ICIC Express Letters ICIC International ©2016 ISSN 1881-803X
Volume **10**, Number **1**, January **2016** pp. **9–14**

GENETIC ALGORITHMS FOR AN INTEGRATED PROBLEM OF BATCH PRODUCTION AND TRUCK DELIVERY SCHEDULING WITH A SINGLE TRUCK

Jaehyun Park, Kyung Park and Byung Soo Kim*

Department of Industrial and Management Engineering
Incheon National University
No. 119, Academy-ro, Songdo-dong, Yeonsu-gu, Incheon 406-772, Korea
*Corresponding author: bskim@inu.ac.kr

Received June 2015; accepted August 2015

Abstract. *In this article, we consider an integrated problem for batch production and truck delivery scheduling in which batch orders are processed and delivered in batches with limited truck capacity. Orders from customers are first manufactured by a single machine with batch processing in a manufacturing plant and then delivered to the corresponding customers by a single truck. The main decisions are to simultaneously determine production scheduling, batching, and truck delivery scheduling to minimize the makespan. To solve the problem, a mathematical model is derived to obtain the optimal solution and genetic algorithms are designed for large-sized problems. The performance of the algorithms is compared by computational experiments using randomly generated examples.*
Keywords: Integrated scheduling problem, Genetic algorithm, Mixed integer programming

1. **Introduction.** Under competitive global market environment, one of interesting challenges requires keeping lower inventory amounts across the supply chain (SC) to improve customer responses and increase throughputs. In order to satisfy this challenge, a closer interaction between production and distribution activities is necessary and SC members must work towards a unified system and coordinate with each other [1]. In this article, we consider an integrated problem for batch production and truck delivery scheduling with a single truck. Orders from customers are first manufactured by a single machine with batch processing in a plant and then delivered to the corresponding customers by a single truck with limited capacity. The main decisions in this problem are to simultaneously determine production scheduling, batching, and truck delivery scheduling to minimize the makespan which includes the sum of processing times of batches, the idle times of the production in the plant during traveling the truck, and the truck delivery time of a last batch. The production and distribution problems have been widely studied independently in operation research. However, the integration of the problems has little attention. One of the main reasons that few studies attempt to address integrated scheduling problem for production and distribution simultaneously is that the problem itself is already extremely hard to solve. Recently, the coordination between activities of a SC has received a lot of attention in production or operations management. Rather than isolating the scheduling problem in production, several researchers have designed models that integrate several functions to minimize the total cost to improve customer service [2]. Chen [7] performed the only comprehensive review on integrated production and outbound distribution scheduling problems within a single time period. There are a few related studies and extensions on integrated production and distribution scheduling under different production types and transportation modes. Cakici et al. [3] addressed the problem of loading and scheduling of batching machines in an environment with job

release times and incompatible job families. Chang et al. [6] considered an integrated production and distribution scheduling problem in which jobs are first processed by one of the unrelated parallel machines and then distributed to corresponding customers by capacitated vehicles without intermediate inventory. Gao et al. [5] studied a variation of the integrated batch production and truck delivery scheduling problem in which batch of orders are manufactured and delivered in a set of batches by single vehicle with limited capacity. They emphasize the no-wait condition between the production and distribution of each batch, and prove that the general version of this integrated operational scheduling problem is strongly NP-hard. They also investigate to explore the optimal solution structures of two special cases and they propose polynomial algorithms for the cases using the optimality structures.

To the best of our knowledge, there is no research of proposing meta-heuristic algorithms for an integrated problem of batch production and truck delivery scheduling even though the problem is strongly NP-hard. Due to this reason, in this article, we propose genetic algorithms for the problem and evaluate the performances of the proposed algorithms through randomly generated examples.

2. Mixed Integer Programming Model.

The parameters and decision variables in a mathematical model are defined as follows:

Set and parameters

N: set of customers arranged the along pre-determined route served, $N = \{1, \ldots, n\}$
B_k: kth potential batch for a set of orders to be processed and delivered, $k \leq n$
C: truck capacity, $C \leq n$
p_j: processing time of order j, $\forall j \in N$
t_j: round trip travel time between the plant and customer j, $\forall j \in N$

Binary variables

x_{jk}: 1 if batch B_k serves order j, 0 otherwise
z_k: 1 if $\max\{x_{jk}, j \in N\}$ is equal to 1, 0 otherwise
y_k: 1 if batch B_k is produced and delivered, 0 otherwise

