A Constraint Programming Model for Food Processing Industry: a Case for an Ice Cream Processing Facility

Ezra Wari, Weihang Zhu
Department of Industrial Engineering
Lamar University
Beaumont, TX 77710, USA

Abstract

This paper presents a Constraint Programming (CP) scheduling model for an ice cream processing facility. CP is a mathematical optimization tool for solving scheduling problems either for optimality (for smaller-size problem) or feasibility (for larger-size problem). In practice, a single solution model can be used to schedule daily production or production horizon extending for months. The proposed model minimizes a make-span objective and consists various processing interval and sequence variables and a number of production constraints for case from a food processing industry. Its performance was compared against a Mix Integer Linear Programming (MILP) model from the literature for optimality, speed and competence using the full and partial capacity of the production facility of the case study. The results demonstrate both the effectiveness and speed of the CP models, especially for large scale models. As an alternative to MILP, CP models can provide a reasonable balance between optimality and computation speed.

Keywords: Constraint Programming; Scheduling; Ice Cream Processing; Food Processing Scheduling

Nomenclature

Sets

$i \in$ Product Mix;

$u \in$ Pasteurizer;

$v \in$ Vessel;

$w \in$ Packaging lines;

Subsets

$PasteurizersFor_i$	Pasteurizers for processing Product Mix *i*;
$VesselsFor_i$	Vessels for processing Product Mix *i*;
$FreezersFor_i$	Freezers for processing Product Mix *i*;
$PackingLinesFor_i$	Packaging lines for processing Product Mix *i*;

Parameters

$Fillingrate_u$	Pasteurization rate for Pasteurizer *u*;
$AgingTime_i$	Aging time for Product Mix *i;*
$FreezerTime_i$	Freezer time for Product Mix *i*;
$Emptyrate_i$	Packaging rate for Product Mix *i*;
$MinVesselSize_i$	Minimum vessel size for Product Mix *i*;
$Demand_i$	Demand for Product Mix *i*;
$ProcessChOTimes_{ii'}$	Process change-over time between Product Mixes *i* and *i'*;
$PackageChOTimes_{ii'}$	Packaging line change-over time between Product Mixes *i* and *i'*;
n	Maximum number of weeks (= 7);
N	Set of processing week (1 to n);
$Week$	Number of working hours per weeks (= 120 hours);
$idle$	Changeover time to idle state (= 2 hours);

Parameter functions

$MinNoBatch_i = {Demand_i}/{MinVesselSize_i}$ Minimum number of batches for Product Mix *i*;

$FillingTime_i = {MinVesselSize_i}/{Fillingrate_i}$ Pasteurization time for Product Mix *i*;

$EmptyTime_i = {MinVesselSize_i}/{EmptyTime_i}$ Packaging time for Product Mix *i*;

$stepFunction\ WeekendBreak = \ stepwise$ {0 −> 0; 1 −> 118; 0 −> 120; 1 −> 238; 0 −> 240; 1 −> 358; 0 −> 360; 1 −> 478; 0 −> 480; 1−> 598; 0 − > 600; 1− > 718; 0 − > 720; 1 -> 838; 0 -> 840; 1 -> 958; 0 -> 960; 1 -> 1078; 0 -> 1080; 1 -> 1198; 0 -> 1200; 1 -> 1318; 0 -> 1320; 1 -> 1438; 0 -> 1440; 0};

Interval Variables (Decision Variable)

$FillProcess_{ibu}$ Pasteurization interval for Product Mix *i*, batch *b* in unit *u* (optional);

$FillAssign_{ib}$ Pasteurization interval for assigning Product Mix *i*, and batch *b*;

$FreezeProcess_{ibx}$ Freezing interval for Product Mix *i*, and batch *b* in freezer *x* (optional);

$FreezeAssign_{ib}$ Freezing interval for assigning Product Mix *i*, and batch *b*;

$EmptyProcess_{ibw}$ Packaging interval for Product Mix *i*, batch *b* in line *w* (optional);

$EmptyAssign_{ib}$ Packaging interval for assigning Product Mix *i*, and batch *b*;

$VesselProcess_{ibv}$ Vessel interval for aging Product Mix *i*, batch *b* in vessel *v* (optional);

$VesselAssign_{ib}$ Vessel interval for assigning Product Mix *i*, and batch *b*;

$WaitProcessV_{ib}$ Waiting interval for assigning Product Mix *i*, and batch (optional);

$AgeProcessV_{ib}$ Aging interval for assigning Product Mix *i*, and batch *b*;

Sequence Variables (Decision Variable)

$PasturSeq_u$ Pasteurization sequence for intervals $FillProcess_{ibu}$ in pasteurizer *u*;

$VesselSeq_v$ Vessel sequence for intervals $VesselProcess_{ibv}$ in vessel *v*;

$PackSeq_w$ Packaging sequence for intervals $EmptyProcess_{ibw}$ in line *w*;

$FreezerSeq_x$ Freezing sequence for intervals $FreezeProcess_{ibx}$ in freezer *x*;

ILOG IBM CPLEX built-in modeling variables and functions (IBM, 2016)

Interval decision variable:- Variable time interval whose exact position is yet to be determined.

Sequence decision variable:- Variable to determine the order of interval decision variables.

stepFunction:- a function to create a step wise function (to create varying value 0-slope graphs)

noOverlap:- constraint function to prevent interval variables overlapping in a sequence.

forbidExtent:- constraint function to prevent interval variables overlapping a given time period.

alternative:- constraint function for creating an encapsulating interval over low-level intervals

startAtStart:- precedence constraint linking the start of a given interval with the start of another

endAtEnd:- precedence constraint linking the end of a given interval with the end of another

startAtEnd:- precedence constraint linking the start of a given interval with the end of another

endBeforeStart:- precedence constraint to arrange the start of a given interval after the end of another

startOf:- variable that gives the starting time of a given interval variable

endOf:- variable that gives the ending time of a given interval variable

lengthOf:- variable that gives the length time of a given interval variable (*endOf – startOf*)

1 Introduction

Companies put a significant effort to utilize production resources efficiently and effectively in their manufacturing activities. Scheduling methods improve resource utilization by providing platforms for planning, managing, and controlling resources. This paper presents a scheduling tool using Constraint Programming (CP) for a scheduling problem from the food processing industries (Wari, Zhu, & Xiang, 2017).

Scheduling problems in the food processing industries must address a combined discrete and continuous production system. Pasteurization, dehydration, and freezing are a few examples of continuous processes whereas packaging process is a good example of discrete processes in the industry. In addition to the optimization objectives (such as production cost/profit, make-span, earliness or lateness) and constraints (assignment of tasks to machines, sequencing and/or timing of tasks, other facility related constraints) they have in common with discrete production systems, continuous systems require decisions on selection and size of processing batches (Harjunkoski, et al., 2014). The perishable nature of food products would be the other factor to consider in scheduling problems in the industry. This factor constrains the manufacturing process to be completed within a limited time window frame. Hence optimization models not only need to define a constraint for this time window but also should complete the optimization run under this time limit. Time for decision making, implementation, and any corrective action would have to be included into this time window as well.

This paper proposes a CP scheduling model for an ice cream processing facility. CP is one of the mathematical programming techniques, along with Mixed Integer Linear Programming (MILP) and Dynamic Programming (DP), that can be applied to an optimization problems (Rossi, Van Beek, & Walsh, 2006; Apt, 2003). CP utilizes an approach where a solution space is reduced using constraints/restrictors before solving the problem with different mathematical programming methods. Problems can be formulated as Constraints Satisfaction Problem where CP aims at finding a feasible solution to the problem, or as Optimization Problem where it tries to find the optimal solution based on a given objective. Scheduling problems constitute one application area for the implementation of CP. Restriction due to manufacturing environments and resource availabilities can be formulated as constraints, and the production scheduling problem can be solved as either constraint satisfaction problem or optimization problem.

The CP model, in this paper, minimizes a make-span objective for a medium size food processing facility (ice cream processing). In our CP model, interval and sequencing variables are created to define the start and completion of processing times and order of processing in a specific machine. Constraints for these variables included no-overlap of processing time, selection and assignment of a task to machines, interlink and order of processing stages for all product types, product processing order in the packaging lines, and cease of production over the weekends. The first experimental run is used to compare the model against a MILP solution model published by Wari and Zhu (2016) for a reduced size manufacturing case study of the facility. The model attained make-span values comparable to those from the MILP model for small size problem instances. For more complex problems, it performed better by solving instances which the MILP model was unable to solve. In the second experimental run, the model solves two demand problem sets for the full-size production facility. It is worth noting that before this paper, no other researchers have been able to solve this problem at its full scale. Vessel capacities are used to define batch sizes and a schedule for month's production horizon is reported.

The rest of the paper is organized as follows. The literature review on scheduling approaches for food processing industries and CP method is presented immediately after this introduction section. The proposed model is described in the third section. Section four discusses the experimental run results of the model. Finally, concluding remarks are given in the last section.

