# INITIAL RESEARCH ON SCHEDULE INTERFACE WITH SHOP FLOOR CONTROL SYSTEM

**Y. SON, J. VENKATESWARAN, K. YASEEN, and R. MOPIDEVI**
*Department of Systems and Industrial Engineering, University of Arizona, Tucson, AZ, USA 85721-0020*

## Abstract

This paper investigates integration of scheduling with a shop floor control system in general, and simulation-based shop floor control system in particular. Simplifications and assumptions made in traditional operations scheduling, such as disregarding material handling and buffers, have created questions concerning the fidelity of scheduling and implementing them in production systems. Since the simulation is used as an online task generator in a simulation-based control system, invalid schedules cause catastrophic results or system deadlocking (blocking) in the shop floor.

In this paper, the assumptions made in scheduling are investigated in order to determine if schedules can be actually implemented as intended. In addition, two prominent scheduling algorithms, Johnson's Algorithm and Jackson's Algorithm, are investigated to determine whether and in what conditions they work properly.

## Keywords
Shop floor control, simulation, integration, and schedule interface.

## 1 Introduction

Several methodologies have been proposed in the scheduling literature, including optimal seeking algorithms, mathematical programming, artificial intelligence (AI) based search techniques, rules or heuristics, and commercial finite capacity schedulers [1, 2, 3, 4, 5, 6]. Due to the many difficulties in scheduling, researchers have tended to simplify the scope of scheduling in order to make analysis more tractable. Characteristics of these traditional scheduling methods include:

- Only material processing activities are considered (disregarding material handling activities).
- They are mostly deterministic schedules since buffers are not considered.
- Performances of generated schedules are more optimistic than actual since less constraints are considered in scheduling. These constraints include constraints associated with material handling activities and constraints associated with deadlock and blocking.
- Systems are always empty and idle at time zero.
- No machine breakdown is considered.
- All jobs are available at time zero.

Little work has been conducted to justify whether these assumptions (simplifications) are valid and generated schedules are feasible or not. Because of the different levels of detail between the scheduler and the controller, it is difficult to implement schedules in the controller as intended. Different characteristics of schedules generated from different methods make the implementation more difficult.

The goal of this paper is to present the results of experiments conducted to investigate the effects of disregarding material handling and buffers in scheduling. Johnson's Algorithm [8] and Jackson's Algorithm [7] have been investigated.

A traditional operations-routing summary is shown in Table 1, and as can be seen it contains only material processing information. Therefore, no material handling resource requirements can be addressed. In this traditional scheduling approach, material handling activities are not planned apriori and not included in production control analysis. A more realistic operations routing summary is shown in Table 2. It contains both the material processing and material handling (MH) activities, and therefore, material handling activities can also be taken into account in operations scheduling. It should be noted in Table 2 that only the activity for material handling is called out. The specific MH resource is not specified.

## 2 Johnson's Algorithm

Johnson (1954) provided optimal solutions for the two-machine scheduling problem under several limiting assumptions [8]. Johnson posed a solution for problems where parts visit two machines in the same sequence (essentially a two-machine flow shop). Johnson's algorithm does not take into account material handling and deadlocking (blocking). In this section, we intend to investigate whether optimal sequences by Johnson's algorithm can be implemented as intended in the production line or simulation. In addition, we also intend to see whether the "optimal" sequences guarantee optimality after considering material handling and blocking effects.

An operations routing for a family of parts is shown in Table 3. According to Johnson's algorithm (1954), the optimal sequence is **P1–P3–P4–P2**. As shown in Fig. 1(a), the resultant make-span of the optimal sequence is 25. This schedule can be implemented only if material handling times are zero and infinite buffers exist so that blocking never occurs. In reality, material handling times are not zero, and buffers may or may not exist. Therefore, the schedule in Fig. 1(a) needs to be modified. If no buffers exist, blocking can occur. Fig. 1(b) illustrates the modified schedule of the schedule from Fig. 1(a) considering blocking interactions. The resultant make-span is 29. After considering blocking effects, the optimal schedule by Johnson's algorithm is no longer optimal. The actual optimal schedule is shown in Fig. 1(c).

