4th Construction Specialty Conference
*4e Conférence spécialisée sur la construction*

Montréal, Québec
May 29 to June 1, 2013 / *29 mai au 1 juin 2013*

# Interruptability Analysis for Linear Scheduling with Singularity Functions

Hong Xian Li[1], Xiang Yu Zhou[1], David Morley[1], Gunnar Lucko[2]
[1]Department of Civil and Environmental Engineering, University of Alberta
[2]Department of Civil Engineering, Catholic University of America

**Abstract:** It is a widely held assumption in the construction industry that activities, especially those that involve labor crews, should be scheduled to perform continuously until they are completed. However, in certain scenarios, depending on the production rates of their adjacent activities, a strategic intentional interruption of activities can create opportunities to improve the schedule by reducing the total duration of the project. This paper analyzes the possibility of interrupting activities with either constant or variable production rates toward the objective of minimizing the total project duration. For activities that exhibit a constant production rate, a heuristic approach determines the feasibility of interrupting activities based on scenario analysis and application of geometry to attain the objective. For activities that exhibit a variable production rate, a mathematical model is developed, extending the geometric approach with singularity functions. A genetic algorithm (GA) is applied in the computer implementation to achieve the objective. Using singularity functions allows modeling complex activities, including all changes in production rates and work breaks, with a single functional expression, greatly reducing the number of constraints that must be implemented for optimization. A complex schedule example is presented to illustrate how the new model can be applied to intentionally interrupt activities to gain a large reduction in total project duration.

## 1 Introduction

The Linear Scheduling Method (LSM) is a two-dimensional and visually-based method to analyze project schedules, wherein one dimension represents work units and the other represents time. LSM provides a convenient tool to express project progress, analyze resource utility, and perform optimization. Much research examined various aspects (Srisuwanrat and Ioannou 2007, Harris and Ioannou 1998, Russell and Caselton 1988). Most studies made the assumption of continuous activities that do not suffer from interruptions and/or that resources are utilized continuously. However, scheduling processes continuously does not guarantee an optimal solution for a repetitive construction project. Moreover, resource continuity cannot be guaranteed due to several factors: (1) Production rates of activities are not always identical, (2) location sequencing may vary among activities, (3) some activities are not performed in certain locations, and (4) logic constraints between activities can vary at different locations (Russell and Caselton 1988). Hegazy and Kamarah (2008) illustrated the principle of how interruptability can reduce the total project duration. Of course, not all activities can or should be interrupted. Long and Ohsato (2009) categorized activities into two types, interruptible and non-interruptible by using several criteria. Additionally, Long and Ohsato (2009) proposed a method to minimize the project duration. Assuming that a range existed for each activity duration and all activities were interruptible, a GA was used to minimize the project duration under the constraints of precedence relationships. The proposed methodology also could be applied to minimize only cost or a combination of cost and duration. Ipsilandis (2007) took into account project duration, resource idle time, unit completion time, and float in proposing a multi-objective linear program

model. The uniform model could be customized to handle different objectives, including project duration, work-break duration, unit completion time, total cost of work-break, project delay cost and work-break cost tradeoff. Note that most of the current research had been limited to only deterministic linear scheduling.

After considering variability in production rates of activities, Srisuwanrat and Ioannou (2007, p. 2153) addressed the "tradeoff between maintaining and relaxing resource continuity constraints … to maximize expected project profit." Their methodology was based on integrating probabilistic simulation with optimization, for which a completed unit algorithm and a GA were implemented. González et al. (2009) addressed the buffer utilization to overcome the production variability, and proposed a multi-objective analytic model that employed simulation toward Pareto optimality. By controlling the buffer between adjacent activities, project objectives of time, cost, and productivity were optimized. Srisuwanrat (2009) systematically examined linear scheduling by applying probabilities. Using probabilistic simulation, a sequence step algorithm was implemented to minimize the total project duration while maintaining a continuous resource utilization. Crew arrival time and crew idle time were scheduled using user-specified confidence levels. The probabilistic approach also allowed the tradeoff between relaxing the resource continuity constraint, project duration, and cost to be analyzed. Srisuwanrat (2009) also discussed the concept of work breaks, deliberate interruptions within a schedule that differ from unintentional idle time. Lucko (2008) introduced a novel way of applying singularity function to linear scheduling, the productivity scheduling method (PSM). Singularity functions originated from structural engineering and are applied in this article. The PSM provides a powerful mathematical model to flexibly model construction activities.

## 2 Research Objectives and Methodology

This paper analyses the interruptability of activities with constant and variable production rates (activities with linear progress or exhibiting a 'zigzag' shape) in the context of linear scheduling to minimize the total project duration and the number of interruptions and their sum. The following objectives will be delivered: (1) Classifying activities into interruptible and non-interruptible by addressing external factors and impacts on project duration for linear scheduling; (2) Generating rules for how and where interruptions should occur based on scenarios analysis; (3) Proposing solution methods for different scenarios and developing their corresponding algorithm; and (4) Applying singularity functions in order to model activity interruptions and to allow efficient optimization computations. Figure 1 represents the methodology of this research.

