

Volume 55 Numbers 1-2 1-15 January 2017

ISSN: 0020-7543

INTERNATIONAL JOURNAL OF

Production Research



Official Journal of the International Foundation for Production Research

Editor-in-Chief: Alexandre Dolgui



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

Journal:	<i>International Journal of Production Research</i>
Manuscript ID	TPRS-2017-IJPR-2425.R2
Manuscript Type:	Original Manuscript
Date Submitted by the Author:	19-Sep-2018
Complete List of Authors:	Wari, Ezra; Lamar University, Industrial Engineering Zhu, Weihang; University of Houston, Engineering Technology

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

Keywords:	SCHEDULING, CONSTRAINT PROGRAMMING
Keywords (user):	Ice Cream Processing, Food Processing Scheduling

SCHOLARONE™
Manuscripts

For Peer Review Only

Dear Editor-in-Chief, Editor and Reviewers:

The reviewer comments are very helpful for improving the papers. We have carefully followed the reviewer comments during the revision process. The major revision paragraphs are highlighted in red color in the paper. Please find below the response to the reviewer comments.

We greatly appreciate your consideration and hope this revision can meet your expectation.

Sincerely

Authors

.....
Comments from the Editor-in-Chief:

Please check also publications in Production Research to have an exhaustive state of the art presentation, see for example:

Parallel machine scheduling with tool loading: a constraint programming approach
Burak Gökgür, Brahim Hnich & Selin Özpeynirci
International Journal of Production Research
Published online: 05 Jan 2018

Transient inter-production scheduling based on Petri nets and constraint programming
Thomas Bourdeaud'huy , Olfa Belkahla , Pascal Yim , Ouajdi Korbaa & Khaled Ghedira
International Journal of Production Research, Volume 49, 2011 - Issue 22

see other recent publication on scheduling in IJPR.

Answer: Thank you for the comments. We have added six more publication (including the above mentioned two).

Reviewer(s)' Comments to Author:

Reviewer: 1

Comments to the Author

The readability of the article has been improved quite a lot since the original submission and it is now easier to understand the CP model, even if, I must admit, someone not familiar with CP Optimizer may find it hard to understand all the details.

In particular I think you should at least define the notion of "optional interval variable" which is the main decision variable you are using in your model and insisting on the notion of optionality (with a special value "absent" in the domain of the decision variable). Otherwise, it is impossible to understand the "alternative" constraints and how the resource allocation part of the problem is formulated in the CP model.

Answer: Thank you for the comments. We have added more explanation on optional interval (the last paragraph in page 12).

Sometimes, on the contrary, the description goes too much into the details, in particular the boolean argument ("1") of the noOverlap constraint (end of p 13) is really not necessary unless you explain in detail what it means (I would just remove this argument in the noOverlap constraints 2-5).

Answer: Thank you for the comments. We have made the correction.

On page 17, the part of the model related with constraints 19-21 is still not very clear. Is it not the case that constraints (19) and (20) are subsumed by constraint (20)? In fact it seems to me that you could use the concept of "intensity function" to formulate these constraints more directly in the model. If the constraints is that a certain amount of time equal to AgingTime_i must elapse between the breaks, then it is exactly what intensity functions stand for. The formulation of these constraints (19-20) result in very weak inferences in the engine so alternative formulations could help a lot here.

Answer: Thank you for the comments. When there is enough time to complete the aging and emptying processes during the working weekdays, the pre-define aging time for each product is assigned to aging interval. This is formulated by equation 19. However, when aging starts at the end of one week and needs to be processed and stored over the weekend, the main objective of the aging interval is to maintain process continuity over the two successive production weeks. Hence it will be difficult to determine the actual length of the aging interval because it is affected by the starting time of the aging, and pre-defined aging time of the product. For example, if a product has 4 hours of aging time and the process started at hour 117 then the length of the aging interval will be 3 hours to get to 120 which is starting point of the following week. The missing hour is completed over the weekend. On the other hand, if the aging time is 2 hours and the process started at 117 then the aging interval will still be 3 hours even though under the normal procedure it should have been 2. In real time, the product stays in the vessel for a total of 51 hours (3 hour over the weekdays + 48 hours over the weekend). This is formulated by equation 20. We could not formulate this feature as neither as intensity function nor as forbidden constraint due this peculiar nature of problem.

Maybe you could cite this very recent article that is a survey of CP Optimizer with a detailed description of the modeling concepts and also some examples of recent applications using this approach:

[1] P. Laborie, J. Rogerie, P. Shaw, P. Vilim . IBM ILOG CP Optimizer for Scheduling. Constraints, Volume 23, Issue 2, pp210-250.

Answer: Thank you for the comments. We have added the recommended article as a reference.

There are some strange statements about CP in the beginning of the article:

- In the abstract: "CP is a mathematical optimization tool for solving problems either for optimality (for small-size problems) or feasibility (for large-size problems)." The term "feasibility" seems to suggest that for large-size problems, CP is good only for producing feasible solutions. I think that it is also good at taking the objective function into account and finding "good quality feasible solutions". And as far as optimality is concerned, there are several types of scheduling problems (for instance classical jobshop or RCPSP) for which CP proves optimality for problems larger than MIP. I would replace the term "feasibility" here by something like "for finding good quality solutions".

Answer: Thank you for the comments. We have made the correction.

- "CP is the only technique which integrates a computer program in its solution model formulation." I do not understand. CP, just like MIP is mostly based on a declarative formulation of your model. I do not understand what is this computer program in the model formulation. Do you mean that usually in CP (but not in CP Optimizer), the user also need to write a search algorithm?

Answer: Thank you for the comments. We referred the computer program to the general CP algorithm and not the optimizer. However, we have replaced this description of the general CP with a better statement that describes the CP optimizer (page 2, paragraph 3, last line).

In several places in the article, you mention "IBM ILOG CP Optimization tool" (page 11), "CPLEX V12.8" (in the experiments), "IBM ILOG CP" (page 22). I think you should homogenize and speak of "CP Optimizer" (or "IBM CP Optimizer") because otherwise it is a bit confusing ("CPLEX" is the Math. Programming engine, "ILOG CP" is the "old" generation of CP Optimizer).

Answer: Thank you for the comments. We have made the correction.

Some other minor comments:

- In the model, I would replace the constraints 'startAtEnd(y,x)' by constraints 'endAtStart(x,y)'. These constraints are just the same but the later version is easier to read as 'x' is before 'y', both in time and in the arguments of the constraints.

Answer: Thank you for the comments. We have made the correction.

- p16, l58: "a non-meta conditional constraint" ? Maybe just "a "conditional constraint" is sufficient? See also my comment above on the inefficiency of this type of formulation in CP.

Answer: Thank you for the comments. We have made the correction.

Reviewer: 2

Comments to the Author

I still believe that the MIP formulation of the CP model should be given in the manuscript. Although one to one correspondence may not be shown, the set of constraints replaced by the CP model could be shown. Also, in the tables the makespan variation should be calculated for the CP model compared to the makespan values obtained by the MIP model. Therefore, minus values should show how better the CP model performs compared to the MIP model.

Answer: Thank you for the comments. We have added the MILP model into the manuscript (Appendix 4). Additional comparison of the model has been also inserted into the paper (page 19 paragraph 1). We have made the commented correction on the makespan table.

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

^aEzra Wari, ^bWeihsang Zhu*
^aDepartment of Industrial Engineering, Lamar University, Beaumont, TX 77710, USA
^bDepartment of Engineering Technology, University of Houston, Houston, TX 77204, USA
*Corresponding author email address: wzhu21@central.uh.edu

Abstract

This paper presents a Constraint Programming (CP) scheduling model for an ice cream processing facility. CP is a mathematical optimization tool for solving problems either for optimality (for small-size problems) or **good quality solutions** (for large-size problems). For practical scheduling problems, a single CP solution model can be used to optimize daily production or production horizon extending for months. The proposed model minimizes a makespan objective and consists of various processing interval and sequence variables and a number of production constraints for a case from a food processing industry. Its performance was compared against a Mixed Integer Linear Programming (MILP) model from the literature for optimality, speed, and competence using the partial capacity of the production facility of the case study. Furthermore, the model was tested using different product demand sizes for the full capacity of the facility. The results demonstrate both the effectiveness, flexibility, 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

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.

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, makespan, 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 (Harjunoski, 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 in this time window as well.

Several mathematical programming techniques have been proposed to solve optimization problems (including scheduling). Some of these techniques include Linear Programming, Mixed Integer Linear Programming (MILP), Dynamic Programming (DP), and CP (Rossi, Van Beek, & Walsh, 2006; Apt, 2003). CP was first developed as a tool for solving combinatorial problems in the artificial intelligence and computer science field of study. Problems in this method are formulated by defining the constraints on the decision variables and then solving these problems using computer programs (procedures) (Bockmayr & Hooker, 2005; I. & JF., 2013). CP

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

constraints can be framed as a linear equation or inequality, conjunctive or disjunctive function, or cumulative function (Sitek & Wikarek, 2015).

CP utilizes an approach where a solution space is reduced using constraints/restrictors before solving the problem with different mathematical programming methods (Rossi, Van Beek, & Walsh, 2006; Apt, 2003). Problems can be formulated as Constraints Satisfaction Problem where CP aims at finding a good quality 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 a constraint satisfaction problem or an optimization problem.

