine m
u^f_{trm}	release time of the tool used in micro-period r in macro-period t on machine m
$U_{trmr'm'}$	$\begin{cases} 1 & \text{if in macro-period } t \text{ micro-period } r \text{ on machine } m \text{ starts after micro-period } r' \text{ of machine } m' \\ 0 & \text{otherwise} \end{cases}$
x_{itrmk}	production quantity of product i on machine m in macro-period t during micro-period r using tool k
y_{trmk}	$\begin{cases} 1 & \text{if the tool } k \text{ is attached to machine } m \text{ in macro-period } t \text{ during micro-period } r \\ 0 & \text{otherwise} \end{cases}$

For Peer Review Only

$$z_{trmkl} \begin{cases} 1 & \text{if there is a switch from tool } k \text{ to tool } l \text{ on machine } m \text{ in macro-period } t \\ & \text{during micro-period } r \\ 0 & \text{otherwise} \end{cases}$$

There are three cost factors occurring in the above problem description, namely inventory cost, backorder cost and setup cost for changing the tools. Hence, the objective is to minimize the sum of these costs

$$\min F_G = \min \sum_{t=1}^{T} \left(\sum_{i=1}^{N} (h_{it} \cdot I_{it} + b_{it} \cdot B_{it}) + \sum_{r=1}^{R} \sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{l=1}^{K} \sigma_{mkl} \cdot z_{trmkl} \right), \tag{1}$$

Since it might happen that we do not switch the tool at the beginning of a certain micro-period (i.e. $z_{trmkk} = 1$) the according cost factor σ_{mkk} must be 0. The backorder costs of the last period b_{iT} should be significantly higher than the costs for the other periods in order to penalize unsatisfied demand at the end of the planning horizon.

Demands are given for each macro-period, therefore inventory and backlog levels have to be computed on that level. Since the production is split up into production on different machines, micro-periods, and tools, it is necessary to sum up all these production quantities

$$I_{it} - B_{it} - I_{i(t-1)} + B_{i(t-1)} - \sum_{m=1}^{M} \sum_{r=1}^{R} \sum_{k=1}^{K} x_{itrmk} + d_{it} = 0 \qquad \forall i, t \tag{2}$$

In order to avoid the simultaneous usage of the same tool on different machines, we have to determine the exact times of using a tool. A micro-period is defined to be the time from the start of the tool exchange until the start of the next tool exchange. Hence, we have to capture the time point when a micro-period starts. A tool is only available to other machines after the complete tool exchange has been performed. Therefore, the release time of a tool on a machine is determined by the start of the next micro-period plus the time for the tool switch. As shown in Figure 2, a micro-period consists of the tool switch, the total production using that tool and possible some idle time, which might be necessary for the synchronization of the tools across different machines.

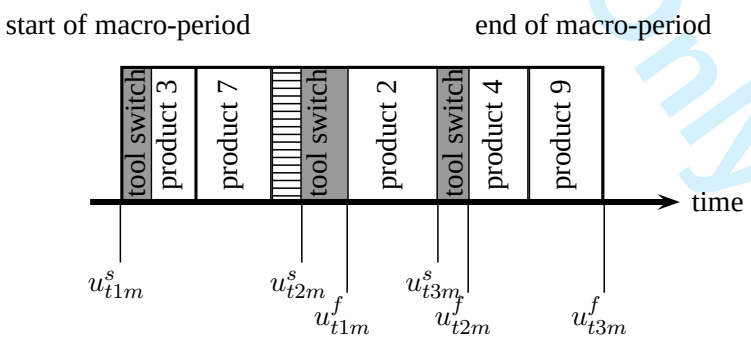

Figure 2: Values of variables u^s_{trm} and u^f_{trm} within a macro-period

The following set of constraints is necessary to calculate the correct starting of the micro-

For Peer Review Only

periods and the release times of the tools:

$$u^s_{t1m} = 0 \qquad \forall t, m \qquad (3)$$

$$u^s_{trm} \geq u^s_{t(r-1)m} + \sum_{k=1}^{K}\sum_{l=1}^{K} s_{mkl} \cdot z_{t(r-1)mkl} + \sum_{i=1}^{N}\sum_{k=1}^{K} p_{imk} \cdot x_{it(r-1)mk} \qquad \forall t, m, r > 1 \qquad (4)$$

$$u^f_{trm} \geq u^s_{t(r+1)m} + \sum_{k=1}^{K}\sum_{l=1}^{K} s_{mkl} \cdot z_{t(r+1)mkl} \qquad \forall t, m, r < R \qquad (5)$$

$$u^f_{tRm} \geq u^s_{tRm} + \sum_{k=1}^{K}\sum_{l=1}^{K} s_{mkl} \cdot z_{tRmkl} + \sum_{i=1}^{N}\sum_{k=1}^{K} p_{imk} \cdot x_{itRmk} \qquad \forall t, m \qquad (6)$$

$$u^f_{trm} \geq u^f_{t(r-1)m} \qquad \forall t, m, r > 1 \qquad (7)$$

$$u^f_{tRm} = 1 \qquad \forall t, m \qquad (8)$$

Constraints (3) force the first micro-period to start at the beginning of the macro-period. If there is no switch of tools at the beginning of the period, a virtual switch from tool k to tool k is performed ($z_{t1mkk} = 1$). The following constraints (4) relate the start times of subsequent micro-periods. The coefficients p_{imk} and s_{mkl} measure the times necessary to produce one unit and to perform the tool exchange, respectively. Note, that the last term of the constraint represents the total production time of all products i produced with the same tool k. In order to allow idle time for gaining flexibility for the synchronization inequalities are used. The release times u^f_{trm} of a tool that is attached to the machine m during micro-period r must be set after the tool has been removed. Hence, we have to consider the start time of the next micro-period and the tool switch occurring at the beginning of that micro-period. This relation is represented by constraints (5). Because after the last micro-period there is no start time in the current macro-period, the last micro-period ends after the start of the last micro-period plus the tool switch and the production, as denoted by constraints (6). As constraints (5) provide only a lower bound for the release times, constraints (7) are necessary to ensure the right ordering. Constraints (8) force the last micro-periods to finish at the end of the macro-periods. The right-hand-side represents the total available capacity (in terms of time units) for each macro-period. Due to this calculation of start and release times, it is not necessary to include the common capacity constraints. The total production and the tool exchanges in all micro-periods cannot consume more time than available in the macro-period.

Now, having the correct start and end times of the tool usage on a machine, we can compare them across machines and avoid an overlap (cf. Figure 1). For that purpose, we need an additional binary variable, which indicates the ordering of the micro-periods on different machines. $U_{trmr'm'} = 1$ means that micro-period r on machine m starts after the micro-period r' on machine m'.

$$U_{trmr'm'} \geq u^s_{trm} - u^s_{tr'm'} \qquad \forall t, r, r', m, m' \neq m \qquad (9)$$

$$U_{trmr'm'} + U_{tr'm'rm} = 1 \qquad \forall t, r, r', m, m' \neq m \qquad (10)$$

In order to synchronize each tool across different machines, we have to enforce that the

For Peer Review Only

finishing time of micro-period r' on machine m' is before the start time of micro-period r on machine m, if the start time of micro-period r' on machine m' is also before the start time of micro-period r on machine m (i.e. $U_{trmr'm'} = 1$). Furthermore, the same tool must be used in both micro-periods.

$$u^f_{tr'm'} - (3 - U_{trmr'm'} - y_{trmk} - y_{tr'm'k}) \leq u^s_{trm} \qquad \forall t, r, r', m, m' \neq m, k \qquad (11)$$

To illustrate those constraints let us assume the situation in Figure 1 representing one macro-period t. Due to (9) $U_{t2113} = 1$, because the first micro-period on machine 3 using tool k starts before the second micro-period on machine 1. Because of (10), $U_{t1321} = 0$ and from (11), $u^f_{t13} - (3 - U_{t2113} - y_{t21k} - y_{t13k}) = u^f_{t13} \leq u^s_{t21}$, ensuring that tool k is not used on machine 1 before its usage on machine 3 ends.