### 2.1 Interruptability of Activities

The classification criteria proposed by Long (2009, p. 502) are adopted here: "If an activity is outsourced, or relied on external resources," or "is performed by a big crew," or incurs a high idle cost, then such activity should be considered as being non-interruptible; else it can be generally considered interruptible.

### 2.2 Interruptability Analysis of Activities with Constant Production Rates

If an activity has a constant production rate, the activity is linear within the graphical LSM. There are two distinct concepts in LSM, the controlling sequence and the critical path. Both concepts have different meanings for repetitive projects. The controlling sequence (Harris and Ioannou 1998) is a continuous path from project start time to finish time, and is determined based on resource continuity. Delaying activities that are part of the controlling sequence will break the resource continuity, but *may not* delay the project. The critical path however, determines the project duration, and any delay of critical activities on it will delay the project. There may not exist a continuous critical path for a repetitive project that encompasses full activities. The four types of activities are defined per Table 1 and illustrated in the following figures.

**Table 1: Activity Type Matrix**

| Sequence | Critical activity | Non-critical activity |
|---|---|---|
| Controlling | Blue | Yellow |
| Non-controlling | Red | Green |

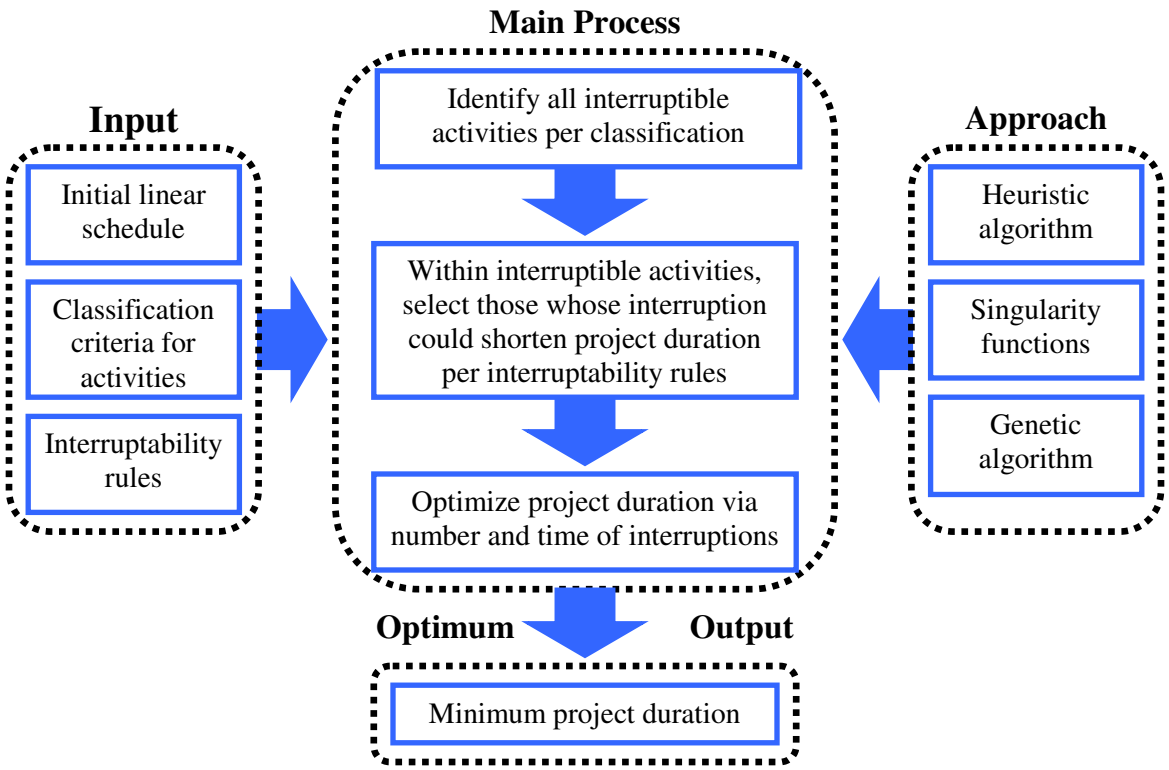
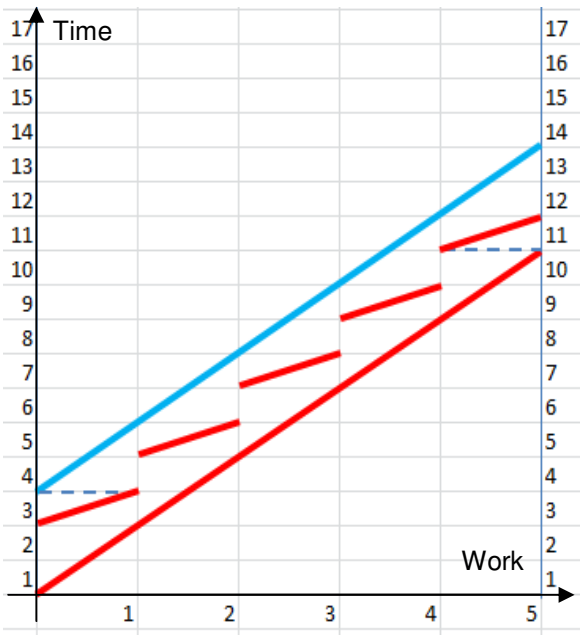