The proposed CP model, in this paper, minimizes a makespan 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 an MILP solution model published by Wari and Zhu (2016) for a reduced size of the case study (Wari, Zhu, & Xiang, 2017). The model attained makespan 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. Two batch sizes were used to break total products demands 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 by adopting either a general machine or production system layout or a specific industrial case study problem approach. The most common general system layout problem formulation is given by a no-wait Flow Shop (FS) and Flexible Flow Shop (FFS) manufacturing setup. Industrial application cases covered a wide range of subsectors. This section presents a brief literature review of these approaches.

Various heuristic and metaheuristic approaches have been utilized to optimize scheduling problems with general system layout. Wang & Liu proposed a Genetic Algorithm (GA) model where jobs were coded as genes in the solution model (Wang & Liu, 2013). In this algorithm, 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 Simulating Annealing (SA - Population-based SA) and Imperialist Competitive Algorithm (ICA) approach in which the earlier explored the solution space whereas the later exploited the neighborhoods. Other similar mixed approaches include Moradinasab, Shafaei, Rabiee, & Ramezani (2013) who developed models using ICA, Ant Colony Optimization (ACO), and GA; Zhou and Gu (2009) who integrated GA and Gaming Theory; and Samarghandi and ElMekkawy (2012) who presented a hybrid Tabu Search (TS) – Particle Swarm Optimization (PSO) approach. Ye, Li, & Miao (2017; 2016) proposed two heuristics methods (based on average idle and departure time) for a no-wait FS. Nagano, Miyata, & Araújo presented a constructive heuristic where the scheduling problem is

broken-down into smaller sized problems 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. However, few mathematical methods can be found which would include a branch and bound presented by Wang, Liu, & Chu (2015) for an FFS manufacturing setup.

Scheduling problems specific to the food processing facilities 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, & Möhring (2011) or production function (cost of distribution) for the case of Karray, Benrejeb & 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 problems. Banerjee et al. (2008) presented an Artificial Bee Colony (ABC) metaheuristic model for solving 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. 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 (the case study for this paper as well) with a makespan 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 runtime, shorter production scheduling horizon and few numbers 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

1
2
3 sessions. These four publications considered part of the production process of the facility and a
4
5 smaller number of products. The proposed CP model in this paper presents an optimization model
6
7 for a larger number of products processed using the full capacity of the facility. It also presents a
8
9 new mathematical approach for solving scheduling problems with the combined continuous and
10
11 discrete production system. Other MILP models publications include Doganis and Sarimveis
12
13 (2007; 2008a; 2008b) and Kopanos, Puigjaner, and Georgiadis (2009) who developed models to
14
15 optimize production cost (yogurt processing facility), Sadi-Nezhad & Darian (2010) who
16
17 presented a model to optimize production capacity (juice processing facility), and Liu, Pinto, and
18
19 Papageorgiou (2010) who proposed a model to maximize profit (an edible oil manufacturing
20
21 facility). New approaches, such as chance-constraint programming and a combined local search
22
23 and machine learning methods, have been used to optimize industrial scheduling problems. For
24
25 example, Wauters et al. presented a scheduling model where data for different food processing
26
27 features were utilized by search heuristics to attain better result (Wauters, Verbeeck, Verstraete,
28
29 Berghe, & De Causmaecker, 2012). Sel, Bilgen, & Bloemhof-Ruwaard demonstrated the
30
31 application of chance-programming for a scheduling problem in a dairy processing facility (Sel,
32
33 Bilgen, & Bloemhof-Ruwaard, 2017). The model applied chance-programming to quality
34
35 decaying properties of dairy products.
36
37
38
39
40
41

42 Since few publications were identified for specific applications in the food processing
43
44 industries, the literature review for CP application has been expanded to include other
45
46 manufacturing industries sectors. For machine scheduling application of CP, Novas and Henning
47
48 (2012), Öztürk et al. (2012), and Zeballos, Quiroga, & Henning (2010) presented a makespan
49
50 minimizing model for an automated wet-etch station (semiconductor manufacturing), flexible
51
52 mixed-model assembly line, and machine-tool allocation and routing of products respectively. The
53
54 constraints in all models include start/completion processing times, the processing order (for
55
56 products) and resource availability. To minimize total weighted completion time for a batch and
57
58
59
60

unary machine, Huang, Shi, & Shi (2018) proposed MILP and CP optimization models. Korbaa, Yim, & Gentina (2000) and Bourdeaud'Huy, Belkahla, Yim, Korbaa, & Ghedira, (2011) presented CP optimization approaches for a deterministic and cyclic transient state scheduling problem in a flexible manufacturing environment. For scheduling a set of tools, Gökgür, Hnich, & Özpeynirci (2018) developed a CP model to optimize a makespan objective. This model formulated constraints for variability and availability of production tools which were required by parallel machines to complete different jobs. 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, Weil, Heus, Francois, & Poujade, (1995) – nurse scheduling problem, and Topaloglu & Ozkarahan (2011) – optimizing the schedule of medical residencies. CP can also be used for fleet scheduling optimization either in a distribution/logistics 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. Meneghetti & Monti (2015) presented a CP model to optimize a refrigerated automated storage and retrieval systems for a food processing supply chain. This model optimized rack sizes, surface area and volume of storage cells with an objective of minimizing total storage costs and maximizing energy efficiency. Meneghetti, Dal Borgo, & Monti (2015) presented similar model focusing only on the rack sizes for the same optimization objectives. Routing, inventory, supply chain management, and combined planning and scheduling problem are a few other application areas for CP (Goel, Slusky, van Hoeve, Furman, & Shao, 2015; Zhang & Wong, 2012; Sitek & Wikarek, 2015). Table 1 summarizes the key relevant publication in the food processing industry.

Table 1 Summary of relevant scheduling problem publication in the food processing industry

Publication	Formulation	Objective	Mathematical			Metaheuristic/Heuristic								Other
			MILP	CP	B&B	GA	SA	ICA	ACO	TS	PSO	ABC	Heuristic	
Wang & Liu, 2013	2-stage no-wait FFS	Makespan				X								
Jolai, Rabiee, & Asefi (2012)	No-wait FFS	Makespan					X	X						
Moradinasab, Shafaei, Rabiee, & Ramezani (2013)	No-wait FFS	Total completion time				X		X	X					
Zhou and Gu (2009)	No-wait FFS	Customer satisfaction				X								X
Samarghandi and ElMekkawy (2012)	No-wait FS	Makespan								X	X			
Ye, Li, & Miao (2017; 2016)	No-wait FS	Makespan											X	
Nagano, Miyata, & Araújo, 2015	No-wait FS	Total flow time											X	
Wang, Liu, & Chu (2015)	2-stage no-wait FFS	Makespan			X									
Shaw et al. (2000)	Batch/Continuous (no-wait FS)	Various production costs				X								
Heinonen & Pettersson (2003)	Batch/Continuous (no-wait FS)	Various production costs				X								
Gellert, Höhn, & Möhring (2011)	Dairy processing facility	Total production cost				X								
Karray, Benrejeb, & Borne (2011)	Agro-food industries	Various production costs				X								
Banerjee et al. (2008)	Milk processing facility	Cost and risk levels										X		
Hecker et al. (2013; 2014)	No-wait FFS (Bakery facility)	Total production cost				X			X		X		X	
Bongers and Bakker (2006)	Ice cream processing facility	Makespan	X										X	
Kopanos, Puigjaner and Georgiadis (2011; 2012)	Ice cream processing facility	Makespan	X										X	
Wari and Zhu (2016)	Ice cream processing facility	Makespan	X											
Doganis and Sarimveis (2007; 2008a; 2008b)	Yogurt processing facility	Production Cost	X											
Kopanos, Puigjaner, and Georgiadis (2009)	Yogurt processing facility	Production Cost	X											
Sadi-Nezhad & Darian (2010)	Juice processing facility	Production Capacity												
Liu, Pinto, and Papageorgiou (2010)	Edible oil manufacturing facility	Profit	X											
Wauters et. al.	Packaging line	Makespan												X
Sel, Bilgen, & Bloemhof-Ruwaard (2017)	Dairy processing facility	Makespan												X

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 three-stage processing with a fewer number of machines, and product mixes mainly due to the limitation of the optimization software. Later, relevant

3.1 Problem Description

As illustrated in Figure 1, production in the ice cream processing facility starts by mixing different ingredients of the ice cream based on receipts (Bongers & Bakker, 2006). This stage is assumed to have no resource limitation and hence excluded from the scheduling problem. Ice cream mixes are pasteurized first using two continuous-pasteurization units. These units expose the mixtures to a high temperature for a short period as the mixtures flow through the equipment. The products flow to the aging vessels where the quality of the ice cream mix is improved, and the mix is cooled-down. Aging vessels are refrigerated tankers with agitators and store ice cream mixtures for a receipt-based period. Each product mixes can be aged in a specific number of vessels. The ice cream mixes are cooled further using product-specific freezers for a predefined period. The production process concludes with the packaging of product mixes into different sizes and shapes at a rate specific to each product mix.