2 Literature Review

Scheduling problems in the food processing industries can be formulated as either theoretical or practical/industrial case study problems. The most common theoretical problem formulation for the industry is given by a no-wait Flow Shop (FS) or Flexible Flow Shop (FFS) manufacturing setup. Industrial application cases covered a wide range of subsector. A summary of optimization approach for both type of problem formulation is given on Figure 1. This section presents a brief literature review of these approaches.

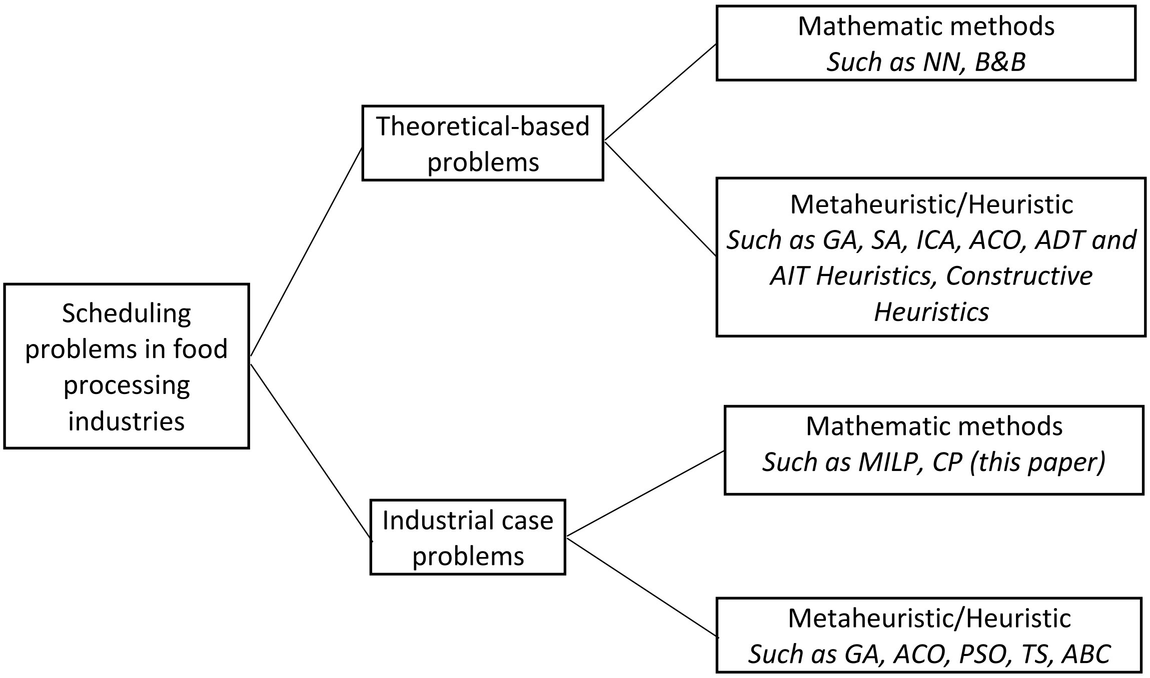


Figure 1. Summary of Scheduling Problem Optimization Approach

Various heuristic and metaheuristic approaches have been adopted to optimize theoretical-based problems. Wang & Liu proposed a GA model where jobs were coded as genes in the solution model (Wang & Liu, 2013). In this algorithms jobs were defined as genes and schedule as chromosomes for a two-stage production process. To utilize the best features of multiple metaheuristic approaches, a number of publications adopted hybrids of heuristics/metaheuristics methods. Jolai, Rabiee, & Asefi (2012) proposed a hybrid SA (Population-based SA) and ICA approach in which the earlier explored the solution space while the later exploited neighbourhood.

Other similar mixed approaches include Moradinasab, Shafaei, Rabiee, & Ramezani (2013) developed models using Imperialist Competitive Algorithm (ICA), ACO, and GA, Zhou and Gu (2009) integrated GA and Gaming Theory, and Samarghandi and ElMekkawy (2012) presented a hybrid TS - PSO approach. Ye, Li, & Miao (2017; 2016) proposed two heuristics methods (based on average idle and departure time, AIT and ADT) for a no-wait FS. Nagano, Miyata, & Araújo presented a constructive heuristic where the scheduling problem is divided into smaller sized problem before being optimized (Nagano, Miyata, & Araújo, 2015). Overall heuristic and metaheuristic dominate the approaches in the literature since most of no-wait FS/FFS scheduling problems were formulated as NP-Hard. Though few, mathematical methods include neural network (Shafaei, Rabiee, & Mirzaeyan (2011) and branch and bound (Wang, Liu, & Chu (2015))

Scheduling problems specific to the food processing industries adopted similar metaheuristics, mathematical, or hybrid methods. The application of GA could be for whole production setup as presented by Shaw et al. (2000), and Heinonen & Pettersson (2003), or specific processing stage (the filling line of dairy plant) as proposed by Gellert, Höhn, and Möhring (2011) or production function (cost of distribution) for the case of Karray, Benrejeb and Borne (2011). In all cases, production constraints such as processing stages and precedence, clean-up/sanitization, machine capacity, and processing time were formulated into the problem. Banerjee et al. (2008) presented an ABC approach to solve a multi-objective scheduling (optimal cost and risk levels) problem for a milk processing industry. Combined metaheuristics approach includes the two publications by Hecker et al. (2013; 2014) which adopted GA, ACO, PSO, and Random Search algorithms. Unlike the case for the theoretical based approach, a wide range of publications can be found for mathematical optimization for the industry. Linear Programming, particularly MILP, dominate the mathematical approaches for specific food processing scheduling solution models. Bongers and Bakker (2006), Kopanos, Puigjaner and Georgiadis (2011), Kopanos, Puigjaner and Georgiadis (2012), and Wari and Zhu (2016) proposed MILP models for a simplified ice cream processing scheduling problem with a make-span optimization objective. Bongers and Bakker (2006), Kopanos, Puigjaner and Georgiadis (2011), Kopanos, Puigjaner and Georgiadis (2012)integrated heuristics to supplement the limitation of the MILP approach such as long optimization run time, shorter production scheduling horizon and few number of product types in the scheduling problem. Wari and Zhu (2016) presented an MILP model for multi-week production horizon with proper weekend break-up points and clean-up sessions. Other MILP models publications include Doganis and Sarimveis (2007; 2008a; 2008b) and Kopanos, Puigjaner, and Georgiadis (2009) who developed models to optimize production cost (yogurt processing facility), Sadi-Nezhad & Darian (2010) who presented a model to optimize production capacity (juice processing facility), and Liu, Pinto, and Papageorgiou (2010) who proposed a model to maximize profit (an edible oil manufacturing facility).

Since no publication could be identified for both the theoretical and industrial application, the literature review for CP application has been expanded to include other manufacturing industries sectors. For machine scheduling application of CP, Novas and Henning (2012), Öztürk et al. (2012), and Zeballos, Quiroga, & Henning (2010) presented a make-span minimizing model for an automated wet-etch station (semiconductor manufacturing), flexible mixed model assembly line, and machine-tool allocation and routing of products respectively . The constraints in all models include start/completion processing times, the processing order (for products) and resource availability. In shift scheduling problem (for employees), the optimization objective includes meeting demand requirements of labor, and reducing costs such as overtime and under-utilized labor. Model constraints may be comprised of factors such as labor cost, workload balance, and allowable working hours. Publications for this instance included Han & Li (2014) – optimizing drivers and operators for a mass rapid transit train system, and Topaloglu & Ozkarahan (2011) – optimizing the schedule of medical residences. CP can also be used for fleet scheduling optimization either in a distribution/logistic problem or an internal material handling system to minimize cost. Unsal & Oguz (2013) and El Hachemi, Gendreau, & Rousseau (2011) presented CP models to demonstrate this application. Routing, inventory, and combined planning and scheduling problem are a few other application areas for CP (Goel, Slusky, van Hoeve, Furman, & Shao, 2015; Zhang & Wong, 2012).

3 Problem Description and Solution Models

The case study for this paper was first proposed by Bongers & Bakker (2006) to improve production scheduling for an ice cream processing facility. Bonger & Bakker reduced the size of the production in the facility to a three-stage processing with fewer number of machines and product mixes mainly due to the limitation of the optimization software. Later, relevant publications have expanded the number of product mixes and demands sizes, while keeping the problem at the reduced size. This paper takes on the challenge of considering the facility at its full size as described in Bongers & Bakker (2006).

3.1 Problem Description

As illustrated in Figure 2, production in the ice cream processing facility starts by mixing different ingredients of the ice cream based on receipts (Bongers & Bakker, 2006). Ice cream mixes are pasteurized first using two pasteurization units and then are transferred to the aging vessels. A group of vessels age specific product mixes for a receipt-based period. Next, product mixes are frozen using several freezer units. The production process completes with the packaging of product mixes into different sizes and shapes.

