

A Scheduling and Rescheduling Decision Support System for Apparel Manufacturing

Ehsan Ardjmand, Ohio University, USA

William A. Young II, Ohio University, USA

Iman Ghalekhondabi, Our Lady of the Lake University, USA

Gary R. Weckman, Ohio University, USA

 <https://orcid.org/0000-0002-2445-4934>

ABSTRACT

The manufacturing environment for apparel is subject to a variety of constraints, stochasticity, and unforeseen events. In order to create an accurate scheduling system for this environment, these complexities must be considered. This article presents the development and the application of a scheduling and rescheduling decision support system for an apparel manufacturer. Furthermore, the results of applying the proposed system are presented and discussed. The scheduling and rescheduling decision support system presented in this article takes advantage of a variable neighborhood search and Monte Carlo simulation in order to minimize tardiness in the presence of different release times, sequence-based setup times, blocking, and resource constraints. The results show that the quality of the schedules generated by the proposed scheduling and rescheduling decision support system is superior to the current firm's scheduling practice, which is based on an earliest due date heuristic. Moreover, the percentage of the realized schedule and overall equipment effectiveness were improved.

KEYWORDS

Knowledge-Driven, Model-Driven, Monte Carlo Simulation, Variable Neighborhood Search

INTRODUCTION

In the apparel market, there are numerous types of products that have a very short life-cycle and this produces fluctuating demand (Mazziotti & Horne Jr, 1997; Mok, Cheung, Wong, Leung, & Fan, 2013). On the other hand, due to the manufacturing constraints, the lead time in the apparel industry is usually longer in comparison to other industries (Guo, Wong, Leung, Fan, & Chan, 2006; Şen, 2008; Taplin, 1997). Manufacturing in this type of environment requires systems that can quickly adapt to the dramatic changes within the market (Chuter, 1995; Şen, 2008). Moreover, within the US alone, there are approximately 30,000 companies that are actively manufacturing apparel (Statistics of US Businesses, 2013). Given this level of competition, it is crucial that organizations within this sector have efficient manufacturing planning systems.

Similar to other manufacturing systems, scheduling plays a vital role in apparel manufacturing. In fact, in order for an apparel manufacturing system to be agile and responsive to changes in the market, it is necessary to have flexible, reliable, and fast scheduling systems (Bruce, Daly, & Towers, 2004; Tomastik, Luh, & Liu, 1996). Thus, a system developed for this type of environment should

DOI: 10.4018/IJORIS.20211001.0a4

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

consider both scheduling and rescheduling aspects of product planning. This kind of dual-focused system is even more critical within the apparel industry where schedules frequently change due to the volatility of deadlines and demand.

An efficient scheduling system for apparel production should consider two sets of constraints. These constraints can be summarized as either general or apparel-specific scheduling constraints. General scheduling constraints are those that are common across various industries such as starting time, precedence, and set up dependency (Pinedo, 2005, 2015). Apparel specific constraints are mostly related to machine availability. The primary concept behind these constraints deals with pneumatic and steam generator systems. For example, sewing machines are usually connected to pneumatic systems that have a certain air generation capacity. Depending on the capacity, sometimes these systems might not be able to generate enough air to operate all of the sewing machines at once during high-load production periods. A similar constraint applies to the irons, which are connected to a central boiler that generates steam. If more than a certain number of irons work simultaneously, there is often not enough steam pressure for all the irons to function properly.

In this article, a scheduling and rescheduling decision support system (SRDSS) for apparel manufacturing is introduced. While considering general and apparel specific constraints, the proposed SRDSS minimizes the tardiness using a variable neighborhood search (VNS) coupled with Monte Carlo simulation.

The contribution of this paper is threefold. First, an SRDSS framework for apparel manufacturing is proposed. Second, a hybrid VNS-Monte Carlo heuristic algorithm for scheduling and rescheduling is introduced that considers different release times, sequence-based setup times, blocking, and other resource constraints. Third, the SRDSS is implemented in a real-world manufacturing plant and the associated results are analyzed and presented.

The remainder of this article is organized as follows. In section 2, the related literature is discussed. In section 3, the SRDSS architecture is introduced. In section 4, the scheduling/rescheduling method (a hybrid VNS-Monte Carlo heuristic) is explained. In section 5, a case study example is solved, and the corresponding results are presented. Finally, in section 6, the overall conclusion of this research is presented along with a discussion about future research considerations.

LITERATURE REVIEW

Similar to other industries, scheduling in apparel and textile manufacturing is addressed in many studies at different levels of the fashion supply chain (Ngai, Peng, Alexander, & Moon, 2014). There is a vast body of scholarly research in the literature on the application of different types of DSSs in scheduling (Ardjmand, Ghalehkhondabi, Weckman, & Young, 2016). One of the major distinguishing features of a DSS is its architecture. A DSS architecture is a determinant of its applicability and reusability under different conditions (Frantzén, 2013; Tolio, et al., 2010). There are many different DSS architectures for the production schedule proposed in the literature. For example, Son, Wysk, and Jones (2003) proposed an architecture for developing a simulation-based shop-floor control system that was applicable to both flow-shop and job-shop environments. Many other references using simulation-based architectures can also be found in the literature (Frantzén, Ng, & Moore, 2011; Horn, Weigert, Schonig, & Thamm, 2006; L. Li, Yumin, & Kai, 2012).

Framinan and Ruiz (2010) proposed a modular production scheduling DSS architecture including user interface, schedule generator, database management, and business logic modules, which could interact with an external information system. One notable property of the architecture proposed by Framinan and Ruiz (2010) was that their system was highly-modular which made it possible to apply the system to various industries and contexts. Frantzén (2013) introduced a web-based scheduling DSS named OPTIMISE (OPTIMisation using Intelligent Simulation and Experimentation) which took advantage of optimization and simulation. Modularity and the ability to connect to external stimulation and optimization systems were among the most notable features of this particular DSS.

Regardless of architecture, most DSSs that are designed for production scheduling can be categorized as knowledge-driven or model-driven DSSs. It is also common for a combination of both types of systems to be developed. Knowledge-driven DSSs take advantage of expert knowledge while model-driven DSSs apply mathematical models or algorithmic procedures to find solutions for scheduling problems (Power, 2002). In the following sections, the literature related to various scheduling decision support systems (DSSs) architectures and the application of model-driven and knowledge-driven DSSs in production schedules is summarized. While DSSs exist within a diverse set of industries, the following sections are focused on DSSs related to the apparel manufacturing industry.

Model-Driven DSSs

Stern and Avivi (1990) proposed an integer programming formulation for a model-driven DSS for the textile industry where they could choose the orders to be processed in the production line using two greedy algorithms for scheduling. The result of their tests using real-world data showed that their algorithm yielded a 4.5 percent deviation from maximum revenue.

Chen, Racine, and Swift (1992) applied a simulated annealing algorithm to the scheduling problem in apparel manufacturing and their method was suitable for medium-sized instances. Using the maximum tardiness as the objective function, Serafini and Speranza (1992) derived solutions for general cases of the problem using an LP-based heuristic.

Tomastik, et al. (1996) proposed an integer programming formulation for an apparel manufacturing scheduling and resource allocation problem where the workload in the production line is divided between cells and each cell requires a specific amount of resources. They solved the model using a Lagrangian relaxation approach and evaluated it on a real production site. Their solution method was able to generate schedules within 16-20% of optimal solutions.

H. Li, Li, Li, and Hu (2000) proposed a production rescheduling expert simulation system, which took advantage of simulation techniques, artificial neural networks, and various dispatching rules. Lee, Abernathy, and Ho (2000) modeled an apparel manufacturing system where there are two production lines having long and short lead times. They evaluated the problem in order to find the fraction of the production load that should be assigned to each line as well as the schedule that maximizes the overall profit. Wong, Chan, and Ip (2001) modeled a flow-shop problem in apparel manufacturing along with master production scheduling that considered the minimization of the cost of tardiness as the objective function.