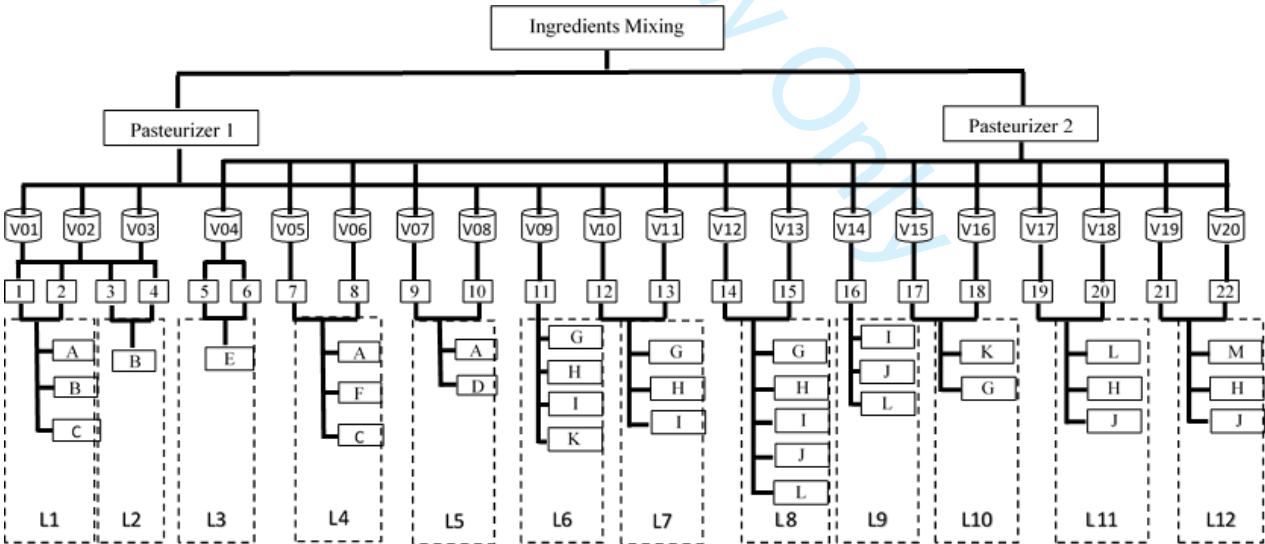


Figure 1. A medium size ice cream processing facility: V01 to V20 are aging vessels; Items 1 to 22 are freezers; Items L1 to L12 are packaging lines; Items A to M are product mix types (Source: Bongers and Bakker (2006))

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 a 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.

The processing flow of the ice cream product mixes is composed of multiple production stages (Figure 2). In the first stage, pasteurization of the product mixes takes place at a rate based on the specific equipment under consideration. Processing inside the vessels consists of multiple steps. First, the vessels would have to be filled up before the aging commences. As a continuous process, the pasteurization and filling processes can be considered as overlapping processes (interval variables for the two would have the same value). Then product mixes are aged and either transferred to the freezer or stored depending on the availability of a freezer and packaging line. When there is no idle freezer or packaging line available, the product is temporarily stored in the vessel (creates waiting interval). When a freezer and packaging line is available to take the mix, continuous freezing and packaging processes commence. Similar to the filling step, the emptying step of a vessel is a continuous process that overlaps with the freezing and packaging stages. Freezing and packaging are independent and consecutive processes.

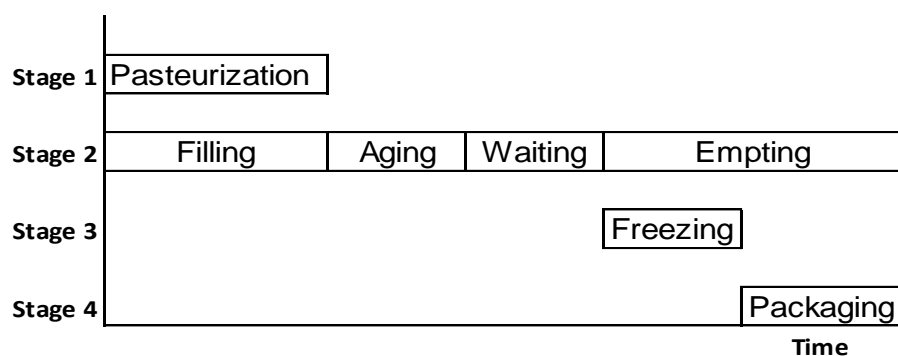


Figure 2. Processing time intervals for different stages of the ice cream production

3.2 CP model

Scheduling problems can be formulated using different approaches in CP. The most common approach is to define processes using bounding, binary, or both types of decision variables and then adopt constraint functions and propagating algorithms to solve the problem (Laborie & Rogerie, 2008). **IBM CP optimization** tool provides a different approach where the processes and respective sequences are formulated as decision variables. This tool incorporates different model formulation approaches which include global constraints construction, modeling layers creation (for processing activities and resources), and methods for generating optional processing activities (Laborie, Rogerie, Shaw, & Vilím, 2009). The proposed CP model creates processing time interval decision variables for all products in all stages. Sequencing decision variable defines the processing orders of products scheduled in each equipment. The complete model is described in this section.

Sets and Subsets

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

Parameters

Parameters define model constants such as processing speeds (filling, aging, emptying), changeover times, working week time horizons, and scheduled weeks whereas parameter functions formulate mathematical relationship among parameters.

Decision Variables

Two types of decision variables, intervals and sequences, are created to formulate different constraint relation which emulates the processing restrictions of the ice cream production. 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*, *PackProcess*). Assignment intervals select a process interval among multiples of

alternatives to create a schedule for each product mix (*VesselAssign*, *FillAssign*, *FreezeAssign*, *PackAssign*).

Figure 3 shows the temporal and logical constraints of the interval variables. The chain processing step is shown with the series of time intervals (linked by straight arrows). *VesselAssign* variables span over all the intervals and align with *FillAssign* at the start and *PackAssign* at the end (linked by double-headed arrows). Filling and emptying steps of the vessel are equivalent to the pasteurization process and the sum of freezing and packaging processes, respectively.

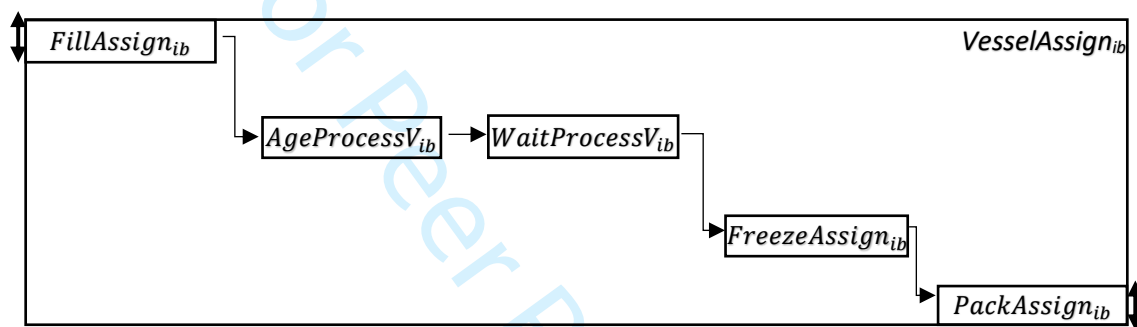


Figure 3. Temporal and spanning decision variables

The processes at all four stages of production can be performed by multiple equipment and the model has to pick only one equipment to create a viable schedule for each product. This feature is represented by formulating all assign intervals (*FillAssign*, *FreezeAssign*, *PackAssign* and *VesselAssign*) as optional intervals. An interval is defined as optional interval when it is created neither as ‘must be scheduled’ (present) nor ‘must not be scheduled’ (absent) interval. The presence/absence of the interval could only be determined when a solution is obtained (IBM, 2016; Laborie, Rogerie, Shaw, & Vilím, 2018). CP ignores absent intervals in any part of the optimization model. Alternative constraints enforce the selection process for optional intervals from among the different potential interval decision variables - process interval (see Figure 4).

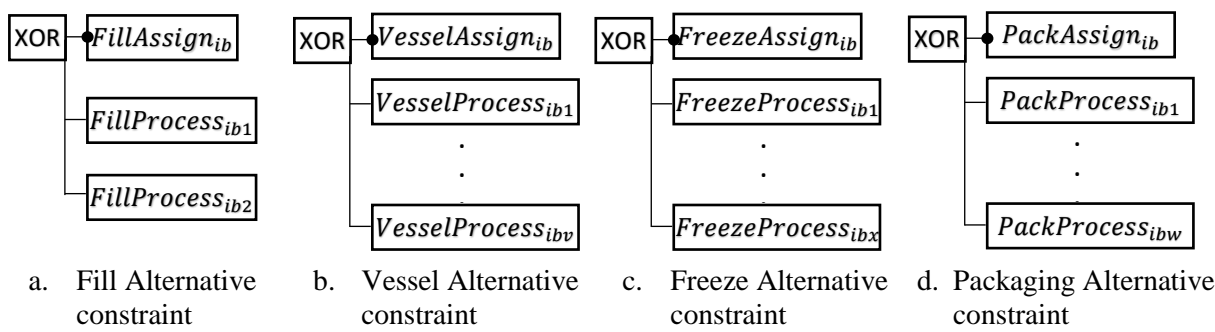


Figure 4. Alternative interval decision variables

Sequence decision variables (sequence interval variables) arrange the processing order of product mixes for each equipment (Laborie, Rogerie, Shaw, & Vilím, 2018). Figure 5 shows the process interval assigned to the sequence variable. It is important to note that *Vesselprocess* combines the filling (pasteurization), aging, waiting, and emptying (freezing + packaging) steps.