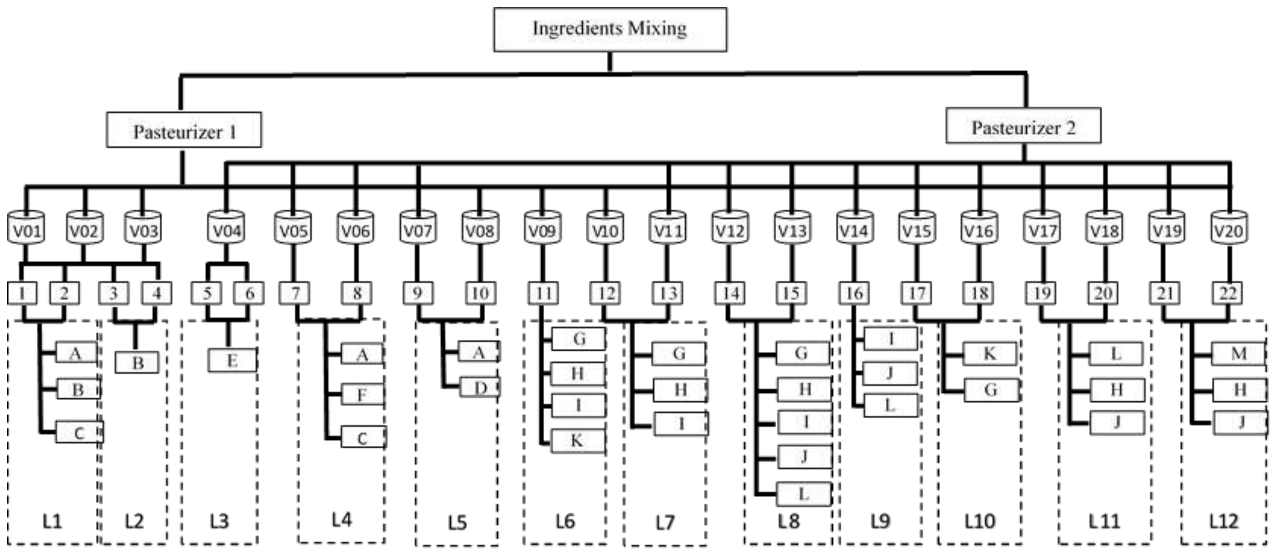


Figure 2. A medium size ice cream processing facility: V01 to V20 are aging vessels; Items 1 to 22 are freezers; Items L1 to L12 are packaging lines; Items A to M are product mix types (Source: Bongers and Bakker (2006)**)**

Product processing speeds have been defined in two methods in the problem. In the first method, machine's processing speed determines the processing rate. For example, the two pasteurization units process all product mixes at the rate of either 4500Kg/hr or 6000Kg/hr. The second method defines the processing speed based on the product mix type. Aging and packaging rate are good examples where product's intrinsic nature determines the processing rate. Most machines can process only a group of product mixes except for Pasteurizer 1 which has the capacity to process all products.

3.2 CP Solution Model

The proposed CP solution model comprises several sets, subsets, parameters, variables and constraints.

- Sets create collections for product mix types, and machines for pasteurization, aging, and packaging.
- Subsets memorize the assignments of processing machines for product mixes.

- Parameters define model constants such as processing speed (filling, aging, emptying), changeover times, working week time horizons, and scheduled weeks whereas parameter functions formulate mathematical relationship among parameters.
- Interval decision variables consist of two types of variables.
  - Process Intervals construct the processing time interval for each product mix in all assigned machines (*VesselProcess*, *FillProcess*, *FreezeProcess*, *EmptyProcess*).
  - Assignment Interval select a process interval among multiples of alternatives to create a schedule for each product mix (*VesselAssign, FillAssign, FreezeAssign, EmptyAssign*).
- Sequence decision variables arrange the processing order of product mixes in each machine in accordance with a predefined rule.

Objective

The model minimizes a make-span objective. It is computed as the maximum of all batches' completion time for the vessel variables (*VesselAssign*).

$$\text{Minimize} \quad \max_{p,b} \quad endOf(VesselAssign_{p,b}) \tag{1}$$

Sequence constraints

Sequence constraints in the model prohibit the overlap of intervals (defined in sequence decision variables) in all machines. The constraints for pasteurizing, aging, freezing, and packaging stages are given in equation 2, 3, 4, and 5 respectively. These constraints also insert the respective changeover times between the processing intervals of consecutive batches ($ProcessChOTimes_{ii'}$ for pasteurization and aging processes, and $PackageChOTimes_{ii'}$ for packaging process).

$$noOverlap(PasturSeq_u, ProcessChOTimes_{ii'}, 1)\ \forall u,\ u \in\ Pasteurizer, i, i' \in Product\ Mix \quad (2)$$

$$noOverlap(VesselSeq_v, ProcessChOTimes_{ii'}, 1)\ \forall v,\ v \in\ Vessels,\ i, i' \in\ ProductMix \quad (3)$$

$$noOverlap(FreezerSeq_x, PackageChOTimes_{ii'}, 1)\ \forall x, x \in Freezers,\ i, i' \in ProductMix \quad (4)$$

$$noOverlap(PackSeq_w, PackageChOTimes_{ii'}, 1)\ \forall w, w \in Packaging\ Lines, i, i' \in ProductMix \quad (5)$$

Interval constraints

Three groups of interval constraints are formulated for the optimization model. The first group constitutes constraints that interlink the start and end of intervals for successive stages of each product to create a processing chain (equations 6, 7, 8 and 9).

Pasteurization is immediately followed by the aging process (equation 6). Filling rate of the vessels is assumed to be equal to the pasteurization speed.

$$startAtEnd(AgeProcessV_{ib}, FillAssign_{ib})\ \ \forall i,\ \ i \in\ Product\ Mix,\ b\ \leq\ MinNoBatch_i \quad (6)$$

Right after the completion of the aging interval, the waiting interval commences (equation 7). Waiting interval is formulated as optional interval. Hence it exists only if there is a waiting period in the cases where the packaging processes could not be started immediately after the aging.

$$startAtEnd(WaitingProcessV_{ib}, AgeProcessV_{ib})\ \ \forall i,\ \ i \in\ Product\ Mix,\ b\ \leq\ MinNoBatch_i \quad (7)$$

Equations 8 and 9 connect the end of the waiting period to the start of the freezing process and freezing to the packaging process (emptying).

$$startAtEnd(FreezeAssign_{ib}, WaitProcessV_{ib})\ \ \forall i,\ \ i \in\ Product\ Mix,\ b\ \leq\ MinNoBatch_i \quad (8)$$

$$startAtEnd(EmptyAssign_{ib}, FreezeAssign_{ib})\ \ \forall i,\ \ i \in\ Product\ Mix,\ b\ \leq\ MinNoBatch_i \quad (9)$$

The processing in the aging vessels contains five distinct steps. These steps are 1) filling up the tanks, 2) aging; 3) waiting 4) freezing and 5) emptying the tanks. All these steps would have to be completed before any succeeding batch could start its processing in the same vessel. An aggregating interval (*VesselAssign*) is created to integrate these steps by aligning it with the start and end of these processing steps. Explicitly, the starts of *VesselAssign* and *FillAssign* are aligned (equation 10) whereas the ends of *VesselAssign* and *EmptyAssign* are aligned (equation 11).

$$startAtStart(FillAssign_{ib}\ , VesselAssign_{ib}\ )\ \forall i,\ \ i \in\ Product\ Mix,\ \ b\ \leq\ MinNoBatch_i \qquad (10)$$

$$endAtEnd(\ VesselAssign_{ib}\ ,\ EmptyAssign_{ib}\ )\ \forall i,\ \ i \in\ Product\ Mix,\ \ b\ \leq\ MinNoBatch_i \qquad (11)$$

In each processing stage, multiple machines are allocated to process each product batch. However, only one of these processing alternatives can be inserted in the final schedule. The third group of interval constraints selects and assigns processing intervals to the schedule (equations 12, 13, 14, and 15). *FillAssign, VesselAssign, FreezeAssign, and EmptyAssign* variables represent the scheduled intervals for pasteurizing, aging, freezing and packaging stages respectively.

$$alternative(FillAssign_{ib}, FillProcess_{ibu})\ \ \forall i,\ i \in\ Product\ Mix,\ \ b\ \leq\ MinNoBatch_i\ ,\ u\ \in\ Pasteurizer) \qquad (12)$$

$$alternative(VesselAssign_{ib}\ ,\ \ VesselProcess_{ibv}\ )\ \forall i,\ \ i \in\ Product\ Mix,\ \ b\ \leq\ MinNoBatch_i\ ,\ v\ \in\ Vessels) \qquad (13)$$

$$alternative(FreezeAssign_{ib}\ , FreezeProcess_{ibx})\ \forall i, i \in Product\ Mix,\ \ b\ \leq\ MinNoBatch_i,\ x \in\ Freezers) \qquad (14)$$

$$alternative(EmptyAssign_{ib}\ , EmptyProcess_{ibw})\ \forall i, i \in Product\ Mix,\ \ b\ \leq\ MinNoBatch_i,\ w \in\ Packaging\ Lines) \qquad (15)$$