Karumanasseri and AbouRizk (2002) introduced a DSS for planning and scheduling of steel fabrication projects, which was based on an order ranking heuristic. Cowling (2003) described a commercial DSS which provided semi-automatic scheduling solutions for steel hot rolling mills. They introduced a DSS that used a Tabu search for solving the scheduling problem. In the author's comparison of the commercial DSS and a manual scheduling system, they found significant scheduling improvement in terms of solution quality and processing time.

Kapanoglu and Miller (2004) proposed an evolutionary algorithm based DSS, which was able to memorize high-quality solutions, which in turn, could be used to seed their algorithm's parameters for future problem formulations. To evaluate their model, a set of artificially generated test problems was explored. Chryssolouris, Papakostas, and Mourtzis (2005) addressed the scheduling problem in a refinery, which included several operations starting from unloading of crude oil to storage tanks to the arrangement of the temperature cut points for each distillation unit. They used a simulation-based approach for solving the scheduling problem and evaluated the results in a real-world scenario.

Guo, et al. (2006) proposed a mathematical model of a non-preemptive job-shop schedule in the apparel assembly process for minimizing the total penalties of earliness and tardiness. They applied a genetic algorithm to solve this particular problem. Petrovic and Duenas (2006) proposed a fuzzy logic-based DSS for parallel scheduling problems in the presence of uncertain disruptions and applied it to a real scenario involving the manufacturing of pottery. They defined the disruption

as the number and duration of glaze shortages and found that their DSS performed well in terms of absorbing the effect of glaze shortages.

Rose and Shier (2007) presented an enumerative approach for finding optimal cutting schedules in apparel manufacturing while having the minimum number of lays. They showed that their method could solve standard test problems in the literature as well as some challenging examples proposed by a global apparel manufacturer. Jeong, Leon, and Villalobos (2007) described an integrated DSS for generating a maintenance schedule, which was capable of diagnosing the fault and advising proper maintenance tasks by using a stochastic sampling method. The authors evaluated the DSS for single pick-and-place machines used in electronic assembly systems.

Hsu, Hsiung, Chen, and Wu (2009) applied a genetic algorithm to a multi-stage yarn-dyed textile manufacturing where the scheduling problem has sequence-dependent setup times and hierarchical product structure. They modeled the problem using mixed-integer programming and reduced the computational complexity of the model by breaking it into several sub-problems. Mahdavi, Shirazi, and Solimanpur (2010) proposed a simulation-based DSS for the production control of a stochastic flexible job-shop and then evaluated the DSS on three stochastic and deterministic test cases. In their research, they found that the DSS was suitable for controlling stochastic flexible shops that have a large number of machines. Krishna, Mahesh, Dulluri, and Rao (2010) designed an online scheduling DSS for a high mix-manufacturing firm. The DSS included two phases, which included the development of a scheduling DSS and the integration with the World Wide Web.

Geiger (2011) reviewed the application of a DSS that uses a metaheuristic based on variable neighborhood search on permutation flow-shop systems. Through testing on several multi-objective test problems, the authors showed that the DSS has the advantage of obtaining high-quality solutions while being straightforward to implement and tune.

Montoya-Torres and Vargas-Nieto (2012) addressed the flow-shop problem in apparel manufacturing using a genetic algorithm. In their study, the objective function was the minimization of makespan and the number of orders with late delivery. Huang and Yu (2013) solved the cutting and sewing scheduling problem using an ant colony optimization algorithm while minimizing earliness, tardiness, and makespan.

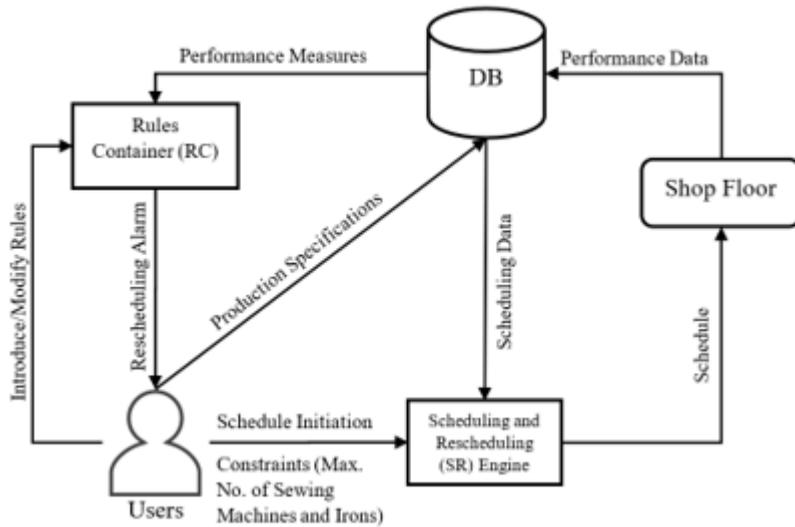
Ardjmand (2015) considered the scheduling problem in apparel manufacturing along with planning, pricing, and inventory decisions. The author proposed an integrated decision support system for apparel production that generated near-optimal schedules based on the data received from the production line and interaction with manufacturing experts. Along with scheduling, many other researchers have also addressed the production planning in apparel manufacturing as well (Ait-Alla, Teucke, Lütjen, Beheshti-Kashi, & Karimi, 2014; Ardjmand, Weckman, Young, Sanei Bajgiran, & Aminpour, 2016; Das & Patnaik, 2015; Mak, 2016; Thomassey, 2014).

Knowledge-Driven DSSs

Compared to model-driven DSSs, knowledge-driven DSSs are less represented in literature. With that said, a few examples found in the literature are summarized in this section. For example, Henning and Cerdá (2000) proposed a knowledge-based DSS framework based on object-oriented technology for production scheduling. The authors implemented the DSS in several plants and observed that the DSS has helped to reduce the number of extra shifts during weekends and work-in-process. Bistline, Banerjee, and Banerjee (1998) designed an interactive DSS for dynamic job-shop scheduling which was capable of responding to machine downtime.

Yamaha, Matsumoto, and Tomita (2008) proposed a method for attaining scheduling rules in a decentralized production environment. The authors built their method based on their previous observations of behavioral modeling of scheduling activities and reinforcement learning. Novas and Henning (2009) introduced a reactive scheduling approach based on domain-knowledge and constraint programming, which was able to advise a new schedule when a disruptive event occurs. Makarouni, Psarras, and Siskos (2015) developed an interactive DSS for helping decision-makers explore the

Figure 1. Proposed SRDSS framework



tradeoffs between solutions found by integer programming. The authors evaluated their method on a large-scale example and found that it was useful in aiding the decision-making process of finding proper solutions when considering computational times.

Although the scheduling problem in apparel manufacturing is well-studied by many researchers, implementing solutions within a real-world setting is still an arduous task and most methods found in literature can be very restrictive or utilize assumptions that do not adequately represent the challenges faced by apparel manufacturers. The goal of this paper is to propose a scheduling and rescheduling method for apparel manufacturing that expands on the limitations of other methods found in literature and apply the methodology to a real-world setting. In this regard, the proposed method will attempt to avoid performance declines by minimizing the tardiness while considering different release times, sequence-based setup times, blocking, and other resource constraints. Moreover, the processing times are considered to be stochastic which is the case in many real scenarios. To the best of authors' knowledge, the summation of all these characteristics is not well-addressed in the literature of apparel manufacturing, which is viewed as the primary contribution of the work presented in this article. With that said, creating a model that addresses these constraints within a production setting presents its own complexities. Hence, this paper aims to consider these constraints by proposing a DSS based on variable neighborhood search and Monte Carlo simulation.

SCHEDULING AND RESCHEDULING DECISION SUPPORT SYSTEM (SRDSS)

In this section, the architecture of the proposed SRDSS and its components will be discussed. Figure 1 depicts the SRDSS framework and its relationship with the shop-floor and user(s).

The SRDSS has three parts that continuously interact with users and the shop-floor. The first part is a database that stores the production specifications and performance data. Production specifications data include parameters such as timeline and working hours, cycle times, machines, operators, setup times and all other necessary information for specifying the production system. This information is entered into the system by the users. In addition to specifying the shop-floor parameters, it is necessary to record the performance of the shop-floor in order to measure the deviations from the expected performance. The performance data will be gathered from the shop-floor and stored in the database. The data stored in the database will later be used for scheduling and rescheduling purposes.

