ISSN 2090-3359 (Print)
ISSN 2090-3367 (Online)

Advances in Decision Sciences

Volume 24
Issue 3
September 2020

Michael McAleer

Editor-in-Chief
University Chair Professor
Asia University, Taiwan

Published by Asia University, Taiwan

ADS@ASIAUNIVERSITY

An Optimization Model for Production Planning in the Synthetic Fertilizer Industry*

A.C. Mahasinghe **

Department of Mathematics
University of Colombo

L.A. Sarathchandra

Department of Mathematics
University of Colombo

Revised: July 2020

*The authors wish to than a reviewer for helpful comments and suggestions.

****** Correspondence: anuradhamahasinghe@maths.cmb.ac.lk

Abstract

Production planning plays a key role in manufacturing. Different industrial requirements impose different conditions, eventually leading to different and non-trivial production plans. In this work, a particular planning problem which arises in synthetic fertilizer manufacturing is considered, to which the industrial conditions in the process of making fertilizer are incorporated. As in standard industrial scheduling problems, we assume a situation where different types of synthetic fertilizer are produced by different mixers having different capacities, in order to fulfil the demand of each vehicle that receives fertilizer from the outlets of those mixers. Loading the fertilizer to vehicles that wait in a queue at the factory premises and cleaning costs when using some mixer to produce two fertilizer types makes our problem significantly different to existing scheduling problems in literature. We first survey related problems and attempt to find the venue of our problem among those standard planning problems. Then we attempt to generate a production plan, addressing the industry-specific conditions for fertilizer manufacturing by reformulating as a binary integer program aimed at minimizing the total waiting time of the vehicles to which the fertilizers are loaded and the machine cleaning cost, satisfying industrial constraints. Due to the non-linearity of the optimization model, we adopt optimization heuristics to generate solutions.

Keywords: Optimization model, production planning, synthetic fertilizer industry, manufacturing, industrial conditions, capacities.

JEL: C61, D24, I52, L52, O21.

1. Introduction

Production planning has become an essential component in industry. Correct and careful decisions made by industrial planners could yield in significant benefits. With the diversity of products and markets available today, manufacturers must deal with a wide range of items and product types in their production lines, which is almost impossible to do manually. This is the context where automated decision support systems have become extremely helpful.

Many traditional decision-making processes are boiled down by these systems into standard machine scheduling problems and production plans are generated accordingly. One may find a variety of papers and books in optimization and decision science literature on production planning and automated decision making in industry, where different versions of standard and other machine scheduling problems are formulated and solved. One may find comprehensive surveys of such systems and methods in Metaxiotis et al. (2002), Pinedo & Hadavi (1992) and Caridi & Cavalieri (2004).

The main reason behind the vast literature on automated production planning is the fact that each industry has its own requirements and goals, making the planning process industry-specific to a certain extent. For instance, in labour-intensive industries such as apparel or porcelain, learning effects of the workers become the most non-trivial and significant factor that must be considered when making decisions on the production (Du et al. 2017, Anderson 2001). Further, in certain industries such as painting or plastic, thorough cleaning is required between operations if two types are produces on the same machine; therefore, cleaning costs affect significantly the decision-making of the planners (Alvarez et al. 2004). Thus, industry-specific factors play a significant role in production planning.

With the demand for agricultural products in the world, even with the trends to confine to natural fertilizers, usage of synthetic fertilizers has never decreased. Fertilizers such as ammonium nitrate, high analysis phosphates, diammonium phosphate, nitric phosphates, ammonium polyphosphate and urea ammonium phosphates have become essential for agricultural farming (Zhang et al.

2020, Russel & Williams 1977). Several synthetic fertilizer manufacturers have come up to serve the markets, in particularly in South Asian countries. These manufacturers do not usually keep stocks but make fertilizer to orders they receive. Those manufacturers produce the above-mentioned synthetic fertilizer types using large mixers (machines) of which the outlets are directed to loading areas in their factories. Each mixer can be used to make different types of fertilizers, but cleaning costs differ when switching from one type to another. T

herefore, minimizing the cleaning cost is one concern in synthetic fertilizer manufacturing. Further, the tipper trucks to which the fertilizers are loaded from the outlets of the mixers usually wait in parking areas for several hours. Thus, another major concern of the production planners is to minimize the waiting time of these trucks. In addition, the demand by each truck must be fulfilled by the production. Also, due to space restrictions, the production plan must ensure that whatever produced for one truck comes along either a single mixer, or two adjacent mixers, in order to make the movement of tipper trucks smooth as much as possible. This has made the task of decision-making in fertilizer industry very challenging for planners.

It is noteworthy that despite several automated decision-support systems are available for different industries such as apparel, textile and plastic, addressing industry-specific requirements (Charka & Jaju 2020, Renata et al. 2020, Dahmen et al. 2020), it is not the case with synthetic fertilizer industry. This is the context we were motivated to introduce an optimization model for a tentative decision support system, particularly intended for synthetic fertilizer industry.

We first set up a mathematical formulation which reflects the practical scenario, by reducing it down eventually into a binary integer program, which turns out to be both non-linear and non-convex. This makes the process of finding an optimum highly non-trivial, motivating us to move away from the search for closed-form solutions and to look upon *heuristic* methods, mainly the *local search* and the *tabu search*. Accordingly, we break down the process of finding an optimal solution into two phases, where in phase 1 the problem is restated as a constraint satisfaction problem and a feasible solution is found satisfying industrial constraints. The second phase is focused on improving the feasible solution in order to minimize two objective functions.

Accordingly, the paper is organized as follows: first we present a brief survey of related industrial problems in section 2. In section 3, we describe our problem and the optimization model with the mathematical formulation. In section 4 we discuss computational experience with a description of the local and tabu search methods. We conclude the paper in section 5.

2. State of the Art

Let us first identify where our problem is placed in the literature on scheduling. In general, a decision-making problem which deals with the allocation of resources to activities over given time periods and its goal is to optimize one or more objectives is regarded a *scheduling problem* (Pinedo & Hadavi 1992). When making a schedule, one should consider an objective to be optimized and the constraints which hinder the optimization process. The machine cleaning cost must be minimized in our problem together with the waiting time of the tipper trucks. Further, it is mainly the machine conditions that restrict the optimization. Thus, we first consider the machine conditions and attempt to place the problem.

Considering machine restrictions, scheduling problems are classified in literature into different classes. There are different types of machine environments such as *single machine*, *identical parallel machines* and *unrelated parallel machines* (Tanaev et al. 2012). Recall the fertilizer mixers in a typical factory have different capacities and they operate independently; it is not difficult to identify our problem belongs to the class of unrelated parallel machines.