Weekend break constraints

In the original problem, production takes place only during the weekdays in the facility. Therefore, the solution model cannot schedule any processing over the weekends and would have to cease production making proper break-up arrangements. The first part of this arrangement is to stop the production process earlier so that shut-down procedures would be completed before the end of the production week. This is expressed as a changeover to the idle state (given as 2 hours) which would effectively reduce the production week from 120 to 118 hours. Any interval for pasteurization, freezing and packaging processes cannot extend over the hours 118 to 120. A '*forbidExtent*' constraint is formulated to embody this scheduling restriction (equations 16, 17 and 18). A step function (*WeekendBreak*) defines these forbidden periods for a given production horizon.

$$forbidExtent(FillAssign_{ib}, WeekendBreak) \quad \forall i,\ i \in \ Product\ Mix,\ \ b \leq \ MinNoBatch_i \quad (16)$$

$$forbidExtent(FreezeAssign_{ib}, WeekendBreak) \quad \forall i, i \in \ Product\ Mix,\ \ b \leq \ MinNoBatch_i \quad (17)$$

$$forbidExtent(EmptyAssign_{ib}, WeekendBreak) \quad \forall i, i \in \ Product\ Mix,\ \ b \leq \ MinNoBatch_i \quad (18)$$

The weekend break arrangement for the aging process follows a different approach. Aging vessels act not only as processing machines but also as storage units. Hence any process started before the beginning of the weekend break can be finished and stored in the units over the weekends. The freezing/packaging process commences at the beginning of the subsequent production week. To represent this feature of the production process, a non-meta conditional constraint (case specific conditional constraint) is formulated (equation 19). The equation sets the completion time for the aging process to the beginning of the succeeding week for those processes that go beyond the end of the preceding weekdays.

$$startOf(AgeProcessV_{ib}) \leq \ n * (Week - Idle) \ \&\& \ endOf(AgeProcessV_{ib}) \ > \ n * Week => $$
$$endOf(AgeProcessV_{ib}) = n * Week \ \ \forall i, n,\ \ i \in \ Product\ Mix, b \ \leq \ MinNoBatch_i, n \in N \quad (19)$$

Processing precedence constraint

This constraint ensures that the processing interval for each product batch corresponds to the actual production sequence, i.e., in the order of pasteurization, aging, freezing, and packaging (equation 20). This is achieved by simply constraining the end of pasteurization interval to be greater than the start of the packaging process. The first group of interval constraints enforces the order when they create the processing chain for the rest of the stages.

$$endOf(FillAssign_{ib}) < startOf(EmptyAssign_{ib}) \;\; \forall i, i \in Product\ Mix,\; b \leq\; MinNoBatch_i \qquad (20)$$

Packaging line constraint

Processing in each packaging line would have to observe a precedence rule for product mix type. Higher priority product must be processed before lower priority products (in descending priority: M-L-K-J-I-H-G-F-E-D-C-B-A). A built-in function ('*endBeforeStart*') constraint is formulated to enforce this processing condition as given in equation 21.

$$endBeforeStart(EmptyProcess_{i'b'w'},\; EmptyProcess_{ibw}) \;\; \forall i, i' \in\; Product\ Mix,\; b, b' \leq MinNoBatch_{i\,i'},\; w, w' \in Packaging\ Lines\colon ord(ProductMix, i) > ord(ProductMix, i')\;\; \&\&\; w = w' \qquad (21)$$

Processing time window constraint

As perishable products, ice creams require a quick processing chain. For the case study in this paper, the maximum processing time window for each batch is 72 hours. Equation 22 constrains the total processing time (from start to finish) for all batches to be within this production time window.

$$endOf(VesselAssign_{ib}) - startOf(VesselAssign_{ib}) < \; 72 \;\; \forall i, i \in Product\ Mix,\; b \leq MinNoBatch_i \qquad (22)$$

4 Experimental Run Result and Discussion

Two sets of experimental runs were used to test the proposed model. The first set compared the model's performance against a published model from the literature. A modified data sets were used to evaluate the performance of these models. The second experimental run focused on solving the scheduling problem for the full-scale ice cream processing facility. For both sets of experiments, CPLEX v12.6 was used to formulate and execute the proposed model. All computations were performed on an Alienware Workstation with Intel® Core™ i7-5820K Processor CPU, 3.6TB Hard Drive, and 32GB RAM, on Windows 8.1 Professional operating systems.

4.1 Experimental Run 1

The publication by Wari and Zhu (2016) was selected as a base for comparison with the proposed CP solution model in this paper. However, the model is modified for fair comparison due to the limitation of CP models in CPLEX solver (the fact that it only optimizes integer models). Specifically, all parameters and parameter functions were converted to integer values. Furthermore, the published model did not consider the freezer stage. Freezer parameter, parameter function and constraints were excluded from the CP model (relax constraints 4, 7, 13, and 16 and modify constraint 8). To test the performance of each solution model (MILP and CP) over longer production horizon, batch sizes for problem instances were stretched from 180 or 200 to 640 batches for all three problem sets. In other words, the modified problem model mixed up part of the data from the published paper and new problem instance with larger batch sizes. The new problem instances were created by random generation of demand size for each product mix. The complete list of these values is given in Appendix 1. Finally, the run parameter limits for CP model were set to 1,000,000 for the fail limit (to improve optimization run speed) and 600 seconds maximum run time, similar to the limit set in Wari and Zhu (2016). Fail limit is a search control parameter that determines the maximum number of restarts a solution search must perform in order to guide it to optimal value.

Table 1 compared the run results of the MILP and CP models. The table showed the make-span results for the three test demand sets (Appendix 1a) and their respective computation time for each problem instances. The last column gave the make-span difference between the two models. Overall, the difference ranged from +14% to -15% which can be attributed to model variation. For small size problem instances, the MILP model attained optimal value faster than the CP model. As the size of these instances increased, the run time needed for MILP increased more quickly than the time needed for the CP model.

Table 1 Run result summary for experiment 1 – for modified problem instances from Wari and Zhu **(2016)**

Set 1 (8 product mixes)

Problem Instances	Total Batch Size	MILP model Make span (in hrs)	MILP model Run time (in s)	CP model Make span (in hrs)	CP model Run time (in s)	Makespan Result Variation (hrs)
1	80	129	15	136	64	-5%
2	100	161	29	172	82	-7%
3	120	191	44	199	112	-4%
4	140	228	58	235	144	-3%
5	160	254	92	269	181	-6%
6	180	295	126	308	222	-4%
7	200	318	141	346	276	-9%
8	220	336	167	366	317	-9%
9	240	381	187	400	389	-5%
10	260	429	221	441	405	-3%
11	280	456	236	478	440	-5%
12	320	512	339	532	600	-4%
13	360	569	588	589	600	-3%
14	400	664	528	683	600	-3%
15	440	-	600	740	600	-
16	480	-	600	796	600	-
17	520	-	600	875	600	-
18	560	-	600	926	600	-
19	600	-	600	994	600	-
20	640	-	600	1085	600	-

Set 2 (16 product mixes)

Problem Instances	Total Batch Size	MILP model Make span (in hrs)	MILP model Run time (in s)	CP model Make span (in hrs)	CP model Run time (in s)	Makespan Result Variation (hrs)
21	80	129	19	141	64	-9%
22	100	162	40	176	93	-8%
23	120	192	75	203	124	-6%
24	140	224	144	242	150	-8%
25	160	249	188	266	199	-7%
26	180	296	512	318	304	-7%
27	200	318	600	365	311	-15%
28	220	348	600	380	348	-9%
29	240	474	600	406	353	14%
30	260	-	600	445	494	-
31	280	-	600	473	587	-
32	320	-	600	549	600	-
33	360	-	600	613	600	-
34	400	-	600	712	600	-
35	440	-	600	765	600	-
36	480	-	600	870	600	-
37	520	-	600	910	600	-
38	560	-	600	1030	600	-
39	600	-	600	1035	600	-
40	640	-	600	1175	600	-

Set 3 (24 product mixes)

Problem Instances	Total Batch Size	MILP model Make span (in hrs)	MILP model Run time (in s)	CP model Make span (in hrs)	CP model Run time (in s)	Makespan Result Variation (hrs)
41	80	140	18	145	57	-4%
42	100	164	37	180	98	-10%
43	120	193	63	210	116	-9%
44	140	223	99	238	163	-7%
45	160	258	185	280	182	-8%
46	180	285	301	317	252	-11%
47	200	334	298	352	296	-5%
48	220	355	600	385	289	-8%
49	240	387	600	410	434	-6%
50	260	-	600	453	455	-
51	280	-	600	495	470	-
52	320	-	600	557	600	-
53	360	-	600	632	600	-
54	400	-	600	691	600	-
55	440	-	600	770	600	-
56	480	-	600	855	600	-
57	520	-	600	906	600	-
58	560	-	600	980	600	-
59	600	-	600	1055	600	-
60	640	-	600	1150	600	-

As seen in Table 1, the MILP model could not solve many relatively larger size problem instances within the 600 second run time limit. Increase in batch sizes within a set and product mix between different sets have made it significantly harder for the MILP model to solve problem instances. On the other hand, the CP model has solved all the problem instances and showed only a run time variation for any increase in either batch sizes or product mixes. This performance variation can be explained by the number of constraints and variables generated by each model for the same problem instance. Table 2, as an example, presented the number of constraints and variables for MILP model which were significantly larger when compared to those for CP model for problem set 1. The rate of increase for this model was also higher as problem size increased.

