A Simulation Architecture for the Fish Farming Production Process

Tellef Solberg, NHH & Øyvind L. Brunstad, NMBU

07/02/2020

1. Introduction

The Aquaculture industry is an industry where the production process from broodstock to distribution and sales of fully grown fish is sequential and chronological. This allows for several optimization techniques from process mining and operations research for increasing the performance of the systems. Discrete Event Simulation (DES) is a widely used analysis-technique in such sequential production systems, and it allows for replicating a production process in a virtual environment and then running scenario analyses and determining the effects of different alterations to the system.

The purpose of this study is to create a general simulation model for a fish farming production process. The simulation is not specific to any particular operation of a particular company, however, the purpose is that the data inputs can be altered for the operation that is being investigated. A great emphasis is put on making the model simplistic and easy to interpret and edit for alterations for the specific system to be analyzed.

Using the block-based simulation programming language aGPSS a proposition is made and commented in short.

2. Review of the production process and simulation model

Using data from laksefakta (2015), the following generic production process is created as seen in the flowchart in figure 1.

Note that these figures are all rough inputs into the simulation, and these times would vary with different production equipment, routines, layouts etc. for the specific operation. The purpose of these figures is to have rough inputs into the simulation model.

For simulation to be an effective analysis tool, there should be one or more stochastic elements. Therefore we make the following assumptions in the inital simulation model:

1. The roe/egg production until smolt is uniformly distributed ~ (390,90)
2. Wellboat transportation to the salmon cage facility is uniformly distributed ~ (2,0.5)
3. The feed fish process in the salmon cages is normally distributed ~ (540, 20)
4. The remainding processes are kept deterministic

The simulation architecture can be found in figure 2 with the corresponding source code towards the end in figure 4.

The simulation architecture model contains two separate segments:

1. The sequential production process (starting from GENERATE 100,10)

Figure 1: Production process from roe/egg production until distribution and sales