A *setup* or *changeover* is defined as any preparatory procedure that needs to be performed whenever a machine switches production between different items (Geoffrion & Graves, 1976). Setup cost is the cost to set up any resource prior to next activity. Setup costs may include obtaining tools, returning tools, setting up suitable environments etc. There are two types of setup costs: *sequence-dependent* and *sequence-independent* (Fleischmann, 1994). If setup cost depends on both preceding task and the task to be processed, it is called sequence dependent. On the other case, if it is only the task to be processed it is sequence independent. Detailed explanations on setup costs can eb found in Allahverdi & Soroush (1998). In our problem, the only significant setup cost is the cleaning cost which is sequential and machine-dependent.

Graham introduced a convenient three field notation $\alpha | \beta | \gamma$ to represent a scheduling problem (Graham et al. 1979). The α field describes the machine environment and the number of machines, β field provides the feasibility constraints and the γ field contains the objective functions. For example, $1||\Sigma wjCj$ thus corresponds to a single machine weighted completion time problem, while $P_{10} | prmp | C_{max}$ denotes 10 identical parallel machines with preemption under minimizing makespan (maximum completion time). Accordingly, the notation relevant to our problem is $R_m | prmp, S_{ijk}, P_k | \sum W_k$ which has m unrelated parallel machines (R_m) with preemption $(prmp)$ and sequence-dependent setups (S_{ijk}) under minimizing total waiting time.

Research efforts on parallel machine scheduling problems have been dealt in a vast range. These problems can be further classified according to the characteristics of machines (identical, uniform, and unrelated machines) and setups (sequence and machine-dependent, machine dependent setups). Consideration of sequence dependent setup times or costs between jobs has not been considered until recently. In Allahverdi et al. (2008), a review of scheduling problems with setup times is presented, including the parallel machine case. Other characteristics are job preemption (preemptive and non-preemptive) and due dates, release dates, and so on. We briefly review existing related works in this direction.

The majority of these published work are aimed at minimizing makespan (maximum completion time) and weighted tardiness (which is related to problems with due dates) as the scheduling objectives. Since scheduling problems are generally regarded non-trivial and challenging in optimization literature in a computational perspective, computational progress of the above-mentioned classes of problems is worth exploring. We present below some computational attempts on parallel machine scheduling problems with sequence-dependent setups.

Helal et al. (2004) attempted minimizing the makespan for the non-preemptive, unrelated parallel machines scheduling problem with sequence-setup times using optimization heuristics. They proposed a *tabu search* algorithm that uses two phases of perturbation schemes: the *intra-machine perturbation*, which optimizes the sequence of jobs on the machines, and the *inter-machine perturbation*, which balances the assignment of the jobs to the machines. They compared the proposed algorithm to an existing one that addressed the same problem. They claim that computational results show that the proposed tabu search procedure generally outperforms the heuristic approach used in Helal & Hosni (2003) for small- and large-sized problems.

Also, Jae-Ho Lee et al. (2013) have used the tabu search approach for unrelated parallel machine scheduling with sequence and machine dependent setups under the objective of minimizing total tardiness. In this study, they suggest a version of tabu search algorithm that incorporates various neighborhood generation methods. They conducted a computational experiment was done on the instances generated by previous research articles and the results show that the tabu search algorithm outperforms the *simulated annealing* algorithm significantly. Also, an additional test was done to compare the performances of the tabu search and the existing iterated greedy algorithms, and the result shows that the tabu search algorithm gives faster solutions than the iterated greedy algorithm although it gives less quality solutions.

Another tabu search based algorithm has developed by Logendran et al. (2007) to minimize the weighted tardiness of jobs in unrelated parallel machine scheduling problem with setups. Since it is difficult to solve industrial-size problems efficiently, six different search algorithms based on tabu search are developed to identify the best schedule that gives the minimum weighted tardiness. To enhance the efficiency of the search algorithms, four different initial solution finding mechanisms, based on dispatching rules, are developed by them. Another noteworthy attempt is the usage of genetic algorithms for the unrelated parallel machine scheduling problem with sequence dependent setup times by Vallada and Ruiz (2010) aimed at minimizing the makespan.

Although plenty works are available in literature on production planning with parallel machines, most of the studies considered machines as the only resource. It is more realistic to suppose manufacturing systems to have other resources such as machine operators, storage facilities and raw materials. A few works have considered non-trivial resource constraints in their scheduling problems. Recently, Afzalirad & Rezaeian (2016) have considered a constrained scheduling problem with unrelated parallel machines and setups. In this work, each job can be processed on a specific subset of machines and all the machines are not capable of handling every job. They have proposed a new integer programming model to minimize the makespan and two meta-heuristics, *genetic algorithm*s and *artificial immune systems* were to find optimal solutions.

In our problem, there are non-trivial constraints to be considered, mainly associated with the loading process. Since there is no storage facility, whatever produced in a mixer should be loaded to relevant truck and due to the smooth movement of trucks, one trucks order should be produced in either a single machine or two adjacent machines. To the best of our knowledge this problem has not been investigated in the literature.

3. Optimization Model

3.1 Inputs

Suppose p fertilizer types are produced on m mixers for l tipper trucks. Each truck has its order consisting of different quantities from different fertilizers. The fertilizer mixers in the factory are unrelated in the sense that processing time of a job depends on the mixer to which the job is assigned. Each mixer is capable of producing any fertilizer type but only one job can be processed on one mixer at a time. That being said, the demand of one truck can come along two different mixers, but they must be adjacent in that case. Job processing is preemptive; that is, jobs can be interrupted at any time and resumed later in same mixer or another. Recall the objectives are minimizing the total waiting time and the cleaning cost for a given set of tipper trucks, cleaning cost is both machine and sequence-dependent. For convenience, setup times are neglected when calculating the waiting time. Order quantities of each should be fulfilled.

The scheduling horizon is divided into a finite number of time intervals with known duration. Each task can only start or finish at the boundaries of these time intervals. Thus, resource constraints are only monitored at predefined and fixed time points. This enables us to define our indices needed for the optimization model.