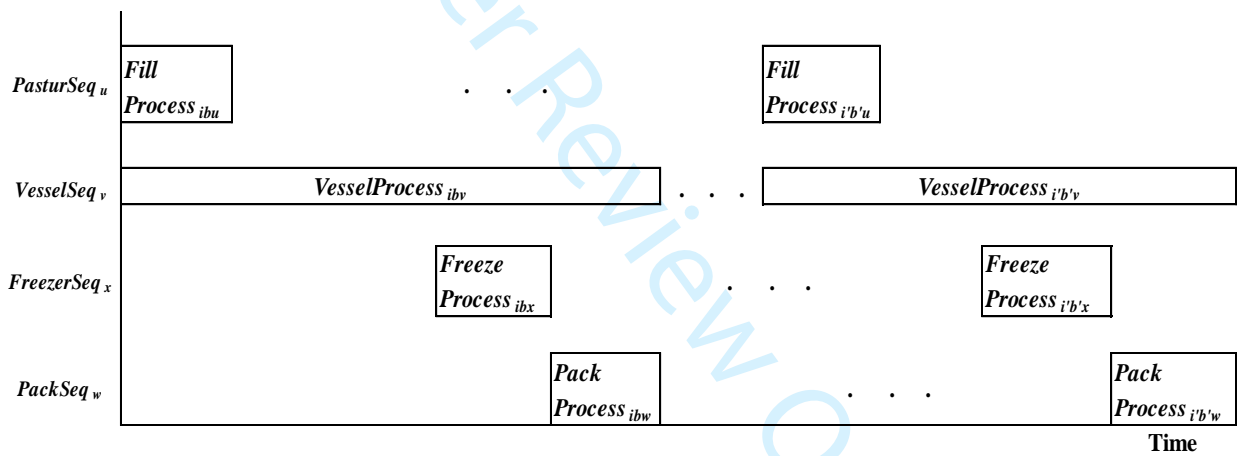


Figure 5. Sequence decision variables

Objective

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

Minimize $\max_{i,b} \text{endOf} (VesselAssign_{i,b}) \quad \forall i, b \text{ where } i \in \text{Product Mix}, b \in 1..MinNoBatch_i \quad (1)$

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 equations 2, 3, 4, and 5 respectively. These constraints also insert the respective changeover times between the processing intervals of consecutive batches. The changeover times required between product mixes are defined by two transition matrices given in Appendix 1c and 1d ($ProcessChOTimes_{i,i'}$ for pasteurization and aging processes, and $PackageChOTimes_{i,i'}$ for packaging process).

$$noOverlap(PasturSeq_u, ProcessChOTimes_{i,i'}) \quad (2)$$

$\forall u \text{ where } u \in \text{Pasteurizer}, i \& i' \in \text{Product Mix}$

$$noOverlap(VesselSeq_v, ProcessChOTimes_{i,i'}) \quad (3)$$

$\forall v \text{ where } v \in \text{Vessels}, i \& i' \in \text{Product Mix}$

$$noOverlap(FreezerSeq_x, PackageChOTimes_{i,i'}) \quad (4)$$

$\forall x \text{ where } x \in \text{Freezers}, i \& i' \in \text{Product Mix}$

$$noOverlap(PackSeq_w, PackageChOTimes_{i,i'}) \quad (5)$$

$\forall w \text{ where } w \in \text{Packaging Lines}, i \& i' \in \text{Product Mix}$

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 are assumed to be equal to the pasteurization speed.

$$endAtStart(FillAssign_{i,b}, AgeProcessV_{i,b}) \quad \forall i, b \text{ where } i \in \text{Product Mix}, b \in 1..MinNoBatch_i \quad (6)$$

Equations 7, 8, and 9 connect the aging, waiting, freezing, and packaging. Right after the completion of the aging interval, the waiting interval commences (equation 7). Following this, equation 8 links the freezing stage with the waiting interval and finally equation 9 connects the freezing stage with the packaging stage.

$$\begin{aligned} & \text{endAtStart}(\text{AgeProcess}_{i,b}, \text{WaitingProcess}_{i,b}) \\ & \forall i, b \text{ where } i \in \text{Product Mix}, b \in 1..MinNoBatch_i \end{aligned} \quad (7)$$

$$\begin{aligned} & \text{endAtStart}(\text{WaitingProcess}_{i,b}, \text{FreezeAssign}_{i,b}) \\ & \forall i, b \text{ where } i \in \text{Product Mix}, b \in 1..MinNoBatch_i \end{aligned} \quad (8)$$

$$\begin{aligned} & \text{endAtStart}(\text{FreezeAssign}_{i,b}, \text{PackAssign}_{i,b}) \\ & \forall i, b \text{ where } i \in \text{Product Mix}, b \in 1..MinNoBatch_i \end{aligned} \quad (9)$$

The processing in the aging vessels contains four distinct steps. These steps are 1) filling up the tanks; 2) aging; 3) waiting; and 4) emptying the tanks. All these steps have to be completed before any succeeding batch can 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 *PackAssign* are aligned (equation 11).

$$\begin{aligned} & \text{startAtStart}(\text{FillAssign}_{i,b}, \text{VesselAssign}_{i,b}) \\ & \forall i, b \text{ where } i \in \text{Product Mix}, b \in 1..MinNoBatch_i \end{aligned} \quad (10)$$

$$\begin{aligned} & \text{endAtEnd}(\text{PackAssign}_{i,b}, \text{VesselAssign}_{i,b}) \\ & \forall i, b \text{ where } i \in \text{Product Mix}, b \in 1..MinNoBatch_i \end{aligned} \quad (11)$$

As described in the previous sections, 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 *PackAssign* variables represent the scheduled intervals for pasteurizing, aging, freezing and packaging stages respectively.

$$\begin{aligned} & \text{alternative}(\text{FillAssign}_{i,b}, \text{FillProcess}_{i,b,u}) \\ & \forall i, b \text{ where } i \in \text{Product Mix}, b \in 1..MinNoBatch_i, u \in \text{Pasteurizer} \end{aligned} \quad (12)$$

$$\begin{aligned} & \text{alternative}(VesselAssign_{i,b}, VesselProcess_{i,b,v}) \\ & \forall i, b \text{ where } i \in \text{Product Mix}, b \in 1..MinNoBatch_i, v \in Vessels \end{aligned} \quad (13)$$

$$\begin{aligned} & \text{alternative}(FreezeAssign_{i,b}, FreezeProcess_{i,b,x}) \\ & \forall i, b \text{ where } i \in \text{Product Mix}, b \in 1..MinNoBatch_i, x \in Freezers \end{aligned} \quad (14)$$

$$\begin{aligned} & \text{alternative}(PackAssign_{i,b}, PackProcess_{i,b,w}) \\ & \forall i, b \text{ where } i \in \text{Product Mix}, b \in 1..MinNoBatch_i, w \in \text{Packaging Line} \end{aligned} \quad (15)$$

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.

$$\begin{aligned} & \text{forbidExtent}(FillAssign_{i,b}, WeekendBreak) \\ & \forall i, b \text{ where } i \in \text{Product Mix}, b \in 1..MinNoBatch_i \end{aligned} \quad (16)$$

$$\begin{aligned} & \text{forbidExtent}(FreezeAssign_{i,b}, WeekendBreak) \\ & \forall i, b \text{ where } i \in \text{Product Mix}, b \in 1..MinNoBatch_i \end{aligned} \quad (17)$$

$$\begin{aligned} & \text{forbidExtent}(PackAssign_{i,b}, WeekendBreak) \\ & \forall i, b \text{ where } i \in \text{Product Mix}, b \in 1..MinNoBatch_i \end{aligned} \quad (18)$$

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 conditional constraint (case specific conditional constraint) is formulated (equation 19 and 20). Equation 19 assigns the normal aging time to the product when it is processed during the working week-days. Whenever the aging process extends over consecutive weeks, equation 20 assigns the time from the end of the fill process to the beginning of the later week as the aging time to product. **As a result, the length of the aging interval may not be equal to the pre-defined aging time. The aging interval in this case serves as a production process link between two consecutive weeks and the start of the freezing process will always fall at the beginning of the later week.** Aging process cannot start during the weekends, and this constraint is enforced by equation 21.

$$\begin{aligned}
 &endOf(FillAssign_{i,b}) \geq (l-1)*Week \text{ AND} \\
 &\quad endOf(FillAssign_{i,b}) + AgingTime_i \geq (l-1)*Week \text{ AND} \\
 &\quad endOf(FillAssign_{i,b}) < (l*Week) - Idle \text{ AND} \\
 &\quad endOf(FillAssign_{i,b}) + AgingTime_i < (l*Week) - Idle \Rightarrow \\
 &\quad sizeOf(AgeProcessV_{i,b}) = AgingTime_i \\
 &\quad \forall i, b, l \text{ where } i \in Product \text{ Mix}, b \in 1..MinNoBatch_i, l \in N
 \end{aligned} \tag{19}$$

$$\begin{aligned}
 &endOf(FillAssign_{i,b}) \geq (l-1)*Week \text{ AND} \\
 &\quad endOf(FillAssign_{i,b}) + AgingTime_i \geq l*Week \text{ AND} \\
 &\quad endOf(FillAssign_{i,b}) < (l*Week) - Idle \text{ AND} \\
 &\quad endOf(FillAssign_{i,b}) + AgingTime_i < ((l+1)*Week) - Idle \Rightarrow \\
 &\quad sizeOf(AgeProcessV_{i,b}) = (l*Week) - endOf(FillAssign_{i,b}) \\
 &\quad \forall i, b, l \text{ where } i \in Product \text{ Mix}, b \in 1..MinNoBatch_i, l \in 1..n-1
 \end{aligned} \tag{20}$$

$$\begin{aligned}
 &forbidStart(AgeProcessV_{i,b}, WeekendBreak) \\
 &\quad \forall i, b \text{ where } i \in Product \text{ Mix}, b \in 1..MinNoBatch_i
 \end{aligned} \tag{21}$$