Table 1. Traditional operations routing summaries

| Part # | Routing Sequence | Times Required |
|---|---|---|
| 1 | M1 – M2 – M3 | 3 – 2 – 1 |
| 2 | M2 – M1 | 4 – 2 |
| 3 | M3 – M2 | 5 – 1 |
| ... | ... | ... |
| n | M3 – M2 – M1 | 2 – 1 – 3 |

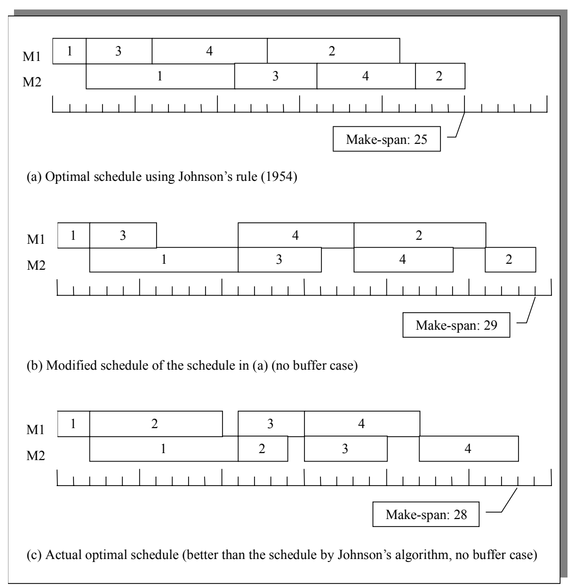
Table 2. Actual operations routing summaries ("t" is material handling time, and its value may vary as a function of the distance between handling origination and destination)

| Part # | Routing Sequence | Times Required |
|---|---|---|
| 1 | MH – M1 – MH – M2 – MH – M3 – MH | t – 3 – t – 2 – t – 1 – t |
| 2 | MH – M2 – MH – M1 – MH | t – 4 – t – 2 – t |
| 3 | MH – M3 – MH – M2 – MH | t – 5 – t – 1 – t |
| ... | ... | ... |
| n | MH – M3 – MH – M2 – MH – M1 – MH | t – 2 – t – 1 – t – 3 – t |

Table 3. Operations routing for a family of parts (M1 – M2)

| Part | P1 | P2 | P3 | P4 |
|---|---|---|---|---|
| M1 | 2 | 8 | 4 | 7 |
| M2 | 9 | 3 | 5 | 6 |

A small experiment was conducted to examine the effects of material handling on the schedule. Table 4 provides the make-span for different configurations. As discussed, the optimal schedule produced by Johnson's rule may not be the actual optimal schedule if no buffers exist. When buffers exist, results vary depending on the material handling time. In the extreme case where material handling time is zero, the resultant make-span agrees with the make-span by Johnson's algorithm. It can be seen that the longer the material handling times are, the less chances are that the schedule produced by Johnson's rule is the actual optimal schedule.



(a) Optimal schedule using Johnson's rule (1954)

(b) Modified schedule of the schedule in (a) (no buffer case)

(c) Actual optimal schedule (better than the schedule by Johnson's algorithm, no buffer case)

Fig. 1: Optimal schedule by Johnson's algorithm and real optimal schedule

Table 4. Make-span for different material handling times

| Sequence | Make-span | | | | | | |
|---|---|---|---|---|---|---|---|
| | Johnson's rule | T(MH) = 0 | | T(MH) = 0.5 | | T(MH) = 2 | |
| | | No buffer | Buffer | No buffer | Buffer | No buffer | Buffer |
| 1-2-3-4 | | *28 | 27 | *37 | 38 | *64 | *64 |
| 1-2-4-3 | | 29 | 28 | 38 | 37 | 65 | 65 |
| 1-3-2-4 | | 32 | 27 | 41 | 36 | 68 | 72 |
| 1-3-4-2 | *25 | 29 | *25 | 38 | *35 | 65 | 71 |
| 1-4-2-3 | | *28 | 26 | *37 | 38 | *64 | 76 |
| 1-4-3-2 | | *28 | *25 | *37 | *35 | *64 | 71 |
| 2-1-3-4 | | 33 | 31 | 42 | 42 | 70 | 76 |
| 2-1-4-3 | | 31 | 31 | 40 | 42 | 68 | 78 |
| 2-3-1-4 | | 32 | 32 | 41 | 41 | 68 | 76 |
| 2-3-4-1 | | 34 | 34 | 43 | 43 | 70 | 74 |
| 2-4-1-3 | | 35 | 35 | 44 | 44 | 71 | 79 |
| 2-4-3-1 | | 35 | 35 | 44 | 45 | 71 | 75 |
| 3-1-2-4 | | 31 | 27 | 40 | 36 | 67 | 75 |
| 3-1-4-2 | | 29 | 27 | 38 | 36 | 65 | 73 |
| 3-2-1-4 | | 30 | 30 | 39 | 41 | 67 | 75 |
| ........ | | | | | | | |
| 4-1-2-3 | | 31 | 30 | 40 | 39 | 67 | 72 |
| 4-1-3-2 | | 33 | 30 | 42 | 39 | 69 | 74 |
| 4-2-1-3 | | 32 | 32 | 41 | 43 | 69 | 77 |
| 4-2-3-1 | | 33 | 33 | 42 | 42 | 69 | 73 |
| 4-3-1-2 | | 30 | 30 | 39 | 40 | 66 | 77 |
| 4-3-2-1 | | 33 | 31 | 42 | 41 | 70 | 82 |