Indices

i: mixer index, $i \in \{1,...,m\}$

j: fertilizer type, $j \in \{1,...,p\}$

k: truck index, k $\in$ $\{1,...,l\}$

t: time slot, t $\in$ $\{1,...,a\}$

Decision variables

Accordingly, we introduce a set of decision variables x_{ijkt} aimed at checking the possibility of making fertilizer j for truck k at a particular time slot t on mixer i. Thus, it a decision variable with degree of freedom equal to four, with corresponding indices: machine, product type, truck and the time slot:

$$x_{ijkt} = \begin{cases} ?1 & \text{if type } j \text{ for truck } k \text{ is produced at the } t^{th} \text{ time slot on machine } i \\ ?0 & \text{otherwise???} \end{cases} \quad \textbf{(1)}$$

Parameters

In addition, we define several industry-specific parameters needed for the optimization model.

r_{ij} : production rate of fertilizer j on mixer i

d_{jk} : order quantity from fertilizer j for truck k

s_{ijk} : cost for cleaning mixer i when switching from fertilizer j to k

u : size of a time slot

a : number of time slots

3.2 Objectives and constraints

Waiting time Recall one objective is minimizing the waiting time, it is necessary to state this objective in terms of the decision variables and parameters. Given a list of fertilizer orders from a fixed number of trucks, the last time slot T_k at which some mixer produces fertilizer for the k^{th} truck can be expressed as follows:

$$T_k = \max\left\{t \mid \exists i \in \{1,..,m\}, \exists j \in \{1,..,p\} \text{ such that } x_{ijkt} = 1\right\}. \qquad \textbf{(2)}$$

It is clear that T_k in Equation (2) gives the waiting time of the k^{th} truck. One objective is minimizing the total waiting time of the trucks, that is the sum of all T_k 's as expressed symbolically by the following function in Equation (3):

$$T = \sum_{k=1}^{l} T_k \qquad \textbf{(3)}$$

Switching cost When two fertilizer types j_1 and j_2 are produced on the same mixer, the cost for cleaning when switched from j_1 to j_2 is given by $s_{j_1 j_2}$. Suppose j_1 was produced on machine i at the time slot t, then for that cost to incur, j_2 must be produced on the same machine at the $t+1$st time slot. In other words, a cost of $s_{j_1 j_2}$ incurs when $x_{ij_1kt} = 1$ and $x_{ij_2k(t+1)} = 1$. Notice that these fertilizer types are intended for truck k. However, in real, they can be produced for two trucks k_1 and k_2. Thus, the cleaning cost $C_{ij_1 j_2 t}$ for machine i when switched from j_1 to j_2 in between time slots t and $t+1$ is expressible using the decision variables as follows:

$$C_{ij_1 j_2 t} = \sum_{k_1=1}^{l} \sum_{k_2=1}^{l} x_{ij_1k_1t} x_{ij_2k_2(t+1)} s_{j_1 j_2} \tag{4}$$

Thus, the total switching cost C_i for machine i is given by Equation (5):

$$C_i = \sum_{j_1=1}^{p} \sum_{j_2=1 (j_2 \neq j_1)}^{p} \sum_{t=1}^{a-1} C_{ij_1 j_2 t}, \tag{5}$$

from which we derive our second objective function of the total cleaning cost given by Equation (6) below:

$$C = \sum_{i=1}^{m} C_i. \tag{6}$$

Constraints Recall all fertilizer orders of l trucks must be served; the relevant set of constraints can be expressed by Equation (7) as follows:

$$\forall j \in \{1,..,p\}\}, \forall k \in \{1,..,l\}, \sum_{i=1}^{m}\sum_{t=1}^{a}\left(ux_{ijkt}r_{ij}\right) = d_{jk}. \tag{7}$$

As in all scheduling problems, even in our problem, a production plan must ensure that at most one fertilizer type must be produced during a given time slot on a mixer. This can be expressed by Equation (8) as follows:

$$\forall i \in \{1,..,m\}, \forall n \in \{1,..,a\} \sum_{j=1}^{p}\sum_{k=1}^{l} x_{ijkt} \le 1 \tag{8}$$

The very specific requirement in fertilizer industry for smooth transition of trucks must be addressed now. That is, if i_1 and i_2 are non-adjacent mixers, no tipper truck must receive fertilizers from both. This can be stated by Equation (9) as follows:

$$\forall k \in \{1,..l\}, \sum_{i_1=1}^{m}\sum_{i_2(|i_1-i_2|>1)}^{m}\sum_{j_1}^{p}\sum_{j_2}^{p}\sum_{t_1}^{a}\sum_{t_2}^{a} x_{i_1 j_1 k t_1} x_{i_2 j_2 k t_2} = 0. \tag{9}$$

Finally, it must be specified that all decision variables are Boolean:

$$\forall i \in \{1,..,m\}, \forall j \in \{1,..p\}, \forall k \in \{1,..l\}, \forall t \in \{1,..,a\}, \; x_{ijkt} \in \{0,1\}. \tag{10}$$

Therefore, the planning problem is thus boiled down to an optimization problem in the form of a binary integer program, aimed at minimizing the objective functions given by Equations (3) and (4), subject to the constraints given by (7), (8), (9) and (10).

3.3 Penalty formulation

It is a well-known fact that binary integer programming is an NP-hard problem. Several approximation algorithms can be found in literature for solving linear binary integer programming problems, to which many standard practical problems are reduced (Schrijver, 1998) Our formulation however is non-linear, due to objective functions in Equations (3), (6) and constraints in Equations (9). Further, non-differentiability rules out *gradient-based* optimization methods from the scope. These factors motivate us to investigate *heuristics*, instead of exact or approximate optimization algorithms.

The field of heuristics and metaheuristics achieved significant progress in the last few years. Several NP-complete and NP-hard problems have been attempted by heuristic algorithms, generating interesting results (Gendreau et al., 1994, Hertz & Wera 1987, Osman 1993, Song et al. 2003, Mahasinghe et al. 2019). In order to attempt an optimization problem using heuristics, it is important to restate the problem in a way compatible with the method. Almost all heuristics are governed by a comparison of values at certain instances. Therefore, keeping constraints as they appear might be problematic, and they must be transformed into a comparable format. We resolve his problem by replacing the constraints by equivalent penalty functions.

Accordingly, constraint in Equation (7) is replaced by the following penalty:

$$P_1 = \left| d_{jk} - \sum_{i=1}^{m} \sum_{t=1}^{a} \left(t * x_{ijkt} * r_{ij} \right) \right| . \qquad \textbf{(11)}$$

Similarly, constraints in Equations (8) and (9) are replaced by the penalties in Equations (12) and (13) respectively:

$$P_2 = \max \left\{ \sum_{j=1}^{p} \sum_{k=1}^{l} x_{ijkt} - 1 \, , \, 0 \right\} . \qquad \textbf{(12)}$$

$$P_3 = \sum_{k=1}^{l} \sum_{i_1, i_2=1}^{m} \{ (\sum_{j=1}^{p} \sum_{t=1}^{a} x_{i_1 jkt}) * (\sum_{j=1}^{p} \sum_{t=1}^{a} x_{i_2 jkt}) \}. \tag{13}$$

Terms P_1, P_2 and P_3 serve as follows: P_1 penalizes if some tipper truck violates the parking restriction; that is, if fertilizer for some tipper truck comes through two non-adjacent mixers then, $P_1>0$ and it increases $G = P_1 + P_2 + P_3$. Similarly, P_2 penalizes if more than one job have assigned at a same time slot of a machine. P_3 penalizes if demand constraint is violated. i.e. if a particular trucks demand does not fulfil or an excess amount is produced by some mixer, then $P_3>0$ and the particular solution becomes infeasible. A feasible solution should have zero penalty values. i.e. $P_1 = P_2 = P_3 = 0$.