Table 2 Constraint and Variable numbers for problem set 1

Problem Instances	Total batch size	Numbe of Constraints		Numbe of Variables		Problem Instances	Total batch size	Numbe of Constraints		Numbe of Variables	
		MILP	CP	MILP	CP			MILP	CP	MILP	CP
1	80	16,055	3,622	11,261	849	11	280	164,967	24,826	51,560	2,943
2	100	23,863	4,916	14,621	1,057	12	320	212,755	31,294	61,549	3,361
3	120	33,266	6,205	18,149	1,265	13	360	267,568	38,455	72,009	3,781
4	140	44,107	7,815	21,845	1,473	14	400	326,432	45,877	83,624	4,195
5	160	56,730	10,003	25,569	1,685	15	440	393,298	54,652	95,344	4,615
6	180	71,388	11,655	29,504	1,895	16	480	463,361	64,964	108,104	5,031
7	200	85,313	13,665	33,941	2,097	17	520	543,853	72,785	121,301	5,449
8	220	101,619	15,950	38,400	2,303	18	560	629,177	83,727	134,881	5,869
9	240	122,790	19,379	42,209	2,525	19	600	716,661	96,863	149,609	6,285
10	260	143,412	22,067	46,744	2,735	20	640	815,331	108,547	164,429	6,705

4.2 Experimental Run 2

The second experimental run implemented the proposed CP model to the complete production setup of the facility considering all the stages and machines. The pasteurization process is synchronized with the filling step of the aging vessels. After completing aging (a discrete process), product mixes are continuously emptied from the vessel, frozen and packed as continuous process to complete the production. Since the processing speed for the freezing and packaging processes vary, the proposed model assumed a speed buffering mechanism (such as temporary storage) exists between these two stages. The complete CP model was implemented and executed with two settings of fail limit: 1,000,000 (1M) and 10,000,000 (10M).

Because no large problem data can be found in the literature, we developed two sets of demand data to test the proposed model. Two capacity sizes were assumed for the vessels in the facility: 4,000 and 8,000 Kg. Each product mix takes only one of these vessel sizes and a multiple of these sizes were used to generate demand volume for the mix. Demand data for the 4K vessels dominated the total demand for each problem instances in set 1. The total batch size for this set started at 40 and grew to 400 batches for the last problem instance. In demand set 2, 8K dominated total demand. The two demand sets and additional parameters for the model are given in Appendix 2. The specific set of machines in which each product mix complete its processing were also given in this appendix.

The make-span results for the two demand sets extended over 4 to 5 production weeks, as shown in Table 3. For the same total batch sizes, the two demand sets attained varying make-span results. The cause for this variation could be associated with the variation of processing time between product mixes of the two vessel types and the number of available processing machines (fewer number of vessels, and packaging lines for 8K). As expected, the computation time in all run scenarios increased as the batch sizes for problem instances increased. For any given problem instance, the higher fail limit run (10M) attained as good as or better results when it is compared with lower fail limit run. Higher fail limits allowed the model to search solution space more for better results. For the smaller total batch size problem instances, the improvement attained with this higher fail limit was so small that it could be ignored, especially when the time to attain them was taken into consideration. However, for the larger total batch size problems, these improvements were significant and could not be ignored. Therefore, large fail limit (long optimization runs) will have to be set for these problems to attain good make-span results.

Table 3 Run result summary for experiment 2

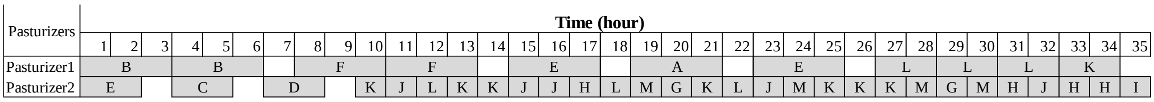
a. Set 1

Problem Instance	Total Batch	CP model (fail limit 1M) Make span (in hrs)	CP model (fail limit 1M) Run time (in s)	CP model (fail limit 10M) Make span (in hrs)	CP model (fail limit 10M) Run time (in s)	Difference Make span (in hrs)	Difference Run time (in s)	Number of constraints	Number of variables
1	40	37	24	36	252	1	-228	1398	857
2	80	70	50	70	495	0	-446	3963	1762
3	120	116	98	114	1034	2	-935	6479	2530
4	160	156	160	156	1602	0	-1442	11648	3556
5	200	228	196	221	2342	7	-2146	15556	4201
6	240	289	218	270	2028	19	-1810	19610	4926
7	280	340	324	340	3457	0	-3133	24444	5629
8	320	373	433	352	4474	21	-4041	30603	6346
9	360	418	553	415	5810	3	-5258	34408	6945
10	400	450	651	442	6790	8	-6139	51342	8209

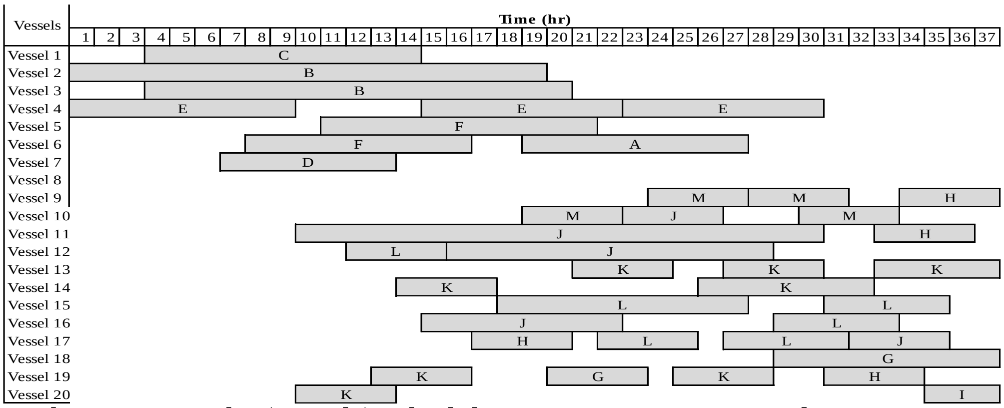
b. Set 2

Problem Instance	Total Batch	CP model (fail limit 1M) Make span (in hrs)	CP model (fail limit 1M) Run time (in s)	CP model (fail limit 10M) Make span (in hrs)	CP model (fail limit 10M) Run time (in s)	Difference Make span (in hrs)	Difference Run time (in s)	Number of constraints	Number of variables
11	40	59	16	57	157	2	-142	1247	772
12	80	115	37	111	397	4	-359	3103	1503
13	120	178	53	178	541	0	-488	4557	2177
14	160	213	94	212	903	1	-809	7971	3020
15	200	253	127	246	1322	7	-1194	11071	3822
16	240	325	194	318	1916	7	-1723	15558	4529
17	280	345	238	330	2184	15	-1946	21432	5458
18	320	415	316	408	2874	7	-2558	22483	5958
19	360	488	306	441	2511	47	-2205	36231	7242
20	400	508	416	498	4550	10	-4134	36995	7645

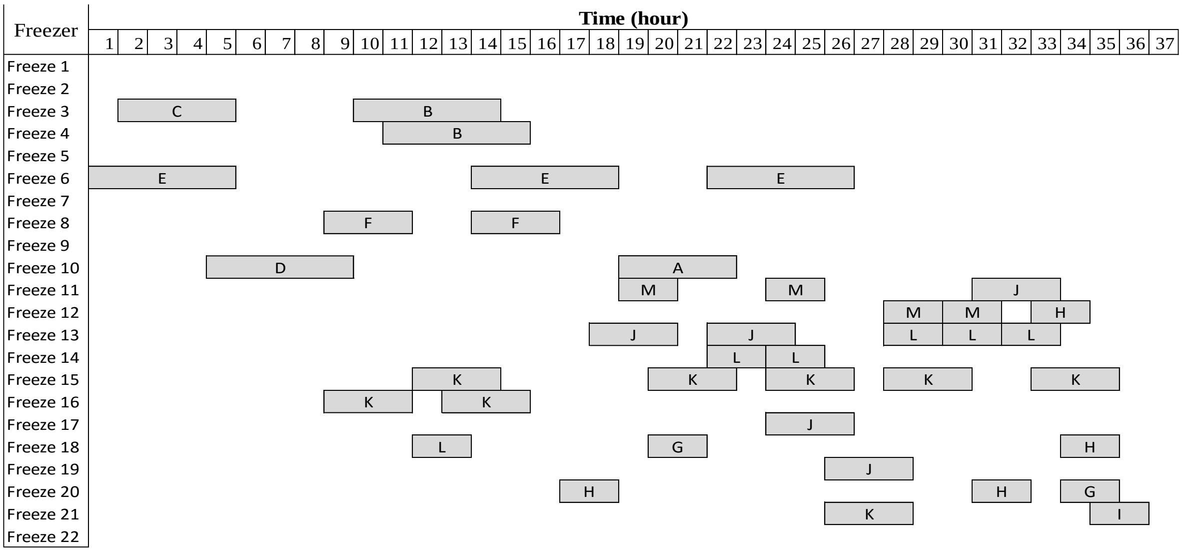
Figure 2 presented the schedule for problem instance 1 (make-span value of 37 hours). The schedule showed the bottleneck machine to be the pasteurizing units, unlike the cases in experiment 1 in which the packaging units were the bottleneck (Wari & Zhu, Multi-week MILP scheduling for an ice cream processing facility, 2016). For the remaining stages, under-utilization of most of the machines was observed. The increased number of available machines gave product mixes additional alternatives to complete their processing. Some machines (such as vessel 8, freezers 1, 2, 5, 7, and 9, and packaging line 2) were even not used at all while others were used for short operational run time over the 48 hours production horizon.



