Emerald Built Environment Project and Asset Management

# Dynamic Scheduling Model for the Construction Industry



| | |
|---|---|
| Journal: | *Built Environment Project and Asset Management* |
| Manuscript ID | BEPAM-02-2019-0021.R2 |
| Manuscript Type: | Research Paper |
| Keywords: | resource constrained project scheduling problem, Dynamic Scheduling, Particle Swarm, Multi-Objective Optimization, Scheduling, Resource Management |
| | |

# Dynamic Scheduling Model for the Construction Industry


## Abstract

**Purpose:**

Basic project control through traditional methods is not sufficient to manage the majority of real-time events in most construction projects. This paper proposes a Dynamic Scheduling (DS) model that utilizes multi-objective optimization of cost, time, resources and cashflow, throughout project construction.

**Design/methodology/approach:**

Upon reviewing the topic of Dynamic Scheduling, a worldwide Internet survey with 364 respondents was conducted to define end-user requirements. The model was formulated and solution algorithms discussed. Verification was reported using predefined problem sets and a real-life case. Validation was performed via feedback from industry experts.

**Findings:**

The need for multi-objective dynamic software optimization of construction schedules and the ability to choose among a set of optimal alternatives were highlighted. Model verification through well-known test cases and a real-life project case study showed that the model successfully achieved the required dynamic functionality whether under the small solved example or under the complex case study. The model was validated for practicality, optimization

of various DS schedule quality gates, ease of use, and software integration with contemporary project management practices.

**Practical/Social implications:**

Optimized real-time scheduling can provide better resources management including labour utilization and cost efficiency. Furthermore, DS contributes to optimum materials procurement, thus minimizing waste.

**Originality/value:**

The paper illustrates the importance of DS in construction, identifies the user needs, and overviews the development, verification and validation of a model that supports the generation of high quality schedules beneficial to large scale projects.

## Keywords

Dynamic Scheduling, Resource-Constrained Project Scheduling Problem (RCPSP), Particle Swarm, Multi-objective Optimization.

## Introduction

Schedule delays and cost overruns are common in construction projects, albeit advances in construction equipment and management techniques, and despite the multitude of efforts devoted to the planning and control aspects of construction management (Lee, Mora, & Park, 2006). Construction project scheduling is inherently complex and dynamic, involving multiple feedback processes and nonlinear relationships, yet problems encountered during construction have been treated statically within a partial view of a project (Lyneis, Cooper, & Els, 2001). Most research and practice efforts concentrate on scheduling analysis in a static deterministic environment, such as Critical Path Method (CPM), Resource-Constrained Project Scheduling (RCPS), Line of Balance (LOB), and Critical Chain Project Management (CCPM), whereas, real-time events cause disruptions to static scheduling (Cowling & Johansson, 2002; Vieira, Hermann, & Lin, 2003; Ouelhadj & Petrovic, 2009). Benham et al. argue that network-based scheduling methods were not originally developed for managing the production phase in construction projects, and were however, designed for projects that are highly predictable and static (Benham, Harfield, & Kenley, 2016). Stochastic methods such as PERT, GERT, and others have been developed to deal with the schedule uncertainties with a probabilistic approach. The information required as inputs for stochastic methods as well as the complexity of the processes involved can deem them impractical for the use in regular day to day scheduling practices, especially in projects of considerable size as with construction. Dynamic Scheduling (DS) can be used to handle real-time events and schedule disruptions in an optimal manner. DS, as in many other scheduling concepts, started and developed in the manufacturing industry with the majority of applications and approaches presented in literature mainly focusing on manufacturing systems (Aytug, Lawley, McKay, Mohan, & Uzsoy, 2005; Herroelen & Leus, 2005; Ouelhadj & Petrovic, 2009; Lagodimos, Mihiotis, & Kosmidis, 2004). Other applications have been cited in computer engineering (Webster & Azizoglu, 2001), in logistics (Liang, Huang, & Yang, 2009), in petroleum (Aissani, Beldjilali, & Trentesaux, 2009), and in aviation (Bennell & Potts, 2017).

DS can be summarized as a continuous dynamic process of updating, checking and revising the schedule according to the selected scheduling architecture and based on predefined rescheduling strategy, policy and rescheduling technique. The scheduling/rescheduling processes involve regular deterministic scheduling methods for the analysis, while effects of real-time events and uncertainties are mitigated with the continuous schedule adjustments and/or optimization (Fahmy, 2014). DS has been categorized in literature into: reactive; proactive; and predictive reactive scheduling (Aytug, Lawley, McKay, Mohan, & Uzsoy, 2005; Herroelen & Leus, 2005; Ouelhadj & Petrovic, 2009). In reactive scheduling no baseline schedule is required, and real-time decisions are made on the resource level in more of a 'dispatching' manner, which can be seen in small construction companies. The proactive approach is based on producing a predictive baseline schedule, with continuous cycles of progress updates usually to activity durations and resource levels, but rarely to logic. This is similar to the process of schedule updating in construction. The predictive reactive approach produces a baseline schedule and is logically revised based on real-time events, similar to the production of revised schedules in the construction industry for major real-time disruptions. These approaches do not however, provide dynamic optimization of schedules.

DS in construction literature has been quite scarce and has focused on the aforementioned approaches using risk management and project time control practices without providing optimized schedules (Vanhoucke M. , 2013; Kerhove & Vanhoucke, 2017; Kumaar & Praveenkumar, 2015). Kumaar and Praveenkumar recognized the notion that almost no project in the construction industry performs totally as planned due to the different uncertainties faced during the execution phase. They attempted, however, to provide dynamic scheduling without dynamic optimization of the schedule (Kumaar & Praveenkumar, 2015). The same approach of performing dynamic scheduling through risk management and project time control was introduced in other literatures (Vanhoucke M. , 2013; Kerhove & Vanhoucke, 2017).

The problem can be summarized as construction project scheduling being inherently complex and dynamic, whereas, most techniques developed are static or deterministic in nature. Furthermore, the information required for stochastic methods and their complexity can deem them impractical for practical everyday use. DS models for construction are scarce, and in the very few instances in literature, the models did not include optimization. Therefore, this paper addresses this gap and aims to introduce a model for dynamic scheduling where schedules are optimized to handle real-time events. This paper includes multi-objective optimization, based on the work of other industries, and building upon current optimization techniques to meet the dynamic needs of the construction industry. The mathematical formulation of the model addressed specific construction-related issues and responded to the needs identified in a questionnaire survey. Optimized real-time scheduling can provide better resources management including labour utilization and cost efficiency, providing financial and time benefits to contractors and owners. Furthermore, DS contributes to optimum materials procurement, thus minimizing waste and improving the sustainability of the construction industry, and hence, affecting society as a whole.