4. Computational Experience

4.1 Heuristic approach

Local Search Local Search (LS) is perhaps the simplest heuristics method, on which several other heuristics are developed. It starts at a given initial solution and at each iteration the algorithm replaces the current solution by a neighboring solution (Arts et al., 2003). A neighborhood is generated by the application of an operator that performs a small change to the current solution. The search process stops when a local optimum has obtained. i.e. all the neighbors are worse than the current solution. However, it does not result in local optimal solution when the problem is non-convex (Liberatore & Camacho-Collados, 2016).

Tabu Search Introduced by Glover as an improvement of LS (Glover, 1990), *tabu search* (TS) is one of the mostly used metaheuristics to find solutions of various combinatorial optimization problems (Løkketangen & Glover, 1998, Molina et al. 2007, Dorne & Hao, 1999). Unlike other techniqes such as *simulated annealing* and *genetic algorithms*, TS technique does not utilize random numbers. This has been shown to be advantageous recently by Helal et al. (2016). TS implementation starts from an initial solution and moves from the current solution to the best one among its neighborhoods at each iteration even local optimality is attained. Thus, it has another advantage of being able to overcome the limitations of local optimality. To avoid cycling, recently visited solutions are set to be in a tabu for a certain number of iterations. Generally, TS application characterized by several factors. They are, *initial solution*, *neighborhood generating method*, *evaluation function* and *termination criterion*.

Initial Solution: A feasible or an infeasible solution can be selected as an initial solution, but it is customary to select the best feasible solution as the initial solution in order to speed up the search for finding the best final solution to the problem. Past researches have used different mechanisms in finding an initial solution. Most of them are based on priority dispatch algorithms.

Neighborhood generation methods: The neighborhood function is the most important part of the TS algorithm, as it significantly affects both the running time and the quality of solutions. A set of neighborhood solutions to the current solution can be created by applying different moves, namely the swap move, insert move and shift move. A swap move is a move that interchanges the positions of two jobs that are either assigned to the same machine or different machines. An insert move is a move that inserts a job to any machine. The shift move is one of simplest which changes the current value of a variable to another.

Evaluation Function: An evaluation function is used to evaluate the current solution at each iteration in order to determine the search direction. It contains all objective functions if it is a multi-objective function case.

Termination Criterion: The usual stopping criterion involved in TS algorithm are the maximum number of iterations, maximum number of iterations without any improvement in evaluation function and the specified amount of computer time allowed.

Accordingly, the basic steps of the TS are outlined below.

Input: An initial solution x_0 and iteration upper bound *num_iter*
Output: Good solution, x^*
Step 0:

```
Set x (0) = x0;  x = x0;  Tabu list,  T (1): = null and let h: =1
Step h:
Find the best neighbor x'∈N (x(h -1)) \T (h-1)
 If  x' is better than x, then update x := x'
stop; otherwise update T(h+1).
If h ≥ num_iter and local optimality has been attained at least once
then Stop; otherwise update T(h+1)
Let h :=h+1 and go to step h.
```

4.2 Solution criteria

Since our problem has multiple objectives and three sets of constraints which are non-linear, it is a challenging task to achieve the feasibility and improve the objectives simultaneously. Thus, solution approach was divided into two phases. First phase was used to find an initial feasible solution by considering constraints alone. For that, constraint satisfaction problem was solved using LS as described in 4.1. Then, using that solution as the initial solution, objective functions were minimized in the next phase using TS.

Phase 1: In this phase main objective is to find a feasible assignment. For that, local search method is used under constraint satisfaction problem (CSP). Starting with the assignment which takes all decision variables are zero, at each iteration it proceeds to find the best neighboring solution of the current solution using shift neighborhood. i.e. at each iteration, a chosen variable changes its value from zero to one. An assignment(S) is represented by a 4-dimensional array where each dimension represent one index of the decision variable (i,j,k,t). Therefore, there are 4 approaches to choose a candidate variable. The order in which a variable is chosen depends on this approach. As you can see in the following figure, in a 2-dimensional array there are two approaches; row-wise(a), column-wise(b) to a variable. Numbers 1,2,3,4 indicates the order in which a variable is chosen to shift. This order depends on the approach.

The solution space(T) for the problem is the set of all assignments. Thus, the problem can be restated as follows:

Minimize:

$$G(S) = P_1(\mathrm{S}) + P_2(\mathrm{S}) + P_3(\mathrm{S})$$
$$\text{subject to } S \in T \qquad \textbf{(14)}$$

We now present a general sketch of our method of solving he problem.

Notations:

S: Current assignment

S*: The best known assignment

S ($x_{ijkt} \leftarrow 1$): variable x_{ijkt} changes its value to 1 in S

N(S): Neighborhoods of S

N(S)= {S ($x_{ijkt} \leftarrow 1$) for some i,j,k,t}

G: Evaluated value of S*

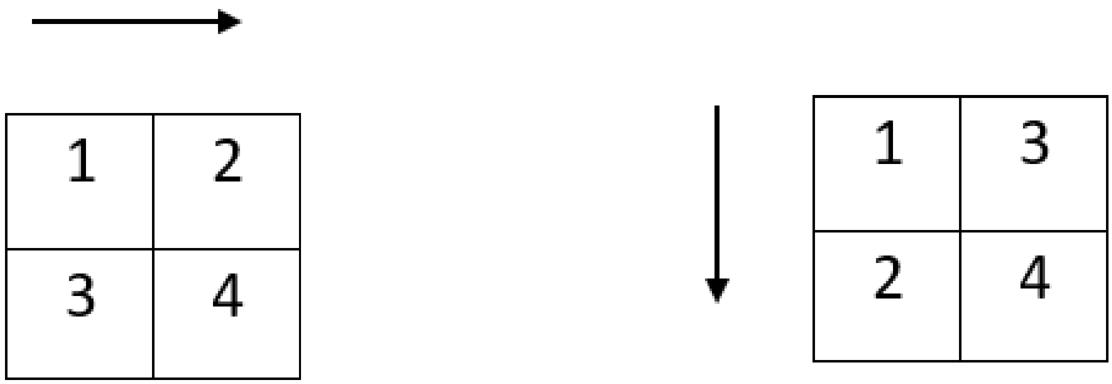
J (0): Set of variables currently equal to zero.