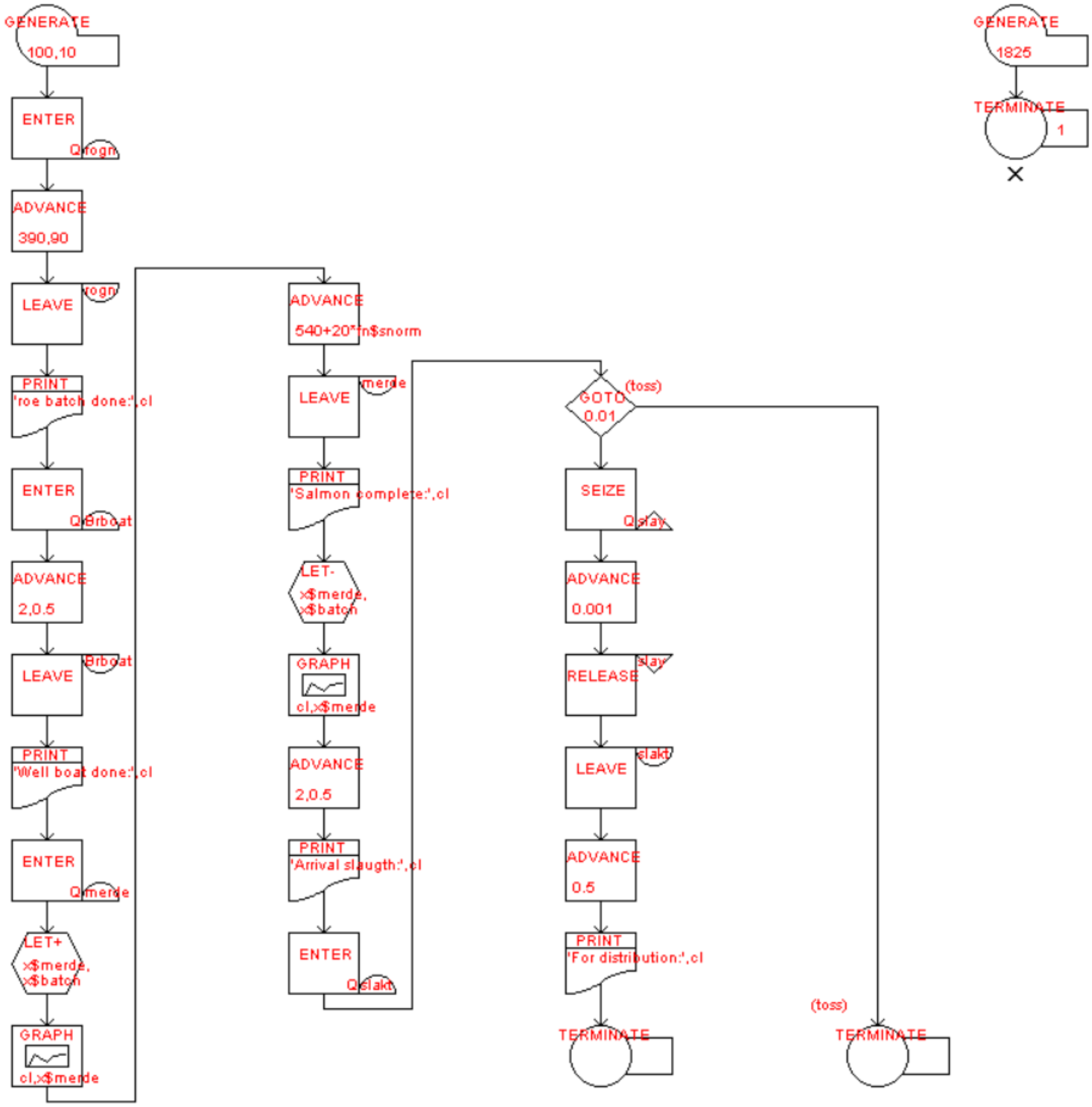

Figure 2: Simulation Architecture in aGPSS

2. The simulation clock of 5 years (starting from GENERATE 1825)

Also note that the longest processing time is by the salmon cage with the normally distributed ~ (540, 20). The levers for dealing with this bottleneck is intuitively to increase the capacity of this station and/or decrease the processing time for the salmons. However, capacity is the lever we acctually can control. Also before the wellboats transport the salmon to the slaughtering facility, there is a given probability that some fish does not fill the requirements for distribution and sales, and they are hence thrown away (as seen in the last terminate block in the production segment)

4. Interpretation of the simulation results

From the block statistics obtained in the simulation, we obtain the following results displayed in figure 3 towards the end.

The block statistics tells us that the simulation ran for 1825 days, and that 5 production units are currently in the salmon cage. All in all, 9 production units finished the whole production process during the 5 years simulation period. Note that the PRINT statements from the program gives us results from when an activity is executed and the corresponding time.

5. Further research

From this rather short report, there are several points to be considered going forward:

1. How does the simulation architecture perform when applying it to a real production operation?
2. How many alterations does the simulation architecture need when applied to a real production operation?
3. How much extra detail does the simulation architecture need for obtaining a suffiecient virtual replication?

References, Source Code and Block Statistics

Laksefakta. (2015, Oktober 26). Bærekraftig laksefôr. Hentet Februar 10, 2016 fra Laksefakta: http://www.laksefakta.no/Norsk-havbruk/Dette-spiser-laksen/Les-mer-om/B%C3%A6rekraftig-laksef%C3%B4r

Laksefakta. (2015, August 25). Fra rogn til mat. Hentet fra Laksefakta: http://www.laksefakta.no/Norsk-havbruk/Laksens-livssyklus/Les-mer-om/Fra-rogn-til-mat

https://github.com/OprSim/Fish-Farming-Simulation

Clock 1825.00

Block counts

Number	Adr.	Oper.	Current	Total
1		GENERA		18
2		ENTER		18
3		ADVANC	4	18
4		LEAVE		14
5		PRINT		14
6		ENTER		14
7		ADVANC		14
8		LEAVE		14
9		PRINT		14
10		ENTER		14
11		LET		14
12		GRAPH		14
13		ADVANC	5	14
14		LEAVE		9
15		PRINT		9
16		LET		9
17		GRAPH		9
18		ADVANC		9
19		PRINT		9
20		ENTER		9
21		GOTO		9
22		SEIZE		9
23		ADVANC		9
24		RELEAS		9
25		LEAVE		9
26		ADVANC		9
27		PRINT		9
28		TERMIN		9
29	TOSS	TERMIN		0
30		GENERA		1
31		TERMIN		1

Figure 3: Block statistics

```
SIMULATE 1

! Smolt batch size:
INPUT X$batch
LET X$cage=100
roe CAPACITY 10
wellB CAPACITY 10
cage CAPACITY 10
slaugh CAPACITY 10
        GENERATE 100,10
        ENTER roe,Q     ! start roe
        ADVANCE 390,90  ! From roe to smolt
        LEAVE roe
        PRINT 'roe batch done  :',cl
        ENTER wellB,Q   ! Well boat acquires smolt
        ADVANCE 2,0.5   ! shipping og handling for transfer
        LEAVE wellB
        PRINT 'Well boat done  :',cl
        ENTER cage,Q    ! Smolt transferred to facility
        LET+ x$cage,x$batch
        GRAPH cl,x$cage
        ADVANCE 540+20*fn$snorm ! Development phase smolt to salmon  (Loc: -8,+2)
        LEAVE cage      ! salmon fully grown
        PRINT 'Salmon complete :',cl
        LET- x$cage,x$batch
        GRAPH cl,x$cage
        ADVANCE 2,0.5   ! transport to slaughtering facility
        PRINT 'Arrival slaugth :',cl
        ENTER slaugh,Q
        GOTO toss,0.01  ! Probability of diseases in batch  (Loc: -6,+2)
        SEIZE slay,Q    ! slaughter time salmon
        ADVANCE 0.001
        RELEASE slay
        LEAVE slaugh
        ADVANCE 0.5     ! Make salmon ready for transport
        PRINT 'For distribution:',cl
        TERMINATE
toss    TERMINATE       ! (Loc: +0,+2)

        GENERATE 1825   ! (Loc: +0,+0)
        TERMINATE 1

START 1
END
```

Figure 4: Source code from simulation