**Figure 1: Methodology**

For linear activities a geometric approach to determine interruptability is applied. To measure the impact of interruption on project duration and generate a rule that allows for optimization, different scenarios are analyzed, each of which generate a 'hinge' configuration so that *B* should indeed be interrupted. Scenario 1 occurs if the slope (inverse of the production rate for vertical time axis) of activity *A* is equal to the slope of *C*, and both are greater than the slope of *B*, as Figure 2 shows. Its heuristic optimization process follows the following steps, where segment numbers indicate intervals on the horizontal work axis and buffers of one work unit are dashed lines: (1) Break *B* into segments and move the critical $B_1$ down to when $A_1$ finishes, then move $B_2$ to $B_4$ respectively; (2) move *C* to the finish of the segments of *B*. In this scenario, all activities and their segments become critical and the optimal number of interruptions is equal to the number of work units minus one. Scenario 2 occurs when the slope of *A* is greater than that of *C*, which is greater than that of *B*, as Figure 3 shows. Its heuristic has these steps: (1) Break *B* into segments and move them down to when the segments of *A* finish; (2) schedule *C* according to the last unit of *B*, which determines the minimum project duration; (3) minimize interruptions in *B* by checking segments $n - 1$ to 1. If it can be moved back up to join with its successor without violating the buffer, then move it to eliminate one interruption; otherwise keep it at the current position and examine the next one.

The aforementioned controlling sequence is related to resource continuity. For example, in Scenario 1 the controlling sequence – specifically allowing for interruptability – consists of segments $A_1$ to $A_5$, $B_4$ to $B_2$ (in inverse order due to the 'hinge' that forms between the three activities with a finish-to-finish relationship from *A* to *B* and a start-to-start relationship from *B* to *C*) and $C_1$ to $C_5$. If they are not delayed, resources, e.g. labor crews, of each activity can proceed continuously. Yet from the view of the total project duration, $B_2$ to $B_4$ (yellow) can be delayed without impacting the project duration. Criticality, on the other hand, is related to project duration. If $B_1$ (red) is delayed, the start of *C* will be delayed, which will result in a delay to the entire project. $B_1$ is non-controlling in Scenario 1, same as $B_5$ (green), but the latter is non-critical. The last activity, here *C* (blue), is always both critical and controlling and under the assumptions of this optimization will never be interrupted. While the controlling sequence is continuous throughout the linear schedule before the optimization process is applied, ultimately such a continuous controlling sequence in its traditional definition will not exist within the linear schedule once all interruptions have been inserted.