Initialization:

```
Choose the initial solution S0
Set S := S0,            S*  := S0,       G = G(S0)
```

Figure 1

Shift neighbourhood



(a) (b)

Search:

```
while |J (0) | > 0 Do,
Generate {N(S)};
Select best S' ∈ {N(S)}
If G(S') <G* then,
Set         G := G(S0)       and        S  := S0    ;
else
stop;
endif
endwhile
```

Solution: S and G

Phase 2 In this phase, the objective is to optimize the feasible solution in order to minimize the waiting time and cleaning cost. In single-objective optimization problems, the superiority of a solution over other solutions is easily determined by comparing their objective function values. But when it comes to multi-objective problems, a single solution which minimizes all objective function values simultaneously may be does not exist. A solution is said to be *Pareto optimal*, if at least one objective function cannot improve without reducing the other objective functions. The weighted sum method is used to improve the current solution using following objective function (15) which need to be minimized:

$$F(x)= w\,(1)\,*T + w\,(2)\,*C. \qquad \textbf{(15)}$$

I

f all of the weights are positive, then minimum of (14) is Pareto optimal [10]. To solve (15) tabu search algorithm is used considering feasible solution arrived at phase 1 as the initial solution. Same neighborhood structure, shift neighborhood was used in this phase also as other neighborhood structures are complicated in this problem environment.

Table 1

Demand of trucks

Demand	A	B	C
P	4250	2500	1000
Q	1000	1450	1200
R	1500	3000	1800

Table 2

Capacities of mixers (amount per unit time)

Capacity	**P**	**Q**	**R**
1	25	100	50
2	50	50	10
3	10	100	60
4	50	30	40

Table 3

Cleaning cost per mixer

Machine 1			
	P	Q	R
P	-	100	200
Q	200	-	100
R	100	150	-
Machine 2			
	P	**Q**	**R**
P	-	300	150
Q	100	-	100
R	200	100	-
Machine 3			
	P	**Q**	**R**
P	-	400	500
Q	100	-	400
R	200	100	-
Machine 4			
	P	**Q**	**R**
P	-	300	400
Q	200	-	400
R	200	100	-

Now we illustrate the computational method using an example. Consider 3 trucks A, B, C with some order quantities from 3 fertilizer types P, Q and R. Assume the factory has 4 mixers with different capacities. Deterministic data relevant to the problem which need in calculations are shown in the following tables.

Table I which contains demands of each truck gives the values of the parameter d_{jk} in our formulation. And table II illustrates capacity r_{ij} of each machine. Capacities are given as the rates at which each type is produced in each machine. These data relevant to the parameter r_{ij} in the formulation. Final table III which contains cleaning costs carrying the values corresponds to the parameter s_{ijk}. Size of time slots and number of time slots are the other two parameter values that we need in calculations. Number of time slots can be chosen some arbitrarily large value. But the size of time slots depends on the problem data such as demands and the capacities of machines. Therefore, it is required to figure out comparing solutions for different values. For that, FIG 2 and FIG 3 were used.

FIG 2 illustrates the variation of waiting time, cleaning cost and penalty value corresponds to demand (P_3) with respect to different sizes of the time slots. Other two constraints P_1 and P_2 have not considered here as they do not depend on the size of time slots. It can be seen in FIG 2 variation of cleaning cost is negligible when t > 5. Other two function values (waiting time, demand violation) are comparatively lower in that time period.

Therefore, to compare waiting time and violation of demand, they are drawn separately in FIG 3. Violation of demand constraint is minimum around 5 and waiting time also comparatively lower at that point. Since violation of demand constraint affect the feasibility of the solution it is the crucial fact which needs to consider. Thus, 5 is chosen as the most appropriate value for the size of time slots for this problem data.

Figure 2

Variation of waiting time, cleaning cost and demand with the size of the time slot