Another experiment has been conducted to come up with conditions where the optimal schedule by Johnson's algorithm is maintained as optimum while considering material handling and buffers. In particular, we intend to come up with the ratio between the average material handling times to average material processing times where Johnson's algorithm works favorably. To see the effect of the variance of processing times, the experiment is composed of two sub-experiments, each having different variance of material processing times. The performance measure used in the experiment is make-span. Operations routing summaries for two families of parts are given in Table 5. For these data, make-span has been collected for different material handling times, and results are shown in Table 6 and 7. As shown in Table 6, optimality by Johnson's algorithm for the first sub-experiment is maintained until material handling time reaches 1.1. Optimality by Johnson's algorithm for the second sub-experiment is maintained until average material handling time is 0.1. Table 8 summarizes the result of two sub-experiments. Note that average material handling times have been doubled because material handling activities are composed of pick and put operations. As shown in the table, as the variance of material processing times becomes smaller, the ratio of material handling times to material processing times also becomes smaller for optimality to be maintained. As future research, more complete experiment will be conducted for more general result regarding the relationship between the variance of material processing times and the ratio between material handling times to material processing times.

Table 5. Operations routing for two families of parts (M1 – M2)

| 1st sub-experiment (VAR T(MP) = 6) | | | | | 2nd sub-experiment (VAR T(MP) = 0.084) | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Part | P1 | P2 | P3 | P4 | Part | P1 | P2 | P3 | P4 |
| M1 | 2 | 8 | 4 | 7 | M1 | 3.2 | 3.1 | 3.7 | 3.5 |
| M2 | 9 | 3 | 5 | 6 | M2 | 3.4 | 3.8 | 3.3 | 3.9 |

Table 6. Make-span for 1st experiment (VAR T(MP) = 6 and Avg. T(MP) = 5.5)

| Sequence | Make-span | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Johnson's Rule | T(MH) = 0 | T(MH) = 0.5 | T(MH) = 1.0 | T(MH) = 1.1 | T(MH) = 1.2 | T(MH) = 1.5 | T(MH) = 2.0 |
| 1-2-3-4 | | 27 | 38 | 50 | ***48** | ***50** | ***55** | ***64** |
| 1-2-4-3 | | 28 | 37 | 49 | 51 | 53 | 59 | 65 |
| 1-3-2-4 | | 27 | 36 | 49 | 51 | 53 | 60 | 72 |
| 1-3-4-2 | ***25** | ***25** | ***35** | ***46** | ***48** | 51 | 58 | 71 |
| 1-4-2-3 | | 26 | 38 | 49 | 53 | 56 | 63 | 76 |
| 1-4-3-2 | | ***25** | ***35** | 47 | 50 | 52 | 59 | 71 |
| 2-1-3-4 | | 31 | 42 | 53 | 52 | 55 | 62 | 76 |
| 2-1-4-3 | | 31 | 42 | 53 | 55 | 57 | 64 | 78 |
| 2-3-1-4 | | 32 | 41 | 53 | 55 | 58 | 65 | 76 |
| ...... | | | | | | | | |

Table 7. Make-span for 2nd experiment (VAR T(MP) = 0.084 and Avg. T(MP) = 3.488)

| Sequence | Make-span | | | | |
|---|---|---|---|---|---|
| | Johnson's Rule | T(MH) = 0 | T(MH) = 0.1 | T(MH) = 0.2 | T(MH) = 0.5 |
| 1-2-3-4 | | 18.1 | 20.2 | 22.8 | 27.3 |
| 1-2-4-3 | | 18.1 | 20.2 | 22.6 | 27.1 |
| 1-3-2-4 | | 18.4 | 20.6 | 22.8 | 27.4 |
| 1-3-4-2 | | 18.6 | 20.4 | 22.2 | 28.6 |
| 1-4-2-3 | | 18.2 | 20 | 22.2 | 29.2 |
| 1-4-3-2 | | 18.2 | 20.4 | 21.8 | 27.2 |
| 2-1-3-4 | | ***18** | 20.1 | 22.5 | 30.3 |
| 2-1-4-3 | ***18** | ***18** | ***19.9** | 22.3 | 30.1 |
| 2-3-1-4 | | ***18** | 20.3 | ***21.7** | 27.1 |
| 2-3-4-1 | | 18.1 | 20.6 | 21.9 | 28.2 |
| 2-4-1-3 | | ***18** | 20.1 | 22.4 | 29.3 |
| 2-4-3-1 | | ***18** | 20.1 | 22.6 | ***27** |
| . | | | | | |