In order to ensure that each machine has a tool attached in each micro-period, we need the following constraints:

$$\sum_{k=1}^{K} y_{trmk} = 1 \qquad \forall t, r, m \qquad (12)$$

Furthermore, it is necessary to enforce a switch of tools if different tools are used in subsequent micro-periods.

$$z_{trmkl} \geq y_{t(r-1)mk} + y_{trml} - 1 \qquad \forall t, r > 1, m, l, k \qquad (13)$$

$$z_{t1mkl} \geq y_{(t-1)Rmk} + y_{t1ml} - 1 \qquad \forall t > 1, m, l, k \qquad (14)$$

Production in a micro-period is only allowed if the corresponding tool is attached to the machine and if no other technical restrictions prohibit it.

$$p_{imk} \cdot x_{itrmk} \leq y_{trmk} \qquad \forall i, t, r, m, k \qquad (15)$$

$$x_{itrmk} = 0 \qquad \forall i, t, r, m, k \notin S_{im} \qquad (16)$$

Finally we end up with the usual non-negativity and binary constraints. Meyr (2002) has already shown that it is not necessary to force the integrality of the z_{trmkl} variables, because due to constraints (13) and (14) and the objective function (1) they can take only the values 0 or 1.

$$I_{it}, B_{it}, u^s_{trm}, u^f_{trm}, x_{itrmk}, z_{trmkl} \geq 0, \quad U_{trmr'm'}, y_{trmk} \in \{0, 1\} \qquad (17)$$

Remark. So far in the model description we have assumed that each machine is always equipped with a tool, which might not be necessary and might make it more difficult to compute feasible solutions. In order to capture also the situation of empty machines (without any tool attached during idle times), we might introduce a virtual tool $k = 1$ which represents the "no-tool" case. Obviously we have to remove the synchronization restriction for that tool, that means that constraints (11) must not hold for $k = 1$.

We refer to this model as general lotsizing and scheduling problem on parallel machines with tool synchronization ($GLSP^{PM}_{ToolSync}$).

For Peer Review Only

CLSP-based model formulation

As referred beforehand, the CLSP is considered to be a big-bucket problem, because several setups may be performed per period. In the presence of significant sequence-dependent setup times and costs, it is necessary to determine the sizes and the sequence of the lots simultaneously.

In this section, we develop an extension of the single-level formulation with sequence-dependent setups (e.g. Almada-Lobo et al., 2007) considering additional constraints related to the tools (synchronization, switching, capacity, among others) and multiple machines.

In order to define the model, we use the following additional decision variables:

μ^s_{tmk} start time of the attachment of tool k to machine m in period t

μ^f_{tmk} end time of the tool exchange where tool k is removed from machine m in period t

μ^{max}_{tm} end time of the last tool exchange performed on machine m in period t

X_{itmk} production of product i produced in period t on machine m with tool k

$$\alpha_{tmk} \begin{cases} 1 & \text{if tool } k \text{ is attached to machine } m \text{ at the beginning of period } t \\ 0 & \text{otherwise} \end{cases}$$

$$T_{tmkl} \begin{cases} 1 & \text{if there is a switch from tool } k \text{ to } l \text{ on machine } m \text{ in period } t \\ 0 & \text{otherwise} \end{cases}$$

$$W_{tmm'k} \begin{cases} 1 & \text{if tool } k \text{ is used on machine } m \text{ after being used on machine } m' \text{ in period } t \\ 0 & \text{otherwise} \end{cases}$$

Note, that $\alpha_{(T+1)mk}$ denotes the final tool attached to each machine at the end of the planning horizon.

The according lotsizing and scheduling problem reads:

$$\min F_C = \min \sum_{i=1}^{N} \sum_{t=1}^{T} (h_{it} \cdot I_{it} + b_{it} \cdot B_{it}) + \sum_{t=1}^{T} \sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{l=1}^{K} \sigma_{mkl} \cdot T_{tmkl} \tag{18}$$

$$I_{it} - B_{it} - I_{i(t-1)} + B_{i(t-1)} - \sum_{m=1}^{M} \sum_{k=1}^{K} X_{itmk} + d_{it} = 0 \quad \forall i, t \tag{19}$$

$$p_{imk} \cdot X_{itmk} \leq \sum_{l=1}^{K} T_{tmlk} + \alpha_{tmk} \qquad \forall i, t, m, k \tag{20}$$

As before, the objective function (18) is to minimize the sum of sequence-dependent setup costs for changing the tools, holding costs and backorder costs. We assume that $T_{tmkk} = 0$ for every tool k. Constraints (19) balance production, inventories and backlogs with demand. We do not impose explicitly capacity constraints for tools and machines. Such requirements are guaranteed later by other constraints. Requirements (20) ensure that a product is produced only if the machine has been set up for the appropriate tool. Constraints (21)-(24) determine the sequence of tools on each machine in each period and keep track of the machine configuration state by recording the tool that is attached to each machine.

For Peer Review Only

$$\sum_{k=1}^{K} \alpha_{tmk} = 1 \qquad \forall t, m \qquad (21)$$

$$\sum_{m=1}^{M} \alpha_{tmk} \leq 1 \qquad \forall t, k \qquad (22)$$

$$\sum_{l=1}^{K} T_{tmlk} + \alpha_{tmk} = \sum_{l=1}^{K} T_{tmkl} + \alpha_{(t+1)mk} \qquad \forall t, m, k \qquad (23)$$

Constraints (21) impose that the machine is set up for exactly one tool at the beginning of each period, and (22) forces each tool to be attached to at most one machine at the beginning of the time bucket. Constraints (23) ensure a balanced network flow of the machine configuration states and carry the setup configuration state of the machine over to the next period. Note that constraints (19)– (23) admit solutions that have at most one path, but potentially many cycles (subtours). In order to obtain a feasible solution, we need to cut off the disconnected subtours, as follows:

$$\mu^f_{tmk} - s_{mkl} \cdot T_{tmkl} + T_{tmkl} - 1 - \alpha_{(t+1)ml} \leq \mu^s_{tml} \qquad \forall t, m, k, l \quad (24)$$

$$\mu^f_{tmk} - s_{mkl} \cdot T_{tmkl} \geq \mu^s_{tml} + T_{tmkl} - 1 - \alpha_{(t+1)ml} \qquad \forall t, m, k, l \quad (25)$$

$$\mu^s_{tmk} + \sum_{i=1}^{N} p_{imk} \cdot X_{itmk} + \sum_{l=1}^{K} (s_{mlk} \cdot T_{tmlk} + s_{mkl} \cdot T_{tmkl}) - \alpha_{(t+1)mk} \leq \mu^f_{tmk} \qquad \forall t, m, k \quad (26)$$

$$\mu^s_{tmk} + \sum_{i=1}^{N} p_{imk} \cdot X_{itmk} + \sum_{l=1}^{K} s_{mkl} \cdot T_{tmkl} \leq \mu^f_{tmk} \qquad \forall t, m, k \quad (27)$$

The aim of the traveling salesman problem (TSP)-related requirements (24) is twofold. First, they ensure that the assignments of setup components to the machines are feasible and do not lead to unreachable sequences or subcycles, i.e., they cut off disconnected subtours (cf. Almada-Lobo et al., 2008a). Taking (27), which ensures the minimum time span between start and release times, to replace μ^f_{tmk} in expression (24), we derive the equivalent requirement:

$$\mu^s_{tmk} + \sum_{i=1}^{N} p_{imk} \cdot X_{itmk} + T_{tmkl} - 1 - \alpha_{(t+1)ml} \leq \mu^s_{tml} \quad \forall t, m, k, l$$

Given a subtour $\Phi : k_1 \rightarrow k_2 \rightarrow \cdots \rightarrow k_l \rightarrow k_1$ with $l < K$ and summing up the previous constraint for every tool belonging to that subtour, we get: $\sum_{k,i} p_{imk} \cdot X_{itmk} \leq \sum_{k \in \Phi} \alpha_{(t+1)mk}$. Observe that the left-hand side of this expression is less or equal to one. Naturally, this inequality only holds in case α equals 1 for one of the tools in the subtour. Therefore, we remove disconnected cycles and also subtours that appear in the middle of a sequence. Hence, $\sum_{l=1}^{K} T_{tmlk} \leq 1$ and $\sum_{l=1}^{K} T_{tmkl} \leq 1$ hold. Second, along with (25), constraints (24) link the finishing time of the usage of a tool k to the start time of a different tool l that is attached to machine m right after tool k in period t. Clearly, in case the two tools are attached contiguously to machine m ($T_{tmkl} = 1$) and tool l is not the last to be attached in period t, then $\mu^f_{tmk} = \mu^s_{tml} + s_{mkl}$.