The second part of the SRDSS will be referred to as the rules container (RC). The RC includes a set of rules, defined by the users that trigger an alarm whenever a particular rule is violated. By continuously querying the database, the RC measures the performance of the shop-floor and compares it against the rules defined by the users. Every time the RC determines that a rule has been violated, it sends an alarm to the users for rescheduling. Depending on the users' preferences, the RC can include several rules. One practical rule would be to initiate an alarm if the shop-floor falls behind the schedule for more than a specified amount of time.

The third component of the SRDSS is the scheduling and rescheduling (SR) engine. Initiated by the users, the SR engine will create a schedule. A user can initiate a scheduling or rescheduling task based on external factors such as plan changes or the alarms received from the RC. If the user the process while another schedule is being processed on the shop-floor, a rescheduling task is created. Otherwise, a scheduling task will be created. Moreover, the user will define the constraints for the SR engine that includes the maximum number of the sewing machines or irons that can operate simultaneously.

In the proposed SRDSS, the user will decide the breaking point for a job that needs to be stopped and the SR engine will reschedule considering the remainder of the job that has stopped as a separate job. For the purpose of scheduling and rescheduling, the SR engine uses a hybrid variable neighborhood search (VNS) and Monte Carlo simulation method that will be discussed in the proceeding section. The proposed SRDSS framework shares some similarities to the framework proposed by Framinan and Ruiz (2010). Both frameworks contain modules that manage an algorithm or search procedure and maintain necessary databases. Moreover, in both frameworks, the users can directly influence the schedule and trigger a rescheduling task. However, in this study in addition to a direct interaction, the users can interact with the system through the definition of a set of dynamic rules in the RC module.

SCHEDULING AND RESCHEDULING METHOD

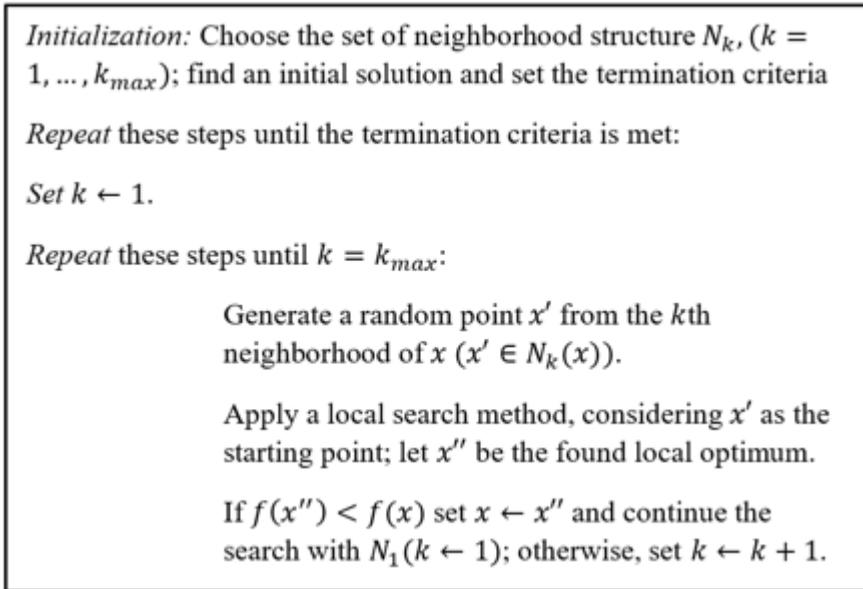
The SR engine takes advantage of a hybrid variable neighborhood search (VNS) and Monte Carlo simulation method for scheduling. VNS is a metaheuristic that operates based upon changing the neighborhood systematically (P. Hansen & Mladenović, 2014; Mladenović & Hansen, 1997). This particular method has been shown to be very effective in various scheduling contexts (Samà, Corman, & Pacciarelli, 2017; Sama, D'Ariano, Corman, & Pacciarelli, 2016). To explain VNS, consider an optimization problem of the form where s is a feasible solution, S is the feasible set, Ω is the solution space, and $N(s)$ is defined as the set of neighborhood structures. For a solution s , there are solutions that are its neighbor and can be reached by the i th neighborhood structure. s is a local minimum with respect to $N(s)$ if no solution can be found such that $f(s) > f(s')$ (P. Hansen & Mladenović, 2014).

In VNS, first, an initial solution s is chosen. Then, having $N(s)$, the neighbors of the generated solution are examined until a local optimum is reached. In the next step, $N(s)$ will be increased by one, and the same procedure is repeated until a local optimum is reached. This procedure continues until the termination criteria are met. Figure 2 shows the procedural flow of VNS.

As mentioned earlier, the SR engine's task is to find a feasible schedule that minimizes the tardiness. The SR engine does this in two steps, which can be summarized as the construction and evaluation steps. In the construction step, VNS is used when a sequence of jobs is created regardless of the constraints. At this level, the sequencing problem is reduced to a simple traveling-salesman problem. To address this problem, VNS is applied to the sequence of the jobs according to the method mentioned in Figure 2 and explained in detail in the study by P. Hansen and Mladenović (2014). Then, the constraints are applied in order to make the sequence feasible. Adding the constraints after a solution is formed is similar to the repair process in the genetic algorithm where an infeasible solution is modified to become feasible (Michalewicz & Nazhiyath, 1995).

In the evaluation step, Monte Carlo simulation is used in such a way where a random number is assigned to the duration of each activity in each job based on its statistical distribution and the

Figure 2. Steps of VNS (obtained from (P. Hansen & Mladenović, 2014))

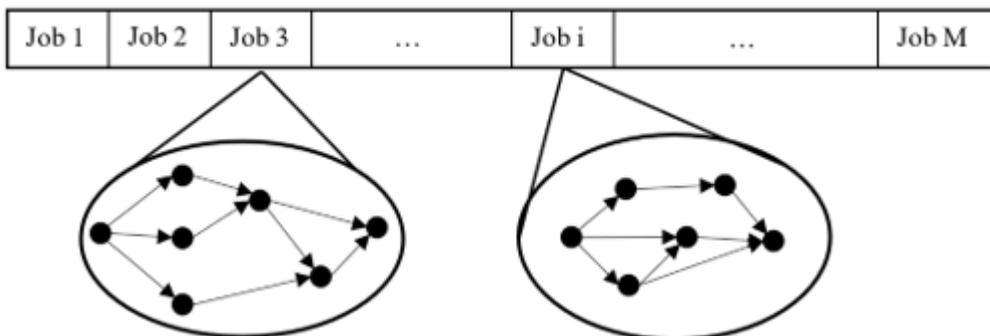


tardiness of the whole sequence is calculated. The evaluation step is performed for a pre-specified number of times and the average tardiness is considered as the tardiness of the sequence.

Figure 3 depicts the representation of a solution in the construction step where precedence constraints are independent of the sequence created by VNS. While a job can have any precedence requirements, VNS deals only with specifying the sequence of jobs without considering any constraints. In the next step, the jobs will be dispatched into the shop-floor according to their position in the sequence and constraints. Hence, if job 1 is placed before job 2 in the sequence, all its activities will be scheduled before job 2 activities.

Similar to precedence constraints, release time, sequence-based setup times, blocking, and resource constraints are imposed on the solution while every single job is scheduled. Considering the precedence constraints, a job can be represented as a network composed of activities that need to be processed on different machines (see Figure 3). From this perspective, scheduling a job is similar

Figure 3. Representation of a solution for scheduling. Each job can be represented by a precedence network composed of nodes (activities/tasks) and arcs. VNS only creates the sequence of the jobs and the constraints will be imposed on the sequence after it is formed.



to assigning resources to different activities in a project management problem. For a project to be completed on time, it is necessary to control the critical path's time and resources. For a product, based on its precedence requirements, the critical path is the set of all the activities that together form the longest path from the beginning to the end of production. Similar to the critical path method in project management, to schedule a job, first the activities on its critical path are scheduled.