Table 8. Summary of results of two sub-experiments

| | 1st experiment | 2nd experiment |
|---|---|---|
| VAR T(MP) | 6 | 0.084 |
| Avg. T(MP) | 5.5 | 3.4875 |
| Avg. T(MH) that optimum by Johnson's algorithm maintained | 2*{(1.1+1.2)/2} = 2.3 | 2*{(0.1+0.2)/2} = 0.3 |
| Avg. T(MH) / Avg. T(MP) | 2.3/5.5 = **0.4181** | 0.3/3.4875 **= 0.086** |

## 3 Modified Johnson's Algorithm

In the previous sections, it has been shown that optimality using Johnson's algorithm is not always maintained when material handling is included. This was due to dynamic buffer interactions and material handling activities. Johnson's algorithm deals with systems with two machines. However, as described before, there is no such two-machine problem because of material handling.

In this section, a modified Johnson's algorithm is proposed. The modified algorithm says that we add material handling times (2t for machine 1 and 4t for machine 2, where t is the time taken to pick or put parts) to each processing time. This modified algorithm has been tested with the examples previously shown. For those examples, performance measures of the schedules from the proposed algorithm were as good as the ones generated by Johnson's algorithm. However, neither of the schedules was always optimal. This is because additional material handling times are involved for buffer interactions. Tests of the proposed algorithm for various configurations and involving buffer interactions are left for future research.

## 4 Jackson's Algorithm

Jackson (1956) provided optimal solutions for the two-machine scheduling problem under several limiting assumptions [7]. Jackson's algorithm developed the two-machine problem to include parts that can be processed at both machines in either an A – B sequence or a B – A sequence, or on a single machine (essentially a two-machine job shop). The algorithm does not take into account material handling and deadlocking (blocking). In this section, we investigate whether optimal sequences generated by Jackson's algorithm can be implemented as intended in the production line or simulation. In addition, we also intend to see whether those optimal sequences guarantee optimality after considering material handling and blocking effects.

An operations routing for multiple parts is shown in Table 9. According to the Jackson's algorithm, the optimal sequence is **P1–P2–P3** for machine 1 and **P3–P4–P1** for machine 2. If buffers do not exist, it is impossible to implement the schedule as specified by Jackson's algorithm. Even if buffers exist, several better schedules may exist including **P1–P2–P3** for machine 1 and **P1–P3–P4** for machine 2. It is obvious that it is more difficult to maintain optimality in the case of Jackson's algorithm than in the case of Johnson's algorithm. This is because of the part interactions (more deadlocking and blocking) associated with random part routing.

Table 9. Operations routing for multiple parts

| Part # | Sequence | Times |
|---|---|---|
| 1 | M1 – M2 | 5 – 1 |
| 2 | M1 | 4 |
| 3 | M2 – M1 | 3 – 4 |
| 4 | M2 | 2 |

## 5 Modeling Material Handling in Scheduling

Effects of disregarding material handling in scheduling have been discussed so far. Even though developing methodologies to involve those assumptions in scheduling is not major purpose of this paper, this section presents a method to incorporate material handling activities in scheduling, which is an initial effort.

In traditional scheduling, the following constraints are considered:

(a) Sequence constraints imposed on the process plans.

(b) Capacity constraints of resources.

The above two constraints are sufficient for scheduling only material processing activities. However, the following additional constraints are required to include the effects of material handling:

(c) A MH task (pick) of a part has to be followed by another MH task (put) of the same part, where another task cannot be inserted between the pick and put for the MH device being used.

(d) The amount of time for a part to occupy a machine depends upon both the processing time and the availability of the next processing or buffer resource needed after processing -- the next resource must be seized before the current resource can be freed.

The additional constraints described above cannot be implemented by simply adding the material handling tasks to the routing of the job. Instead, the following rules are needed:

(1) Rule 1: pick tasks appear explicitly in the routing summary of a job.