For Peer Review Only

During an attachment or removal of a tool on a machine, the tool is blocked and cannot be used somewhere else. In other words, both an input setup to tool k and an output setup from k on a machine consume both resources' capacity. Therefore, to compute the release time of a tool (see constraint (26)), we have to add both switchover and production times to the start time. We do not need to impose an equality here, as the tool may remain idle for some time after the production.

Constraints (26) ensure the correct time spanning from μ^s_{tmk} to μ^f_{tmk}, but do not fix the exact timing of the usage of the tool.

$$\mu^s_{tmk} \leq 1 - \alpha_{tmk} \qquad \forall t, m, k \qquad (28)$$

$$\mu^f_{tmk} \geq \alpha_{(t+1)mk} - \sum_{l=1}^{K} T_{tmlk} \qquad \forall t, m, k \qquad (29)$$

Requirements (28) impose that the start time of a tool on a machine is zero, in case it was the last one attached to the machine in the previous period. In addition, (29) forces μ^f_{tmk} to be one if k is the only tool attached to machine m in period t. Hence, in this situation we have $\mu^s_{tmk} = 0$ and $\mu^f_{tmk} = 1$.

In order to synchronize the tools across different machines, i.e., to ensure that the same tool is not used simultaneously on two or more machines, we use the following constraints:

$$\mu^f_{tm'k} \leq \mu^s_{tmk} + 1 - W_{tmm'k} \qquad \forall t, m, m' \neq m, k \qquad (30)$$

$$W_{tmm'k} + \mu^s_{tm'k} \geq \mu^f_{tmk} \qquad \forall t, m, m' \neq m, k \qquad (31)$$

$W_{tmm'k}$ equals to one if tool k is used on machine m later than on machine m' in period t. Depending on $W_{tmm'k}$, for the same quadruple (m, m', k, t), only one of the constraints (30) and (31) is active. Remark that constraints (24)–(26) are non-active when tool l is the last tool to be attached to a machine in a period (i.e., $\alpha_{(t+1)ml} = 1$), therefore μ^s_{tml} and μ^f_{tml} are not defined. However, the minimum time span from μ^s_{tml} to μ^f_{tml} is determined by (27) – these constraints are redundant in any other case. In order to overcome this issue, we use variables μ^{max}_{tm}, that denote the release time of the last but one tool on machine m in period t. μ^{max}_{tm} are computed as follows:

$$\mu^{max}_{tm} \geq \mu^f_{tmk} + T_{tmkl} + \alpha_{(t+1)ml} - 2 \qquad \forall t, m, k, l \qquad (32)$$

$$\mu^f_{tmk} \geq \mu^{max}_{tm} + T_{tmkl} + \alpha_{(t+1)ml} - 2 \qquad \forall t, m, k, l \qquad (33)$$

$$\mu^{max}_{tm} \leq 1 - \sum_{i=1}^{N} p_{imk} \cdot X_{itmk} + (1 + \alpha_{tmk} - \alpha_{(t+1)mk}) \qquad \forall t, m, k \qquad (34)$$

$$\mu^{max}_{tm} \leq 1 \qquad \forall t, m \qquad (35)$$

Both (32) and (33) only apply when tool l is the last in period t. In such a case, it is easy to see that $\mu^{max}_{tm} = \mu^f_{tmk}$, where k is the tool that precedes l on machine m. Constraints (34) and (35) bound μ^{max}_{tm} from above considering the amount produced with the last tool. We assume

For Peer Review Only

that in case of a cycle of size greater than one on tool k (i.e, $\alpha_{tmk} = \alpha_{(t+1)mk} = 1$), there is only one batch of production with tool k, which takes place the first time this tool is attached to the machine (observe that (34) are loose for such cases). Now we just need to synchronize the last tool across the machines, as follows:

$$\mu^{max}_{tm'} - \sum_{l=1}^{K} s_{m'lk} \cdot T_{tm'lk} + (2 - 2 \cdot \alpha_{(t+1)m'k}) \geq \mu^{f}_{tmk} + \left(\sum_{l=1}^{K} T_{tmlk} + \alpha_{tmk} - 1 \right) \quad \forall t, m, m' \neq m, k \quad (36)$$

If machine m is set up for tool k at the end of period t, then (36) imposes that the start time (given by the term $\mu^{max}_{tm'} - \sum_l s_{m'lk} \cdot T_{tm'lk}$) occurs after the release time of the same tool on other machines in the same time period. The term $\left(\sum_{l=1}^{K} T_{tmlk} + \alpha_{tmk} - 1\right)$ ensures that the inequality is inactive if tool k is not used on machine m in period t.

Finally, we impose technological constraints (37), preventing the assignment of products with some tools to machines, and the variables domain.

$$X_{itmk} = 0 \qquad \forall i, t, m, k \notin S_{im} \quad (37)$$

$$I_{it}, B_{it}, X_{itmk}, \mu^{s}_{tmk}, \mu^{f}_{tmk}, \mu^{max}_{tm} \geq 0; \alpha_{tmk}, T_{tmkl}, W_{tmm'k} \in \{0, 1\} \qquad \forall i, t, m, m', k, l \quad (38)$$

Remark. Both machine and component capacity constraints do not need to be explicitly defined in the model. On one hand, constraints (34) and (35) force the total production and setup time in each period not to exceed the available machine capacity. On the other hand, the correct synchronization of the tools across the machine (guaranteed by (30), (31) and (36)) clearly avoids that a tool is used longer than its capacity.

We refer to this model as a capacitated lotsizing and scheduling problem on parallel machines with tool synchronization ($CLSP^{PM}_{ToolSync}$).

Comparison of the formulations

Illustrative example

We will start with a small example to figure out the main differences of both formulations. The example consists of two machines, three tools (but only two of them are used for production), five products, and a planning horizon of two macro-periods. The restrictions of the production as well as demand, holding and backorder costs, and resource requirements are shown in Table 1. Setup times and costs are depicted in Table 2. Machine 1 is initially equipped with tool A and machine 2 is equipped with tool B.

Considering the $GLSP^{PM}_{ToolSync}$, we have at first to decide about the number of micro-periods. If the synchronization of the tools is not considered, three micro-periods are enough. Since there are only three tools available, without synchronization the exact timing of the production within

For Peer Review Only

Table 1: Demand and technical restrictions of products for the illustrative example

product	restrictions	d_{i1}	d_{i2}	$h_{i1} = h_{i2}$	b_{i1}	b_{i2}	p_{imk}
1	machine 1 / tool B	0.1	0	5	50	5000	1
2	machine 2 / tool B	0.3	0	5	50	5000	1
3	machine 2 / tool C	0.2	0	5	50	5000	1
4	machine 1 / tool C	0.15	0.7	5	50	5000	1
5	machine 2/ tool B	0	1.25	5	50	5000	1

Table 2: Setup times / setup costs for the illustrative example

tool	A	B	C
A	0/0	0.07/1.6	0.07/1.6
B	0.07/1.6	0/0	1/3
C	0.07/1.6	1/3	0/0

the macro-period does not matter. But in the case of synchronization, it might be optimal to attach the same tool several times to the same machine. For this small example it is necessary to consider at least four micro-periods. Figure 3 shows the optimal tooling and production schedule for both machines. We see, that for machine 1 in period 1 the tool sequences A→B→A→C and for machine 2 is B→C→B. The attachment of tool A (which is not used for production) is mandatory, because tool B is blocked on machine 2 to produce product 2 that cannot be assigned to machine 1. Due to the tight capacity in the second period it is necessary to go into the second period again with tool B attached to machine 2. The total cost of this schedule is 12.05 consisting of 1.25 inventory costs and 10.8 due to tool switches.