## Schedule Optimization Review

Project scheduling optimization research has mostly aimed to either optimize resource utilization or provide the optimum time-cost trade off. The Resource-Constrained Project Scheduling Problem (RCPSP) problem, in principle, is an improved version of the original Critical Path Method (CPM) after taking the resource limitations into consideration (Hartmann & Briskorn, 2010). The Time-Constrained Project Scheduling Problem (TCPSP) aims to handle the balance between resource availability constraints of renewable resource and the project time frame (Ranjbar & Kianfar, 2007). The Time-Cost Trade-Off problem (TCTP) aims to minimize the project overall duration, in terms of project cost. Various models were generated to present and solve the TCTP (Demeulemeester, de Reyck, Foubert, Herroelen, & Vanhoucke, 1998; Ranjibar, Kianfar, & Shadrokh, 2008). Critical Chain Project Management (CCPM) focuses on the constraints of a project which prevent achieving its goals (Rabbani, Ghomi, Jolai, & Lahiji, 2007).

Researches have expressed its pitfalls in oversimplification of the problem (Herroelen & Leus, 2005). The, resource optimization problems although well-defined and properly modelled are Non-Polynomial Hard (NP-Hard) (Arigues, Demassey, & Neron, 2008) and can be quite costly in terms of processing time with large scale construction projects, if an exact solution is required. This can be unacceptable with the requirements of dynamic construction projects.

Construction project scheduling is inherently complex and dynamic, involving real-time disruption, multiple feedback processes and nonlinear relationships, yet problems encountered during construction have been treated statically within a partial view of a project (Lyneis, Cooper, & Els, 2001). A research need, thus, exists to attend to the dynamic and complex nature of the construction industry. This need has been addressed by a few scholars, but the approaches were mostly in terms of risk management and project time control, without dynamic optimization of the schedule (Vanhoucke M. , 2013; Kerhove & Vanhoucke, 2017; Kumaar & Praveenkumar, 2015).

## Dynamic Scheduling Review

The types of real-time events and DS categories are discussed in this section. Two main components of any DS system are when and how to respond to real-time events. Accordingly, DS the rescheduling policy and strategy are reviewed. Finally, the rescheduling techniques used in literature are overviewed.

Real-time events which cause disruptions to static scheduling have been discussed and categorized differently in several researches (Suresh & Chaudhuri, 1993; Stoop & Wiers, 1996; Cowling & Johansson, 2002; Vieira, Hermann, & Lin, 2003; Ouelhadj & Petrovic, 2009). Based on these researches and from a construction industry point of view, real-time events can be classified into three main categories:

- Project related events: Additions or omissions to the project's original scope (through change orders, or design changes), changes to the project's due dates or milestones, changes to the predefined sequence of work due to changes in priorities of the project's deliverables, delays in governmental or authorities approvals, effects of inclement weather, force majeure events (ex. floods or earthquakes), …etc.

- Resource related events: Shortages of material, arrival of defective material/equipment, breakdowns of construction machinery on site, delayed arrivals of specialized resources, insufficient capacities of assigned resources, sickness or death of key resources, …etc.
- Operations related events: Quality rejection of outputs, changes in deliverables specifications, prolongations in operations durations (due to incorrect estimates for resources productivities, incorrect estimates for equipment set-up times, or manpower learning curves), unexpected behaviour of predefined design elements (for example unsatisfactory results of soil tests after the completion of ground improvement works), …etc.

The categorization of DS is based on the strategy of how or if the schedule baseline is generated and how to respond to real-time events. The three main categories of DS have been previously discussed in the Introduction section of this paper. The most common DS system is the predictive-reactive scheduling approach (Ouelhadj & Petrovic, 2009) and is thus further explained in this section. In predictive-reactive scheduling a predictive baseline schedule is initially generated, then rescheduled (logically revised) based on real-time events and progress data, as illustrated in Figure 1.

Three rescheduling policies were presented in literature to define when to respond to real-time events (Church & Uzsoy, 1992; Sabuncuoglu & Bayiz, 2000; Vieira, Hermann, & Lin, 2003; Aytug, Lawley, McKay, Mohan, & Uzsoy, 2005):

- Periodic rescheduling policy: Rescheduling process is started every predefined time interval regardless of the amount of real-time events occurring during this period.
- Event-driven rescheduling policy: Rescheduling process is triggered with the occurrence of any disruptive real-time event.
- Hybrid rescheduling policy (Rolling time horizon): Rescheduling process takes place periodically regardless of the in between events; however, certain predefined events can trigger the start of a new intermediate rescheduling process.

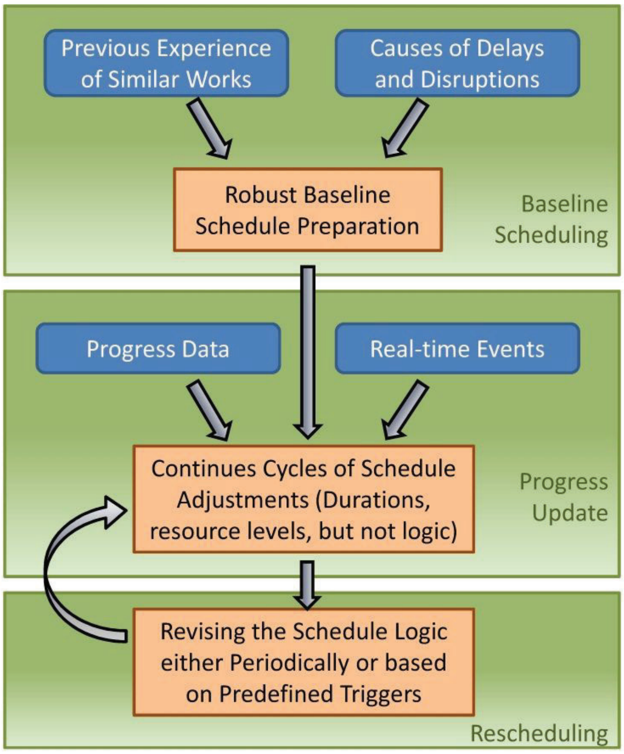


**Figure 1:** Predictive-Reactive Dynamic Scheduling

The rescheduling strategy is concerned with the mass of the changes to be made. There are two main rescheduling strategies (Sabuncuoglu & Bayiz, 2000; Cowling & Johansson, 2002; Vieira, Hermann, & Lin, 2003): Schedule Repair where real-time events are mitigated through minimum adjustments to the schedule portion related to the event; and Complete Rescheduling where the project schedule is regenerated from scratch. In practice, the latter is practically not preferred due to the required time and effort, despite of the fact that it helps in maintaining the near-optimum solution.