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 22.

$$\begin{aligned} & \text{endBeforeStart}(\text{PackProcess}V_{i',b',w}, \text{PackProcess}V_{i,b,w}) \\ & \forall i, i', b, b' \text{ where } i \& i' \in \text{Product Mix}, b \& b' \in 1..\text{MinNoBatch}_i, \\ & w \in \text{Packaging Lines}, \text{ord}(\text{ProductMix}, i) > \text{ord}(\text{ProductMix}, i') \end{aligned} \quad (22)$$

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 23 constrains the total processing time (from start to finish) for all batches to be within this production time window.

$$\text{lengthOf}(\text{VesselAssign}_{i,b}) < 72 \quad \forall i, b \text{ where } i \in \text{Product Mix}, b \in 1..\text{MinNoBatch}_i \quad (23)$$

4 Experiment Result and Discussion

Two sets of experimental runs were used to test the proposed model. The first set compared the model's performance with a published model from the literature. A modified data set was used to evaluate the performance of published and the proposed CP models. The second experimental run focused on solving the scheduling problem for the full-scale ice cream processing facility. This experiment aimed at finding the maximum number of product mixes that can be scheduled with the proposed model (disregarding limitations due to computing capacity and optimization software). For both sets of experiments, **IBM CP Optimizer** was used to formulate and execute the proposed model. All computations were performed on an Alienware Workstation with Intel® Core™ i7-5820K Processor CPU, 1TB HSD/3.6TB HDD, and 32GB RAM, on Windows 10 Enterprise operating systems.

4.1 Experimental Run 1

The publication by Wari and Zhu (2016) was selected as the base for comparison of the performance of the proposed CP model in this paper. This study presented an MILP model to schedule the partial production processing the ice cream facility. It formulated sets and subsets, parameters and parameter functions, decision variables, and objective and constraint functions to represent the production environment. Sets, subsets, parameters, and parameter functions have been created to define product-mix type, available processing equipment for each stage, processing parameter such as processing time, and setup time, scheduling parameter (including time horizon). Decision variables consist of the start and completion of processing times, waiting times, a binary variable for product processing order, and a binary variable for selection of aging vessel. A makespan objective was used. Four groups of constraints were formulated to solve the problem (Wari & Zhu, 2016). The first group created process intervals by defining the start and completion time for each product batch and then creating processing chains for all products and batches (Appendix 4B, equations 1 to 11). These constraints serve similar purpose to interval variables and temporal constraints. The second group of constraints (specific to the MILP model) focused on the allocation of batches to vessels (Appendix 4B, equations 12 to 20). These constraints assigned each batch to a vessel in a cyclic manner. The next group rearranged product batches assigned to an equipment based on predefined priority level (Appendix 4B, equations 21 to 28). On the proposed CP model, these are partly formulated by temporal and ‘noOverlap’ constraints. The last group consisted of miscellaneous constraints such as makespan, lower bound and domain sets for decision variables (Appendix 4B, equations 29 to 34).

Due to the limitation of each model, the test data used by Wari and Zhu was modified in this paper to compare the two models. Since the CP model can be formulated as integer model, all parameters and parameter function were converted to integer values for the MILP model too. The problem model for the MILP considered only one pasteurization unit and no freezer stage. To

1
2
3 accommodate this reduction in model size, the CP model considered only one pasteurization
4 equipment. Furthermore, the freezing stage was excluded from the original model by relaxing all
5 constraints relating to the stage (constraints 4, 8, 14, and 17) and modifying constraint 9. To test
6 the performance of each solution model (MILP and CP) over longer production horizon, batch
7 sizes for problem instances were stretched from 180 or 200 to 640 batches for all three problem
8 sets. In other words, the modified problem model mixed up part of the data from the published
9 paper and new problem instance with larger batch sizes. The new problem instances were created
10 by a random generation of demand size for each product mix. The complete list of these values is
11 given in Appendix 1. Finally, the run parameter limits for CP model were set to 600 seconds
12 maximum run time, similar to the limit set in Wari and Zhu (2016).
13
14
15
16
17
18
19
20
21
22
23
24
25

26 Table 2 compared the run results of the MILP and CP models. The table showed the
27 makespan results for the three test demand sets (Appendix 1a) and their respective computation
28 time for each problem instances. The last column gave the makespan difference between the two
29 models. Overall, the difference ranged from -3% to -15% for most problem instants. For a
30 maximum of 10 minutes of run time, the CP model could not achieve the optimal value. For small
31 size problem instances, the MILP model attained optimal value faster than the CP model. As the
32 size of these instances increased, the runtime needed for MILP to solve the problem increased
33 sharply than the time needed for the CP model. The MILP model failed to obtain any results for
34 large problems given the time limit while the CP model was able to obtain results in all the
35 instances.
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

Table 2 Run result summary for experiment 1 – for modified problem instances from Wari and Zhu (2016)

Set 1 (8 product mixes)							Set 2 (16 product mixes)						
Problem Instances	Total Batch Size	MILP model		CP model		Makespan Result Variation (hrs)	Problem Instances	Total Batch Size	MILP model		CP model		Makespan Result Variation (hrs)
		Make span (in hrs)	Run time (in s)	Make span (in hrs)	Run time (in s)				Make span (in hrs)	Run time (in s)	Make span (in hrs)	Run time (in s)	
1	80	129	15	138	600	9	21	80	129	19	144	600	15
2	100	161	29	173	600	12	22	100	162	40	182	600	20
3	120	191	44	203	600	12	23	120	192	75	214	600	22
4	140	228	58	245	600	17	24	140	224	144	247	600	23
5	160	254	92	271	600	17	25	160	249	188	280	600	31
6	180	295	126	308	600	13	26	180	296	512	317	600	21
7	200	318	141	345	600	27	27	200	318	600	352	600	34
8	220	336	167	378	600	42	28	220	348	600	388	600	40
9	240	381	187	404	600	23	29	240	474	600	420	600	-54
10	260	429	221	442	600	13	30	260	-	600	455	600	-
11	280	456	236	482	600	26	31	280	-	600	498	600	-
12	320	512	339	544	600	32	32	320	-	600	561	600	-
13	360	569	588	612	600	43	33	360	-	600	633	600	-
14	400	664	528	697	600	33	34	400	-	600	700	600	-
15	440	-	600	745	600	-	35	440	-	600	770	600	-
16	480	-	600	814	600	-	36	480	-	600	838	600	-
17	520	-	600	884	600	-	37	520	-	600	906	600	-
18	560	-	600	954	600	-	38	560	-	600	990	600	-
19	600	-	600	1022	600	-	39	600	-	600	1055	600	-
20	640	-	600	1090	600	-	40	640	-	600	1124	600	-

Set 3 (24 product mixes)						
Problem Instances	Total Batch Size	MILP model		CP model		Makespan Result Variation (hrs)
		Make span (in hrs)	Run time (in s)	Make span (in hrs)	Run time (in s)	
41	80	140	18	151	600	11
42	100	164	37	180	600	16
43	120	193	63	213	600	20
44	140	223	99	250	600	27
45	160	258	185	284	600	26
46	180	285	301	322	600	37
47	200	334	298	357	600	23
48	220	355	600	391	600	36
49	240	387	600	417	600	30
50	260	-	600	453	600	-
51	280	-	600	505	600	-
52	320	-	600	577	600	-
53	360	-	600	645	600	-
54	400	-	600	699	600	-
55	440	-	600	772	600	-
56	480	-	600	863	600	-
57	520	-	600	910	600	-
58	560	-	600	995	600	-
59	600	-	600	1060	600	-
60	640	-	600	1141	600	-

Overall, the proposed model provided more flexibility for schedulers by relaxing the constraints to assign product mixes before optimizing the production. Batches of different products can be arranged freely without the need for prioritization of products. However, it made the CP

model more complex. Evident from the results, the proposed model attained good schedules for longer production horizons where the MILP model failed to attain any.

4.2 *Experimental Run 2*

The second experimental run tested the performance of the proposed CP model to optimize production schedules for the full capacity of the facility considering all the stages and equipment. 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 through the freezers and packaging lines as a continuous process to complete the production. The complete list of processing speed or time for each product and equipment are given in Appendix 2. It is important to note that the processing speed for the freezing and packaging processes vary and the proposed model assumes a speed buffering mechanism (such as temporary storage) exists between these two stages.