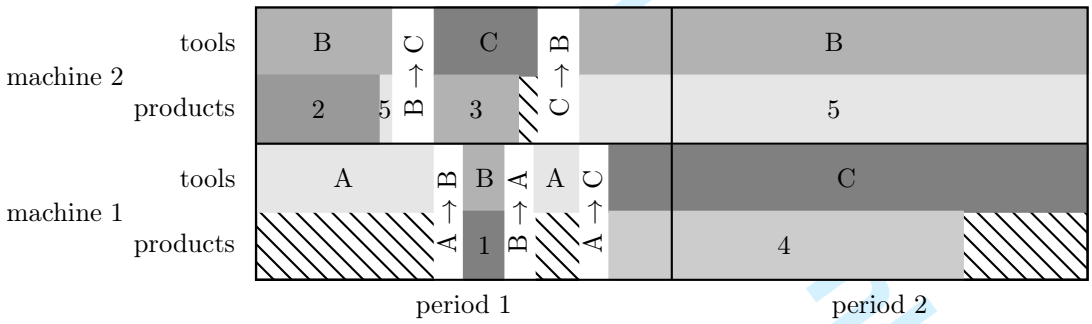

Figure 3: Solution of the $GLSP^{PM}_{ToolSync}$

The $CLSP^{PM}_{ToolSync}$ has no parameter to be determined in advance, but due to the concept of subtour elimination, a tool can only be attached twice to the same machine, in case it is the first and the last in a period. Hence, the optimal solution of the $CLSP^{PM}_{ToolSync}$ shown in Figure 4 is different, as the sequence A→B→A→C is not allowed here. Instead, we have the cycle A→B→A on machine 1 in the first period and the necessary switch to tool C is only performed in the second period. The tool sequence on machine 2 is the same as that of the $GLSP^{PM}_{ToolSync}$. The resulting total costs are 19.55 due to backorder costs of 7.5.

To illustrate the fact that without synchronization one might get solutions which are not im-

For Peer Review Only

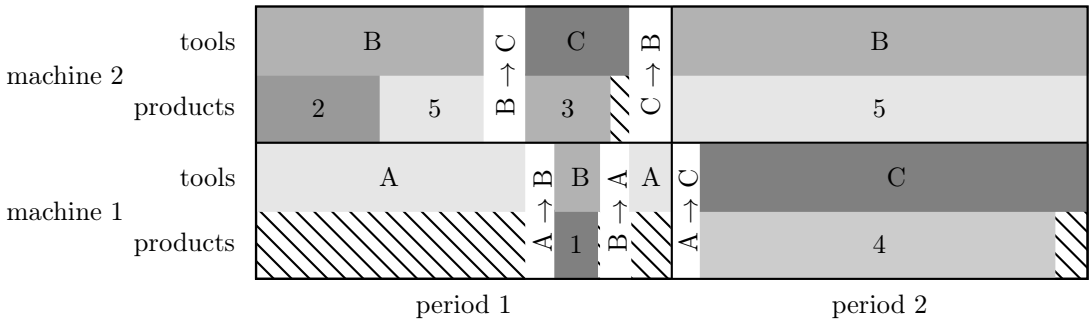

Figure 4: Solution of the $CLSP^{PM}_{ToolSync}$

plementable in practice, we solved the above example using $CLSP^{PM}_{ToolSync}$ without constraints (30), (31) and (36) which take care of the synchronization, and adding instead capacity constraints for the machines and tools:

$$\sum_{i=1}^{N}\sum_{k=1}^{K} p_{imk}x_{itmk} + \sum_{k=1}^{K}\sum_{l=1}^{K} s_{mkl}T_{tmkl} \leq 1 \qquad \forall t, m$$

$$\sum_{i=1}^{N}\sum_{m=1}^{M} p_{imk}x_{itmk} + \sum_{m=1}^{M}\sum_{l=1}^{K} (s_{mkl}T_{tmkl} + s_{mlk}T_{tmlk}) \leq 1 \qquad \forall t, k$$

As we have two machines and two tools and setups for product switches using the same tool are negligible, one may think that a feasible solution to that problem would be to dedicate each tool to a machine. However, from the three products to be produced with tool B (see Table 1) contrarily to products 2 and 5, product 1 has to be manufactured on machine 1. Therefore the construction of an initial feasible solution is not straight forward. Such situations of *non-transitive* production requirements for tools and machines are typical for the semiconductor industry.

The optimal solution for that model without synchronization leads to a tool sequence of A→B→C for machine 1 and B→C→B for machine 2. Since either tool B or tool C is always attached to machine 2, it is not possible to make a changeover from tool B to tool C on machine 1, because for a changeover both tools must be available at the same time.

$CLSP^{PM}_{ToolSync}$ is a special case of $GLSP^{PM}_{ToolSync}$

The above example indicates that the two formulations might lead to different solutions. The $CLSP^{PM}_{ToolSync}$ seems more restrictive raising the question if it is a special case of the $GLSP^{PM}_{ToolSync}$ or if the two models are structurally different from each other. Beforehand we have to analyse the structure of the solutions. For a given set of parameters there might be decision variables which only appear in inactive constraints, i.e. their values are not well defined by the model. For example in the $CLSP^{PM}_{ToolSync}$, the start and end times μ^s_{tmk} and μ^f_{tmk} of a tool k which is not attached to machine m in period t are not determined. Their values have no impact on the solution itself. So we define a feasible solution of the $CLSP^{PM}_{ToolSync}$ as follows:

Definition 1. A *feasible equivalence solution set* ξ_C of the $CLSP^{PM}_{ToolSync}$ is a set of equivalent

For Peer Review Only

feasible solutions $(\mathbf{I}, \mathbf{B}, \mathbf{X}, \mathbf{T}, \alpha, \mu^{\mathbf{s}}, \mu^{\mathbf{f}}, \mu^{\mathbf{max}})$ which fulfill all constraints, have the same objective value and differ in variables only appearing in inactive constraints. $\mathcal{X}_C$ is the set of all feasible equivalence solution sets ξ_C.

Analogously we define for the $GLSP^{PM}_{ToolSync}$:

Definition 2. A *feasible equivalence solution set* ξ_G of the $GLSP^{PM}_{ToolSync}$ is a set of equivalent feasible solutions $(\mathbf{I}, \mathbf{B}, \mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}^{\mathbf{s}}, \mathbf{u}^{\mathbf{f}})$ which fulfill all constraints, have the same objective value and differ in variables only appearing in inactive constraints, respectively in zero-length micro-periods. $\mathcal{X}_G$ is the set of all feasible equivalence solution sets ξ_G.

Further on we consider any feasible realisation of the decision variables as a representative of the feasible equivalence solution set it is element of. Now, we postulate the following theorem:

Theorem 3. *$CLSP^{PM}_{ToolSync}$ is a special case of the $GLSP^{PM}_{ToolSync}$ with $R \geq K+1$ and there exists an injective mapping from the set of feasible equivalence solution sets $\mathcal{X}_C$ of the $CLSP^{PM}_{ToolSync}$ into the set of feasible equivalence solution sets $\mathcal{X}_G$ of the $GLSP^{PM}_{ToolSync}$ $f : \mathcal{X}_C \to \mathcal{X}_G; \xi_C \mapsto \xi_G$ such that the values of the objective functions are the same $F_C(\xi_C) = F_G(f(\xi_C))$.*

Remark. The binary variables $W_{tmm'k}$ and $U_{trmr'm'}$ are not considered in the mapping, because those are only auxiliary variables and their values are determined by the start and end times. Furthermore, we use the same notation for inventory and backorder levels in the $CLSP^{PM}_{ToolSync}$ and $GLSP^{PM}_{ToolSync}$ because they have the same meaning and are identical in both models.