The rescheduling technique represents the method or algorithm which a computerized system will use to repair/reschedule the project plan. Heuristic algorithms are used as a rescheduling technique, which are simple techniques that seek good solutions at a reasonable computational cost without being able to guarantee feasibility nor optimality (Reeves, 1995).

## Research Methodology

In order to develop the DS model, the following research methods were adopted, and are expressed in following sections of the paper:

Defining end-user requirements: A questionnaire survey was implemented to investigate the problem under study from a practical perspective, and identify certain features for the proposed model. The survey was issued on the Internet to ease the process of invitations distribution as well as the responses collection, to obtain opinions that are not biased towards a particular surrounding environment, and to have wide geographical spread and several opinions from different expertise levels and roles.

Model formulation and solution: Based on the defined user requirements and literature review a generalized problem mathematical model was compiled with multiple objectives of time, cost, resources and cashflow. An optimization technique was developed that suits the required operability of the DS model.

Model verification: The model is checked using predefined examples to produce acceptable solutions in terms of optimization quality and in terms of analysis of time, whereas, these solutions are compared to the model output.

Model validation: A case study using actual data from a real project is checked under real-time conditions and the optimization capabilities of the model under dynamic environment. Further, feedback of field experts is sought concerning the model and its practicality via a questionnaire survey.

## User Requirements of DS in Construction

The user requirements of DS have been investigated in a questionnaire survey to the construction industry. The survey reviewed the scheduling/rescheduling problem from practical construction point of view and the suitability and practicality of a dynamic scheduling solution to the day-to-day scheduling works. The survey also reviewed the expectations of the functions/features to be present in any proposed dynamic scheduling solution. A webpage was developed for the questionnaire survey and published on the internet to facilitate its spreading. Then invitations were sent to major construction companies and consultancy offices; in addition, other invitations were sent to members of few popular planning/project management forums. The 364 responses of the survey covered construction planners/schedulers in various types of construction organizations with differing seniority and experience levels. The minimum sample size was computed to be 100 for infinite samples (Easterby-Smith, Thorpe, & Lowe, 2002). The participants were geographically distributed as following: Africa 15.9%; Asia 34.1%; Europe 32.1%; North America 9.1%; South America 3.0%; and Oceania 5.8%. Participants had various seniority levels: top management 11.8%; department management 25.6%; senior level 56.0%; and junior level 6.6%. The experience levels of participants were 34.3% with high experience (over 15 years); 48.1% with medium experience (5 to 15 years); and 17.6% with low experience (less than 5 years). Full details of the survey can be found in Fahmy, Hassan and Bassioni (2014-a) and Fahmy, Hassan and Bassioni (2014-b).

The most relevant outcome of the survey was the welcoming of optimization during scheduling by 96% of the respondents. Real time events were selected as the main cause of disruption to schedules, where less than 0.8% of the participants stated that real-time events does not impact schedule integrity, while more than 85% selected that this disruption usually/always happens. Thus, directly pointing to the need for dynamic scheduling with relevant optimization.

Other relevant results included defining can be summarized in the following points:

1. Beside the classical scheduling objective of minimize time, more than 83% of the survey responses suggested that cost optimization objective should be also used. Adding additional optimization objectives to identify the schedule's flexibility and/or criticality was a general requirement by more than 50% of the survey's participants. Accordingly, two more objectives should be considered while modeling.
2. With respect to resources analysis, 97.8% selected that it is required to be performed before baseline schedule submission, from which 87.5% acknowledged that the time was always

not sufficient to review all resources distribution. Thus, resources distribution (or leveling) is one of the most important objectives for an automated scheduling system.

3. For cash flow analysis, 91.4% selected that it is required, while 75.8% acknowledged that the time was always not sufficient to change the schedule accordingly. Similar response ratios were given to the same issues during schedule updates, but with less importance to cash flow analysis where 16.8% selected that it is not required during schedule updates. Accordingly, cash flow should also be modeled as one of the optimization objectives.
4. Optimization of medium and large scale schedules involves large computational burden, and the number of activities/resources to be optimized is exponentially proportional to the optimization time. More than 59% of participants responded to the fact that the optimization algorithm can concentrate on critical/near critical activities leading to project milestones; while for the rest of responses were more oriented towards optimizing all activities rather than critical activities only. This can be modeled in any proposed solution to be optional based on the requirements of each project.
5. With respect to activity modes, about 58% accepted the practicality of using several execution modes for activities; which makes it an optional input to the model.
6. For solution architecture, nearly 82% suggested that an appropriate solution should be a separate software tool to deal with the optimization process and communicates with the planning software for the transfer of optimized solutions; from which 55% accepted this tool to be fully integrated with the database of the planning software.
7. Finally, most participants suggested that any proposed dynamic scheduling solution should suggest several optimized alternatives for the planner to review and make his choice.

~~7.~~

# Dynamic Scheduling Model

## *Mathematical Formulation*

The DS model was mathematically formulated based on the well-accepted Resource Constrained Project Scheduling Problem (RCPSP). Extensions to the RCPSP were taken into account to match the real life problem's characteristics, such as pre-emption (Ballestin, Valls, & Quintanilla, 2008), generalized precedence relations & minimum/maximum time lags (Chassiakos & Sakellaropoulos, 2005; Vanhoucke M. , 2006), mode scheduling (Alcaraz, Maroto, & Ruiz, 2003), time/resource trade-off problem (TRTP) (Ranjbar & Kianfar, 2007). Accordingly, the main objectives of the model were formulated in line with the RCPSP and in line with the outcomes of the user requirements survey as follows:

1) Minimize Project time span:

$$Minimize \quad T = S_n$$

2) Minimize Project overall cost, which implicitly includes the Resource Investment or Time-Resource Trade-Off problem objective, as well as Time-Cost Trade-Off problem objective (note: $FC_i$ is the fixed cost of activity *i* , costs which are irrelevant to main resources defined in schedule):

$$Minimize \quad C = \sum_{i \in V} FC_i + \sum_{k \in R} c_k * max \left\{ \sum_{i=1}^{n} r_{ik} \right\}$$

3) Maximize resources utilization (i.e. minimize Resources Levels):

$$Minimize \quad RLI = log \left( \sum_{k \in R} (c_r / T_{min}) \times \sum_{t=0}^{T} \sum_{i \in V} (r_{ikt})^2 + \sum_{k \in N} u_k \times \sum_{t=0}^{T} \sum_{i \in V} (r_{ikt})^2 \right)$$

4) Cash flow improvement (i.e. minimize Negative Cash Flows):

$$Minimize \quad NCF = abs\left(\sum_{t=1}^{T} min\,\{0, CF_t\}\right)$$

5) Maximizing schedule stability/robustness (represented in its simplest form as the sum of total floats):

$$Maximize \quad SS = \sum_{i \in V} TF_i$$

Where:

T = Time span of the schedule

C = Project overall cost

RLI = Resource level

NCF = Negative cash flows

SS = Schedule stability

V = Set of activities 1 to n, where 1 & n are dummy activities added for simplicity of calculations

H = Set of pairs of activities indicating their precedence

K = Set of renewable resources

$F_j$ = Finish date for activity j

$d_j$ = Duration of activity j

$S_t$ = Set of activities in progress within time interval [t-1, t]

$r_{ijkm}$ = Section j of activity i per period requirement from resource k in execution mode

$ma_k$ = Available units from resource k

$FC_i$ = the fixed cost of activity i

$TF_i$ = Total lfoat of activity i

Any other financial related objectives such as Net Present Value (NPV), Discounted Cash-flow (DC) …etc., were not considered in the model to avoid unnecessary financial complications. Extensions were made to the DS model for minimizing schedule's deviation, maximizing schedule flexibility, model simplification into optimizable and non-optimizable activities, and splitting of resources into scarce and abundant resources. Constraints have been incorporated to account for the data date which sets the start of the first activity at time interval zero, the precedence which defines all schedule logic relations, and the sectional sequence which introduces logic ties between activities' sections to assure their sequential order if pre-emption is allowed. Optional constraints mainly represent project specific requirement, such as target completion, target budgeted cost, maximum liquidity available for the project or maximum negative cash flow, set of activities time constraints, and two sets of resources availability constraints. A full account of the mathematical formulation was not included for paper size considerations, but can be found in Fahmy (2014).

### *DS Model Solution*

The Particle Swarm Optimization (PSO) was selected as a model solution for the following reasons:

1. All of the meta-heuristic methods were proven in literature to perform efficiently. However, PSO requires only primitive and simple mathematical operators, and it is

computationally inexpensive in terms of both memory requirements and time (Moslehi & Mahnam, 2011);

2. Hardly any meta-heuristic, other than PSO, was presented to solve both problem categories (RCPSP & TCPSP) with the same algorithm;

3. PSO has been shown to perform well with respect to other methods (Chen, 2011).

An optimization algorithm was developed with the ability to explore the search space generated by the formulated mathematical model for finding the optimum solution as per the model's objectives. A computer program was developed to solve the model via a meta-heuristic solution using Particle Swarm Optimization (PSO) with stacking justification. Furthermore, an improvement to the original PSO was devised and termed The Differential Density PSO (DDPSO) algorithm ~~was developed~~ including the following contributions: Stacking Justification (SJ), a heuristic technique to improve solutions quality for resource-constrained problems; Float Justification (FJ), a heuristic technique to improve solutions quality for time-constrained problems; Rectified Schedule Generation Scheme (RSGS), an improved version of the original SGS, which improves the distribution of the search space; Combined Priority Rules (CPR), a technique to initialize particle swarm initialization for proper spreading of swarm particles among good quality areas of the search space; and Differential Density Particle Swarm Optimization (DDPSO), a modified PSO with the introduction of density parameters to swarm particles to overcome the algorithm's early convergence.

~~(Fahmy, Hassan, & Bassioni, Improving RCPSP solutions quality with stacking justification - Application with particle swarm optimization, 2014 c).~~

~~The Differential Density PSO (DDPSO) algorithm was developed including the following contributions: Stacking Justification (SJ), a heuristic technique to improve solutions quality for resource-constrained problems; Float Justification (FJ), a heuristic technique to improve solutions quality for time-constrained problems; Rectified Schedule Generation Scheme (RSGS), an improved version of the original SGS, which improves the distribution of the search space; Combined Priority Rules (CPR), a technique to initialize particle swarm initialization for proper spreading of swarm particles among good quality areas of the search space; and Differential Density Particle Swarm Optimization (DDPSO), a modified PSO with the introduction of density parameters to swarm particles to overcome the algorithm's early convergence.~~

The concept behind the proposed model modification is that a high density swarm particle should move slower than a low density particle, and accordingly explore the search space in a higher intensity; so, the density of particle i is inversely proportional to the particles velocity. The final formulation of the *DDPSO* is as follows:

$$P = No.\,of\ generated\ schedules\ /\ Schedules\ limit$$

$$V_{ij}^{t} = \left(w \times V_{ij}^{t-1} + r_1 c_1\left(L_{ij}^{t-1} - X_{ij}^{t-1}\right) + r_2 c_2\left(G_j^{t-1} - X_{ij}^{t-1}\right)\right) / \delta_i$$

$$X_{ij}^{t} = X_{ij}^{t-1} + V_{ij}^{t}$$

Where $V_i$ is the velocity of particle *i* ($i \in M$ particles), $V_{ij}^{t}$ & $V_{ij}^{t-1}$ are the velocities of component *j* of particle *i* in iterations *t* & *t-1*; $r_1$ & $r_2$ are two random numbers (from 0 to 1); $c_1$ & $c_2$ are two learning coefficients which define the influence of the local and global best solutions on the new velocities; $X_{ij}^{t}$ & $X_{ij}^{t-1}$ are the positions of component *j* of particle *i* in iterations *t* & *t-1*; $L_{ij}^{t-1}$ is the position of component *j* in the positions vector of the best solution found by particle *i* until iteration *t-1*; $G_j^{t-1}$ is the position of component *j* in the positions vector of the best solution found globally in the swarm until iteration *t-1*. The pseudocode for the DDPSO algorithm can be expressed as the following:

```
While total generated schedules < schedules limit
    t = 1
    For each particle i in forward & backward swarms
        If  t = 1 Initialize DD swarm particles using CPR
        Update V_i^t &  X_i^t
        Generate schedule using RSGS
        Apply selected justification scheme
        Map justified solution into  X_i^t
        Calculate particle's fitness
        Update local best L_i^t & global best G^t
    End for
    t = t + 1
End while
```

The selected priority can be either a single priority rule or the CPR and the justification scheme can be original justification, stacking justification or a combination of both. The Rectified Schedule Generation Scheme (RSGS) is an unfair distribution of priorities lists after eliminating lists which do not respect precedence, mainly caused by incorrect order of activities.

The optimization outputs vary depending on the quality of outputs achieved. If the algorithm was able to find any feasible solution, then the Pareto Front (PF) would be presented as the optimization output; otherwise, a set of best achieved Non-Feasible Solutions (NFS) would be presented. The PF consists of a set of pairs of objective (or combined objective) and the corresponding best solution achieved; while the NFS consists of a set of best achieved solutions ordered by their degree of feasibility.

## Dynamic Scheduling Model Verification

The DS model numerical performance was checked against different static scheduling problem sets to confirm the multi-objective solution capabilities of the model from an operational research perspective. The model was further tested with real projects data to verify its solution capabilities under dynamic environment and from a construction industry perspective.