Because no large problem data can be found in the literature, we developed two sets of demand data to test the proposed model. Two batch sizes were assumed for the vessels in the facility: 4,000 and 8,000 Kg. Each product mix takes only one of these batch sizes, and multiple batches were used to generate demand volume for the mix. The first demand set predominantly consists of the 4K-batch of product mixes. 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 resulting schedules for the two demand sets extended over 4 to 5 production weeks, as shown in Table 3. For the same total batch sizes, the model attained varying makespan results for two demand sets. It can also be observed that the model obtained **good quality feasible** solutions for more problem instances in set 1 than set 2. The cause for these variations could be associated

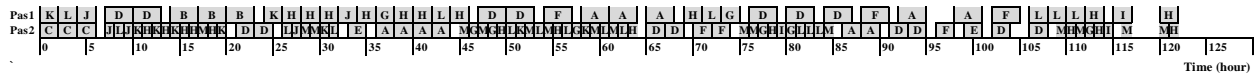
with the variation of processing time between product mixes of the two batch sizes and the number of available processing machines (fewer number of vessels, and packaging lines for 8K). Like the first experiment, the proposed model could not obtain optimal values within the allowed 10 minutes runtime for any of the demand sets. The two test sets also showed maximum problem sizes that the model can handle. The model solved problem instants with batch sizes up to 360 for set 1 (39000 constraints and 7000 variable) whereas it attained results for batch sizes 240 for set 2 (17000 constraints and 4000 variable). To provide some insights on the optimization size of each problem instant, the last column of each set in Table 3 gave the number of constraints and variables generated to solve the problem by the IBM CP Optimizer.

Table 3 Run result summary for experiment 2

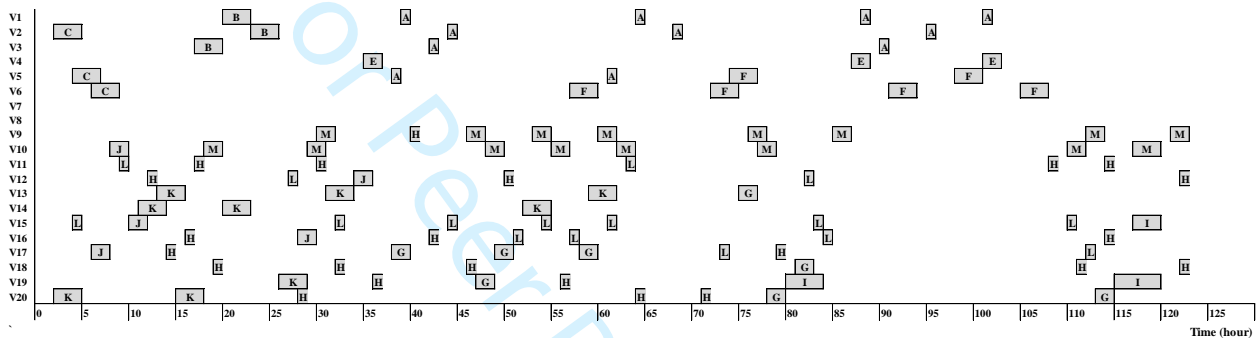
a. Set 1					b. Set 2				
Problem Instance	Total Batch	CP model		Number of constraints/variables	Problem Instance	Total Batch	CP model		Number of constraints/variables
		Make span (in hrs)	Run time (in s)				Make span (in hrs)	Run time (in s)	
1	40	40	600	1719/857	11	40	70	600	1569/772
2	80	72	600	4604/1762	12	80	139	600	3745/1503
3	120	127	600	7440/2530	13	120	318	600	5519/2177
4	160	178	600	12929/3556	14	160	357	600	9253/3020
5	200	235	600	15674/4131	15	200	357	600	12673/3822
6	240	286	600	23647/5156	16	240	460	600	17480/4529
7	280	331	600	27807/5682	17	280	No solution	600	22010/5254
8	320	396	600	31607/6199	18	320	No solution	600	25045/5958
9	360	678	600	37289/6945	19	360	No solution	600	39113/7242
10	400	No solution	600	54543/8209	20	400	No solution	600	40197/7645

Figure 6 presented the schedule for problem instance 3 (makespan value of 127 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, 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. The figure also showed the weekend breaks for the pasteurization, freezer and packaging line

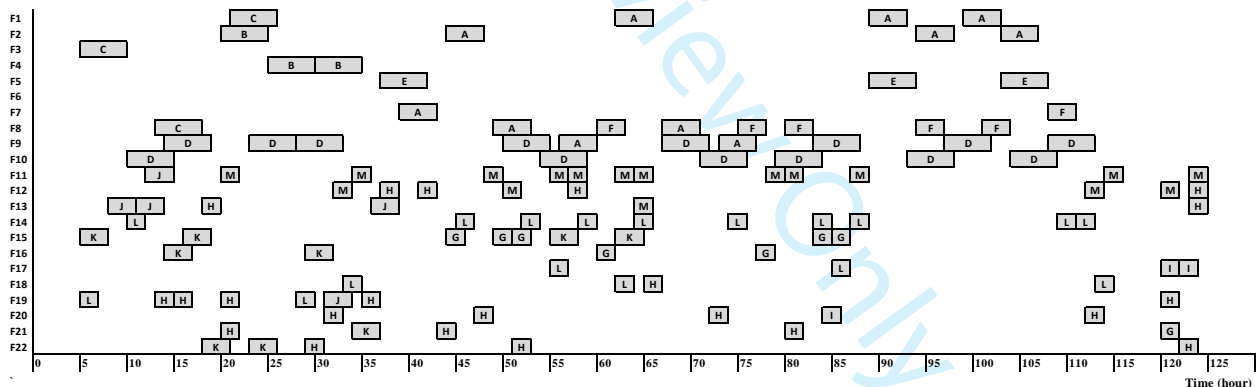
processing stage where production is halted and restarted on the following week. Some aging vessels (specifically vessel 10, 15 and 19) stored products over the weekends before the production commenced the freezing and packaging processes at the beginning of the second week.



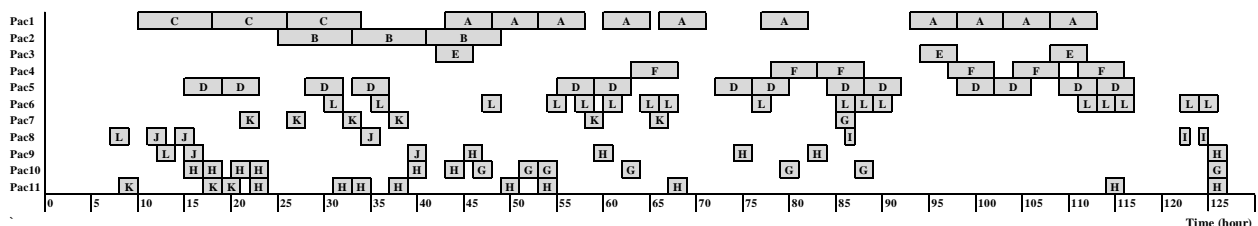
a. Pasteurizers schedules: Items A to M are product mix types



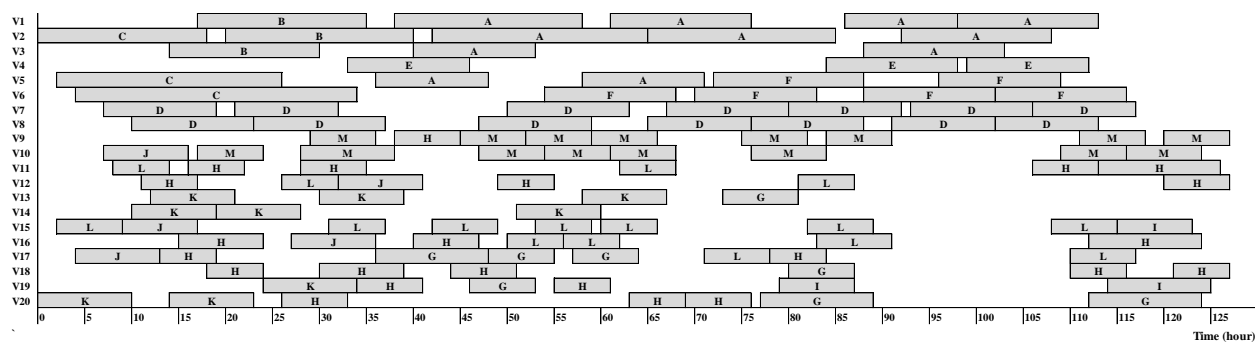
b. Aging tanks schedules: Items A to M are product mix types



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



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



e. Vessel schedules: Items A to M are product mix types

Figure 6. The production schedule for problem instance 3

5 Conclusion

The proposed CP model in this paper solved a large scheduling problem 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 makespan objective, the model was compared with an MILP model using a modified larger demand data from the literature. The comparison showed that the model attained a makespan value comparable (about 15% deviation) to the MILP model. However, for long scheduling horizon, it showed 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. The CP model reported a respective schedule extending over multiple production weeks. It also showed that the model could not solve large product mixes for both sets.