The outline of the proof is that we define the mapping that translates a feasible solution of the $CLSP^{PM}_{ToolSync}$ into a representation of $GLSP^{PM}_{ToolSync}$. Based on this mapping it is possible to prove that all the constraints of the $GLSP^{PM}_{ToolSync}$ are fulfilled, and that the values of the objective functions are identical. A detailed proof can be found in the appendix available at `http://prolog.univie.ac.at/research/publications/downloads/Alm_2010_397.pdf`.

Remark. Since $CLSP^{PM}_{ToolSync}$ is a special case of $GLSP^{PM}_{ToolSync}$, we might introduce additional constraints for the $GLSP^{PM}_{ToolSync}$ to force both approaches to deliver identical solutions. For that purpose, it is necessary to avoid tooling sequences with subcycles in each macro-period or in other words, for each tool at most one switch from another tool is allowed.

$$\sum_{r=1}^{R} \sum_{\substack{l=1 \\ l \neq k}}^{K} z_{trmlk} \leq 1 \qquad \forall t, m, k$$

In the case of a cycle the $CLSP^{PM}_{ToolSync}$ allows only production at the beginning of the period. So we have to add

$$p_{imk} \cdot x_{itrmk} \leq 1 - y_{t1mk} \quad \forall i, t, m, k, r > 1.$$

Valid inequalities

In this section we present classes of valid inequalities to tighten the $CLSP^{PM}_{ToolSync}$ and $GLSP^{PM}_{ToolSync}$ formulations.

For Peer Review Only

Lemma 4. *The inequalities*

$$z_{trmkk} \leq u^s_{trm} \quad \forall t, r > 1, m, k \tag{39}$$

are valid for $GLSP^{PM}_{ToolSync}$.

Proof. To show that they hold, we use the equivalence sets of Definition 2. Recall that the lengths of micro-periods are variable and are determined by production and switchover times. Requirements (39) force empty (zero-length) micro-periods to be placed at the end of macro-periods. Start times for these last micro-periods equal one and, as there are neither production nor tool switches, finishing times are also one. In case of a sequence of micro-periods within a macro-period with the same tool attached to the machine, (39) reduce the solution space by forcing the whole batch to be produced in one micro-period or tool k to remain attached until the end of the macro-period. In other words, these inequalities avoid a start-up to be preceded by a phantom setup (excepting the first micro-period), as follows:

$$\sum_{k=1}^{K} \sum_{l=1(l\neq k)}^{K} z_{t(r-1)mkl} \geq \sum_{k=1}^{K} \sum_{l=1(l\neq k)}^{K} z_{trmkl} \qquad \forall t, r > 2, m$$

□

Lemma 5. *The following sets of inequalities*

$$W_{tmm'k} \leq \sum_{l=1}^{K} T_{tmlk} + \alpha_{tmk} \qquad \forall t, m, m' \neq m, k \tag{40}$$

$$W_{tmm'k} \leq 2 - \alpha_{tmk} - \sum_{l=1}^{K} T_{tm'lk} \quad \forall t, m, m' \neq m, k \tag{41}$$

are valid for $CLSP^{PM}_{ToolSync}$.

Proof. Note that from (30) and (31) follows that $W_{tmm'k} + W_{tm'mk} \leq 1$ (this result is somehow different from the requirement (10) of $GLSP^{PM}_{ToolSync}$). In fact, if tool k is not attached to machines m and m' in period t, then $W_{tmm'k} = W_{tm'mk} = 0$, and in case it is not attached just to m' then $W_{tm'mk} = 0$ and $W_{tmm'k} = 1$. Furthermore, relying on (22) and on the fact that subtours do not appear in the middle of any sequence (such cycles are cut off by (24)), it is easy to see that the right-hand side of (41) is always nonnegative. Now, to prove that (40) and (41) are valid, it suffices to consider the case $W_{tmm'k} = 1$. If $W_{tmm'k} = 1$ then tool k has to be attached in period t to machine m, i.e., $\sum_{l=1}^{K} T_{tmlk} + \alpha_{tmk} \geq 1$, therefore the inequalities (40) are satisfied.

Requirements (30) and (31) define $W_{tmm'k}$ to be one if tool k is used on machine m after it is used on machine m' in period t or in case m' is not set up to k. Consider the case both machines are set up to tool k in period t. If $\alpha_{tmk} = 1$, then tool k is the first to be attached to machine m in t, so clearly tool k is scheduled on machine m' afterwards (i.e. $W_{tmm'k} = 0$ and $W_{tm'mk} = 1$), which validates (41). In case tool k is not attached to one of machines m and m', constraints (41) are non-active. □

For Peer Review Only

Tests and Results

In this section, we compare the efficiency between $GLSP^{PM}_{ToolSync}$ and $CLSP^{PM}_{ToolSync}$ and evaluate computationally the impact of different instance parameters. In designing our study we constructed 6 different classes and generated randomly 10 instances for each parameter combination within the class, to a total of 480 test instances (see Table 3). For each of the 6 classes (determined by the number of products, machines and tools) we used a time horizon of 4 and 8 periods, an approximate capacity utilization of 75% and 95%, and low (S1) and high (S5) setup costs. The parameter ranges are given in Table 4. For the $GLSP^{PM}_{ToolSync}$ we set the number of micro-periods to $R = K + 1$ so that the $CLSP^{PM}_{ToolSync}$ is in fact a special case of the $GLSP^{PM}_{ToolSync}$.

Table 3: Classes of test instances

class	N	M	K	T	capacity utilization	setup cost
N10-M2-K4-Tx-Cx-Sx	10	2	4	4, 8	75%, 95%	1, 5
N10-M2-K8-Tx-Cx-Sx	10	2	8			
N25-M2-K8-Tx-Cx-Sx	25	2	8			
N25-M4-K8-Tx-Cx-Sx	25	4	8			
N25-M4-K12-Tx-Cx-Sx	25	4	12			
N25-M8-K12-Tx-Cx-Sx	25	8	12			

Table 4: Parameter ranges used for the test instances

Parameter	Ranges and values used for the test instances
B_{i0}, I_{i0}	$= 0$
b_{it}	$= 3h_{it}$ for $t < T$ and the last period: $b_{iT} = 300 * h_{iT}$
d_{it}	$\frac{\bar{d}_i}{2} + U[0; \bar{d}_i]$, where $\bar{d}_i$ is the average demand computed based on the given capacity utilization
h_{it}	$= h_i$ chosen randomly from the set $\{1, 2, 3\}$
p_{imk}	$U[0.008; 0.012]$
S_{im}	randomly generated, such that on average 20% of the machine-tool combinations can be used for product i. The same S_{im} is used for each of the 10 instances of each parameter variation.
s_{mkl}	$U[0.005; 0.01]$, i.e. a setup takes 0.5-1% of the periods length
σ_{mkl}	$= 10000 s_{mkl}$ (S1) or $= 50000 s_{mkl}$ (S5)
y_{mk0}	$= \begin{cases} 1 & m = k \\ 0 & \text{otherwise} \end{cases}$

Computational experiments were executed on an Intel Pentium D with 3.2 GHz CPU and 4 GB of random access memory. CPLEX 11.1 from ILOG was used as the MIP solver and the models were coded in OPL version 6.0.1. Tables 5-7 display the percentage of instances that CPLEX found a feasible solution (*%feas*), a capacity feasible solution (*%capfeas*), i.e. solutions without backlog at the end of the planning horizon, and the optimal solution (*%opt*), as well as the average gap (*avg. gap*) between the upper and lower bounds (in the case of capacity feasible

For Peer Review Only

solutions) and the average CPU times (*avg. time*) in seconds. For each instance we allow a maximum CPU time of 1 hour. Furthermore, the last column of Table 5 presents the difference (%) between the best solution obtained with $CLSP^{PM}_{ToolSync}$ to that with $GLSP^{PM}_{ToolSync}$.