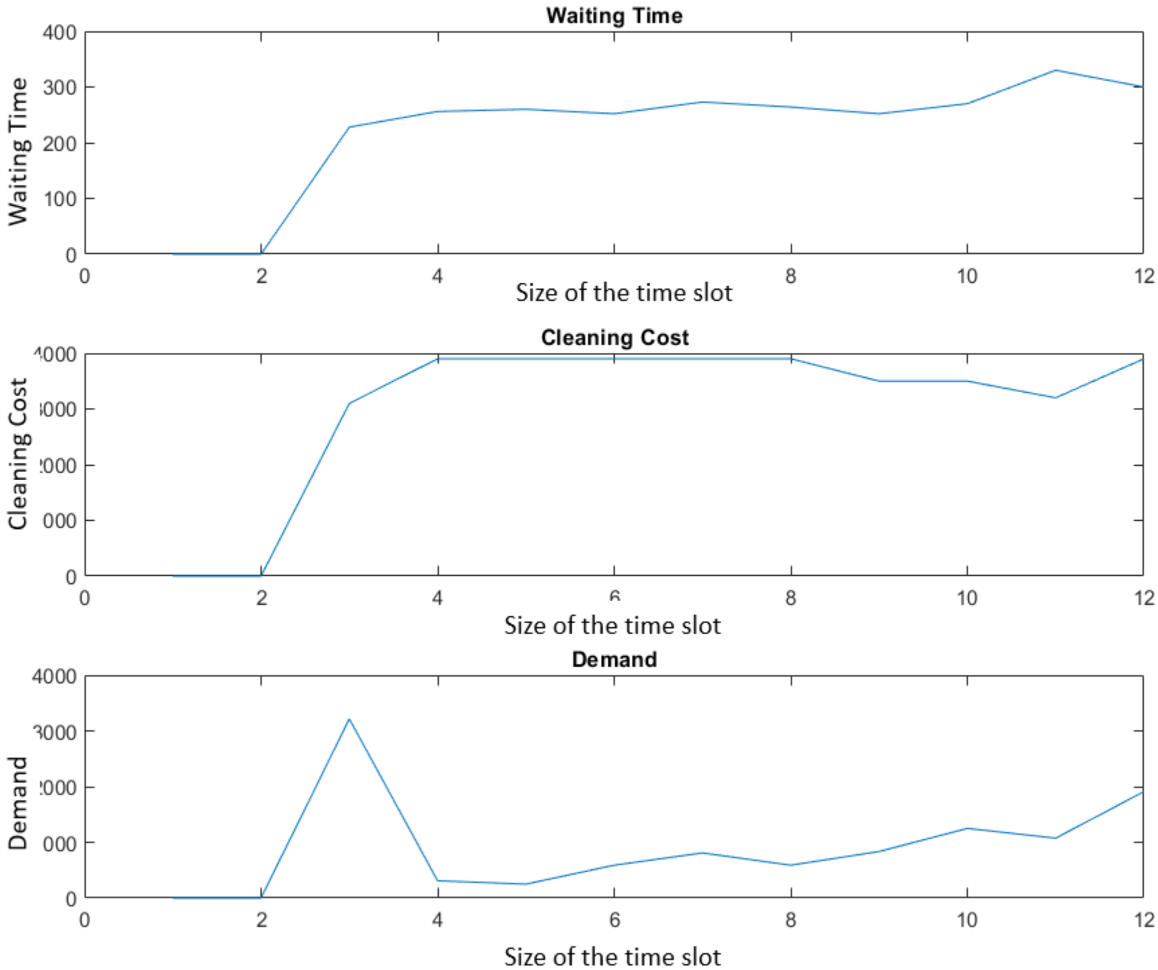


Figure 3

Violation of constraints with size of the time slot

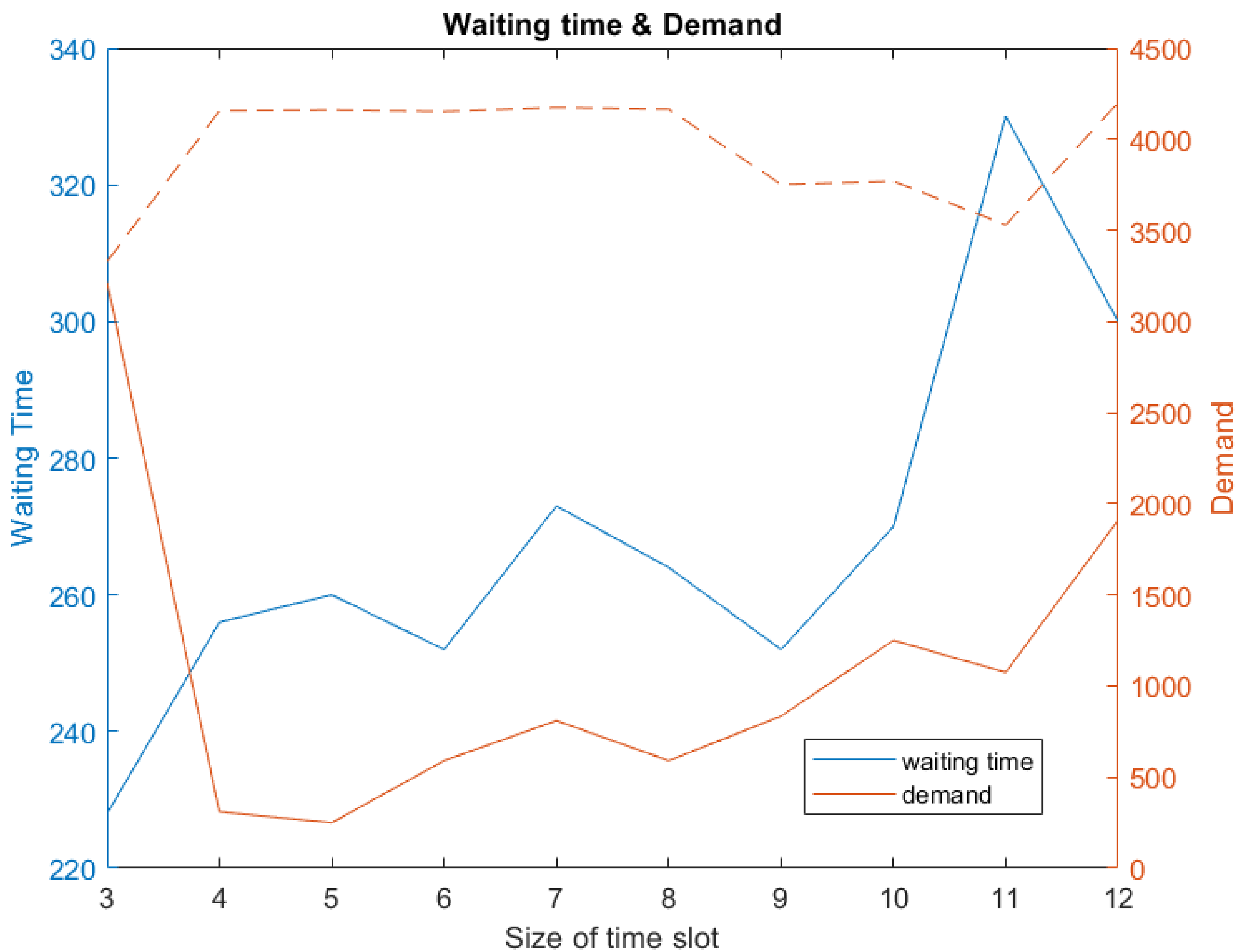


Figure 4

Schedule 1

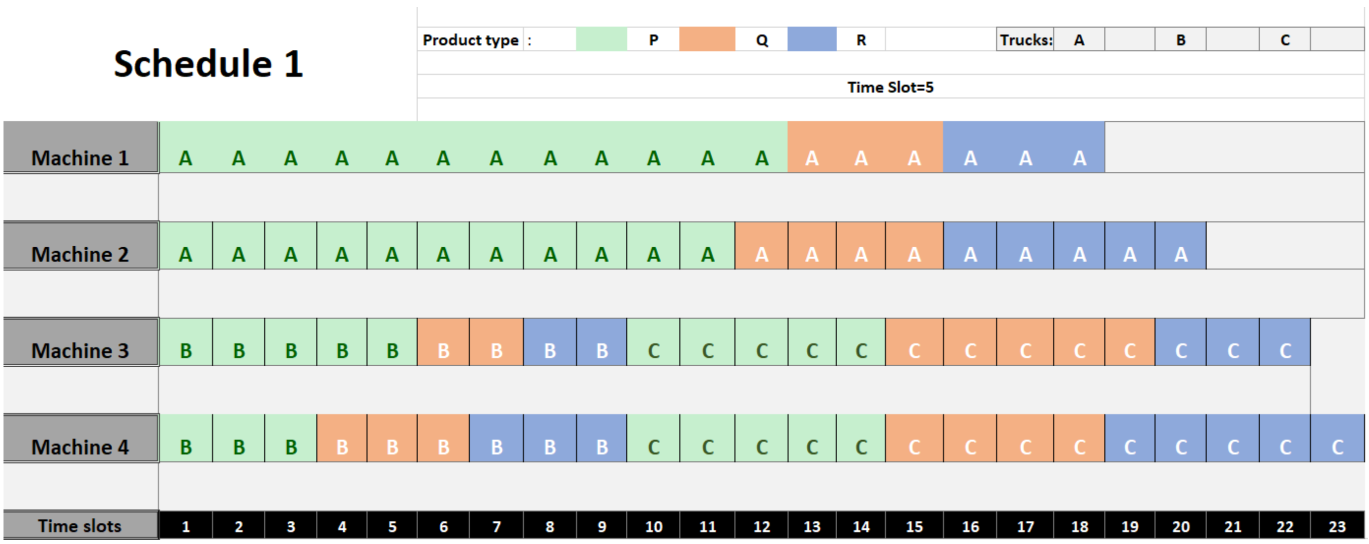
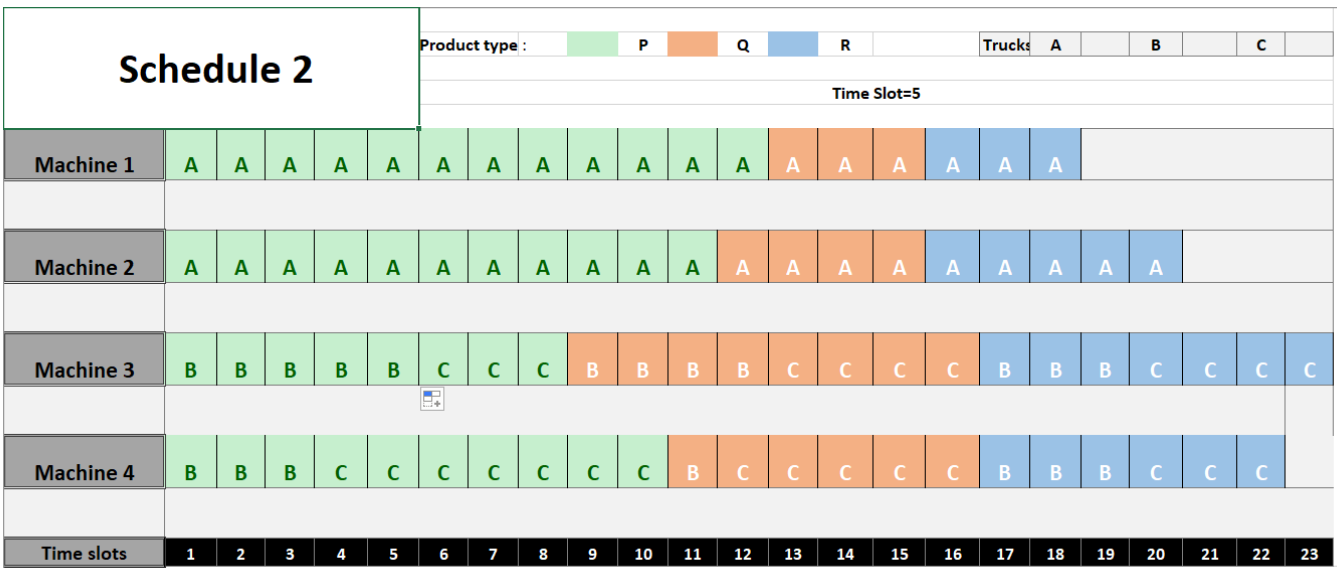


Figure 5

Schedule 2



After figuring out formulation parameters for the relevant problem, it can be solved using the MATLAB code implemented. As explained before, in first phase, it is focused on finding an initial feasible solution. FIG 4 and FIG 5 depict two schedules obtained in phase 1. Schedule 1 was obtained by giving priority on trucks. It chooses one truck and assigns its jobs to suitable time slots and then choose another truck, reducing the waiting times of trucks. When producing schedule 2, priority was given to the fertilizer type. i.e. one type is produced for all the trucks and then switch to another type. Thus, the cleaning cost is lower in schedule 2.

After figuring out formulation parameters for the relevant problem, it can be solved using the MATLAB code implemented. As explained before, in first phase, it is focused on finding an initial feasible solution. FIG 4 and FIG 5 depict two schedules obtained in phase 1. Schedule 1 was obtained by giving priority on trucks. It chooses one truck and assigns its jobs to suitable time slots and then choose another truck, reducing the waiting times of trucks. When producing schedule 2, priority was given to the fertilizer type. i.e. one type is produced for all the trucks and then switch to another type. Thus, the cleaning cost is lower in schedule 2.

As seen from tables IV and V, schedule 1 has 260 units of total waiting time and 4000 units of cleaning cost. For the schedule 2, these values are 310 units and 2100 units respectively. If waiting time alone was considered, schedule 1 is better than the schedule 2 and the cleaning cost alone, it is the other way round. On the other hand, both schedules 1 and 2 produce some excess amounts of products. For the schedule 1, it