Variables

P_k: processing time of batch B_k, and $P_k = \sum_{j \in B_k} p_j$
T_k: travel time of trip B_k, and $T_k = \max\{t_j, j \in B_k\}$
τ_k: production idle time between trips of batches B_{k-1} and B_k, $\tau_k = \{T_{k-1} - P_k\}^+$

Mixed integer programming model

$$Min \quad \sum_{k=1}^{n} (P_k + \tau_k) + T_n \tag{1}$$

$$\text{s.t.} \quad \sum_{k=1}^{n} x_{jk} = 1 \qquad \forall j \in N \tag{2}$$

$$\sum_{j \in N} x_{jk} \leq C \qquad \forall k = 1, \ldots, n \tag{3}$$

$$z_k = \max_{j \in N} x_{jk} \qquad \forall k = 1, \ldots, n \tag{4}$$

$$z_k = y_k \qquad \forall k = 1, \ldots, n \tag{5}$$

$$y_k \geq y_{k+1} \qquad \forall k = 1, \ldots, n-1 \tag{6}$$

$$\sum_{j \in N} p_j \cdot x_{jk} = P_k \qquad \forall k = 1, \ldots, n \tag{7}$$

$$\max_{j \in N} t_j \cdot x_{jk} = T_k \qquad \forall j \in N, \ \forall k = 1, \ldots, n \tag{8}$$

$$T_{k-1} - P_k = \tau_k \quad \forall k = 2, \ldots, n \tag{9}$$

Objective function (1) minimizes the total makespan of the integrated scheduling for batch production and distribution. Constraint sets (2) ensure that each order of customer is delivered only once. Constraint sets (3) limit the capacity of a truck containing in each delivery trip. Constraint sets (4) define z_k which indicates whether at least one order is allocated in B_k. Constraint sets (5) indicate that the batches which serve an order should be produced and delivered. Constraint sets (6) force that empty batches do not exist in the service sequence. Constraint sets (7)-(9) calculate P_k, T_k, and τ_k.

3. **Genetic Algorithms.** Genetic algorithms (GAs) are stochastic search techniques based on the mechanism of natural selection and genetics [4]. In GAs, the representation of solution called *chromosome* has a great influence on the performance of the algorithm. In this study, two GAs are proposed by using different chromosome-types. They are called GA with single string (GA_SS) and GA with double string (GA_DS). In both GAs, every *chromosome* is encoded into a structure that represents its properties, and the set of *chromosomes* forms a population. The chromosome used in GA_SS is described in Figure 1(a). It consists of a single-dimensional string array – batch index array. It is expressed by n digits from 1 to n, where n is the number of potential (imaginary) batches. The digit represents batch index and the index of each cell represents order index. The chromosome used in GA_DS is described in Figure 1(b). It consists of a double-dimensional string array – batch index array and order index array. The batch index array is expressed by n digits from 1 to n, where n is the number of potential batches and the order index array is expressed by n digits from 1 to n without duplication. The same example of the corresponding batch formation from the two chromosome types is illustrated in Figure 1.