It is clear that $CLSP^{PM}_{ToolSync}$ is always more efficient than $GLSP^{PM}_{ToolSync}$. In fact, CPLEX finds feasible solution to $CLSP^{PM}_{ToolSync}$ on every instance type and capacity feasible solutions for almost all instances with $K < 12$ and $K = 12$, $M = 4$, and $K = 12$, $M = 8$, $T = 4$. Observe that $GLSP^{PM}_{ToolSync}$ is not able to solve within one hour instances with $K > 8$ and $K = 8$, $M = 4$ for $T = 8$, and to find any feasible solution for instances with $M > 2$ (that account for 50% of the total number of instances). Clustering only the instances that both models are able to find capacity feasible solutions, the overall average gap between upper and lower bounds is 23% for $CLSP^{PM}_{ToolSync}$ and 79% for $GLSP^{PM}_{ToolSync}$, and the gap between the two models is -81%. When both models were solved to optimality, $CLSP^{PM}_{ToolSync}$ had also significantly lower solution times than the $GLSP^{PM}_{ToolSync}$. For example, for the N10-M2-K4-T4-C75-S5 instance $CLSP^{PM}_{ToolSync}$ and $GLSP^{PM}_{ToolSync}$ average solution times were 8 and 871 seconds, respectively, both delivering the same solution.

The number of products N influences the performance of CPLEX on both models. Comparing N25-M2-K8-T4-Cx-Sx and N25-M2-K8-T8-Cx-Sx to N10-M2-K8-T4-Cx-Sx and N10-M2-K8-T8-Cx-Sx (see Table 5), respectively, clearly the problem becomes harder as N increases. From Table 5, and as expected, the performance deteriorates significantly with the number of machines M. This is mainly caused by the synchronization of tools across the machines, as the number of binary variables $U's$ and $W's$ (respectively from $GLSP^{PM}_{ToolSync}$ and $CLSP^{PM}_{ToolSync}$) increases exponentially. Moreover, the number of tools K also influences negatively the results, as well as does the number of periods T.

Tables 6 and 7 illustrate the impact of the proportion between setup costs and setup times (S) and of the capacity utilization (C) for classes that CPLEX is able to find capacity feasible solutions for at least 50% of the instances. As setup costs start dominating holding costs and as instances get more tightly capacitated, the results worsen. In fact, the instance type Nx-Mx-Kx-Tx-Cx-S1 reports an average gap of 42% against 45% for the Nx-Mx-Kx-Tx-Cx-S5, while the gaps of Nx-Mx-Kx-Tx-C75-Sx and Nx-Mx-Kx-Tx-C95-Sx goes from 40% to 46%. The results are in line with others published in the literature.

For none of the test instances $GLSP^{PM}_{ToolSync}$ delivered a better solution than the $CLSP^{PM}_{ToolSync}$, although the latter one is a special case of the first one and imposes several additional restrictions to the solution. Definitely, this numerical study highlights the need for solution approaches to tackle this challenging and important practice-driven problem.

Conclusions and final remarks

The small illustrative example presented here shows that capturing the usage of secondary resources is necessary for generating feasible production plans in practice. Naturally, if such resources are expensive, only a limited number of them are available and their consideration in the production planning process is mandatory.

We present a new approach to tackle this kind of problem by tracing the exact timing of the setup operations, which allows us to schedule the secondary resources independently of the

For Peer Review Only

Table 5: Solutions of the test instances clustered by capacity utilization and setup costs.

	$CLSP^{PM}_{ToolSync}$					$GLSP^{PM}_{ToolSync}$					
	%feas	**%capfeas**	**%opt**	**avg. gap**	**avg. time**	**%feas**	**%capfeas**	**%opt**	**avg. gap**	**avg. time**	**diff. CLSP-GLSP**
N10-M2-K4-T4-Cx-Sx	100%	100%	100%	0.0%	12	100%	100%	40%	14.2%	2659	-3.3%
N10-M2-K4-T8-Cx-Sx	100%	100%	60%	5.4%	2264	100%	100%	0%	75.3%	3707	-33.4%
N10-M2-K8-T4-Cx-Sx	100%	100%	80%	2.6%	1458	100%	100%	0%	90.6%	3683	-100.3%
N10-M2-K8-T8-Cx-Sx	100%	100%	0%	43.2%	3648	100%	100%	0%	99.0%	3695	-147.2%
N25-M2-K8-T4-Cx-Sx	100%	100%	0%	27.0%	3644	100%	100%	0%	95.5%	3669	-88.2%
N25-M2-K8-T8-Cx-Sx	100%	100%	0%	61.2%	3642	100%	70%	0%	99.5%	3677	-117.8%
N25-M4-K8-T4-Cx-Sx	100%	100%	0%	49.6%	3643	100%	0%	0%	-	3633	-
N25-M4-K8-T8-Cx-Sx	100%	83%	0%	80.2%	3641	0%	0%	0%	-	-	-
N25-M4-K12-T4-Cx-Sx	100%	100%	0%	74.7%	3640	3%	0%	0%	-	3636	-
N25-M4-K12-T8-Cx-Sx	100%	73%	0%	93.0%	3632	0%	0%	0%	-	-	-
N25-M8-K12-T4-Cx-Sx	100%	68%	3%	70.7%	3592	0%	0%	0%	-	-	-
N25-M8-K12-T8-Cx-Sx	100%	10%	0%	60.1%	3626	0%	0%	0%	-	-	-

For Peer Review Only

Table 6: Solutions of $CLSP^{PM}_{ToolSync}$ of the test instances clustered by time periods and capacity utilization.

	%feas	%capfeas	%opt	avg. gap	avg. time
N10-M2-K4-Tx-Cx-S1	100%	100%	75%	3.0%	1355
N10-M2-K4-Tx-Cx-S5	100%	100%	85%	2.4%	921
N10-M2-K8-Tx-Cx-S1	100%	100%	35%	22.5%	2701
N10-M2-K8-Tx-Cx-S5	100%	100%	45%	23.3%	2404
N25-M2-K8-Tx-Cx-S1	100%	100%	0%	36.7%	3644
N25-M2-K8-Tx-Cx-S5	100%	100%	0%	51.6%	3642
N25-M4-K8-Tx-Cx-S1	100%	93%	0%	62.0%	3643
N25-M4-K8-Tx-Cx-S5	100%	90%	0%	64.9%	3640
N25-M4-K12-Tx-Cx-S1	100%	90%	0%	84.0%	3635
N25-M4-K12-Tx-Cx-S5	100%	83%	0%	80.7%	3636

Table 7: Solutions of $CLSP^{PM}_{ToolSync}$ of the test instances clustered by time periods and setup costs.

	%feas	%capfeas	%opt	avg. gap	avg. time
N10-M2-K4-Tx-C75-Sx	100%	100%	95%	0.3%	602
N10-M2-K4-Tx-C95-Sx	100%	100%	65%	5.1%	1674
N10-M2-K8-Tx-C75-Sx	100%	100%	50%	17.1%	1957
N10-M2-K8-Tx-C95-Sx	100%	100%	30%	28.8%	3149
N25-M2-K8-Tx-C75-Sx	100%	100%	0%	38.3%	3641
N25-M2-K8-Tx-C95-Sx	100%	100%	0%	49.9%	3645
N25-M4-K8-Tx-C75-Sx	100%	90%	0%	62.1%	3642
N25-M4-K8-Tx-C95-Sx	100%	93%	0%	64.7%	3642
N25-M4-K12-Tx-C75-Sx	100%	88%	0%	81.9%	3635
N25-M4-K12-Tx-C95-Sx	100%	85%	0%	83.0%	3637

period length of the underlying lotsizing and scheduling model. First we develop a GLSP-based formulation that is able to capture the production process with every necessary detail. The second CLSP-based approach imposes some restrictions, but has significant advantages in terms of computational tractability.

Introducing time variables for the tool changeovers allows to include also other restrictions of practical relevance. For example, it is possible to consider downtime of machines (e.g. for maintenance) by blocking that machine using an artificial blocking tool and impose the start and finishing times of that tool. Another example is the consideration of precedence rules of the production process within a time-period in contrast to delaying subsequent production steps to the next period.