Figure 4 shows the SR engine process, where it is shown that the SR first gets initiated by the user (also see Figure 1). Then, using VNS, several sequences of jobs are created and scheduled. Each job is composed of several activities that combined together, form a network (see Figure 3) that has a critical path. VNS determines the sequence of jobs to be dispatched into the shop-floor. In order to schedule a job, the priority will be given to the critical activities and their precedent tasks.

When an activity is scheduled, the activity is placed in the schedule in such a way that all the constraints are satisfied. For this purpose, an activity is placed in the timeline so that it starts after the release time, does not overlap with any blocked time on the timeline, is placed after the required setup-time (depending on the previous activity), and there are enough resources (operators, sewing machines and irons) to process the activity. Moreover, in order to finish the activity as soon as possible, it is scheduled for the first available machine where all the constraints are satisfied. Note that for scheduling, the average time for an activity is considered. However, for evaluation of tardiness, a distribution of times is used.

An Example

In this section, a small numerical example will be solved in order to show how the SR engine operates. First, assume that there are three jobs to be scheduled where the release times and deadlines of jobs 1, 2, and 3 are 0, 0, and 2, respectively. In this scenario, the deadline for all the jobs is 25. Figure 5 depicts the scenario of the three jobs as a network where the associated activities are represented by circles. For this example, the processing times of activities are assumed to have a uniform distribution, which is shown in the figure by each activity. In the figure, the green arrows represent the critical path in each network, when the average processing times are considered. Furthermore, the black circle in each network represents a dummy start activity for each job with zero processing time.

Setup time is dependent on the sequence of jobs. Table 1 shows the setup times that will be used in this example. Setup time is the time associated with the situation when a job is changed from one job to another on a particular machine. To simplify the example, the setup times are assumed to be constant and are only used if a job is changed on a machine. To continue this example, consider that there are four operators, four sewing machines, four irons available, and each machine in this scenario is controlled by a single operator. Finally, assume that a maximum of three sewing machines and three irons can work together at the same time and that the time between 9 to 11 is blocked.

To schedule the jobs and following the SR engine process depicted in Figure 4, the first step is construction. In this step first a neighborhood structure is defined. Let us assume that . i.e. only the first type of the neighborhood structures will be explored. Then, an initial solution such as will be created. According to , job 1 will be the first job for scheduling. To schedule job 1, critical path activities should be given priority. These activities are 12, 13, 15 and 16. Note that at this step, only the average processing times are used. Prioritizing the critical path activities, Figure 6 shows the schedule for sequence . If the precedent activities of a critical activity are not scheduled, they need to be scheduled first (see Figure 4). For example, activity 11 needs to be scheduled first in order to schedule activity 13. Note that the activity of a job will be scheduled in a way to satisfy all the constraints and be finished as soon as possible. For instance, although scheduling activity 15 between time 9 to 12 could decrease the finishing time of job 1 by 2 since the time between 9 and 11 is blocked, it could not be scheduled there. Hence, the earliest time for scheduling activity 15 would be 11. In other words, given a schedule status, an activity will be scheduled in the earliest time slot where no constraint is violated. Remember that the priority of activity for scheduling is determined by its relation to the critical path of its job precedence network.

Figure 4. SR engine process

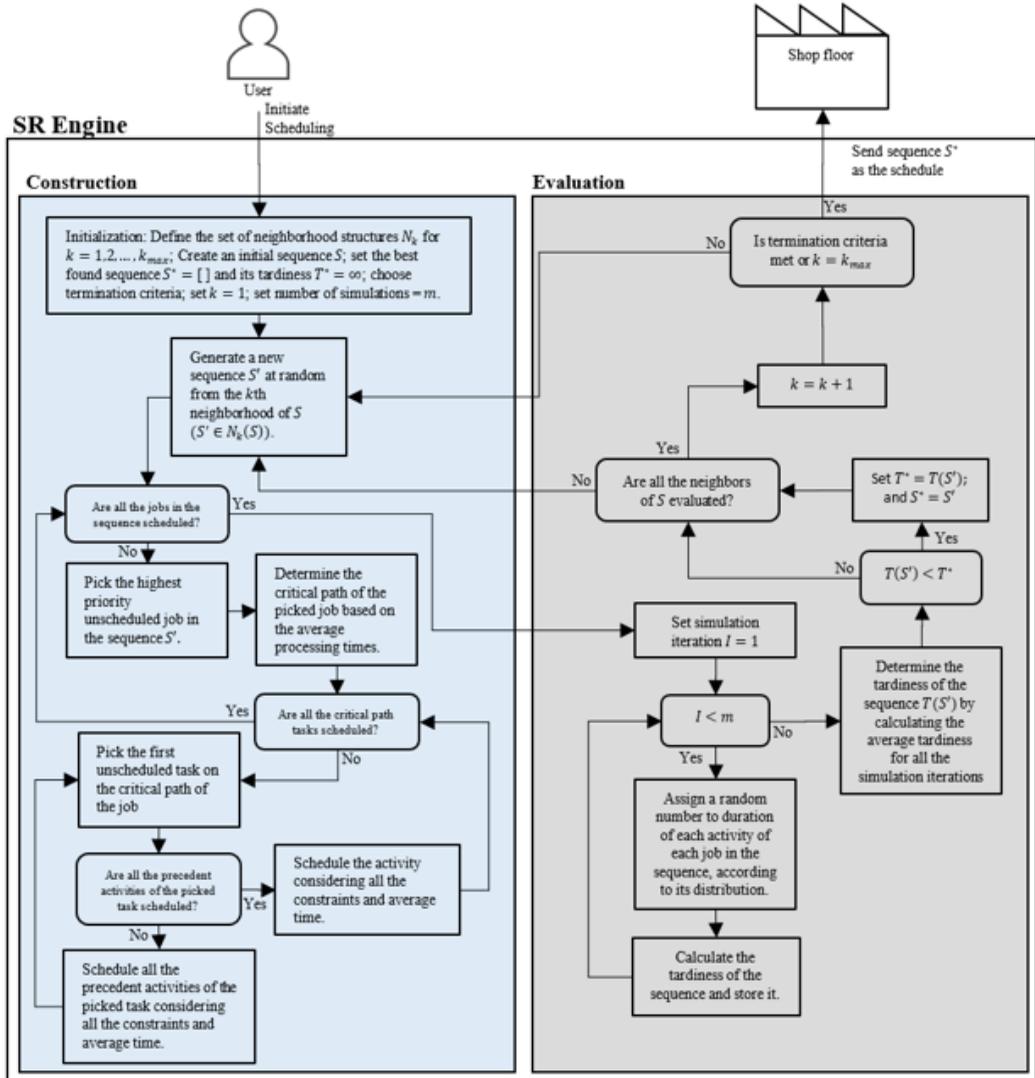


Figure 5. Jobs and their network of activities. Each circle and arrow show activity and a precedence constraint, respectively. Blue activities need sewing machines while red ones need iron. The processing times have uniform distribution and are shown above each activity. The critical paths in the case of average processing times are shown by green arrows