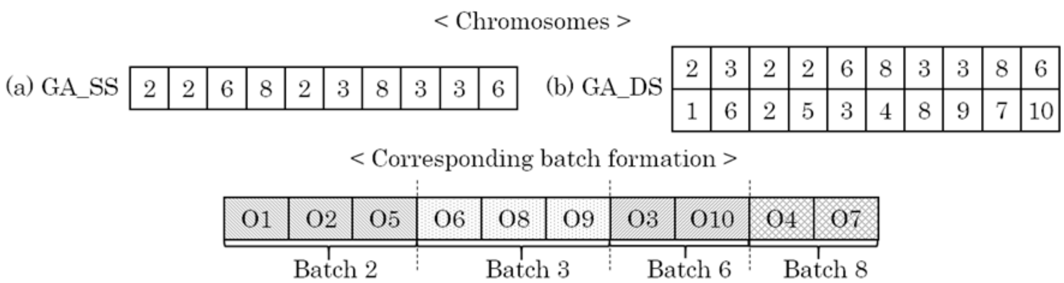

FIGURE 1. Two chromosome representations and their corresponding schedule

GA has two evolutionary operators – crossover and mutation. These operators enhance the performance of solutions by propagating similarities and unexpected genetic characteristics to offspring. For a crossover, uniform crossover is used for batch index array and position-based uniform crossover is used for order index array in GA_SS and GA_DS. For mutation, swap mutation is used for batch index array and order index array in GA_SS and GA_DS. An example of genetic operators is described in Figures 2 and 3. An initial population is generated randomly for the first generation. The chromosomes in the population are evaluated using the makespan as the measure of fitness. Using three genetic operators (crossover, mutation, and reproduction operator), the selected parents reproduce new chromosomes (i.e., children) to generate a population for the next generation. Then the best solution is automatically cloned to one of chromosomes in the next generation and the rest of chromosomes in the next generation is probabilistically formed by roulette wheel method. Then the generation is evaluated and this process is repeated until a stopping criterion (i.e., maximum number of generations) is met. The overall procedure of GAs is shown in Figure 4.

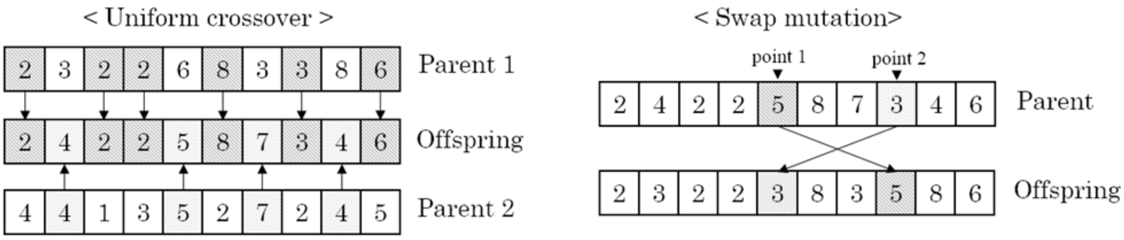

FIGURE 2. Uniform crossover and swap mutation for batch index array

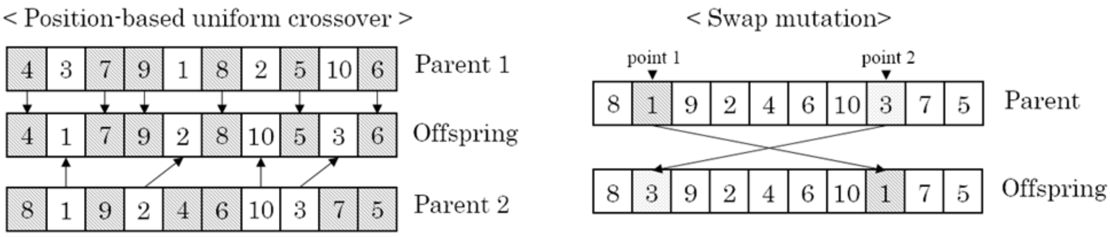

FIGURE 3. Position-based uniform crossover and swap mutation for order index array

```
Generate_InputData();
Initialize_Population();
Update
    {
    Decode_Chromosome();
    Evaluate_Fitness();
    }
while(not terminate condition) do
    Clone_BestSolution();
    Select_Parents();
    UniformCrossover();
    SwapMutation();
    Update();
endwhile      4
Generate_OutputData();
```

FIGURE 4. Overall procedure of GAs

4. **Computation Experiments.** The proposed mixed integer programming was solved by ILOG CPLEX 12.5 and GAs were solved by C#. All experiments were executed on a PC with 3.10GHz Intel® Core^{TM} Processor, and 4.00GB RAM. In less than 20 orders problems, both GAs achieve the optimal solution in most of problems within several seconds. The mixed integer programming model is not tractable for the problems over 20 orders by CPLEX because of the computation time (more than 1 hour). Thus, we focus on developing effective genetic algorithms (GAs) for testing large sized problem instead. The problem becomes more complex as order size and truck capacity are increased. So, we test the GAs 12 combinations of the number of orders (30, 40, 50, and 60) and the capacity of trucks (6, 9, and 12) and repeat each combination 10 times. The production times of orders were generated from a random integer value between $Uniform[7, 13]$ and the delivery times of orders were generated in a random value of $Uniform[5, 25]$. In GAs, the population size is set by $2 \times n$ and the generation size is set by $50 \times n$.