But definitely, it is necessary to strengthen the model formulations by deriving more valid inequalities and to develop solution approaches that generate high-quality solutions for larger instances.

For Peer Review Only

Acknowledgments

This work was partly financed by the Austrian Science Foundation under the contract number J2815-N13. The authors want to thank Maria Antónia Carravilla and José Fernando Oliveira for many helpful discussions and comments on this work.

References

Akturk, M., Onen, S., 2002. Dynamic lot sizing and tool management in automated manufacturing systems. Computers & Operations Research 29 (8), 1059–1079.

Almada-Lobo, B., Carravilla, M., Oliveira, J., 2008a. A note on "The capacitated lot-sizing and scheduling problem with sequence-dependent setup costs and setup times". Computers & Operations Research 35 (4), 1374–1376.

Almada-Lobo, B., Klabjan, D., Carravilla, M., Oliveira, J., 2007. Single machine multi-product capacitated lot sizing with sequence-dependent setups. International Journal of Production Research 45 (20), 4873–4894.

Almada-Lobo, B., Oliveira, J., Carravilla, M., 2008b. Production planning and scheduling in the glass container industry: A VNS approach. International Journal of Production Economics 114 (1), 363–375.

Billington, P., McClain, J., Thomas, L., 1983. Mathematical programming approach to capacity-constrained MRP systems: Review, formulation and problem reduction. Management Science 29 (10), 1126–1141.

Buschkühl, L., Sahling, F., Helber, S., Tempelmeier, H., 2010. Dynamic capacitated lot-sizing problems: A classification and review of solution approaches. OR Spectrum 32 (2), 231–261.

Dastidar, S., Nagi, R., 2005. Scheduling injection molding operations with multiple resource constraints and sequence dependent setup times and costs. Computers & Operations Research 32 (11), 2987–3005.

Drexl, A., Kimms, A., 1997. Lot sizing and scheduling - survey and extensions. European Journal of Operational Research 99 (2), 221–235.

Ellis, K. P., Lu, Y., Bish, E. K., 2004. Scheduling of wafer test processes in semiconductor manufacturing. International Journal of Production Research 42 (2), 215–242.

Ferreira, D., Morabito, R., Rangel, S., 2009. Solution approaches for the soft drink integrated production lot sizing and scheduling problem. European Journal of Operational Research 196 (2), 697–706.

Fleischmann, B., Meyr, H., 1997. The general lotsizing and scheduling problem. OR Spektrum 19, 11–21.

Haase, K., 1996. Capacitated lot-sizing with sequence dependent setup costs. Operations Research Spektrum 18, 51–59.

For Peer Review Only

Jans, R., Degraeve, Z., 2004. An industrial extension of the discrete lot-iszing and scheduling problem. IIE Transactions 36 (1), 47–58.

Jans, R., Degraeve, Z., 2008. Modeling industrial lot sizing problems: A review. International Journal of Production Research 46 (6), 1619–1643.

Kang, S., Malik, K., Thomas, L., 1999. Lotsizing and scheduling on parallel machines with sequence-dependent setup cost. Management Science 45 (2), 273–289.

Karimi, B., Fatemi Ghomi, S., Wilson, J., 2003. The capacitated lot sizing problem: review of models and algorithms. OMEGA 31 (5), 365–378.

Menezes, A. A., Clark, A., Almada-Lobo, B., 2010. Capacitated lotsizing and scheduling with sequence-dependent, period overlapping and non-triangular setups. Journal of Scheduling, doi: 10.1007/s10951–010–0.

Meyr, H., 2002. Simultaneous lotsizing and scheduling on parallel machines. European Journal of Operational Research 139 (2), 277 – 292.

Mönch, L., 2004. Scheduling-Framework für Jobs auf parallelen Maschinen in komplexen Produktionssystemen. Wirtschaftsinformatik 46 (6), 470–480.

Pochet, Y., Wolsey, L. A., 2006. Production Planning by Mixed Integer Programming. Springer Series in Operations Research and Financial Engineering. Springer, New York.

Quadt, D., Kuhn, H., 2005. Conceptual framework for lot-sizing and scheduling of flexible flow lines. International Journal of Production Research 43 (11), 2291–2308.

Quadt, D., Kuhn, H., 2008. Capacitated lot-sizing with extensions: A review. 4OR 6 (1), 61–83.

Suerie, C., Stadtler, H., 2003. The capacitated lot-sizing problems with linked lot sizes. Management Science 49 (8), 1039–1054.

Tempelmeier, H., Buschkühl, L., 2008. Dynamic multi-machine lotsizing and sequencing with simultaneous scheduling of a common setup resource. International Journal of Production Economics 113 (1), 401–412.

Toledo, C., Franca, P., Morabito, R., Kimms, A., 2009. Multi-population genetic algorithm to solve the synchronized and integrated two-level lot sizing and scheduling problem. International Journal of Production Research 47 (11), 3097–3119.

For Peer Review Only

Synchronization of Scarce Resources for a Parallel Machine Lotsizing Problem

Appendix

by Christian Almeder and Bernardo Almada-Lobo

Proof for Theorem 3

Proof. Let us consider a feasible solution of the $CLSP^{PM}_{ToolSync}$. Due to the subtour elimination guaby Christian Almeder and Bernardo Almada-Loboranteed by constraints (23), (24), and (27) there exists for each period t a well defined sequence of tools $\Phi_{tm} : k_{tm1} \to k_{tm2} \cdots \to k_{tm\lambda_{tm}}$ of length λ_{tm} that are attached to machine m. Such a sequence contains each tool at most once, except the last tool which might be identical with the first tool ($k_{tm1} = k_{tm\lambda_{tm}}$). These sequences are determined by the values of α_{tmk} and T_{tmkl}, i.e. if one of the variables are changed, the corresponding sequence changes and vice versa. From now on, we omit the indices t and m of $k_{tm\nu}$ and λ_{tm} for an easier representation.

We have to set the number of micro-periods in the $GLSP^{PM}_{ToolSync}$ to $R \geq K+1$ (note that $\lambda \leq K+1$) and define now a mapping f as follows (for an easy representation all variables of the $GLSP^{PM}_{ToolSync}$ not appearing in that mapping are set to 0):

$$
f\begin{pmatrix} I_{it} \\ B_{it} \\ X_{itmk} \\ \alpha_{tmk} \\ T_{tmkl} \\ \mu^s_{tmk} \\ \mu^f_{tmk} \\ \mu^{max}_{tm} \\ \Phi_{tm} \end{pmatrix} \mapsto \begin{pmatrix}
I_{it} \\
B_{it} \\
x_{it1mk_1} = X_{itmk_1}, x_{it2mk_2} = X_{itmk_2}, \ldots, x_{it(\lambda-1)mk_{\lambda-1}} = X_{itmk_{\lambda-1}}, \\
\qquad x_{it\lambda mk_\lambda} = \begin{cases} X_{itmk_\lambda} & k_\lambda \neq k_1 \\ 0 & k_\lambda = k_1 \end{cases} \\
y_{t1mk_1} = y_{t2mk_2} = \cdots = y_{t\lambda mk_\lambda} = y_{t(\lambda+1)mk_\lambda} = \ldots y_{tRmk_\lambda} = 1 \\
z_{t1mk_1k_1} = z_{t2mk_1k_2} = z_{t3mk_2k_3} = \cdots = z_{t\lambda mk_\lambda k_\lambda} = 1, \\
\qquad z_{t(\lambda+1)mk_\lambda k_\lambda} = \cdots = z_{tRmk_\lambda k_\lambda} = 1 \\
u^s_{t1m} = \mu^s_{tmk_1}, \ldots, u^s_{t(\lambda-1)m} = \mu^s_{tmk_{\lambda-1}}, u^s_{t\lambda m} = \mu^{max}_{tm} - s_{mk_{\lambda-1}k_\lambda}, \\
\qquad u^s_{t(\lambda+1)m} = \cdots = u^s_{tRm} = 1 \\
u^f_{t1m} = \mu^f_{tmk_1}, \ldots, u^f_{t(\lambda-1)m} = \mu^f_{tmk_{\lambda-1}}, u^f_{t\lambda m} = \cdots = u^f_{tRm} = 1
\end{pmatrix}
$$

According to the sequence of tools Φ_{tm} of the $CLSP^{PM}_{ToolSync}$ solution, the first λ micro-periods of the $GLSP^{PM}_{ToolSync}$ are used each for the production with the corresponding tool. During the remaining micro-periods no tool exchange is performed and the length of those micro-periods is set to zero ($u^s_{trm} = u^f_{trm} = 1$, $r > \lambda$). All decision variables of the $CLSP^{PM}_{ToolSync}$ appearing in active constraints are used in the mapping, i.e. if any of those variables changes in a feasible way, also the mapped solution of the $GLSP^{PM}_{ToolSync}$changes. Hence, the mapping is injective.