The proposed CP models can optimize more complex scheduling problems and attain better results within a relatively short run time limit. It also gives practitioners the option of choosing between “optimal result but small-size problem capacity” and “good results but large-size problem

capacity” 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 of 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 runtime 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 highly 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.
- Bockmayr, A., & Hooker, J. N. (2005). Constraint programming. In G. N. K. Aardal, *Handbooks in Operations Research and Management Science* (pp. 559-600).
- 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.
- Bourdeaud'Huy, T., Belkahla, O., Yim, P., Korbaa, O., & Ghedira, K. (2011). Transient inter-production scheduling based on Petri nets and constraint programming. *International Journal of Production Research*, 6591-6608.
- 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.
- Gökgür, B., Hnich, B., & Özpeynirci, S. (2018). Parallel machine scheduling with tool loading: a constraint programming approach. *International Journal of Production Research*, 1-17.
- 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.
- Harjunkski, 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.
- Huang, Z., Shi, Z., & Shi, L. (2018). Minimising total weighted completion time on batch and unary machines with incompatible job families. *International Journal of Production Research*, 1-15.
- I., L., & JF., P. (2013). Constraint Programming. In G. S.I., & F. M.C., *Encyclopedia of Operations Research and Management Science* (pp. 267-273). Boston, MA: Springer.
- 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.
- Korbaa, O., Yim, P., & Gentina, J. C. (2000). Solving transient scheduling problems with constraint programming. *European Journal of Control*, 511-520.
- Laborie, P., & Rogerie, J. (2008). Reasoning with Conditional Time-Intervals. *FLAIRS conference*, (pp. 555-560).
- Laborie, P., Rogerie, J., Shaw, P., & Vilím, P. (2009). Reasoning with Conditional Time-Intervals. Part II: An Algebraical Model for Resources. *FLAIRS conference* (pp. 201-206). Florida, USA: AAAI Publications.
- Laborie, P., Rogerie, J., Shaw, P., & Vilím, P. (2018). IBM ILOG CP optimizer for scheduling. *Constraints*, 210-250.
- 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.
- Meneghetti, A., & Monti, L. (2015). Greening the food supply chain: an optimisation model for sustainable design of refrigerated automated warehouses. *International Journal of Production Research*, 6567-6587.
- Meneghetti, A., Dal Borgo, E., & Monti, L. (2015). Rack shape and energy efficient operations in automated storage and retrieval systems. *International Journal of Production Research*, 7090-7103.
- 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., & Örneş, 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.
- Sel, Ç., Bilgen, B., & Bloemhof-Ruwaard, J. (2017). Planning and scheduling of the make-and-pack dairy production under lifetime uncertainty. *Applied Mathematical Modelling*, 129-144.
- 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.
- Sitek, P., & Wikarek, J. (2015). A hybrid framework for the modelling and optimisation of decision problems in sustainable supply chain management. *International Journal of Production Research*, 6611-6628.
- 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 System Engineering Research Conference*. Pittsburgh, PA.
- Wauters, T., Verbeeck, K., Verstraete, P., Berghe, G. V., & De Causmaecker, P. (2012). Real-world production scheduling for the food industry: An integrated approach. *Engineering Applications of Artificial Intelligence*, 222-228.
- Weil, G., Heus, K., Francois, P., & Poujade, M. (1995). Constraint programming for nurse scheduling. *Engineering in medicine and biology magazine* (pp. 417-422). IEEE .
- 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.

For Peer Review Only

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)

i	Problem Instant set 1									
	1	2	3	4	5	6	7	8	9	10
A	8	16	88	48	64	152	136	112	224	208
B	16	48	24	152	192	200	128	216	160	296
C	8	32	24	64	128	88	104	80	96	168
D	8	16	112	96	64	128	296	384	280	448
E	24	32	24	64	128	112	120	96	408	96
F	16	16	48	56	64	120	176	232	112	224
G	8	48	32	80	96	100	96	72	76	100
H	16	24	88	120	48	144	136	104	128	184
I	4	16	12	16	32	36	80	140	96	264
J	20	64	20	96	128	120	72	76	80	96
K	32	32	36	40	60	64	140	172	64	120
L	24	24	68	16	52	40	60	68	200	32
M	16	32	64	32	64	56	56	88	156	84
\sum 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

i	Problem Instant set 2									
	1	2	3	4	5	6	7	8	9	10
A	32	96	80	120	136	168	160	112	472	248
B	8	48	120	72	160	152	168	248	168	408
C	48	88	96	96	112	96	112	320	272	112
D	64	80	56	184	232	120	288	168	304	176
E	16	48	192	216	216	272	392	408	208	304
F	72	120	96	112	104	312	160	184	176	512
G	4	8	8	48	16	72	96	80	92	60
H	12	12	12	16	24	48	60	72	108	132
I	4	8	20	64	52	96	100	48	96	84
J	4	8	32	32	120	32	40	84	68	112
K	8	24	64	8	60	24	44	120	84	184
L	4	12	8	56	32	80	88	60	104	92
M	4	8	16	16	16	48	52	96	88	56
\sum 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 (in Vessel size in kg and the rest in hours)

i	MinVessel	Fillingrate		AgingTime	Freeingrate	PackTime	Idle
	Size	1	2				
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. Product specific processing rates

Product Code	Aging Period (hr)	Freezer Rate (Kg/hr)	Packaging Rate (Kg/hr)	Vessel Size (Kg)
A	1	2200	1750	8000
B	3	1600	1500	8000
C	3	1500	1000	8000
D	0	1500	1850	8000
E	2	1750	1900	8000
F	3	2400	1500	8000
G	2	2300	2000	4000
H	1	2100	2000	4000
I	4	2500	2750	4000
J	2	1250	1800	4000
K	3	1500	2300	4000
L	1	2300	2250	4000
M	2	1750	1800	4000

e. 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

f. 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

Appendix 3: Nomenclature

Sets

- $i \in$ Product Mix;
- $b \in$ Batch of a 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 ;
- $Packrate_i$ 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

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

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

Packaging time for Product Mix i ; $PackTime_i = MinVesselSize_i / Packrate_i$

Working hours: 1 when the facility is open & 0 when the facility is closed for weekend breaks;

$$stepFunction\ WeekendBreak = \begin{cases} 1 & \text{if } 0 < t \leq (l * Week) - 2 \\ 0 & \text{if } (l * Week) - 2 < t \leq l * Week \end{cases} \quad \forall t, t \in 1..n * Week, l \in N$$

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 ;

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

$PackAssign_{ib}$ 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;

$AgeProcessV_{ib}$ 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 ;

<i>VesselSeq_v</i>	Vessel sequence for intervals <i>VesselProcess_{ibv}</i> in vessel <i>v</i> ;
<i>PackSeq_w</i>	Packaging sequence for intervals <i>EmptyProcess_{ibw}</i> in line <i>w</i> ;
<i>FreezerSeq_x</i>	Freezing sequence for intervals <i>FreezeProcess_{ibx}</i> in freezer <i>x</i> ;

IBM CP Optimizer 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

endAtStart:- 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 of time for a given interval variable (*endOf* – *startOf*)

Appendix 4: MILP ice cream Model by (Wari & Zhu, Multi-week MILP scheduling for an ice cream processing facility, 2016)

A. Nomenclature

Sets	$\beta Itransfer_i^{min}$ Minimum number of batches transferred from preceding week for product i
$b, b', b'' \in B$ Batches of products	$\gamma_{i'j}$ Sequence-dependent change-over time between product i and i' in units j
$i, i' \in I$ Product types	γ_j^{min} Minimum sequence-dependent change-over time in packaging units j
$j \in J$ Processing units	$IdleGamma_i$ Sequence-dependent change over time from product to idle state
$s \in S$ Processing stage	ϵ_i^{life} Shelf life for product i in processing
$Itransfer \in I$ Product types transferred from preceding week	ζ_i Demand for product i
Subsets	θ_i Priority of product i in the packaging unit j
I_j Product i processed in unit j	ω Long production horizon (Big M value)
I_i^{Suc} Immediate successor of product i	λ_i Total number of aging vessels for product i
I_i^{Succ} Successors of product i	μ_j^{max} Maximum capacity of aging vessel j
I_i^{Pred} Predecessor of product i	ρ_{ij} Processing rate of product i in process line or packaging line j
I_i^{SP} Products that share the same packaging line with product i	τ_i^{ag} Minimum aging time for product i
J_i Units j that process product i	τ_i^{empty} Emptying time in aging vessel for product i
$J2_i$ Units j that process product i in the second stage	τ_i^{fill} Filling time in aging vessel for product i
$Mach_s$ Units j that process in stage s	ϕ_j^{min} Minimum wait time to begin packaging in line j
J_{ij}^{last} Last unit j that process product i	$Workweeklength$ Available production horizon
Parameters	$WeekNumber$ Processing week
α_j^{min} Minimum number of products assigned to packaging line j	
β_i^{min} Minimum number of batches for product i	
Parameter Function	
$\beta_i^{min} = \zeta_i / \mu_j^{max}$ where $i \in I, j \in J2_i$	

$$\beta Itransfer_i^{min} = \zeta_i / \mu_j^{max} \text{ where } i \in Itransfer, j \in J2_i$$

$$\tau_i^{empty} = \mu_j^{max} / \rho_{ij} \text{ where } i \in I, j \in Mach[s]:s = 3$$

$$\tau_i^{fill} = \mu_j^{max} / \rho_{ij} \text{ where } i \in I, j \in Mach[s]:s = 1$$

$$\omega = 1.2 (\phi_j^{min} + (\alpha_j^{min} - 1)\gamma_j + \min_j(\sum_{i \in I_j} \tau_i^{empty} \beta_i^{min})) \text{ where } j \in Mach[s]:s = 3$$

Decision Variables

C_{ibs} Completion time for stage s of batch b of product i	$L_{0\ ibs}$ Starting time for stage s of transferred batch b of product i
$C_{0\ ibs}$ Completion time for stage s of transferred batch b of product i	W_{ibs} Waiting (standing) time for stage s of batch b of product i
C_{max} Makespan	$W_{0\ ibs}$ Waiting (standing) time for stage s of transferred batch b of product i
$CWeek_{ibs}$ Processing completion week for stage s batch b of product i	$\bar{X}_{ibib'}$ (Binary) 1 if batch b of product i processed before batch b' of product i'
L_{ibs} Starting time for stage s of batch b of product i	Y_{ibsj} (Binary) 1 if batch b of product i in stage s is processed in unit j