### *Verification Using Benchmark Problem Sets*

Most of the researches published in the scheduling context have used benchmarks problem sets generated by two well-known libraries: the PSPLib, and the PSPLib/max. These libraries were generated using the problems generators PROGEN (Kolisch, 1997) & PROGEN/max (Kolish, Schwindt, & Spreecher, 1998). The optimal values are not known for all these instances, thus the best known solutions were used for performance comparison of the calculated lower bounds. The PSPLIB's SRCPSP j-30 (480 instances), j-60 (480 instances), j-90 (480 instances) & j-120 (600 instances) were used for testing the RCPSP category. While for the TCPSP category, the SRIP/max j-10 (270 instances), j-20 (270 instances) & j-30 (270 instances) were used.

The testing of the model performance is fully reported in Fahmy, Hassan, & Bassioni (2014-c). The algorithm of the DS model was compared to the highest performing and highly ranked stateof-the-art algorithms for solving single mode RCPSPs and was shown to outperform many of these models, whereas, it ranked third among the top 20 models for SRCPSP j-30, the sixth among the top 22 model for SRCPSP j-60, and SRCPSP j-120. The use of the DS model was proven to have significant improvement to results, especially for small number of generated schedules; while the improvement decreased (but still existed) with the amount of generated schedules. This behavior is suitable for practical applications where achieving quick good-to-high quality solutions is necessary.

## *Verification on Real-Life Project*

To further verify the model, an example model application on a portion of a real-life project, whose identity is confidential, was solved manually and using the DS model software. The detailed case study is available in Fahmy (2014). The project was for the development of an international airport and considered a mega project including new airfield, terminal building, traffic control tower and airport management buildings, together to the refurbishment of the existing airfield. The contract value was about 1.3 Billion USD. The Work Break-Down Structure (WBS) of the project extended to eight levels. The project schedule contained hundreds of resources. Not all resources were marked for optimization throughout the case study testing as: constrained resources due to resources availability; constrained resources due to plant productivity constraints; and resource to be levelled to minimize resource costs.

The application of the DS model on the project involved three main stages corresponding to three schedule quality gates:

1. DS project creation, definition of additional project data and optimization inputs; and then performing the baseline schedule optimization and selecting an optimized alternative.

2. Using the selected optimum baseline for progress updating using available historical records for progress, variation orders, and delay events. Then optimization of these updated schedules for the purpose of creating construction look-ahead schedules.

3. Selecting one of the progress updates with a major delay event in order to check the functionality for preparing an optimized what-if schedule mitigating this delay event(s).

The main targets for the optimization process were resources levelling, and minimization of project's direct costs. In addition, minimizing schedule deviation and minimizing total float consumption were added with small weightages to the optimization objectives to avoid the addition of unnecessary resource logic which might be added for resource levelling fine tuning; thus optimum solutions are achieved with the best possible resource levelling with minimum additional logic. During progress updates, minimizing schedule changes weight was increased to avoid large schedule disruption. Finally, minimizing time was not included to objectives because the project works was already time constrained with 9 contractual milestones.

According to the project requirements, the baseline optimization process was performed using the following project parameters:

***Objectives:*** Minimize Cost [C] (weight=60%), Resource Levelling [RLI] (35%), Minimize Schedule Deviation [SD] (2%) & Min. Total Float Consumption [TFC] (3%)

***Constrained Activities:*** 9 milestones

***Constrained Resources:*** 15 resources

***Optimizable Activities:*** All (7604 activities)

***Optimizable Resources:*** 23 resources (15 constrained & 8 to be levelled)

The Pareto Front consisted of 15 solutions, corresponding to all possible combinations of the optimization objectives (C, RLI, SD & TFC). The most important solutions to discuss are the SD-TFC & the C-RLI-SD-TFC. The SD-TFC solution had 43 schedule changes (42 new & 1 deleted resources logic); where these logic changes are the minimum changes needed to achieve resources constraints. The best achieved solution in the baseline optimization process was selected for progress updates. The DS model was utilized in a couple of updates. The latter of which more than 100 variation orders were received, including major changes to utility networks, additional surcharging to roads and buildings, modifications to stone columns ground improvement areas, changes in specifications and so forth. Figure 2 illustrates the original and optimized schedule for a several types of resources. What-if analysis was conducted using the DS model to check the possibility of mitigating delays by increasing the number of stone columns installation crews. The resulting schedule reduced the delay in the contractual milestone by more than 4 months, with an increase of about 49% in the number of stone columns crews and 19 schedule changes (8 added and 11 deleted resource logic relations). The test cases and the case study showed that the model successfully achieved the required dynamic functionality whether under the small solved example or under the complex case study.