Now we show that all constraints of the $GLSP^{PM}_{ToolSync}$ are fulfilled. The inventory balance constraints (2) hold, because the I_{it} and B_{it} are not changed, and the total production during a

For Peer Review Only

macro-period for each item is the same in both model formulations

$$\sum_{m=1}^{M}\sum_{r=1}^{R}\sum_{k=1}^{K} x_{itrmk} = \sum_{m=1}^{M}\sum_{\nu=1}^{\lambda} X_{itmk_\nu} - \alpha_{tmk_\lambda} \cdot X_{itmk_\lambda} = \sum_{m=1}^{M}\sum_{k=1}^{K} X_{itmk}.$$

Due to constraints (20) and (37) of the $CLSP^{PM}_{ToolSync}$ also the constraints (15) and (16) of the $GLSP^{PM}_{ToolSync}$ hold and because of the construction of the the mapping also (12), (13) and (14) are fulfilled and we can state that $z_{t\nu mk_{\nu-1}k_\nu} = T_{tmk_{\nu-1}k_\nu} = 1$ and all other $T_{tmkl} = 0$.

Because of constraints (28) also constraints (3) are fulfilled. If we consider constraints (26) and (24) we may conclude that

$$\mu^s_{tmk_\nu} + \sum_{i=1}^{N} p_{imk_\nu} \cdot X_{itmk_\nu} + \sum_{l=1}^{K} (s_{mlk_\nu} T_{tmlk_\nu} + s_{mk_\nu l} T_{tmk_\nu l}) - s_{mk_\nu k_{\nu+1}} T_{tmk_\nu k_{\nu+1}} =$$

$$\mu^s_{tmk_\nu} + \sum_{i=1}^{N} p_{imk} \cdot X_{itmk_\nu} + s_{mk_{\nu-1}k_\nu} T_{tmk_{\nu-1}k_\nu} \leq \mu^s_{tmk_{\nu+1}} \quad \forall t, m, \nu < \lambda - 1$$

holds for all subsequent tools in the path Φ_{tm} except for the last two tools $k_{\lambda-1}$and k_λ. Hence, constraints (4) are fulfilled for all $r < \lambda$, respectively also for all $r > \lambda+1$ because all start times for those last micro-periods are 1 and there are no production and no tool switches. Because of (32)-(35) and (26) the following constraints can be derived for all t and m

$$u^s_{t\lambda m} = \mu^{max}_{tm} - s_{mk_{\lambda-1}k_\lambda} = \mu^f_{tmk_{\lambda-1}} - s_{mk_{\lambda-1}k_\lambda} \geq$$

$$\mu^s_{tmk_{\lambda-1}} + \sum_{i=1}^{N} p_{imk_{\lambda-1}} \cdot X_{itmk_{\lambda-1}} + \sum_{l=1}^{K} s_{mlk_{\lambda-1}} T_{tmlk_{\lambda-1}} =$$

$$u^s_{t(\lambda-1)m} + \sum_{i=1}^{N} p_{imk_{\lambda-1}} \cdot x_{it(\lambda-1)mk_{\lambda-1}} + s_{mk_{\lambda-2}k_{\lambda-1}} z_{t(\lambda-1)mk_{\lambda-2}k_{\lambda-1}}$$

$$u^s_{t(\lambda+1)m} = 1 \geq \mu^{max}_{tm} \geq \mu^f_{tmk_{\lambda-1}} - s_{mk_{\lambda-1}k_\lambda} \geq$$

$$\mu^s_{tmk_{\lambda-1}} + \sum_{i=1}^{N} p_{imk_{\lambda-1}} \cdot X_{itmk_{\lambda-1}} + \sum_{l=1}^{K} s_{mlk_{\lambda-1}} T_{tmlk_{\lambda-1}} =$$

$$u^s_{t(\lambda-1)m} + \sum_{i=1}^{N} p_{imk_{\lambda-1}} \cdot x_{it(\lambda-1)mk_{\lambda-1}} + s_{mk_{\lambda-2}k_{\lambda-1}} z_{t(\lambda-1)mk_{\lambda-2}k_{\lambda-1}}$$

Hence, constraints (4) hold also for $r = \lambda$ and $r = \lambda + 1$.

Constraints (5) are valid, because if we rewrite (24) for two subsequent tools k_ν and $k_{\nu+1}$ (except the last two tools) of the path Φ_{tm}, we will get for all t and m

$$\mu^f_{tmk_\nu} \leq \mu^s_{tmk_{\nu+1}} + s_{mk_\nu k_{\nu+1}} \cdot T_{tmk_\nu k_{\nu+1}} - T_{tmk_\nu k_{\nu+1}} + 1 + \alpha_{(t+1)mk_{\nu+1}} = \mu^s_{tmk_{\nu+1}} + s_{mk_\nu k_{\nu+1}}$$

So (5) hold for $r < \lambda-1$ and because of constraints (34) they are fulfilled for $r = \lambda$. For $r > \lambda$ the relations hold because $u^s_{t(r+1)m} = u^f_{trm} = 1$ and $\sum_{k=1}^{K}\sum_{l=1}^{K} s_{mkl} \cdot z_{t(r+1)mkl} = 0$. Constraints (6)

http://mc.manuscriptcentral.com/tprs Email: ijpr@lboro.ac.uk

For Peer Review Only

are valid, because either $\lambda < R$ and $u^s_{tRm} = u^f_{tRm} = 1$ and $x_{itRmk} = 0$ or $\lambda = R$ and constraints (32)–(34) and (27) ensure the necessary relations between u^s_{tRm} and u^f_{tRm}. Constraints (7) and (8) follow directly from (24)–(27), (32)–(35) and the definition of the mapping f.

In the remaining part of the proof we will show that the synchronization constraints are valid. Constraints (11) together with (9) and (10) enforce, that either the start time of micro-period r on machine m has to be greater or equal than the finishing time of micro-period r' on a different machine m' if the same tool is used in that period or the other way around, depending which micro-period starts first. Since we only compare micro-periods where the same tools k are used, the mapping f identifies corresponding start and finishing times μ^s_{tmk} and $\mu^f_{tm'k}$ of the $CLSP^{PM}_{ToolSync}$. If tool k is the last tool on machine m and the considered micro-period is $r = \lambda$, we have to take $\mu^{max}_{tm} - \sum_l s_{m'lk} \cdot T_{tm'lk}$ as start time and 1 as finishing time. Because of constraints (30), (31), and (36) it is guaranteed, that the start time on one machine is always after the finishing time on another machine, or the other way around.

Finally, we show that the values of the objective functions are identical. Inventory and backorder levels are the same and as previously noticed $z_{trmkl} = 1$ only if a corresponding $T_{tmkl} = 1$ (for $k \neq l$). Since the z_{trmkl} values are determined by the path Φ_{tm} and because of the subtour elimination no tool exchange can occur more than once, we can state that $\sum_{r=1}^{R} z_{trmkl} = T_{tmkl}$ (for $k \neq l$). So together with the fact that $s_{tmkk} = \sigma_{tmkk} = 0$ it follows directly that also the setup cost parts of the objective functions are identical. □

For Peer Review Only