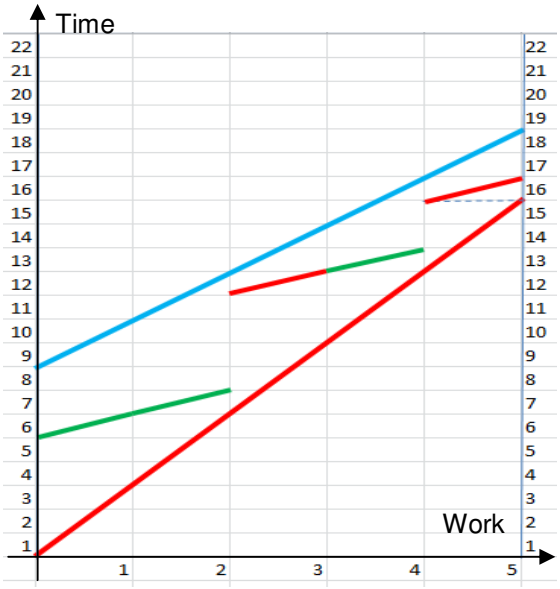

**2a: Before Interruption**




**2b: After Interruption**

**Figure 2: Scenario 1: Slope B < Slope A = Slope C**

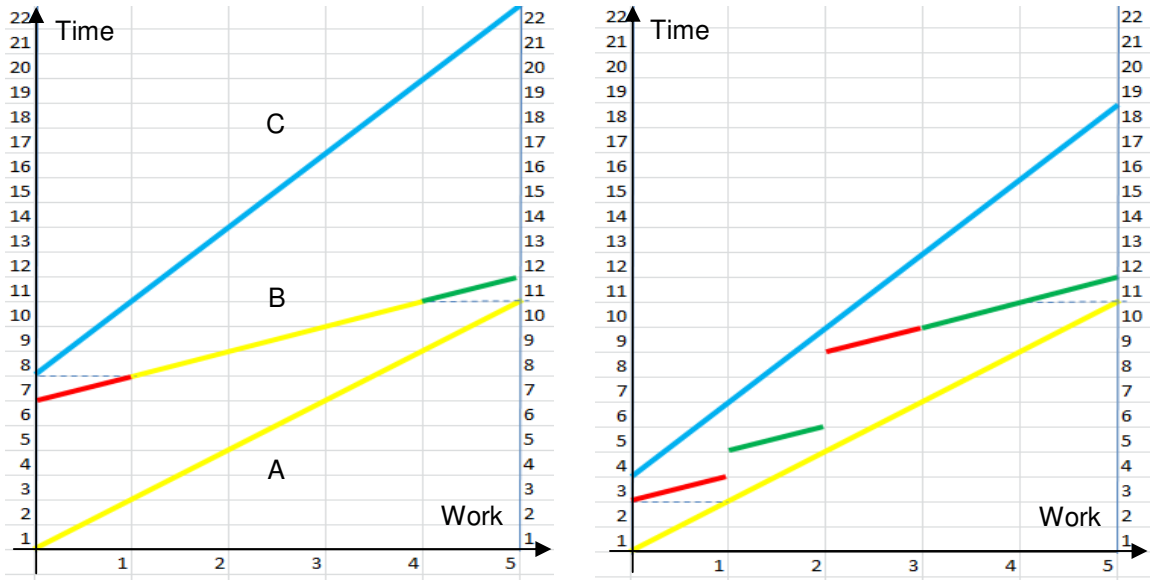

**3a: Before Interruption**




**3b: After Interruption**

**Figure 3: Scenario 2: Slope B < Slope C < Slope A**

Scenario 3 occurs when the slope of *C* is greater that of *A*, which is greater than that of *B*, as Figure 4 shows. Scenario 3 thus represents the inverse case of Scenario 2 during the optimization. Of course, the optimization toward the minimum number of interruptions will depend on the specifics of each scenario.




**4a: Before Interruption** **4b: After Interruption**

**Figure 4: Scenario 3: Slope B < Slope A < Slope C**

For this scenario, the optimization process has the following steps: (1) Break *B* into segments and move them down to when the segments of *A* finish; (2) schedule activity *C* according to the first unit of *B*, which determines the project duration; (3) minimize interruptions in *B* by checking segments *n* to 2. If it can be moved back up to join with its successor without violating the buffer, then move it to eliminate one interruption; otherwise keep it at the current position within the schedule and examine the next segment.

### 2.3 Interruptability Rule for Activities with Constant Production Rates

Based on the scenario analysis, the interruption rule for the activities with constant production rates can be formulated as follows: (1) The interruptability of activities can be optimized toward the minimum project duration and minimum number and duration of interruptions by analyzing triplets (predecessor, activity, and predecessor); (2) an activity should be interrupted *if and only if* its slope is smaller than that of both its predecessor and successor, i.e. its production rate is higher than its neighbours in the LSM diagram.

### 2.4 Heuristic Algorithm for Activities with Constant Production Rates

Using the aforementioned approach, a heuristic algorithm is created to perform the desired minimization of the project duration as follows: (1) Based on the initial linear schedule, check which activities could be interrupted per the rule above; (2) break interruptible activity into segments; (3) schedule its successor activity as early as possible; (4) optimize the interruptible activity by checking each segment. If it can be move back up to join with its successor, then move it; otherwise do nothing; (5) repeat for all interruptible activities sequentially in the schedule. The complete algorithm is presented as a flowchart in Figure 5.

### 2.5 Interruptability Analysis of Activities with Variable Production Rates

For activities with a variable production rate, the situation is more complex. The first rule for interrupting activities with varying production rates is that the activity is a candidate for interruption if and only if: (1) The start of an activity segment is after the finish of its predecessor segment – the *possibility condition*; (2) segments per (1) occur earlier (to the left) of the final critical point that is restricted by its predecessor – *the necessity condition*; and (3) the first and the last activities are not considered interruptible in terms of minimizing the project duration. Since the production rates are variable within activities themselves, the logic constraints should be examined not just at the activity level, but among adjacent individual segments during an optimization. Equation 1 provides the maximum number of constraints for adjacent segments.