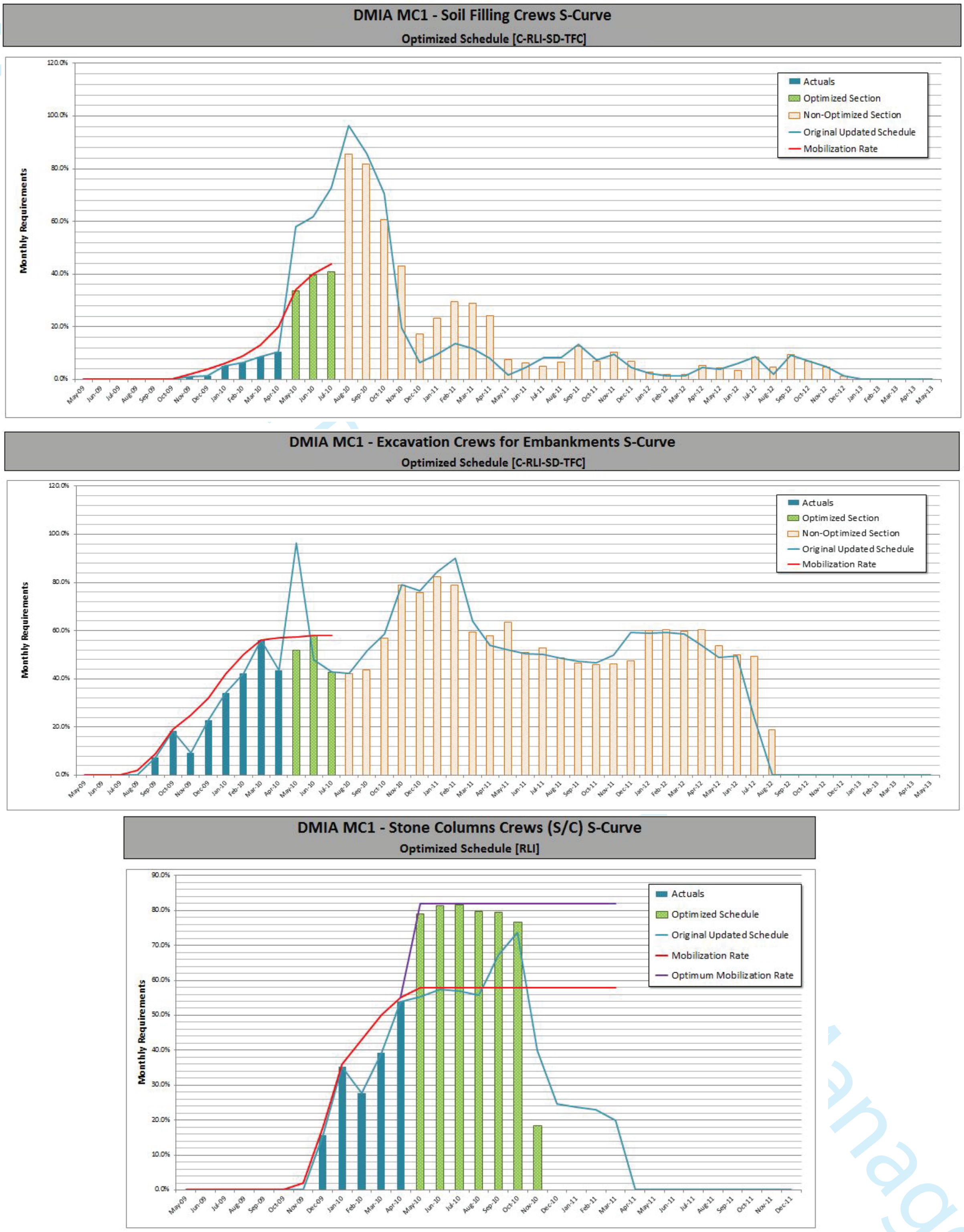


**Figure 2:** Sample Original and Optimized Histograms for a Progress Update and What-If schedule Optimization in the Case Study

# Model Validation

## *Validation Approach*

The validation of a model can be achieved if it is accepted as reasonable for its intended purpose by people who are knowledgeable about the system under study (McGraw-Hill Encyclopedia of Science and Technology, 2002). This type of validation has been termed as 'Face Validity' of research, where experts observe the model and agree it represents a high degree of what happens in reality and its usability (Lucko & Rojas, 20101). The validity of a study can be conducted using face validity through obtaining the subjective judgement of a non-statistical nature that seeks the opinion of non-researchers regarding the validity of a particular study (Leedy & Ormrod, 2001). Therefore, the DS model was validated by seeking project management field experts' judgement. The model was presented to field experts with verification results, and their feedback obtained concerning model validity and implementation practicality. Feedbacks were received from 15 participants, who have previously worked in more than 39 countries. The experts worked as project management consultants (33%), engineering consultants (27%), general contractors (27%), and specialized contractors (13%). Experts with more than 15 years of experience comprised 60% of the sample, and with 10 to 15 years another 27%. The occupation level included 47% of the sample working as department management, 33% as top management and the remaining 20% senior level. Feedback forms were distributed by emails, with phone interviews when necessary to provide guidance, and responses were collected and analyzed. Upon providing a brief summary of DS, the proposed DS model, and the verification results, the form included questions concerning:

a) The need for an optimized DS system to improve scheduling process of construction industry;

b) Assessing the practicality of the proposed DS model within current project management practices;

c) Investigating the efficiency of the developed model in achieving the objectives of the proposed solution; and

d) Soliciting comments and/or improvements via open-ended questions on the DS model.

### *Validation Results*

The general outcome of the validation was very positive from most experts/practitioners. The summary of participations' quantitative ratings is illustrated in Table 1. The majority of experts confirmed the necessity for a new system to optimize schedules: 80% for baselines; 87% for revisions; 53% for updates and look-ahead; and 67% for what-if schedules. Furthermore, experts evaluated the practical application of the proposed DS model: 80% for baselines; 87% for revisions; 67% for updates and look-ahead; and 60% for what-if schedules.

Almost all participants acknowledged the effectiveness of the proposed DS system under various schedule quality gates. The model was seen as strongly effective in baseline/revision optimization by 67% and somewhat effective by 33% of the sample. Effectiveness in progress updates/look-ahead optimization was rated as strongly effective by 33%, somewhat effective by 47% of the sample, and 20% providing a not sure answer. The effectiveness of what-if optimization was assured by 60% as strongly effective, by 20% as somewhat effective, by 7% as not effective and

13% not sure of the effectiveness. More than 73% felt that the proposed system can be easily used, with 47% requiring some practicing.

In the feedback on the software, almost all participants acknowledged the effectiveness of the software tool in providing the required functionalities as per the designed DS system's framework with 60% responding strongly effective, 33% somewhat effective, and 7% responding as not being sure. In terms of the ability to define/modify additional project optimization data 33% gave a strongly effective feedback, 67% as somewhat effective and 7% not being sure. Finally, 27% deemed the DS model as strongly effective in its ability to view optimized alternatives, 67% as somewhat effective, and 7% not being sure.

The validation survey forms included two open-ended questions for participants to include their improvement suggestions or comments for the proposed system. Responses to these questions are listed in the following feedback and explanation/discussion:

1. Added logic might extend the project duration and change the project's critical path, which might not be acceptable in some projects.

   Explanation: With respect to project duration, in the model and the software tool's design, it is up to the user whether or not to constrain the project's end date, and accordingly the optimization algorithm will not extend the project's duration. For the critical path changes, the algorithm will make some changes to the critical path only if it originally contains resource logic relations which can be altered to improve the schedule's quality; and the improved alternative will be presented to the user to confirm whether or not these changes can be accepted as per the project's conditions and requirements.

2. The system should allow manual modifications by users beyond the modifications proposed in the optimized alternatives.

   Explanation: As per the system's general framework, proposed solutions are exported back to the main planning software, where the user reviews and confirm the proposed solutions, and can perform any additional manual modifications.

3. The system's model should include in the schedule's optimization process the associated schedule risks.

   Discussion: There are few additional complexities in the construction project management processes (such as Risk Management) which can be included in future improved versions of the proposed model.

4. The system should verify that the hard logic & the contractual constraints are not violated.

   Explanation: The system does not alter any of the schedule's hard logic relations; it only alters the relations marked by the user as soft logic. For contractual constraints, all constraints added by user (time constraints, resource levels, budget, liquidity ...etc.) are considered during the analysis; and includes a numerical value (the schedule's feasibility rate) which indicates that, if equals to 100%, then the algorithm was able to optimize the schedule in the proposed solution without violating any of the provided constraints.