is 200 of amount and for schedule 2, it is 250. Thus, to overcome this issue the second phase was implemented aimed at finding a Pareto optimal solution. Giving equal priorities to both objective functions in equation (4.12), i.e. w (1) = w (2) phase 2 was executed. This improved the computation up to certain extent, and it would be a future research task to improve it further using moderated tabu search algorithms as in Cordeu et al. (2008).

5. Conclusion

Different industries impose different conditions on production planning. Despite the commendable progress made in other industries on automated production planning with industry-specific constraints, no attention has been paid to this in synthetic fertilizer industry, although the planners in fertilizer factories find it overwhelmingly difficult to handle the decision-making process manually. We considered the decision-making process of such a planner from a mathematical point of view and stated the problem in verbal form. We then surveyed related industrial planning problems and attempted to identify where our problem could be placed. Then we considered the specific industrial requirements in a synthetic fertilizer factory; and formulated a mathematical program in which an optimal production plan in embedded. Out formulation turned out to be a non-linear Boolean optimization problem, thus we attempted solving it using optimization heuristics, namely, the local search and the tabu search.

Table 4

Waiting times of trucks

Truck	**A**	**B**	**C**	**Total**
Schedule 1	100	45	115	260
Schedule 2	100	95	115	310

Table 4

Cleaning Costs of machines

Machine	Cleaning cost in schedule 1	Cleaning cost in schedule 2
1	200	200
2	400	400
3	1800	800
4	1600	700
Total	4000	2100

References

Afzalirad, M., & Rezaeian, J. (2016). Resource-constrained unrelated parallel machine scheduling problem with sequence dependent setup times, precedence constraints and machine eligibility restrictions. *Computers & Industrial Engineering*, *98*, 40-52.