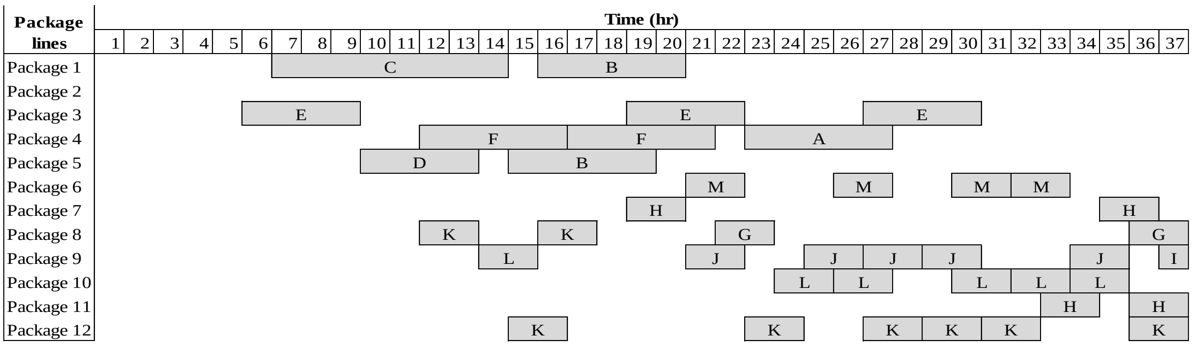
a. Pasteurizers schedules: Items A to M are product mix types



b. Aging tanks (vessels) schedules: Items A to M are product mix types



c. Freezer schedules: Items A to M are product mix types



d. Packaging lines schedules: Items A to M are product mix types

Figure 3. Production schedule for problem instance 1

5 Conclusion

The proposed CP model in this paper solved a scheduling model previously considered too complex and large due to optimization method limitations. Processing interval for each product mix and sequencing of these intervals in their respective machines were formulated as variables in the model. Constraints to prevent overlapping of processing interval, link processing stage, order processing stages, and restrict production only to the weekdays were integrated to emulate some limitations of the production environment. For a make-span objective, the model was compared with an MILP model using a modified larger demand data from the literature. The comparison showed that the model required fewer number of variables and constraints to optimize the similar scheduling problem and reported reasonably good results for short scheduling horizons. However, for long scheduling horizon it showed a better performance by solving all problem instance where the MILP failed to attain results. For the primary challenge of this paper, the model was applied to a scheduling problem for multi-stage ice cream processing with multiple machines in each stage for a full-size production facility. Two demand data sets were developed to test the performance of the model. For all problem instances in these sets, the CP model reported a respective schedule extending over multiple production weeks. Regarding run setups, short optimization horizon (1-2 weeks of production) can be solved with low fail limit values. However longer horizon required larger fail limits to attain better final results.

CP model formulation in CPLEX is limited to integer optimization. Hence scheduling problems must be formulated with integer value parameters and variables. However, CP models can optimize more complex scheduling problems and attain better results with a relative ease and shorter run time limit. It also gives practitioners the option choosing between "fast but lesser quality results" and "optimal results but long run time" approaches. The method could easily be integrated into an existing production system for a company. Future research directions could explore technical and financial aspects of this integration. Other optimization objectives, such as Tardiness and Earliness, are also promising directions to extend research in this area. It would also be interesting to see the application CP scheduling in other food processing industries (other processing industries) with even more manufacturing constraints. Scheduling problems could be combined with other production function such as inventory, transportation and distribution, and production planning. The performance of CP approach in these areas could also be a topic of interest.

The CP model in this paper obtained results within a short optimization run time for large problems. This could be used to provide timely inputs for decision making in production planning and scheduling activities of any company. Such inputs can give companies a competitive edge in the high dynamic market currently observed in the industry.

References

Apt, K. (2003). *Principles of constraint programming.* Cambridge University Press.

Banerjee, S., Dangayach, G. S., Mukherjee, S. K., & Mohanti, P. K. (2008). Modelling process and supply chain scheduling using hybrid meta-heuristics. *Metaheuristics for scheduling in industrial and manufacturing applications*, 277-300.

Bongers, P. M., & Bakker, B. H. (2006). Application of multi-stage scheduling. *16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering* (pp. 1917–1922). Boston: Elsevier.

Doganis, P., & Sarimveis, H. (2007). Optimal scheduling in a yogurt production line based on mixed integer linear programming. *Journal of Food Engineering*, 445-453.

Doganis, P., & Sarimveis, H. (2008a). Mixed Integer Linear Programming Scheduling in the Food Industry. In D.-W. Sun, & F. Erdogdu, *Optimization in food engineering* (p. 305). New York: Taylor & Francis Group, LLC.

Doganis, P., & Sarimveis, H. (2008b). Optimal production scheduling for the dairy industry . *Annals of Operations Research*, 315-331.

El Hachemi, N., Gendreau, M., & Rousseau, L. (2011). A hybrid constraint programming approach to the log-truck scheduling problem. *Annals of Operations Research*, 163-178.

Gellert, T., Höhn, W., & Möhring, R. H. (2011). Sequencing and scheduling for filling lines in dairy production. *Optimization Letters* , 491-504.

Goel, V., Slusky, M., van Hoeve, W., Furman, K., & Shao, Y. (2015). Constraint programming for LNG ship scheduling and inventory management. *European Journal of Operational Research*, 662-673.

Han, A., & Li, E. (2014). A constraint programming-based approach to the crew scheduling problem of the Taipei mass rapid transit system. *Annals of Operations Research*, 173-193.

Harjunkoski, I., Maravelias, C., Bongers, P., Castro, P., Engell, S., Grossmann, I., . . . Wassick, J. (2014). Scope for industrial applications of production scheduling models and solution methods. *Computers & Chemical Engineering*, 161-93.

Hecker, F. T., Hussein, W. B., Paquet-Durand, O., Hussein, M. A., & Becker, T. (2013). A case study on using evolutionary algorithms to optimize bakery production planning. *Expert Systems with Applications*, 6837-6847.

Hecker, F. T., Stanke, M., Becker, T., & Hitzmann, B. (2014). Application of a modified GA, ACO and a random search procedure to solve the production scheduling of a case study bakery. *Expert Systems with Applications*, 5882-5891.

Heinonen, J., & Pettersson, F. (2003). Scheduling a specific type of batch process with evolutionary computation. *Evolutionary Computation* (pp. 966-970). IEEE.

IBM. (2016). IBM ILOG CPLEX Optimization Studio CP Optimizer User’s Manual. *Version 12 Release 7*. Armonk, NY, USA: IBM Corp.

Jolai, F., Rabiee, M., & Asefi, H. (2012). A novel hybrid meta-heuristic algorithm for a no-wait flexible flow shop scheduling problem with sequence dependent setup times. *International Journal of Production Research*, 7447-66.

Karray, A., Benrejeb, M., & Borne, P. (2011). New parallel genetic algorithms for the single-machine scheduling problems in agro-food industry. *Communications, Computing and Control Applications (CCCA)* (pp. 1-7). Hammamet: IEEE.

Kopanos, G. M., Puigjaner, L., & Georgiadis, M. C. (2009). Optimal production scheduling and lot-sizing in dairy plants: the yogurt production line. *Industrial & Engineering Chemistry Research*, 701-718.

Kopanos, G. M., Puigjaner, L., & Georgiadis, M. C. (2011). Production scheduling in multiproduct multistage semicontinuous food processes. *Industrial & Engineering Chemistry Research*, 6316-6324.

Kopanos, G. M., Puigjaner, L., & Georgiadis, M. C. (2012). Efficient mathematical frameworks for detailed production scheduling in food processing industries. *Computers & Chemical Engineering*, 206-216.

Liu, S., Pinto, J. M., & Papageorgiou, L. G. (2010). Single-stage scheduling of multiproduct batch plants: An edible-oil deodorizer case study. *Industrial & Engineering Chemistry Research*, 8657-8669.

Moradinasab, N., Shafaei, R., Rabiee, M., & Ramezani, P. (2013). No-wait two stage hybrid flow shop scheduling with genetic and adaptive imperialist competitive algorithms. *Journal of Experimental & Theoretical Artificial Intelligence*, 207-25.