5. The software tool needs some improvement in the graphical reports.

   Discussion: The software tool is a prototype for optimized DS in construction projects. Any software developed for commercial use should include several other improvements with respect to user interface and graphical presentation of solutions and reports.

## Conclusion and Recommendations

Construction projects are extremely dynamic, and the integrity of construction projects' schedules is vulnerable to large disruptions due to real-time events. Thus, a predefined Dynamic Scheduling (DS) strategy for mitigation of schedule disruptions is necessary to ensure efficient planning within construction projects. The few instances of DS in construction literature has focused on proactive and predictive reactive approaches using risk management and project time control practices without providing optimized schedules. This paper proposed a DS model with optimized schedules to manage real-time events.

The user requirements of DS have been investigated in a worldwide Internet survey of the construction industry with 364 responses from varied backgrounds and international experiences. The survey identified the need for multi-objective optimization various baseline and progress schedules, namely: time, cost, resources and cash flow. Additionally, the need for software tools to support the practical implementation of DS and the suggestion of several optimized alternatives for the planner to review and choose among. Views of respondents were divided on the optimization focus to be critical/near critical activities leading to project milestones, or optimizing all project activities.

A multi-objective DS model was mathematically formulated based on the well-accepted Resource Constrained Project Scheduling Problem (RCPSP) with famous extensions for minimum time, minimum cost, maximum resource utilization, minimum negative cash flows, and maximum schedule stability/robustness in terms of total floats. A computer program was developed to solve the model via a meta-heuristic solution using Particle Swarm Optimization (PSO) with stacking justification. The Pareto Front (set of feasible solutions), or the best achieved Non-Feasible Solutions (NFS) are given to the scheduler to choose from, as indicated in the user requirements survey.

The model was verified against the well-known scheduling problem sets PSPLib, and the PSPLib/max. The DS model was shown to outperform many of these models, whereas, it ranked third among the top 20 models for SRCPSP j-30, the sixth among the top 22 model for SRCPSP j-60, and SRCPSP j-120. The model was further verified by applying to a 1.3 Billion USD international infrastructure real-life project. The optimized baseline was developed and several progress update optimizations, in addition to a What-If analysis.

Finally, the model was validated via expert feedback on the practicality of the model, effectiveness of optimization of various DS schedule quality gates, ease of use, and software integration with contemporary project management practices. Feedback comments of the experts highlighted the issues of possible extension of project duration and contractual consequences, allowance of modifications by users to optimized alternatives, integration of schedule risks in the optimization process, verification that hard logic and contractual constraints are not violated, and required improvements in graphical report. Each of these comments were responded to by either a feature in the software, or the interoperable capability of the software with professional scheduling packages, or on the basis of the software being a prototype with possible improvements upon commercialization.

It can be concluded that the suggested DS model can provide better resources management including labour utilization and cost efficiency. The model can also contribute to optimum materials procurement, thus minimizing waste. Further, it has opened the door to more practical optimized schedules in construction processes and to a plethora of knowledge gaps for DS applications in construction which require further research. For example, contractual issues associated with DS, the use of Artificial Intelligence (AI) applications in DS such as knowledge-based systems, case-based reasoning, neural networks, fuzzy logic and hybrid systems. Other areas include the integration of Risk Management and tools such as Critical Chain Project Management (CCPM) in DS.

The DS model has several limitations. First, and although a large, international and diversified sample was used to define user requirements, the model will always be limited by the sample used and subject to further improvements. Second, the model solution is limited by the algorithm utilized. Third, other research gaps identified in literatures can provide further limitations to the suggested model. Fourth, although the software tool utilized is not very complicated, but might require some training to non-experienced planners. Finally, the balance between schedule changes and their benefits requires sound knowledge of the overall project's requirements and its contractual scheduling limitations, and a good understanding of the DS solution to be able to translate the project requirements into a good estimate for the optimization objectives' weights, which can serve as another limitation to in-experienced planners.

## References

Aissani, N. B., Beldjilali, B., & Trentesaux, D. (2009). Dynamic scheduling of maintenance tasks in the petroleum industry: A reinforcement approach. *Engineering Applications of Artificial Intelligence, 22*(7), 1089-1103.

Alcaraz, J., Maroto, C., & Ruiz, R. (2003). Solving the multi-mode resource-constrained project scheduling problem with genetic algorithms. *Journal of the Operational Research Society, 54*, 614-626.

Arigues, C., Demassey, S., & Neron, E. (2008). *Resource-Constrained Project Scheduling.* London: John Wiley and Sons.

Aytug, H., Lawley, M. A., McKay, K., Mohan, S., & Uzsoy, R. (2005). Executing production schedules in the face of uncertainties: A review and some future directions. *European Journal of Operational Research, 161*(1), 86-110.

Ballestin, F., Valls, V., & Quintanilla, S. (2008). Pre-emption in resource-constrained project scheduling. *European Journal of Operational Research, 189*(3), 1136-1152.

Benham, A., Harfield, T., & Kenley, R. (2016). Linear vs. systems approach to production scheduling in construction. *CIB W78 Conference.* Brisbane, Australia.

Bennell, J. A., & Potts, C. N. (2017, April). Dynamic scheduling of aircraft landings. *European Journal of Operational Research, 258*(1), 315-327.

Chassiakos, A., & Sakellaropoulos, S. (2005). Time-cost optimization of construction projects with generalized activity constraints. *Journal of Construction Engineering and Management, 131*, 1115-1124.

Chen, R. M. (2011). Particle swarm optimization with justification and designed mechanisms for resource-constrained project scheduling problem. *Expert Systems with Applications, 38*, 7102-7111.

Church, L., & Uzsoy, R. (1992). Analysis of periodic and event-driven rescheduling policies in dynamic shops. *International Journal of Computer Integrated Manufacturing, 5*, 153-163.

Cowling, P., & Johansson, M. (2002). Using real time information for effective dynamic scheduling. *European Journal of Operational Research, 139*(2), 230-244.

Demeulemeester, E., de Reyck, B., Foubert, B., Herroelen, W., & Vanhoucke, M. (1998). New computational results on the discrete time/cost trade-off problem in project networks. *Journal of the Operational Research Society., 49*, 614-626.

Easterby-Smith, M., Thorpe, R., & Lowe, A. (2002). *Management Research: An Introduction.* London, UK: Sage Publications.

Fahmy, A. M. (2014). *A Dynamic Scheduling Model for Construction Enterprises.* Loughborough, UK: PhD Thesis, Loughborough University.

Fahmy, A. M., Hassan, T. M., & Bassioni. (2014-b). Questionnaire survey on dynamic scheduling. *Project management World Journal, 3*(5), May.