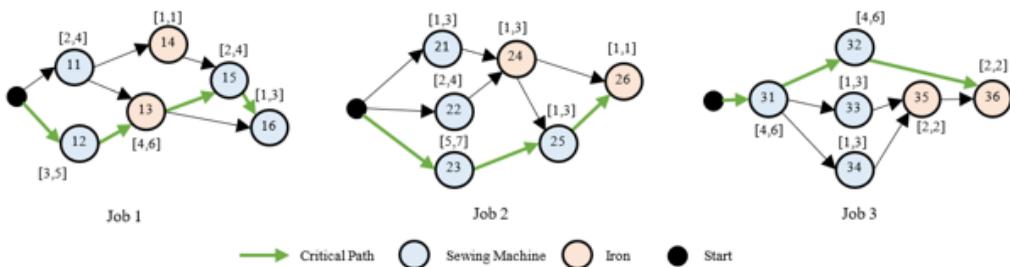


Table 1. Setup times

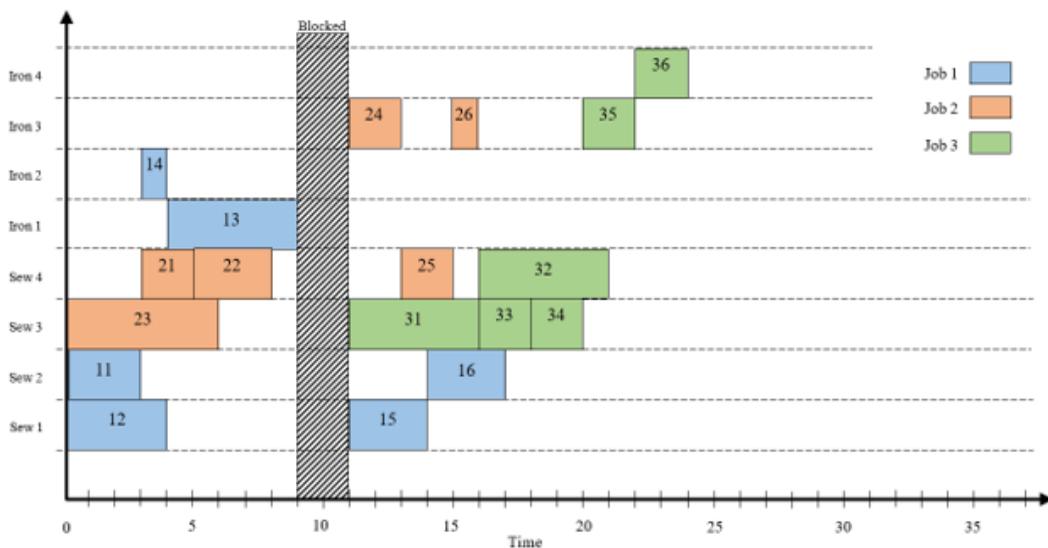
	Job 1	Job 2	Job3
Job 1	-	1	2
Job 2	1	-	1
Job 3	2	1	-

Another example of constraint satisfaction can be found in activity 21. Although this activity could be scheduled from time 0, it had to be delayed until time 3 because the maximum three sewing machines can work simultaneously. All the activities are scheduled to make sure that all the jobs are started after the release time and there is enough time for setup. Human resource and iron machine constraints are also taken into consideration in Figure 6. As it can be observed in Figure 6, the tardiness considering the average processing times is 0. However, this might not hold if the processing times are stochastic.

In the evaluation step and in order to assess the tardiness considering stochastic processing times, the schedule of Figure 6 will be simulated using the processing time distributions (see Figure 5). For this purpose, in each simulation iteration, a random time will be assigned to the activities and considering all the constraints, the schedule will be modified. The average tardiness of all simulation steps will be considered as the tardiness of the solution. One interesting scenario in the valuation step would occur if assigning a new processing time to an activity make the schedule infeasible. For instance, in Figure 6, if the processing time of activity 13 increases, then it will overlap with the blocked time and the schedule becomes infeasible. In this case, since activity cannot be broken, activity 13 will be scheduled after the blocked time.

After the evaluation step, considering the neighborhood structure, a new job sequence will be constructed. If for example, the new sequence is , then job 2 will be the first job to be scheduled. Following the example above, after the activates are scheduled (considering all constraints and prioritizing the critical path activities), the tardiness will be evaluated by simulating the schedule

Figure 6. Scheduling sequence considering all the constraints. Tardiness is 0



that is obtained using average processing times and time distributions. The process of construction and evaluation will continue until the termination criteria are met. Then, the best-found sequence will be sent to the shop-floor as the schedule.

CASE STUDY

In this chapter, the results of implementing the proposed SRDSS in an apparel manufacturing plant are explained. The manufacturing site chosen as the testing ground is in the women's clothes market and has five production classes, including jackets, raincoats, winter coats, trousers, and shirts. For each production class, a separate line is created. Each class has several designs for each year and new designs are introduced to the shop-floor each season. Some of the classes, such as winter coats and raincoats, are produced in a limited time during the year and have highly seasonal demand.

In this research, the winter coat line is chosen as a pilot for implementing the SRDSS. The planning period starts from November 1st and consists of four periods. Each period is considered to be one month, and hence, the planning horizon will be ending on March 1st. The reason for choosing this time interval is because the company introduces its new winter coats into the market around the end of November and the demand starts to decline sharply at the end of February. As the demand decreases, the company has to reduce its prices in order to sell more winter coats. The coats that remain unsold are not kept for the next year and are sold at a salvage price. Hence, it is very important to have a schedule that minimizes the tardiness and meet the market demand before the sales season is finished.

A winter coat consists of a "front", "back", "sleeve", "hem", "lining" and "collar". Each part of the coat is produced with a separate activity using one or more machines (sewing machines and irons). Hence, the production line has six activities that manufacture different parts of coats. In addition to these six activities, there are five more necessary activities called "support", "supplementary lining", "body assembly", "supplementary 1" and "supplementary 2". In support activity, the initial and small parts of a coat, such as pockets and belts, are produced. In body assembly, the main parts of the coats are attached together. In supplementary lining, the coats' lining is attached to the main body, while in supplementary 1 and 2, the final touches on producing a coat are done. These preparations mainly include ironing, covering, and packaging.

Depending on the design of the product, coats can have different precedence networks. For example, for some designs, it is necessary for the product to meet the 'lining' station before the 'collar' and for some, it might be the opposite. Figure 7 depicts a precedence network for one of the products of the case study manufacturing plant where a critical path is shown in green.

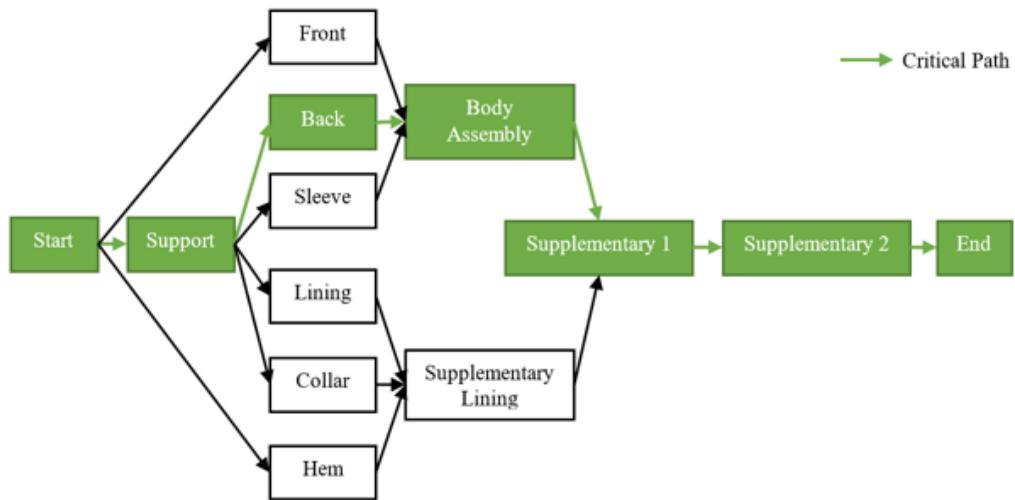
According to the scheduling method explained earlier (see Figure 4), to schedule a job, its critical path should be scheduled prior to other activities. Hence, while scheduling the job of the coat shown in Figure 7, 'back' activity needs to be given priority compared to 'front' or 'sleeve' activity and available operators, sewing machines and irons will be assigned to it first.

To assess the effectiveness of the proposed SRDSS, two sets of experiments have been done. The remaining of this section explains the numerical experiments conducted.

Experiment 1

In the first experiment, it is tried to evaluate the ability of SRDSS in finding 'good' feasible solutions. For this purpose, a sample of 20 schedules generated by the SR engine is selected randomly. Then using the same inputs and earliest due date (EDD) heuristic, another schedule is created. EDD is chosen for comparison because of its popularity and closeness to the way that scheduling has been done in the manufacturing site before implementing the SRDSS. Table 2 lists the tardiness of the solutions proposed by the SR engine and EDD as well as the run time in each case. As can be observed, SR engine solution outperforms EDD by 6.9% on average. However, EDD was much faster. The average SR engine run time was 101 seconds.