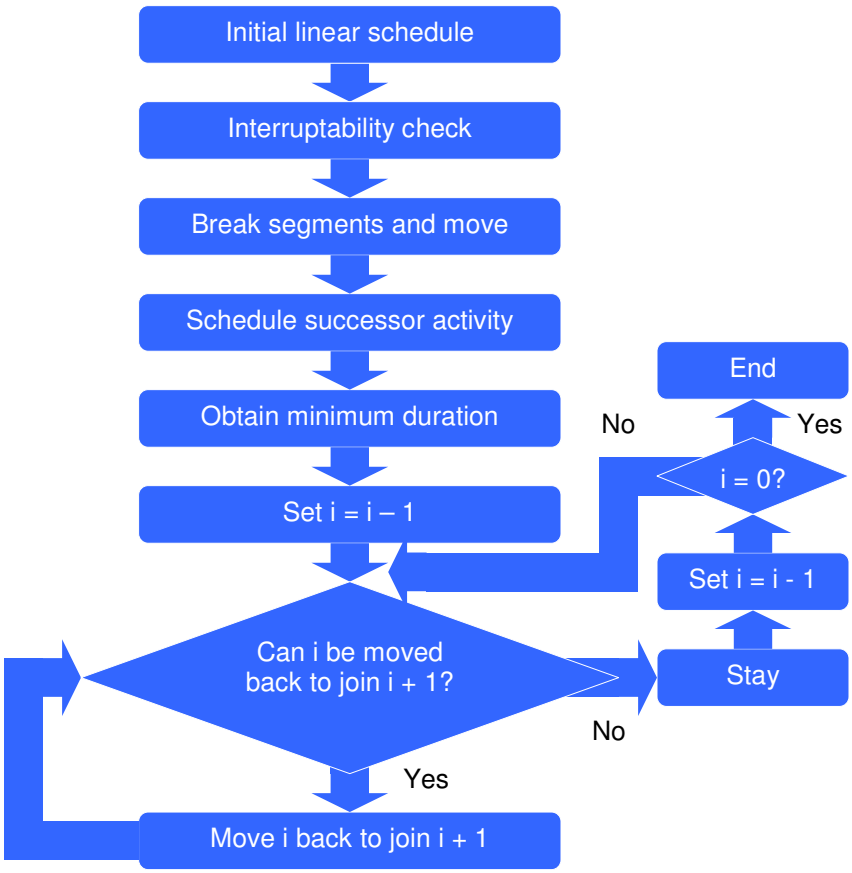



**Figure 5: Flowchart of Heuristic Algorithm**

$$[1] \qquad N_c = 2 \cdot (N_a - 2) \cdot (N_s - 1) + (N_a - 2) + N_s$$

Where $N_c$ is the number of constraints, $N_a$ is the number of activities, and $N_s$ is their number of segments. For example, for Scenario 1 with $N_a = 3$ activities with $N_s = 5$ segments each exist up to $N_c = 2 \cdot (3 - 2) \cdot (5 - 1) + (3 - 2) + 5 = 2 \cdot 1 \cdot 4 + 1 + 5 = 14$ links exist, i.e. constraints within the system to be optimized. The first eight links consist of four links within $B_1$ to $B_5$ plus four links between intermediate segments of *A* and *B*, the next one is to the start of *B*, and the last five are to and within the non-interruptible last activity.

### 2.6 Modeling Utilizing Singularity Functions for Activities with Variable Production Rates

Modeling using singularity functions provides a uniform and flexible mathematical model to incorporate the interruptions directly into LSM and allows easy computer implementation. The basic form of singularity functions is provided by Equation 2, which contains the activation cutoff *a* and the behaviour exponent *n*. This basic form allows customization to express a multiplicity of behaviours. Using singularity function to express activities with breaks within a linear schedule is demonstrated in the following schedule example.

$$[2] \qquad \langle x - a \rangle^n = \begin{cases} 0 & for\ x < a \\ (x - a)^n & for\ x \geq a \end{cases}$$

### 2.7 Genetic Algorithm

GA is a feasible method to this problem, which is performed with a population of 50, 25 generations, a stochastic uniform selection, crossover rate of 0.6, and mutation rate of 0.05 per the flowchart of Figure 6.