Table 1 is presented in the three measures for evaluating the performance of GAs. The first column in each GA, Relative Percentage Deviation (RPD) is measured, which means the ratio of the deviation of the solution in each GA by comparing to the best feasible solution for every repetition in GAs. This expression is expressed as follows:

$$RPD(\%) = \frac{MH_{sol} - Best}{Best} \times 100$$

where, MH_{sol} is obtained by using GAs, and $Best$ denotes the best feasible solution for every repetition in GAs.

The second column in each GA, Mean Absolute Deviation (MAD) is calculated as a measure to evaluate the variance of algorithms. The MAD is defined as the median of the absolute deviations from the data's median. The third column in each GA, we measured the average computation time of GAs. In Table 1, there are the solutions of large-sized problems using GA_SS and GA_DS. We can see that GA_SS is more effective with low variation than the GA_DS. Average RPD and MAD of GA_SS are 4.90 and 0.88, respectively. Meanwhile, RPD and MAD of GA_DS are 5.94 and 0.96. Furthermore, the average computing time of GA_SS is better than GA_DS.

Table 1. The test results of large-sized problems

				GA_SS			GA_DS		
No.	orders	Capacity	Best	RPD (%)	MAD (%)	Time (Sec)	RPD (%)	MAD (%)	Time (Sec)
1	30	6	653	6.22	0.67	14.21	3.14	0.92	15.99
2		9	670	3.64	0.74	14.41	1.45	0.75	16.10
3		12	654	6.01	0.97	15.42	3.06	1.01	16.04
4	40	6	899	3.37	1.32	36.41	2.01	0.73	43.83
5		9	882	6.12	0.68	35.86	3.64	1.52	43.54
6		12	874	6.46	0.99	35.65	4.20	1.23	44.38
7	50	6	1,077	1.90	0.79	83.66	0.79	0.43	96.07
8		9	1,060	3.47	0.91	75.10	1.09	1.04	95.29
9		12	1,058	3.78	1.00	75.10	0.97	0.85	95.17
10	60	6	1,252	10.18	0.67	140.87	16.71	1.13	197.01
11		9	1,282	3.20	0.97	143.44	17.12	0.82	198.75
12		12	1,269	4.41	0.91	138.78	17.15	1.03	194.75
Average				4.90	0.88	67.41	5.94	0.96	88.08

5. **Conclusions.** In this paper an integrated problem for production and truck delivery scheduling in which orders are processed and delivered in batches with limited truck capacity is considered. The objective of this problem is to simultaneously determine production scheduling, batching, and truck delivery scheduling to minimize the makespan. Two different solution approaches are proposed. The first approach is based on a mixed integer programming model. Since the mathematical model is not tractable for the problems over total 20 orders, we propose GAs (i.e., GA_SS and GA_DS) to increase solution efficiency. The test results in large-sized problems indicate that GA_SS is very effective and efficient algorithm with low variation for the integrated scheduling problem for production and distribution.

In the future, the problem may be interesting to examine an integrated problem for production and truck delivery scheduling considering heterogeneous truck types and truck routing strategy instead of direct shipping delivery strategy.

Acknowledgment. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future (Grant number: NRF-2013R1A1A1012454).

REFERENCES

[1] A. K. Arshinder and S. G. Deshmukh, Supply chain coordination: Perspectives, empirical studies, and research directions, *International Journal of Production Economics*, vol.115, no.2, pp.316-335, 2008.

[2] C. N. Potts and V. A. Strusevich, Fifty years of scheduling: A survey of milestones, *Journal of the Operational Research Society*, vol.60, pp.41-68, 2009.

[3] E. Cakici, S. J. Mason, J. W. Fowler and H. N. Geismar, Batch scheduling on parallel machines with dynamic job arrivals and incompatible job families, *International Journal of Production Research*, vol.51, no.8, pp.2462-2477, 2013.

[4] M. Gen and R. Cheng, *Genetic Algorithms & Engineering Design*, vol.3029, 1997.

[5] S. Gao, L. Qi and L. Lei, Integrated batch production and distribution scheduling with limited vehicle capacity, *International Journal of Production Economics*, vol.160, pp.13-25, 2015.

[6] Y.-C. Chang, V. C. Li and C.-J. Chiang, An ant colony optimization heuristic for an integrated production and distribution scheduling problem, *Engineering Optimization*, vol.46, no.4, pp.503-520, 2014.

[7] Z. L. Chen, Integrated production and outbound distribution scheduling: Review and extensions, *Operations Research*, vol.58, no.1, pp.131-148, 2010.