Figure 7. A precedence network for one of the case study winter coats. The critical path is shown in green.



Experiment 2

In the second experiment, the ability of SRDSS in rescheduling and avoiding the performance deviations in the production line was evaluated. For this purpose, two criteria, namely the percentage of the realized schedule and overall equipment effectiveness (OEE) were used. As the name suggests, the percentage of realized schedule measures the percentage of the total time that the shop-floor has been able to produce according to the schedule. OEE measures the percentage of the times that a machine has been working effectively (producing no defects and without stoppage) compared to its total available time (Dal, Tugwell, & Greatbanks, 2000; Godfrey, 2002; R. Hansen, 2001; Muchiri & Pintelon, 2008). For instance, if a sewing machine works effectively for 5 hours in a day, then its OEE would be $5/24 = 20.83\%$. Remember that the proposed SRDSS measures the percentage of the realized schedule and OEE and sends an alarm to the user if there is a deviation more than a pre-specified threshold.

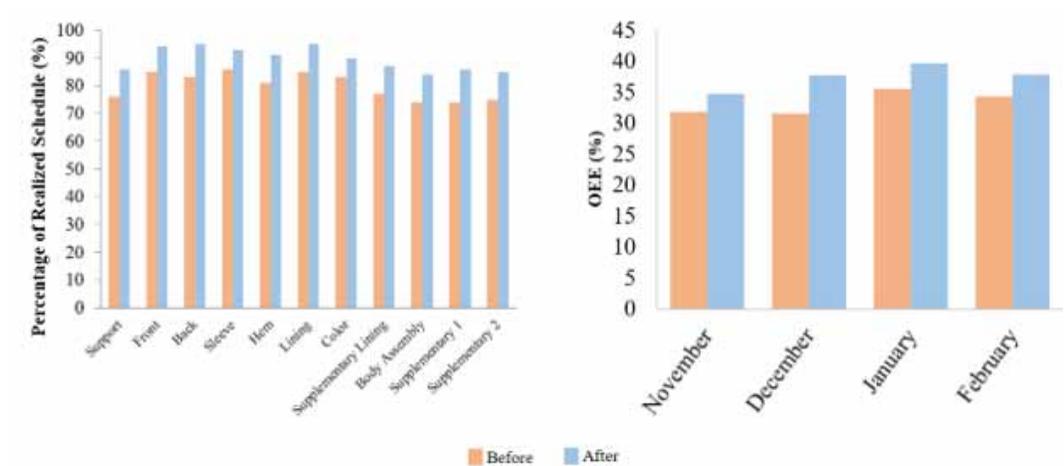
Figure 8 shows the percentage of realized schedule for each activity (aggregated for 4 periods) and OEE of all machines each month before and after SRDSS implementation at the same time of two consecutive years. As can be observed, after the SRDSS implementation OEE has increased in all of the periods (5% on average). However, the trend of OEE has remained almost the same. In both scenarios, the OEE increased until January but declined in February. The average realized schedule has increased by 10% after implementing the SRDSS. Although there is an increase in this indicator, the activities that used to have the lowest percentages before the implementation, have the lowest percentage after implementation as well. The improvement in the percentage of the realized schedule can be because of avoiding the deviations measured by RC and rescheduling when necessary. However, the OEE improvement seems to be due to reduced idle times and more balanced machines' workloads as a result of rescheduling.

Note that since the rules in the RC section of SRDSS and the thresholds are changed several times during the implementation, it is not possible to report a fixed rule or threshold in this section. However, the improved percentage of realized schedule and OEE show the efficiency of SRDSS in mitigating the performance decline by rescheduling.

Table 2. Comparison of schedules generated by SR engine and EDD heuristic

Schedule No.	Tardiness (Days)		Gap(%)	Run Time (s)	
	SR Engine	EDD		SR Engine	EDD
1	2.2	2.2	0.0	85	< 1
2	5.6	6.6	15.2	99	< 1
3	1.2	1.3	7.7	87	< 1
4	6.0	6.2	3.2	75	< 1
5	0.0	0.0	0.0	67	< 1
6	3.9	4.1	4.9	90	< 1
7	0.0	0.0	0.0	121	< 1
8	6.7	6.8	1.5	104	< 1
9	6.0	6.9	13.0	101	< 1
10	0.0	0.0	0.0	117	< 1
11	4.0	4.7	14.9	61	< 1
12	1.2	1.4	14.3	92	< 1
13	6.8	7.3	6.8	123	< 1
14	3.0	3.1	3.2	72	< 1
15	12.0	14.8	18.9	215	< 1
16	4.6	4.8	4.2	126	< 1
17	4.1	4.6	10.9	123	< 1
18	3.7	4.0	7.5	98	< 1
19	0.0	0.0	0.0	122	< 1
20	3.5	4.0	12.5	49	< 1

Figure 8. Percentage of realized schedule for each activity (aggregated for 4 periods) and OEE of all machines each month before and after SRDSS implementation at the same time of two consecutive years.



LIMITATIONS, CONCLUSION AND FUTURE RESEARCH

In this study, a scheduling and rescheduling decision support system (SRDSS) for apparel manufacturing is proposed. An apparel manufacturing environment is subject to a variety of constraints, stochasticity, and unforeseen events that might interrupt the production process and make the scheduling problem exceedingly difficult. The introduced SRDSS, as the implementation results suggest, was an effective tool for scheduling in such an environment.

The proposed SRDSS can handle minimizes the tardiness while considering two sets of constraints. The first set includes generic constraints that are common among a variety of industries such as release times, sequence-based setup times, blocking, and human resource constraints. The second set includes industry-specific constraints that are mostly related to machines and require that only up to a certain number of machines can work at the same time. To the best of authors' knowledge, the summation of all these characteristics is not addressed in the literature of apparel manufacturing.

The SRDSS has three parts, namely, a database, rule container (RC) and scheduling and rescheduling (SR) engine. The database stores production-specific data such as timeline and working hours, cycle times, machines, operators, setup times and all other necessary information for specifying the production system. RC contains user-defined rules and fires a rescheduling alarm if any of them are violated. Defining the constraints and initiated by the user, the SR engine generates a schedule.

For the purpose of scheduling and rescheduling, the SR engine uses a hybrid variable neighborhood search (VNS) and Monte Carlo simulation method. A schedule is generated by the SR engine in two steps, namely, construction and evaluation. In the construction step and using VNS, a sequence of jobs is formed. This sequence defines the order by which the jobs are dispatched into the production line. A job is composed of multiple activities that together form a precedence network. The network of a job defines the priority of the activities for scheduling. To schedule a job, using the critical path method (CPM) in project control, critical activities will be given higher priority.

After the sequence of the jobs is formed and all the activities of the jobs scheduled, in the evaluation step a Monte Carlo simulation is performed to calculate the average total tardiness of the sequence considering stochastic processing times of activities in each job. Construction and evaluation steps will be repeated until the termination criteria are met and the best solution found will be sent to the shop-floor for execution.

To evaluate the efficiency of the proposed SRDSS, it was implemented in an apparel manufacturing site and its performance was monitored over four months. To assess the scheduling ability, a sample of 20 schedules generated by the SR engine was selected randomly and compared against the schedule generated using the earliest due date (EDD) heuristic. EDD was chosen because of its similarity to the existing scheduling practice in the firm. The results showed that the SR engine solution outperforms EDD by 6.9% on average. However, EDD was observed to be much faster. The average SR engine run time was 101 seconds.

To assess the rescheduling ability percentage of the realized schedule and overall equipment effectiveness (OEE) were monitored in four months. It was observed that on average the percentage of the realized schedule and the OEE have improved 5% and 10% accordingly. This improvement was due to controlling the deviation from the schedule by rescheduling according to the rules in RC.

The contribution of this paper is threefold. First, an SRDSS framework for apparel manufacturing is proposed. Second, a hybrid VNS-Monte Carlo heuristic algorithm for scheduling and rescheduling is introduced. Third, the SRDSS is implemented in a manufacturing plant and the obtained results over a period of time are analyzed.