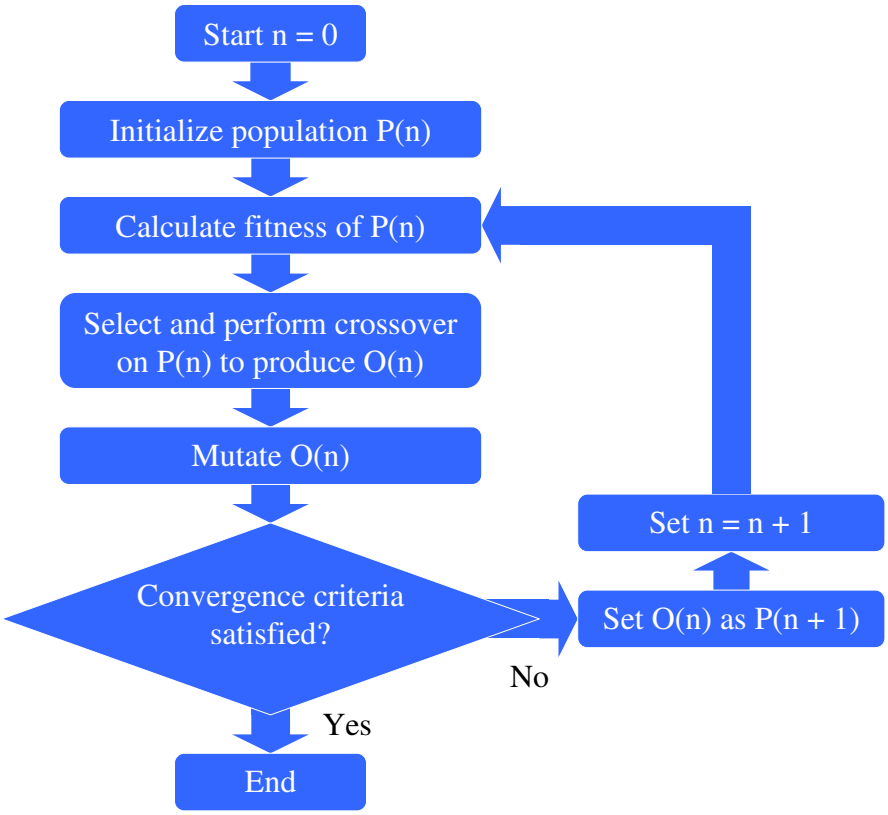


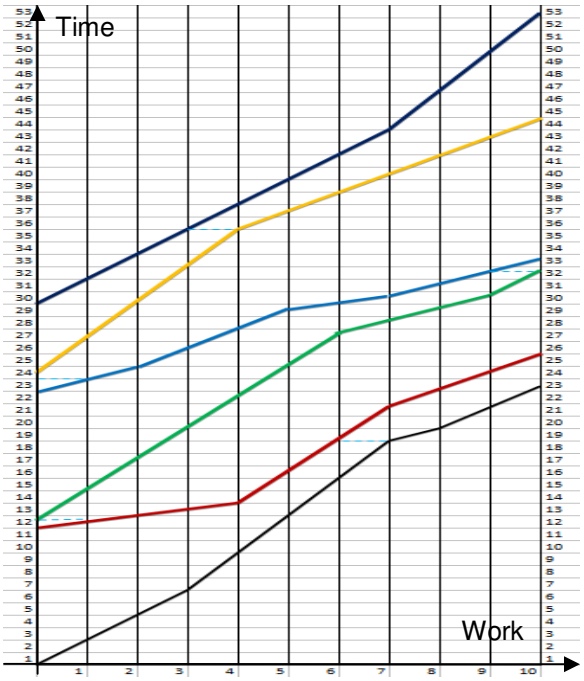

**Figure 6: Flowchart of Genetic Algorithm**

## 3 Schedule Example

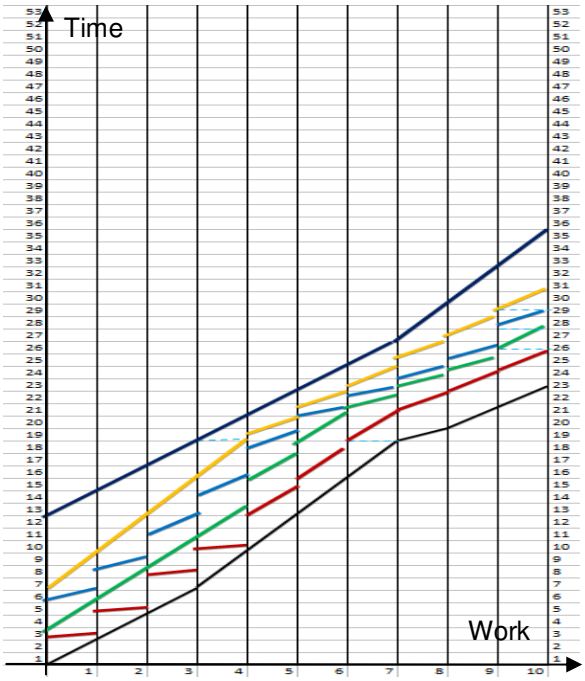
A larger scheduling example than the previously described scenarios was designed with six activities with varying production rates and ten segments to demonstrate the methodology and test its functioning. The initial schedule is provided in Table 2 and Figure 7. How activities can be modeled for all possible breaks is shown by the singularity functions of Equations 3 through 9. The initial project duration is 52 weeks.

**Table 2: Activity Information**

| Activity | Successor | Time Distance at Each Segment (Weeks) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| A | B | 2.0 | 2.0 | 2.0 | 3.0 | 3.0 | 3.0 | 3.0 | 1.0 | 1.5 | 1.5 |
| B | C | 0.5 | 0.5 | 0.5 | 0.5 | 2.5 | 2.5 | 2.5 | 1.5 | 1.5 | 1.5 |
| C | D | 2.5 | 2.5 | 2.5 | 2.5 | 2.5 | 2.5 | 1.0 | 1.0 | 1.0 | 2.0 |
| D | E | 1.0 | 1.0 | 1.5 | 1.5 | 1.5 | 0.5 | 0.5 | 1.0 | 1.0 | 1.0 |
| E | F | 3.0 | 3.0 | 3.0 | 3.0 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 |
| F | - | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 3.0 | 3.0 | 3.0 |




**7a: Before Interruption**




**7b: After Interruption**

**Figure 7: Activities with Variable Production Rates Example**

[3] $Minimize\{y(10)_F\}$ The total project duration

Subject to:

[4] $$y(x)_A = 2\cdot\langle x-0\rangle^1 - (2-3)\cdot\langle x-3\rangle^1 - (3-1)\cdot\langle x-7\rangle^1 - (1-1.5)\cdot\langle x-8\rangle^1$$ Non-interruptible