Nagano, M. S., Miyata, H. H., & Araújo, D. C. (2015). A constructive heuristic for total flowtime minimization in a no-wait flowshop with sequence-dependent setup times. *Journal of Manufacturing Systems, 36*, 224-230. Retrieved 11 22, 2017, from http://sciencedirect.com/science/article/pii/s0278612514000739

Novas, J., & Henning, G. (2012). A comprehensive constraint programming approach for the rolling horizon-based scheduling of automated wet-etch stations. *Computers & Chemical Engineering*, 189-205.

Öztürk, C., Tunalı, S., Hnich, B., & Örnek, M. (2012). A Constraint Programming Model for Balancing and Scheduling of Flexible Mixed Model Assembly Lines with Parallel Stations. *IFAC Proceedings*, (pp. 420-425.). Bucharest, Romania.

Rossi, F., Van Beek, P., & Walsh, T. (2006). *Handbook of constraint programming.* Elsevier.

Sadi-Nezhad, S., & Darian, S. B. (2010). Production scheduling for products on different machines with setup costs and times. *International Journal of Engineering and Technology*, 410-418.

Samarghandi, H., & ElMekkawy, T. Y. (2012). A meta-heuristic approach for solving the no-wait flow-shop problem. *International Journal of Production Research* , 7313-7326.

Shafaei, R., Rabiee, M., & Mirzaeyan, M. (2011). An adaptive neuro fuzzy inference system for makespan estimation in multiprocessor no-wait two stage flow shop. *International Journal of Computer Integrated Manufacturing*, 888-899.

Shaw, K. J., Lee, P. L., Nott, H. P., & Thompson, M. (2000). Genetic algorithms for multiobjective scheduling of combined batch/continuous process plants. *Evolutionary Computation* (pp. 293-300). La Jolla, CA: IEEE.

Topaloglu, S., & Ozkarahan, I. (2011). A constraint programming-based solution approach for medical resident scheduling problems. *Computers & Operations Research*, 246-255.

Unsal, O., & Oguz, C. (2013). Constraint programming approach to quay crane scheduling problem. *Transportation Research Part E*, 108-122.

Wang, S., & Liu, M. (2013). A genetic algorithm for two-stage no-wait hybrid flow shop scheduling problem. *Computers & Operations Research*, 1064-1075.

Wang, S., Liu, M., & Chu, C. (2015). A branch-and-bound algorithm for two-stage no-wait hybrid flow-shop scheduling. *International Journal of Production Research*, 1143-1167.

Wari, E., & Zhu, W. (2016). Multi-week MILP scheduling for an ice cream processing facility. *Computers & Chemical Engineering*, 141-156.

Wari, E., Zhu, W., & Xiang, Y. (2017). A Constraint Programming Model for Ice Cream Processing. *Industrial and Systems Engineering Research Conference.* Pittsburgh, PA, USA.

Ye, H., Li, W., & Abedini, A. (2017). An improved heuristic for no-wait flow shop to minimize makespan. *Journal of Manufacturing Systems, 44*, 273-279. Retrieved 11 22, 2017, from http://sciencedirect.com/science/article/pii/s0278612517300468

Ye, H., Li, W., & Miao, E. (2016). An effective heuristic for no-wait flow shop production to minimize makespan. *Journal of Manufacturing Systems*, 2-7.

Zeballos, L., Quiroga, O., & Henning, G. (2010). A constraint programming model for the scheduling of flexible manufacturing systems with machine and tool limitations. *Engineering Applications of Artificial Intelligence*, 229-248.

Zhang, L., & Wong, T. (2012). 2012. Solving integrated process planning and scheduling problem with constraint programming. *In Proceedings of the Asia Pacific Industrial Engineering & Management Systems Conference* (pp. 1525-1532). Phuket, Thailand: APIEMS.

Zhou, Y., & Gu, X. (2009). Research on no-wait flow shop scheduling problem with fuzzy due date based on evolution games. *Computer Science and Information Technology* (pp. 495-499). Beijing: IEEE.

Appendix

Appendix 1: Experiment 1 data

a. Modified problem instant data for three set types from Wari & Zhu (2016) (The numbers are in 1,000Kg)

Product Mix	Problem Instance (Set 1)																			
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
A	48	96	32	48	80	96	200	208	128	144	192	160	160	280	176	304	240	360	352	368
B	16	16	80	96	16	64	120	88	32	80	80	32	112	168	240	208	304	232	336	352
C	64	72	32	64	80	128	88	32	144	192	200	160	176	264	224	216	264	264	240	328
D	32	24	112	96	160	88	40	176	192	120	112	320	304	144	296	304	312	344	368	328
E	48	68	124	124	120	252	160	132	168	300	320	240	292	440	500	272	440	452	524	528
F	32	40	16	176	68	104	60	88	100	136	144	136	116	332	296	428	568	176	304	640
G	80	76	144	92	120	52	248	136	200	132	128	240	344	168	288	368	352	592	524	392
H	80	112	68	16	164	124	108	272	244	204	236	328	312	232	208	336	160	420	400	312
Batch Total	80	100	120	140	160	180	200	220	240	260	280	320	360	400	440	480	520	560	600	640

Product Mix	Problem Instance (Set 2)																			
	21	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**
A	56	64	48	16	32	64	128	80	96	136	64	80	96	176	216	176	104	200	176	176
B	16	8	48	80	88	16	16	56	96	48	104	96	136	72	120	112	152	72	152	248
C	8	16	32	56	40	48	32	64	64	88	80	104	120	96	80	64	120	88	144	176
D	24	16	8	56	16	64	32	48	16	40	104	120	104	88	72	144	112	192	160	208
I	12	68	40	100	8	152	136	88	80	84	56	252	188	260	156	92	136	260	284	372
J	20	24	88	20	36	20	48	96	176	124	72	40	116	132	104	176	188	112	204	152
K	44	20	12	136	28	60	40	60	24	68	116	196	196	92	112	156	156	180	160	388
L	32	8	12	20	84	60	16	108	24	160	100	80	128	80	132	80	172	104	116	136
E	16	16	64	16	24	16	32	48	128	32	64	32	64	96	72	216	216	168	160	128
F	8	24	24	8	112	48	48	56	48	40	104	56	64	96	168	72	144	112	192	120
G	24	32	16	56	8	128	64	80	32	104	48	184	136	176	96	136	152	256	192	224
H	40	72	32	24	16	64	144	80	64	120	48	88	104	168	200	224	216	288	240	200
M	20	4	16	20	176	28	8	60	32	36	120	48	80	48	204	176	164	172	100	148
N	8	8	16	68	12	20	16	80	32	80	124	88	148	44	36	112	136	116	192	232
O	60	104	104	20	68	140	208	64	208	108	156	160	84	348	336	380	328	460	412	224
P	28	40	56	20	60	16	80	68	112	76	68	36	88	112	168	176	192	148	224	168
Batch Total	80	100	120	140	160	180	200	220	240	260	280	320	360	400	440	480	520	560	600	640

Product Mix	Problem Instance (Set 3)																			
	41	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**	**58**	**59**	**60**
A	24	24	16	56	32	40	56	40	80	72	64	64	88	64	112	136	128	144	144	152
B	16	8	40	40	32	8	56	48	48	80	72	64	88	72	88	128	120	104	120	104
C	24	40	8	16	64	16	56	48	56	24	56	128	120	72	40	136	128	136	104	176
D	8	16	48	48	32	64	80	64	64	96	64	64	112	144	160	168	144	152	192	192
I	24	8	8	24	56	24	8	16	32	32	56	112	64	64	56	40	96	152	112	96
J	24	8	8	16	16	64	16	16	24	24	56	32	32	56	88	56	48	72	104	104
K	16	16	24	16	48	40	64	40	32	40	56	96	112	104	80	144	104	152	128	168
L	8	32	8	16	8	72	8	40	48	24	24	16	16	64	96	24	64	72	104	120
Q	8	8	16	16	16	8	16	24	24	32	32	32	32	56	40	40	56	72	56	48
R	16	8	56	40	8	16	40	64	48	96	88	16	48	104	112	96	112	88	120	72
S	16	40	16	16	16	16	16	56	56	32	48	32	32	56	48	48	88	80	64	88
T	8	16	24	16	40	80	64	40	32	40	40	80	104	72	120	136	96	88	160	200
E	20	16	36	40	60	88	136	52	56	76	76	120	196	132	164	292	152	180	224	300
F	4	36	28	40	8	4	8	64	76	68	36	16	16	60	72	20	112	80	80	56
G	12	12	20	40	44	8	64	32	52	60	44	88	108	64	68	140	116	136	112	128
H	20	36	32	20	4	84	8	68	56	52	72	8	12	96	136	36	92	56	140	132
M	40	12	24	40	8	4	4	36	52	64	104	16	12	104	68	48	84	128	76	28
N	12	12	12	40	148	136	80	24	52	52	36	296	228	100	188	172	212	276	336	376
O	20	20	4	20	36	8	40	24	40	24	44	72	76	68	32	100	80	120	68	104
P	8	8	4	40	12	20	28	12	48	44	20	24	40	96	64	64	64	108	76	68
U	24	36	52	20	12	68	80	88	56	72	100	24	92	100	140	184	120	124	152	196
V	4	12	4	40	8	16	8	16	52	44	12	16	16	68	60	20	64	92	68	44
W	52	36	40	40	80	16	80	76	76	80	144	160	160	104	96	212	196	184	176	212
X	8	52	88	20	36	44	24	140	72	108	104	72	60	144	152	56	196	100	188	156
Batch Total	80	100	120	140	160	180	200	220	240	260	280	320	360	400	440	480	520	560	600	640