This study has two limitations. First, the monitoring period is limited to four months and only includes the winter season. The same monitoring process for other seasons might provide the authors with more insights. Second, due to confidentiality reasons, the sampling process and access to the site information was limited and it was possible to collect a limited number of samples.

As future research directions, this study can be extended in three domains. First, the rule container of the SRDSS can be modified to include fuzzy rules and define more complex types of rules. Combining the proposed scheduling problem with planning, pricing and inventory can be the second and a natural extension of this study. Adding planning, inventory control and pricing capabilities to the proposed SRDSS can coordinate the marketing, manufacturing, and inventory decision-making process.

The current VNS algorithm does not consider the constraints of the problem directly which can potentially cause the search process cannot explore some promising feasible areas. From a solution method perspective, improving the VNS in order to generate a feasible solution while incorporating the constraints directly, or using a different solving method with such a capability can be a promising third future research.

REFERENCES

- Ait-Alla, A., Teucke, M., Lütjen, M., Beheshti-Kashi, S., & Karimi, H. R. (2014). Robust production planning in the fashion apparel industry under demand uncertainty via conditional value at risk. *Mathematical Problems in Engineering*. doi:10.1155/2014/901861
- Ardjmand, E. (2015). *An Interactive Intelligent Decision Support System for Integration of Inventory, Planning, Scheduling, and Revenue Management*. Ohio University.
- Ardjmand, E., Ghalekhondabi, I., Weckman, G. R., & Young, W. A. (2016). Application of decision support systems in scheduling/planning of manufacturing/service systems: A critical review. *International Journal of Management and Decision Making*, 15, 248–276. doi:10.1504/IJMDM.2016.080703
- Ardjmand, E., Weckman, G. R., Young, W. A., Sanei Bajgiran, O., & Aminipour, B. (2016). A robust optimization model for production planning and pricing under demand uncertainty. *International Journal of Production Research*, 1–21.
- Bistline, W. G. Sr, Banerjee, S., & Banerjee, A. (1998). RTSS: An interactive decision support system for solving real-time scheduling problems considering customer and job priorities with schedule interruptions. *Computers & Operations Research*, 25(11), 981–995. doi:10.1016/S0305-0548(97)00092-0
- Bruce, M., Daly, L., & Towers, N. (2004). Lean or agile: A solution for supply chain management in the textiles and clothing industry? *International Journal of Operations & Production Management*, 24(2), 151–170. doi:10.1108/01443570410514867
- Chen, C., Racine, R., & Swift, F. (1992). A practical approach to the apparel production-planning and scheduling problem. *International Journal of Clothing Science and Technology*, 4(2/3), 9–17. doi:10.1108/eb002988
- Chryssolouris, G., Papakostas, N., & Mourtzis, D. (2005). Refinery short-term scheduling with tank farm, inventory and distillation management: An integrated simulation-based approach. *European Journal of Operational Research*, 166(3), 812–827. doi:10.1016/j.ejor.2004.03.046
- Chuter, A. (1995). *Introduction to clothing production management*. Blackwell science.
- Cowling, P. (2003). A flexible decision support system for steel hot rolling mill scheduling. *Computers & Industrial Engineering*, 45(2), 307–321. doi:10.1016/S0360-8352(03)00038-X
- Dal, B., Tugwell, P., & Greatbanks, R. (2000). Overall equipment effectiveness as a measure of operational improvement-A practical analysis. *International Journal of Operations & Production Management*, 20(12), 1488–1502. doi:10.1108/01443570010355750
- Das, S., & Patnaik, A. (2015). Production planning in the apparel industry. *Garment Manufacturing Technology*, 81.
- Framinan, J. M., & Ruiz, R. (2010). Architecture of manufacturing scheduling systems: Literature review and an integrated proposal. *European Journal of Operational Research*, 205(2), 237–246. doi:10.1016/j.ejor.2009.09.026
- Frantzen, M. (2013). *A real-time simulation-based optimisation environment for industrial scheduling*. Academic Press.
- Frantzen, M., Ng, A. H., & Moore, P. (2011). A simulation-based scheduling system for real-time optimization and decision making support. *Robotics and Computer-integrated Manufacturing*, 27(4), 696–705. doi:10.1016/j.rcim.2010.12.006
- Geiger, M. J. (2011). Decision support for multi-objective flow shop scheduling by the Pareto iterated local search methodology. *Computers & Industrial Engineering*, 61(3), 805–812. doi:10.1016/j.cie.2011.05.013
- Godfrey, P. (2002). Overall equipment effectiveness. *Manufacturing Engineer*, 81(3), 109–112. doi:10.1049/me:20020302
- Guo, Z., Wong, W. K., Leung, S., Fan, J., & Chan, S. (2006). Mathematical model and genetic optimization for the job shop scheduling problem in a mixed-and multi-product assembly environment: A case study based on the apparel industry. *Computers & Industrial Engineering*, 50(3), 202–219. doi:10.1016/j.cie.2006.03.003

- Hansen, P., & Mladenović, N. (2014). Variable neighborhood search. In *Search methodologies* (pp. 313–337). Springer. doi:10.1007/978-1-4614-6940-7_12
- Hansen, R. (2001). *Overall equipment effectiveness*. Industrial Press.
- Henning, G. P., & Cerdá, J. (2000). Knowledge-based predictive and reactive scheduling in industrial environments. *Computers & Chemical Engineering*, 24(9-10), 2315–2338. doi:10.1016/S0098-1354(00)00589-5
- Horn, S., Weigert, G., Schonig, P., & Thamm, G. (2006). Application of Simulation-Based Scheduling in a Semiconductor Backend Facility. In *Electronics Systemintegration Technology Conference, 2006* (Vol. 2, pp. 1122-1126): IEEE. doi:10.1109/ESTC.2006.280150
- Hsu, H.-M., Hsiung, Y., Chen, Y.-Z., & Wu, M.-C. (2009). A GA methodology for the scheduling of yarn-dyed textile production. *Expert Systems with Applications*, 36(10), 12095–12103. doi:10.1016/j.eswa.2009.04.075
- Huang, R.-H., & Yu, S.-C. (2013). Clarifying cutting and sewing processes with due windows using an effective ant colony optimization. *Mathematical Problems in Engineering*, 2013, 2013. doi:10.1155/2013/182598
- Jeong, I., Leon, V., & Villalobos, J. (2007). Integrated decision-support system for diagnosis, maintenance planning, and scheduling of manufacturing systems. *International Journal of Production Research*, 45(2), 267–285. doi:10.1080/00207540600678896
- Kapanoglu, M., & Miller, W. A. (2004). An evolutionary algorithm-based decision support system for managing flexible manufacturing. *Robotics and Computer-integrated Manufacturing*, 20(6), 529–539. doi:10.1016/j.rcim.2004.07.008
- Karumanasseri, G., & AbouRizk, S. (2002). Decision support system for scheduling steel fabrication projects. *Journal of Construction Engineering and Management*, 128(5), 392–399. doi:10.1061/(ASCE)0733-9364(2002)128:5(392)
- Krishna, L. S. R., Mahesh, V., Dulluri, S., & Rao, C. (2010). Implementation of an online scheduling support system in a high mix manufacturing firm. *International Journal of Engineering Science and Technology*, 2, 90–103.
- Lee, L. H., Abernathy, F. H., & Ho, Y.-C. (2000). Production scheduling for apparel manufacturing systems. *Production Planning and Control*, 11(3), 281–290. doi:10.1080/095372800232243
- Li, H., Li, Z., Li, L. X., & Hu, B. (2000). A production rescheduling expert simulation system. *European Journal of Operational Research*, 124(2), 283–293. doi:10.1016/S0377-2217(99)00381-1
- Li, L., Yumin, Q. F. M., & Kai, Y. (2012). Simulation-Based Modular Scheduling System of Semiconductor Manufacturing. In *Production Scheduling*. InTech. doi:10.5772/26257
- Mahdavi, I., Shirazi, B., & Solimanpur, M. (2010). Development of a simulation-based decision support system for controlling stochastic flexible job shop manufacturing systems. *Simulation Modelling Practice and Theory*, 18(6), 768–786. doi:10.1016/j.simpat.2010.01.015
- Mak, L. C. L. (2016). *Modeling of apparel production system and optimization of lot size scheduling in apparel industry*. The Hong Kong Polytechnic University.
- Makarouni, I., Psarras, J., & Siskos, E. (2015). Interactive bicriterion decision support for a large scale industrial scheduling system. *Annals of Operations Research*, 227(1), 45–61. doi:10.1007/s10479-013-1406-4
- Mazziotti, B. W., & Horne, R. E. Jr. (1997). Creating a flexible, simulation-based finite scheduling tool. In *Proceedings of the 29th conference on Winter simulation* (pp. 853-860). IEEE Computer Society. doi:10.1145/268437.268665
- Michalewicz, Z., & Nazhiyath, G. (1995). Genocop III: A co-evolutionary algorithm for numerical optimization problems with nonlinear constraints. In *Evolutionary Computation, 1995., IEEE International Conference on* (Vol. 2, pp. 647-651). IEEE. doi:10.1109/ICEC.1995.487460
- Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, 24(11), 1097–1100. doi:10.1016/S0305-0548(97)00031-2