[5] $$y(x)_B = y(0)_B\cdot\langle x-0\rangle^0 + 0.5\cdot\langle x-0\rangle^1 - (0.5-2.5)\cdot\langle x-4\rangle^1 - (2.5-1.5)\cdot\langle x-7\rangle^1 + \sum_{i=1}^{n-1} b_i\cdot\langle x-i\rangle^0$$

[6] $$y(x)_C = y(0)_C\cdot\langle x-0\rangle^0 + 2.5\cdot\langle x-0\rangle^1 - (2.5-1)\cdot\langle x-6\rangle^1 - (1-2)\cdot\langle x-9\rangle^1 + \sum_{i=1}^{n-1} c_i\cdot\langle x-i\rangle^0$$

[7] $$y(x)_D = y(0)_D\cdot\langle x-0\rangle^0 + 1\cdot\langle x-0\rangle^1 - (1-1.5)\cdot\langle x-2\rangle^1 - (1.5-0.5)\cdot\langle x-5\rangle^1 - (0.5-1)\cdot\langle x-7\rangle^1 + \sum_{i=1}^{n-1} d_i\cdot\langle x-i\rangle^0$$

[8] $$y(x)_E = y(0)_E\cdot\langle x-0\rangle^0 + 3\cdot\langle x-0\rangle^1 - (3-1.5)\cdot\langle x-4\rangle^1 + \sum_{i=1}^{n-1} e_i\cdot\langle x-i\rangle^0$$

[9] $$y(x)_F = y(0)_F\cdot\langle x-0\rangle^0 + 2\cdot\langle x-0\rangle^1 - (2-3)\cdot\langle x-7\rangle^1$$ Non-interruptible

[10] $$y(x)_j \geq y(x+1)_i$$ Predecessor-successor sequence constraint

Where $x$ is the work unit, $y(x)$ is a singularity function that calculates values on the curves in Figure 7, $y(0)_B$ to $y(0)_F$ are the starts of activities $B$, $C$, $D$, $E$, and $F$, respectively, that are varied during optimization, and $b_i$, $c_i$, $d_i$, and $e_i$ are the respective breaks at segment $i$ of activity $B$, $C$, $D$, and $E$ that are also varied. Sums in Equations 5 through 8 allow that all possible interruptability scenarios can be expressed with breaks at their respective integer work units. In general, values of these five starts and $4 \cdot 9 = 36$ breaks could take on any positive or even negative real number, but for practical purposes it is useful to restrict them to multiples of the time unit for $y(x)$, even though this might limit the search for minima somewhat. Per Equation 10, for a buffer of one work unit, a start of a successor segment $j$ must be larger than or equal to a finish of its predecessor segment $i$ on the time axis. Singularity functions enable automatically building numerous constraints into the mathematical model, which reduces the effort to code them to well under half (47.7%), as Equation 1 would else yield 86 constraints. This schedule example is optimized with a GA, whose corresponding performance and results are depicted in Figures 7 and 8, and numerical values are listed in Tables 3 and 4. Its minimized total project duration becomes 35 weeks with 28 breaks.

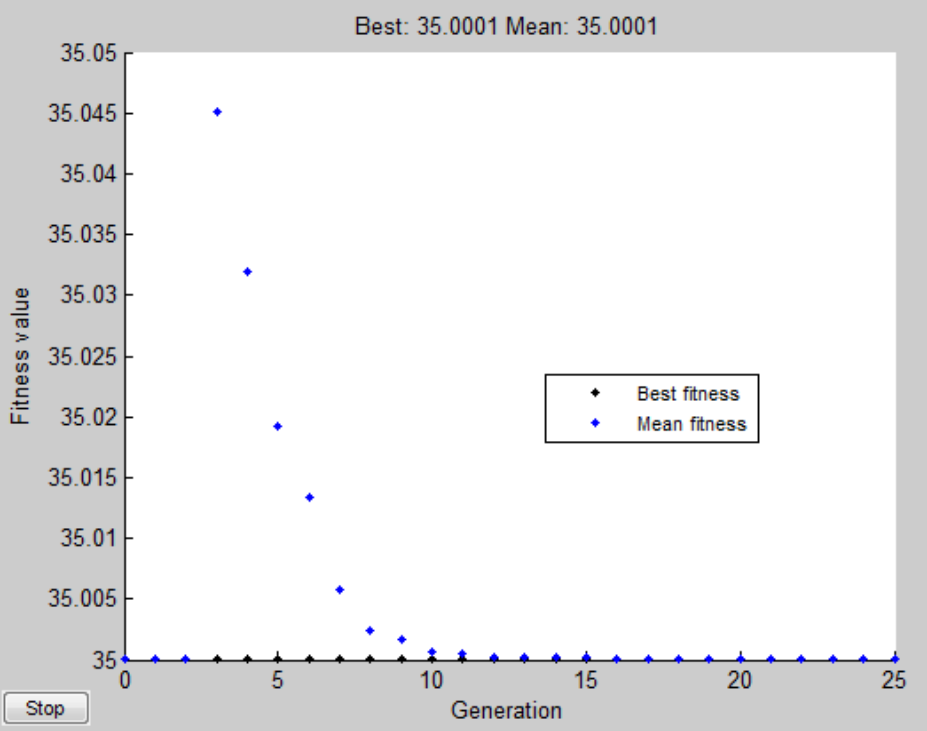



**Figure 8: Optimization Process**

**Table 3: Optimization Result 1**

| Activity | B | C | D | E | F |
|---|---|---|---|---|---|
| Start [Weeks] | 2.0 | 2.5 | 5.0 | 6.0 | 12.0 |