Allahverdi, A., & Soroush, H. M. (2008). The significance of reducing setup times/setup costs. *European Journal of Operational Research*, *187*(3), 978-984.

Alvarez, D., Garrido, N., Sans, R., & Carreras, I. (2004). Minimization–optimization of water use in the process of cleaning reactors and containers in a chemical industry. *Journal of cleaner production*, *12*(7), 781-787.

Andersen, T. J., & Segars, A. H. (2001). The impact of IT on decision structure and firm performance: evidence from the textile and apparel industry. *Information & Management*, *39*(2), 85-100.

Aarts, E., Aarts, E. H., & Lenstra, J. K. (Eds.). (2003). *Local search in combinatorial optimization*. Princeton University Press.

Caridi, M., & Cavalieri, S. (2004). Multi-agent systems in production planning and control: an overview. *Production Planning & Control*, *15*(2), 106-118.

Charkha, P. G., & Jaju, S. B. (2020). Decision Support System for Supply Chain Performance Measurement: Case of Textile Industry. In *New Paradigm of Industry 4.0* (pp. 99-131). Springer, Cham.

Cordeau, J. F., Laporte, G., & Pasin, F. (2008). Iterated tabu search for the car sequencing problem. *European Journal of Operational Research*, 191(3), 945-956.

Dahmen, S., Rekik, M., Soumis, F., & Desaulniers, G. (2020). A two-stage solution approach for personalized multi-department multi-day shift scheduling. *European Journal of Operational Research*, *280*(3), 1051-1063.

Dorne, R., & Hao, J. K. (1999). Tabu search for graph coloring, T-colorings and set T-colorings. In *Meta-heuristics* (pp. 77-92). Springer, Boston, MA.

Du, W., Tang, Y., Leung, S. Y. S., Tong, L., Vasilakos, A. V., & Qian, F. (2017). Robust order scheduling in the fashion industry: a multi-objective optimization approach. *arXiv preprint arXiv:1702.00159*.

Fleischmann, B. (1994). The discrete lot-sizing and scheduling problem with sequence-dependent setup costs. *European Journal of Operational Research*, *75*(2), 395-404.

Gendreau, M., Hertz, A., & Laporte, G. (1994). A tabu search heuristic for the vehicle routing problem. *Management science*, *40*(10), 1276-1290.

Geoffrion, A. M., & Graves, G. W. (1976). Scheduling parallel production lines with changeover costs: Practical application of a quadratic assignment/LP approach. *Operations Research*, *24*(4), 595-610.

Glover, F. (1990). Artificial intelligence, heuristic frameworks and tabu search. *Managerial and Decision Economics*, *11*(5), 365-375.

Graham, R. L., Lawler, E. L., Lenstra, J. K., & Kan, A. R. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. In *Annals of discrete mathematics* (Vol. 5, pp. 287-326). Elsevier.

Helal, M. E., & Hosni, Y. A. (2003). A tabu search approach for the non-identical parallel-machines scheduling problem with sequence-dependent setup times. In *IIE Annual Conference. Proceedings* (p. 1). Institute of Industrial and Systems Engineers (IISE).

Helal, M., Rabadi, G., & Al-Salem, A. (2006). A tabu search algorithm to minimize the makespan for the unrelated parallel machines scheduling problem with setup times. *International Journal of Operations Research*, *3*(3), 182-192.

Hertz, A., & de Werra, D. (1987). Using tabu search techniques for graph coloring. *Computing*, *39*(4), 345-351.

Lee, J. H., Yu, J. M., & Lee, D. H. (2013). A tabu search algorithm for unrelated parallel machine scheduling with sequence-and machine-dependent setups: minimizing total tardiness. *The International Journal of Advanced Manufacturing Technology*, *69*(9-12), 2081-2089.

Liberatore, F., & Camacho-Collados, M. (2016). A comparison of local search methods for the multicriteria police districting problem on graph. *Mathematical Problems in Engineering*, *2016*.

Logendran, R., McDonell, B., & Smucker, B. (2007). Scheduling unrelated parallel machines with sequence-dependent setups. *Computers & Operations Research*, *34*(11), 3420-3438.

Løkketangen, A., & Glover, F. (1998). Solving zero-one mixed integer programming problems using tabu search. *European journal of operational research*, *106*(2-3), 624-658.

Mahasinghe, A., Hua, R., Dinneen, M. J., & Goyal, R. (2019). Solving the Hamiltonian cycle problem using a quantum computer. In *Proceedings of the Australasian Computer Science Week Multiconference* (pp. 1-9).

Metaxiotis, K. S., Askounis, D., & Psarras, J. (2002). Expert systems in production planning and scheduling: A state-of-the-art survey. *Journal of Intelligent Manufacturing*, *13*(4), 253-260.

Molina, J., Laguna, M., Martí, R., & Caballero, R. (2007). SSPMO: A scatter tabu search procedure for non-linear multiobjective optimization. *INFORMS Journal on Computing*, *19*(1), 91-100.

Osman, I. H. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of operations research*, *41*(4), 421-451.

Pinedo, M., & Hadavi, K. (1992). Scheduling: theory, algorithms and systems development. In *Operations Research Proceedings 1991* (pp. 35-42). Springer, Berlin, Heidelberg.

Renata, I., Halim, S., & Yahya, B. N. (2020). Solving a Real Problem in Plastic Industry: A Case in Trim-loss Problem. *KnE Life Sciences*, 119-127.

Russel, D. A., & Williams, G. G. (1977). History of Chemical Fertilizer Development 1. *Soil Science Society of America Journal*, *41*(2), 260-265.

Schrijver, A. (1998). *Theory of linear and integer programming*. John Wiley & Sons.

Song, C. H., Lee, K., & Lee, W. D. (2003, July). Extended simulated annealing for augmented TSP and multi-salesmen TSP. In *Proceedings of the International Joint Conference on Neural Networks, 2003.* (Vol. 3, pp. 2340-2343). IEEE.

Tanaev, V., Gordon, W., & Shafransky, Y. M. (2012). *Scheduling theory. Single-stage systems* (Vol. 284). Springer Science & Business Media.

Vallada, E., & Ruiz, R. (2011). A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times. *European Journal of Operational Research*, *211*(3), 612-622.

Zhang, X., Fang, Q., Zhang, T., Ma, W., Velthof, G. L., Hou, Y., & Zhang, F. (2020). Benefits and trade-offs of replacing synthetic fertilizers by animal manures in crop production in China: A meta-analysis. *Global Change Biology*, *26*(2), 888-900.