- Mok, P., Cheung, T., Wong, W. K., Leung, S., & Fan, J. (2013). Intelligent production planning for complex garment manufacturing. *Journal of Intelligent Manufacturing*, 24(1), 133–145. doi:10.1007/s10845-011-0548-y
- Montoya-Torres, J. R., & Vargas-Nieto, F. (2012). Solving a bi-criteria hybrid flowshop scheduling problem occurring in apparel manufacturing. *Management Innovations for Intelligent Supply Chains*, 214.
- Muchiri, P., & Pintelon, L. (2008). Performance measurement using overall equipment effectiveness (OEE): Literature review and practical application discussion. *International Journal of Production Research*, 46(13), 3517–3535. doi:10.1080/00207540601142645
- Ngai, E., Peng, S., Alexander, P., & Moon, K. K. (2014). Decision support and intelligent systems in the textile and apparel supply chain: An academic review of research articles. *Expert Systems with Applications*, 41(1), 81–91. doi:10.1016/j.eswa.2013.07.013
- Novas, J. M., & Henning, G. P. (2009). A reactive scheduling approach based on domain-knowledge. *Computer-Aided Chemical Engineering*, 27, 765–770. doi:10.1016/S1570-7946(09)70348-7
- Petrovic, D., & Duenas, A. (2006). A fuzzy logic based production scheduling/rescheduling in the presence of uncertain disruptions. *Fuzzy Sets and Systems*, 157(16), 2273–2285. doi:10.1016/j.fss.2006.04.009
- Pinedo, M. (2005). *Planning and scheduling in manufacturing and services* (Vol. 24). Springer.
- Pinedo, M. (2015). *Scheduling*. Springer.
- Power, D. J. (2002). *Decision support systems: concepts and resources for managers*. Greenwood Publishing Group.
- Rose, D. M., & Shier, D. R. (2007). Cut scheduling in the apparel industry. *Computers & Operations Research*, 34(11), 3209–3228. doi:10.1016/j.cor.2005.12.001
- Samà, M., Corman, F., & Pacciarelli, D. (2017). A variable neighbourhood search for fast train scheduling and routing during disturbed railway traffic situations. *Computers & Operations Research*, 78, 480–499. doi:10.1016/j.cor.2016.02.008
- Sama, M., D'Ariano, A., Corman, F., & Pacciarelli, D. (2016). Metaheuristics for efficient aircraft scheduling and re-routing at busy terminal control areas. *Transportation Research Part C: Emerging Technologies*.
- Şen, A. (2008). The US fashion industry: A supply chain review. *International Journal of Production Economics*, 114(2), 571–593. doi:10.1016/j.ijpe.2007.05.022
- Serafini, P., & Speranza, M. G. (1992). Production scheduling problems in a textile industry. *European Journal of Operational Research*, 58(2), 173–190. doi:10.1016/0377-2217(92)90205-N
- Son, Y. J., Wysk, R. A., & Jones, A. T. (2003). Simulation-based shop floor control: Formal model, model generation and control interface. *IIE Transactions*, 35(1), 29–48. doi:10.1080/07408170304428
- Statistics of US Businesses. (2013). US Census Bureau.
- Stern, H. I., & Avivi, Z. (1990). The selection and scheduling of textile orders with due dates. *European Journal of Operational Research*, 44, 11–16.
- Taplin, I. M. (1997). *Backwards into the future: new technologies and old work organization in the US clothing industry. Rethinking global production*. Ashgate.
- Thomassey, S. (2014). A simulation based comparison: Manual and automatic distribution setup in a textile yarn rewinding unit of a yarn dyeing factory. *Simulation Modelling Practice and Theory*, 45, 80–90.
- Tolio, T., Ceglarek, D., ElMaraghy, H., Fischer, A., Hu, S., Laperriere, L., Newman, S. T., & Vánca, J. (2010). SPECIES—Co-evolution of products, processes and production systems. *CIRP Annals-Manufacturing Technology*, 59, 672–693.
- Tomastik, R. N., Luh, P. B., & Liu, G. (1996). Scheduling flexible manufacturing systems for apparel production. *IEEE Transactions on Robotics and Automation*, 12, 789–799.
- Wong, W., Chan, C., & Ip, W. (2001). A hybrid flowshop scheduling model for apparel manufacture. *International Journal of Clothing Science and Technology*, 13, 115–131.

Yamaha, H., Matsumoto, S., & Tomita, S. (2008). An attempt to obtain scheduling rules of network-based support system for decentralized scheduling of distributed production systems. In *Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference on* (pp. 506-511). IEEE.

Ehsan Ardjmand is an Assistant Professor of Analytics and Information Systems at Ohio University. His teaching and research interests include predictive and prescriptive modeling in business, machine learning, soft computing, decision support systems, and operations research. He earned his Ph.D. in Industrial and Systems Engineering from Ohio University. Dr. Ardjmand has more than eight years of experience in manufacturing and service industries in various operations management and analytics related positions. He has led several research projects related to business intelligence and data science and their applications in manufacturing, supply chain, social media, and stock markets, to name a few.

William A. Young II is a Charles G. O'Brien Associate Professor of Business Analytics in the Department of Analytics and Information Systems at Ohio University. Young earned his doctorate degree in Mechanical and Systems Engineering from Ohio University in 2010. William also received a bachelor's and master's degree in Electrical Engineering at Ohio University in 2002 and 2005, respectively. William has collaborated with multidisciplinary teams of faculty, students, and professionals on projects and programs. Young's primary research and teaching interests relate to operation management, healthcare services, and environmental systems, as well as specific interests in quantitative sports analysis, and educational technologies and techniques for innovative curriculum development and teaching instruction.

Iman Ghalekhondabi is an Assistant Professor of Management at the School of Business and Leadership, Our Lady of the Lake University. His research focuses on a range of topics, including supply chain management, pricing, game theory, and sustainability. Dr. Ghalekhondabi has published more than 25 peer-reviewed academic journal papers and conference proceedings. He served as the reviewer of various journals such as Cleaner Production, Intelligent Manufacturing, Utilities Policy, and Energies.

Gary Weckman is an Associate Professor in the Department of Industrial and Systems Engineering in the Russ College of Engineering and Technology at Ohio University. Dr. Weckman has an extensive research background in data analysis, with specific training and expertise in key research areas of nonlinear modeling. He has authored or co-authored more than 100 peer-reviewed articles in journals and conferences. Currently, he has been researching multidisciplinary applications utilizing knowledge extraction techniques with artificial neural networks (ANN). Dr. Weckman has experience in and possesses an internationally-renown reputation for neural network modeling of complex information systems, both traditional (e.g. applied applications of reliability analysis and telecommunications) and non-traditional (e.g. network modeling applications for financial and ecological monitoring). Prior to returning to university for an academic career, he has industrial engineering experience with over 12 years at such firms as General Electric Aircraft Engines.