**Table 4: Optimization Result 2**

| Activity | Break Between Adjacent Segments [Weeks] | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1-2 | 2-3 | 3-4 | 4-5 | 5-6 | 6-7 | 7-8 | 8-9 | 9-10 |
| B | 1.55 | 2.79 | 1.54 | 2.11 | 0.50 | 0.50 | 0.00 | 0.16 | 0.00 |
| C | 0.00 | 0.50 | 0.49 | 1.01 | 0.00 | 0.80 | 0.70 | 1.16 | 0.00 |
| D | 1.50 | 2.00 | 1.49 | 2.00 | 1.00 | 1.63 | 1.25 | 0.89 | 0.89 |
| E | 0.00 | 0.00 | 0.00 | 0.50 | 0.49 | 0.14 | 0.87 | 0.40 | 0.26 |

## 4 Conclusion and Future Research

Within the context of minimizing the total project duration and the number of breaks and their sum, the interruptability of activities with constant and varying production rates has been analysed. For activities with a constant production rate, a rule has been proposed to determine the interruptability of activities based on scenario analysis, and a heuristic algorithm has been designed to perform this multi-objective optimization based on a geometrical approach. For activities with varying production rates an integrated model using singularity functions has been proposed, and a GA has been implemented. The contribution to the body of knowledge is that the new approach has demonstrated that the interruptability of complex linear schedules that contain activities with variable production rates can be uniformly and dynamically modeled with singularity functions, which reduces the number of constraints within the linear schedule significantly and benefits an intelligent optimization. Since interruptions can have many side effects on projects besides enabling a minimized total project duration for certain scenarios, such as the interruption cost and loss of a learning effect, such detriments should be taken into account under future research.


## Acknowledgement

This paper was prepared while the last author was on sabbatical at the University of Alberta in Edmonton, Alberta, Canada. Faculty, staff, and students are thanked for their hospitality.


## References


González V., Alarcón L. F., Molenaar, K. R. (2009). "Multiobjective design of work-in-process buffer for scheduling repetitive building projects." *Automation in Construction* 18(2): 95-108.

Harris, R. B., Ioannou, P. G. (1998). "Scheduling projects with repetitive activities." *Journal of Construction Engineering and Management* 124(4): 269-278.

Hassanein A., Moselhi O. (2004). "Planning and scheduling highway construction." *Journal of Construction Engineering and Management* 130(5): 638-646.

Hegazy, T., Kamarah, E. (2008). "Efficient repetitive scheduling for high-rise construction." *Journal of Construction Engineering and Management* 134(4): 253-264.

Hyari, K., El-Rayes, K. A. (2006). "Optimal planning and scheduling for repetitive construction projects." *Journal of Management in Engineering* 22(1): 11-19.

Ipsilandis, P. G. (2007). "Multiobjective linear programming model for scheduling linear repetitive projects." *Journal of Construction Engineering and Management* 133(6): 417-424.

Long, L. D., Ohsato, A. (2009). "A genetic algorithm-based method for scheduling repetitive construction projects." *Automation in Construction* 18(4): 499-511.

Lucko, G. (2009). "Productivity scheduling method: Linear schedule analysis with singularity functions." *Journal of Construction Engineering and Management* 135(4): 246-253.

Russell A. D., Caselton W. F. (1988). "Extensions to linear scheduling optimization." *Journal of Construction Engineering and Management* 114(1): 36-52.

Srisuwanrat, C., Ioannou, P. G. (2007). "Optimal scheduling of probabilistic repetitive projects using completed unit and genetic algorithms." *Proceedings of the 2007 39th Winter Simulation Conference*, eds. Henderson, S. G., Biller, B., Hsieh, M.-H., Shortle, J., Tew, J. D., Barton, R. R., Washington, District of Columbia, December 9-12, 2007, Institute of Electrical and Electronics Engineers, Piscataway, New Jersey: 2,151-2,158.

Srisuwanrat, C., Ioannou, P. G. (2007). "The investigation of lead-time buffering under uncertainty using simulation and cost optimization." *Proceedings of the 15th Annual Conference of the International Group for Lean Construction*, eds. Pasquire, C., Tzortzopoulos, P., July 18-20, 2007, East Lansing, Michigan: 580-589.

Srisuwanrat, C. (2009). "The sequence step algorithm: A simulation-based scheduling algorithm for repetitive projects with probabilistic activity durations." A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Civil Engineering) in The University of Michigan, University of Michigan, Ann Arbor, Michigan: 415 p.