(2) Rule 2: put tasks (except the last put task to the unloading station) do not appear in the routing summary for a job. Instead, put tasks are modeled as a setup activity for the associated material processing tasks, so they are scheduled just before processing tasks. Since the last put task is not relevant to any of the processing tasks, it appears explicitly in the routing summary.

(3) Rule 3: after a pick task, the material handler must be held (remain in the busy state) until the processing task is scheduled to begin. Otherwise, constraint (c) may be violated.

(4) Rule 4: after a processing task, the material processor must be held (remain in the busy state) until the next pick task is scheduled to begin. Otherwise, constraint (d) may be violated.

A sample schedule is shown in Fig. 2 to illustrate the additional constraints (c and d) and the rules for material handling activities. The effect of holding the MP resource while the material handling is being performed is to lengthen the effective process time required.

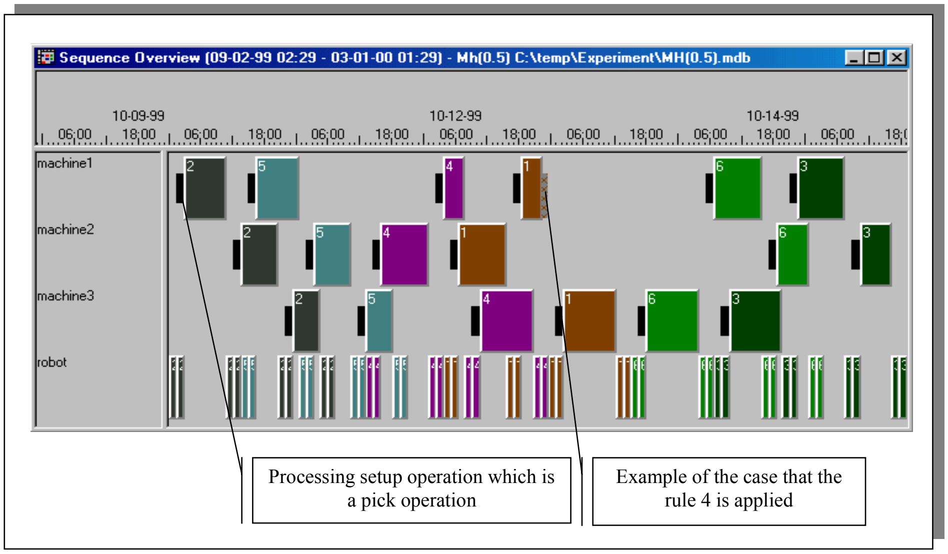



Fig. 2. A schedule with material handling activities

## 6 Conclusion

Execution interfaces with scheduling was investigated in this paper. The assumptions made in scheduling were investigated in order to determine if schedules can be actually implemented as intended. Scheduling algorithms, including Johnson's Algorithm (1954) and Jackson's Algorithm (1956), were investigated to determine whether and in what conditions the algorithms work properly.

In order to accomplish a "seamless" integration between a scheduler and a shop floor control system, either all the simplifications made in traditional scheduling must be removed or a control system must have certain intelligence to incorporate traditional schedules. Research has been conducted in both directions, and this paper presented part of the efforts, a method incorporating material handling activities in scheduling. A complete integration of scheduling with a shop floor control system will be presented in another paper.

## References


[1] E. H. L. Aarts, M. J. Frans, and E. H. A. Habers, Parallel Implementations of the Statistical Cooling Algorithm, The VLSI Journal, Vol. 4, No. 3, pp. 209-238, 1986.

[2] T. Aoki, S. Nakayama, M. Yamamoto, M. Hashimoto, and J. Tanaka, Combinatorial scheduler: simulation & optimization algorithm, Proceedings of the 1991 Winter Simulation Conference. pp. 280-288, 1991.

[3] H. Cho and R. A. Wysk, A Robust Adaptive Scheduler for an intelligent Workstation Controller, International Journal of Production Research, Vol. 31, No. 4, pp. 771-789, 1993.

[4] W.J. Davis and A.T. Jones, Real-Time Simulation and Production Scheduling Systems, NIST report NISTIR 89-4070, 1989.

[5] F. Glover, Future Paths for Integer Programming and Links to Artificial Intelligence, Computers and Operations Research, Vol. 13, No. 5, pp. 533-549, 1986.

[6] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, England, 1989.

[7] J.R. Jackson, Networks of Waiting Lines, Operations Research, Vol. 5, pp. 518-521, 1957.

[8] S.M. Jonhson, Optimal Two and Three Stage Production Scheduled with Setup Times Included, Naval Research Logistics Quarterly, Vol. 1, No. 1, pp. 61-68, 1954.