b. Processing parameters data

	Aging Time	Min VesselSize	Filling rate	Empty rate	Filling Time	Empty Time
A	1	8000	4500	1750	2	5
B	3	8000	4500	1500	2	5
C	3	8000	4500	1000	2	8
D	0	8000	4500	1500	2	5
I	2	8000	4500	1750	2	5
J	3	8000	4500	1500	2	5
K	2	8000	4500	2000	2	4
L	1	8000	4500	2000	2	4
Q	4	8000	4500	2500	2	3
R	2	8000	4500	1250	2	6
S	3	8000	4500	1500	2	5
T	1	8000	4500	2250	2	4
E	2	4000	4500	1750	1	2
F	2	4000	4500	2000	1	2
G	2	4000	4500	2000	1	2
H	2	4000	4500	2000	1	2
M	3	4000	4500	2250	1	2
N	2	4000	4500	2000	1	2
O	3	4000	4500	1750	1	2
P	2	4000	4500	2250	1	2
U	1	4000	4500	1500	1	3
V	2	4000	4500	2000	1	2
W	2	4000	4500	1750	1	2
X	2	4000	4500	2750	1	1

c. Processing changeover times (in hours)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
A	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
B	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
C	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
D	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
E	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
F	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
G	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
H	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
I	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
J	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
K	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
L	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
M	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
N	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
O	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
P	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Q	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
R	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
S	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
T	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
U	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
V	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
W	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
X	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0

d. Packaging line changeover times (in hours)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
A	0	1	1	1	1	1	1	1	1	1	1	1												
B	0	0	1	1	1	1	1	1	1	1	1	1												
C	0	0	0	1	1	1	1	1	1	1	1	1												
D	0	0	0	0	1	1	1	1	1	1	1	1												
E	0	0	0	0	0	1	1	1	1	1	1	1												
F	0	0	0	0	0	0	1	1	1	1	1	1												
G	0	0	0	0	0	0	0	1	1	1	1	1												
H	0	0	0	0	0	0	0	0	1	1	1	1												
I	0	0	0	0	0	0	0	0	0	1	1	1												
J	0	0	0	0	0	0	0	0	0	0	1	1												
K	0	0	0	0	0	0	0	0	0	0	0	1												
L	0	0	0	0	0	0	0	0	0	0	0	0												
M													0	1	1	1	1	1	1	1	1	1	1	1
N													0	0	1	1	1	1	1	1	1	1	1	1
O													0	0	0	1	1	1	1	1	1	1	1	1
P													0	0	0	0	1	1	1	1	1	1	1	1
Q													0	0	0	0	0	1	1	1	1	1	1	1
R													0	0	0	0	0	0	1	1	1	1	1	1
S													0	0	0	0	0	0	0	1	1	1	1	1
T													0	0	0	0	0	0	0	0	1	1	1	1
U													0	0	0	0	0	0	0	0	0	1	1	1
V													0	0	0	0	0	0	0	0	0	0	1	1
W													0	0	0	0	0	0	0	0	0	0	0	1
X													0	0	0	0	0	0	0	0	0	0	0	0

Appendix 2: Experiment 2 data

a. Problem instance data for two set types (The numbers are in 1,000Kg)

	Problem Instant set 1									
i	1	2	3	4	5	6	7	8	9	10
A	8	16	88	48	80	240	136	144	224	208
B	16	48	24	152	32	64	72	280	160	296
C	8	32	24	64	128	112	96	136	96	168
D	8	16	112	96	80	128	120	96	280	448
E	24	32	24	64	120	80	360	128	408	96
F	16	16	48	56	200	176	176	336	112	224
G	8	48	32	80	132	40	96	92	76	100
H	16	24	88	120	80	64	136	56	128	184
I	4	16	12	16	36	180	92	196	96	264
J	20	64	20	96	72	24	72	132	80	96
K	32	32	36	40	40	80	160	64	64	120
L	24	24	68	16	80	112	36	60	200	32
M	16	32	64	32	40	60	48	120	156	84
∑ Min NoBatch	40	80	120	160	200	240	280	320	360	400
8k	10	20	40	60	80	100	120	140	160	180
4k	30	60	80	100	120	140	160	180	200	220

	Problem Instant set 2									
i	1	2	3	4	5	6	7	8	9	10
A	32	96	80	120	136	168	296	112	472	248
B	8	48	120	72	160	152	392	248	168	408
C	48	88	96	96	112	96	96	320	272	112
D	64	80	56	184	232	120	112	168	304	176
E	16	48	192	216	216	272	216	408	208	304
F	72	120	96	112	104	312	168	184	176	512
G	4	8	8	48	16	72	16	80	96	60
H	12	12	12	16	24	48	96	72	108	132
I	4	8	20	64	52	96	32	48	92	84
J	4	8	32	32	120	32	112	84	104	112
K	8	24	64	8	60	24	64	120	84	184
L	4	12	8	56	32	80	56	60	68	92
M	4	8	16	16	16	48	104	96	88	56
∑ Min NoBatch	40	80	120	160	200	240	280	320	360	400
8k	30	60	80	100	120	140	160	180	200	220
4k	10	20	40	60	80	100	120	140	160	180

b. Processing parameters data

i	MinVessel Size	Fillingrate 1	Fillingrate 2	AgingTime	Freeingrate	EmptyTime	Idle
A	8000	3	2	1	4	5	2
B	8000	3	2	3	5	5	2
C	8000	3	2	3	5	8	2
D	8000	3	2	0	5	4	2
E	8000	3	2	2	5	4	2
F	8000	3	2	3	3	5	2
G	4000	2	1	2	2	2	2
H	4000	2	1	1	2	2	2
I	4000	2	1	4	2	1	2
J	4000	2	1	2	3	2	2
K	4000	2	1	3	3	2	2
L	4000	2	1	1	2	2	2
M	4000	2	1	2	2	2	2

c. Machine assignment for each product mix

	Pasturizer	Vessel	Freezer	Packaging
A	1, 2	1, 2, 3, 5, 6, 7, 8	1, 2, 7, 8, 9, 10	1, 4, 5
B	1	1, 2, 3	1, 2, 3, 4	1, 2, 5
C	1, 2	1, 2, 3, 5, 6	1, 2, 7, 8	1, 4
D	1, 2	7, 8	9, 10	5
E	1, 2	4	5, 6	3
F	1, 2	5, 6	7, 8	4
G	1, 2	20, 19, 18, 17, 16, 14, 13	22, 21, 20, 19, 18, 16, 15	8, 10 11, 12
H	1, 2	20, 19, 18, 17, 16, 12, 11, 10, 9	22, 21, 20, 19, 18, 14, 13, 11, 12	6, 7, 10, 11, 12
I	1, 2	20, 19, 18, 17, 16, 15	22, 21, 20, 19, 18, 17	9, 10, 11, 12
J	1, 2	17, 16, 15, 12, 11, 10, 9	19, 18, 17, 14, 13, 12, 11	10, 9, 7, 6
K	1, 2	20, 19, 14, 13	22, 21, 16, 15	12, 8
L	1, 2	17, 16, 15, 12, 11	19, 18, 17, 14, 13	10, 9, 7
M	2	10, 9	12, 11	6

d. Processing changeover times (in hours)

	A	B	C	D	E	F	G	H	I	J	K	L	M
A	0	1	1	1	1	1	1	1	1	1	1	1	1
B	1	0	1	1	1	1	1	1	1	1	1	1	1
C	1	1	0	1	1	1	1	1	1	1	1	1	1
D	1	1	1	0	1	1	1	1	1	1	1	1	1
E	1	1	1	1	0	1	1	1	1	1	1	1	1
F	1	1	1	1	1	0	1	1	1	1	1	1	1
G	1	1	1	1	1	1	0	0	0	0	0	0	0
H	1	1	1	1	1	1	0	0	0	0	0	0	0
I	1	1	1	1	1	1	0	0	0	0	0	0	0
J	1	1	1	1	1	1	0	0	0	0	0	0	0
K	1	1	1	1	1	1	0	0	0	0	0	0	0
L	1	1	1	1	1	1	0	0	0	0	0	0	0
M	1	1	1	1	1	1	0	0	0	0	0	0	0

e. Packaging changeover times (in hours)

	A	B	C	D	E	F	G	H	I	J	K	L	M
A	0	2	2	2	2	2							
B	1	0	2	2	2	2							
C	1	1	0	2	2	2							
D	1	1	1	0	2	2							
E	1	1	1	1	0	2							
F	1	1	1	1	1	0							
G							0	2	2	2	2	2	2
H							1	0	2	2	2	2	2
I							1	1	0	2	2	2	2
J							1	1	1	0	2	2	2
K							1	1	1	1	0	2	2
L							1	1	1	1	1	0	2
M							1	1	1	1	1	1	0