B. Model

Objective: Min $Cmax$

Constraints:

$$L_{ibs} + \tau_i^{fill} = C_{ibs} \quad \forall i, b \leq \beta_i^{min}, s = 1 \tag{1}$$

$$L_{ibs} + \tau_i^{fill} + \tau_i^{ag} + W_{ibs} + \tau_i^{empty} = C_{ibs} \quad \forall i, b \leq \beta_i^{min}, s = 2 \tag{2}$$

$$W_{ibs} \leq \varepsilon_i^{life} - \tau_i^{ag} \quad \forall i, b \leq \beta_i^{min}, s = 2 \tag{3}$$

$$L_{ibs} + \tau_i^{empty} = C_{ibs} \quad \forall i, b \leq \beta_i^{min}, s = 3 \tag{4}$$

$$L_{ibs} = L_{ibs-1} \quad \forall i, b \leq \beta_i^{min}, s = 2 \tag{5}$$

$$C_{ibs} = C_{ibs-1} \quad \forall i, b \leq \beta_i^{min}, s = 3 \tag{6}$$

$$C_{ibs} \leq L_{ib+1s} \quad \forall i, b \leq \beta_i^{min}, s = 3 \tag{7}$$

$$C_{0\ ibs} = W_{0\ ibs} + \tau_i^{empty} \quad \forall i, b \leq \beta Itransfer_i^{min}, s = 2 : i \text{ in } Itransfer \tag{8}$$

$$C_{0ibs} = L_{0ibs} + \tau_i^{empty} \quad \forall i, b \leq \beta Itransfer_i^{min}, s = 3 : i \text{ in } Itransfer \quad (9)$$

$$C_{0ibs} = C_{0ib{s-1}} \quad \forall i, b \leq \beta Itransfer_i^{min}, s = 3 : i \text{ in } Itransfer \quad (10)$$

$$C_{0ibs} = L_{0ib+1s} \quad \forall i, b \leq \beta Itransfer_i^{min} - 1, s = 3 : i \text{ in } Itransfer \quad (11)$$

$$\sum Y_{ibsj} = 1 \quad \forall i, i \in I, b \leq \beta_i^{min}, s = 2, \quad (12)$$

$$Y_{ibsj} = 1 \quad \forall i, i \in Itransfer, b = 1, s = 2, j = first(J2_i) \quad (13)$$

$$Y_{ibsj} = Y_{ib+1sj+1} \quad \forall i, i \in Itransfer, b \leq \beta Itransfer_i^{min} - 2, s = 2, j \in J2_i : \beta Itransfer_i^{min} \leq \beta_i^{min} \quad (14)$$

$$Y_{ibsj} = Y_{ib+1sj+1} \quad \forall i, i \in Itransfer, b \leq \beta_i^{min} - 2, s = 2, j \in J2_i : \beta Itransfer_i^{min} > \beta_i^{min} \quad (15)$$

$$Y_{ibsj} = Y_{i'b'sj+1} \quad \forall i, i \in Itransfer, b = \min(\beta Itransfer_i^{min} - 1, \beta_i^{min}), i' \in IPred_i, b' = \beta Itransfer_{i'}^{min}, s = 2, j \in J2_i : \text{card}(IPred_i) > 0 \quad (16)$$

$$Y_{ibsj} = Y_{i'b'sj+1} \quad \forall i, i \in Itransfer, b = \beta Itransfer_i^{min} - 1, i' \in IPred_i, b' = \beta Itransfer_{i'}^{min}, s = 2, j \in J2_i : \beta Itransfer_i^{min} \leq \beta_i^{min}, \text{card}(IPred_i) = 0 \quad (17)$$

$$Y_{ibsj} = Y_{i'b'sj+1} \quad \forall i, i \in Itransfer, b = \beta Itransfer_i^{min} - 1, i' \in ISuc_i, b' = \beta Itransfer_{i'}^{min}, s = 2, j \in J2_i : \beta Itransfer_i^{min} - 1 \leq \beta_i^{min}, \text{card}(IPred_i) = 0 \quad (18)$$

$$Y_{ibsj} = Y_{i'b'sj+1} \quad \forall i, i \in I, b = \beta_i^{min}, i' \in ISuc_i, b' = \beta Itransfer_{i'}^{min}, s = 2, j \in J2_i \quad (19)$$

$$Y_{ibsj} = Y_{ib+1sj+1} \quad \forall i, i \in I, \beta Itransfer_i^{min} \leq b \leq \beta_i^{min}, s = 2, j \in J2_i \quad (20)$$

$$L_{ibrs} \geq C_{ibs} + \gamma_{ii'} - \omega(1 - \bar{X}_{ibi'b'}) \quad \forall i, b \leq \beta_i^{min}, i' \in I, b' \leq \beta_{i'}^{min}, s = 1, j \in J_i \cap J_{i'} \cap Mach[s] : i < i' \quad (21)$$

$$L_{ibrs} \geq C_{ibrs} + \gamma_{ii'} - \omega \bar{X}_{ibi'b'} \quad \forall i, b \leq \beta_i^{min}, i' \notin I, b' \leq \beta_{i'}^{min}, s = 1, j \in J_i \cap J_{i'} \cap Mach[s] : i < i' \quad (22)$$

$$L_{ibrs} \geq C_{ibs} + \gamma_{ii'} \quad \forall i, b \leq \beta_i^{min}, i' \in I_i^{SP}, b' \leq \beta_{i'}^{min}, s \neq 2, j \in J_i \cap J_{i'} \cap Mach[s] : \theta_i < \theta_{i'} \quad (23)$$

$$L_{ibrs} \geq C_{ibs} + \gamma_{ii'} - \omega(2 - Y_{ibsj} - Y_{i'b'sj}) \quad \forall i, b \leq \beta_i^{min}, i' \in I_i^{SP}, b' \leq \beta_{i'}^{min}, s = 2, j \in J_i \cap J_{i'} \cap Mach[s] : \theta_i < \theta_{i'} \quad (24)$$

$$L_{ibrs} \geq C_{ibs} \quad \forall i, b \leq \beta_i^{min}, b' \leq \beta_{i'}^{min}, s = 2, j \in J_i \cap Mach[s] : b < b' \quad (25)$$

$$L_{ibrs} \geq C_{ibs} - \omega(2 - Y_{ibsj} - Y_{i'b'sj}) \quad \forall i, b \leq \beta_i^{min}, b' \leq \beta_{i'}^{min}, s = 2, j \in J_i \cap Mach[s] : b < b' \quad (26)$$

$$L_{ibrs} = C_{0ibs} + \gamma_{iir} - \omega(2 - Y_{ibsj} - Y_{i'b'sj}) \quad \forall i, b \leq \beta Itransfer_i^{min} - 1, i' \in I, \beta_i^{min} \leq b' \leq \beta_{i'}^{min}, s = 2, j \in J_i \cap Mach[s] : i \in Itransfer \quad (27)$$

$$L_{i'b's} = C_{0\,ibs} + \gamma_{i'ib'} \quad \forall i, b \leq \beta_{Itransfer_i}^{min} - 1, i' \in I, \beta_{i'}^{min} \leq b' \leq \beta_{i'}^{min}, s = 3, j \in J_i \cap Mach[s] : i \in Itransfer \quad (28)$$

$$C_{Max} \geq C_{ibs} \quad \forall i, b \leq \beta_i^{min}, s \geq 2 \quad (29)$$

$$C_{max} = \phi_j^{min} + (\alpha_j^{min} - 1)\gamma_j + \min \sum_{i \in I_j} \tau_i^{fill} \beta_i^{min} \quad \forall j \in Mach[s] : s = 3 \quad (30)$$

$$Y_{ibsj} \in \{0,1\} \quad \forall i, b \leq \beta_i^{min}, s = 2, j \in J_i \cap Mach[s] \quad (31)$$

$$\bar{X}_{ibi'b'} \in \{0,1\} \quad \forall i, b \leq \beta_i^{min}, i' \in I_i^{SP}, b \leq \beta_{i'}^{min} : i < i' \quad (32)$$

$$L_{ibs}, C_{ibs}, C_{0ibs} \geq 0 \quad \forall i, b \leq \beta_i^{min}, s \in S \quad (33)$$

$$W_{ibs}, W_{0ibs} \geq 0 \quad \forall i, b \leq \beta_i^{min}, s = 2 \quad (34)$$

$$L_{ibs} \leq (n * Workweeklength) \&\& C_{ibs} \geq ((n * Workweeklength) - IdleGamma_i) => L_{ibs} \geq (n) * Workweeklength \quad \forall i, j \in J_s, i \in I, \beta_{Itransfer_i}^{min} \leq b \leq \beta_{i'}^{min}, s = 3, n \in WeekNumber \quad (35)$$

$$L_{ibs} \geq ((n * Workweeklength) - IdleGamma_i) \&\& L_{ibs} \leq (n * Workweeklength) => L_{ibs} \geq (n) * Workweeklength \quad \forall i, j \in J_s, i \in I, \beta_{Itransfer_i}^{min} \leq b \leq \beta_{i'}^{min}, s = 1, n \in WeekNumber \quad (36)$$

$$CWeek_{ibs} \leq (C_{ibs} / (Workweeklength - IdleGamma_i)) + 1 \quad \forall i, i \in I, j \in J, s \in S \quad (37)$$

$$CWeek_{ibs} \geq C_{ibs} / (Workweeklength - IdleGamma_i) \quad \forall i \in I, j \in J, s \in S \quad (38)$$