Fahmy, A. M., Hassan, T. M., & Bassioni, H. (2014-a). Prospects of dynamic scheduling concept in construction industry. *Proceedings of Technology Innovation & Industrial Management (TIIM-2014)*, (pp. S4 (154-179)). Seol.

Fahmy, A. M., Hassan, T. M., & Bassioni, H. (2014-c). Improving RCPSP solutions quality with stacking justification - Application with particle swarm optimization. *Expert Systems with Applications, 41*(1), 5870-5881.

Hartmann, S., & Briskorn, D. (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research, 207*(1), 1-14.

Herroelen, W., & Leus, R. (2005). Project scheduling under uncertainty; Survey and research potentials. *European Journal of Operational Research, 165*(2), 289-306.

Kerhove, L., & Vanhoucke, M. (2017). Optimised scheduling for weather sensitive offshore construction projects. *Omega, 66*(A), 58-78.

Kolisch, R. S. (1997). PSPLIB - A project scheduling problem library: OR software - ORSEP operations research software exchange program. *European Journal of Operational Research, 96*, 205-216.

Kolish, R., Schwindt, C., & Spreecher, A. (1998). Benchmark instances of project scheduling problems. In J. Weglarz, *Handbook on Recent Advances in Project Scheduling* (pp. 197-212). Kluwer.

Kumaar, M., & Praveenkumar, T. (2015). A Dynamic Scheduling technique to control time in construction projects. *International Journal of Applied Engineering Research, 10*(62), 338-344.

Lagodimos, A., Mihiotis, A., & Kosmidis, V. (2004). Scheduling a multi-stage fabrication shop for efficient subsequent assembly operations. *International Journal of Production Economics, 90*(3), 345-359.

Lee, S., Mora, F., & Park, M. (2006). Dynamic planning and control methodology for strategic and operational construction project management. *Automation in Construction, 15*(1), 84-97.

Leedy, P. D., & Ormrod, J. E. (2001). *Practical research planning and design* (7th Edition ed.). Upper Saddle River, N.J.: Prentice Hall.

Liang, C., Huang, Y., & Yang, Y. (2009). A quay crane dynamic scheduling problem by hybrid evolutionary algorithm for berth allocation planning. *Computers & Inustrial Engineering, 56*(3), 1021-1028.

Lucko, G., & Rojas, E. M. (20101). Research validation: Challenges and opportunities in the construction domain. *Journal of Construction Engineering and Management, 136*(1), 127-135.

Lyneis, J., Cooper, K., & Els, S. (2001). Strategic management of complex projects: a case study using system dynamics. *System Dynamics Review, 34*(3), 237-260.

*McGraw-Hill Encyclopedia of Science and Technology.* (2002). McGraw-Hill Education.

Moslehi, G., & Mahnam, M. (2011). A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search. *International Journal of Production Economics, 129*, 14-22.

Ouelhadj, D., & Petrovic, S. (2009). A survey of dynamic scheduling in manufacturing systems. *Journal of Scheduling, 12*, 417-431.

Rabbani, M., Ghomi, S., Jolai, F., & Lahiji, N. (2007). A new heuristic for resource-constrained project scheduligN in stochastic networks using critical chain concept. *European Journal of Operational Research, 176*(2), 794-808.

Ranjbar, M., & Kianfar, F. (2007). Solving the discrete time/resource trade-off problem in project scheduling with genetic algorithms. *Applied Mathematics and Computation, 191*(2), 451-456.

Ranjibar, M., Kianfar, F., & Shadrokh, S. (2008). Solving the resource availability cost problem in project scheduling by path relinking and genetic algorithm. *Applied Mathematics and Computation, 196*(2), 879-888.

Reeves, C. R. (1995). *Modern Heuristic Techniques for Combinatorial Problems.* London, UK: John Wiley & Sons.

Sabuncuoglu, I., & Bayiz, M. (2000). Analysis of reactive scheduling problems in a job shop environment. *European Journal of Operational Research, 126*(3), 567-586.

Stoop, P., & Wiers, V. (1996). The complexity of scheduling in practice. *International Joural of Operations & Production Management, 16*(10), 37-53.

Suresh, V., & Chaudhuri, D. (1993). Dynamic scheduling - a survey of research. *International Journal of Production Economics, 32*(1), 53-63.

Vanhoucke, M. (2006). Work continuity constraints in project scheduling. *Journal of Construction Engineering & Management, 132*, 14-25.

Vanhoucke, M. (2013). *Project Management with Dynamic Scheduling: Baseline Scheduling, Risk Analysis and Project Control.* London.: Springer Heidelberg.

Vieira, G. E., Hermann, J. W., & Lin, E. (2003). Rescheduling manufacturing systems: a framework of strategies, policies and methods. *Journal of Scheduling, 6*(1), 36-92.

Webster, S., & Azizoglu, M. (2001). Dynamic programming algorithms for scheduling parallel machines with family setup times. *Computers & Operations Research, 28*(2), 127-137.

**Table 1.** Expert Validation Results

| Necessity for New System for Schedule Optimization | % |
|---|---|
| Baseline schedules preparation | 80.0% |
| Revised schedules preparation | 86.7% |
| Progress updates & look-ahead schedules preparation | 53.3% |
| What-If schedules preparation | 66.7% |

| Practical application of proposed DS system | % |
|---|---|
| Baseline schedules preparation | 80.0% |
| Revised schedules preparation | 86.7% |
| Progress updates & look-ahead schedules preparation | 66.7% |
| What-If schedules preparation | 60.0% |

| Effectiveness in Baseline/Revision Optimization | % |
|---|---|
| Strongly effective | 66.7% |
| Seems effective | 33.3% |
| Not sure | 0.0% |
| Not effective | 0.0% |

| Effectiveness in Progress Updates/Look-Ahead Optimization | % |
|---|---|
| Strongly effective | 33.3% |
| Seems effective | 46.7% |
| Not sure | 20.0% |
| Not effective | 0.0% |

| Effectiveness in What-if Optimization | % |
|---|---|
| Strongly effective | 60.0% |
| Seems effective | 20.0% |
| Not sure | 13.3% |
| Not effective | 6.7% |

| Ease of Use by Construction Projects’ Planning Teams | % |
|---|---|
| Easy to use | 26.7% |
| Easy to use after practicing | 46.7% |
| Complicated and requires a lot of practice | 20.0% |
| Only very experienced planners can use it | 6.7% |

| Effectiveness of Software Integration with Current Project Management Practices | % |
|---|---|
| Strongly effective | 60.0% |
| Seems effective | 33.3% |
| Not sure | 6.7% |
| Not effective | 0.0% |

| Ability to define/modify additional project/optimization data | % |
|---|---|
| Strongly effective | 33.3% |
| Seems effective | 66.7% |
| Not sure | 0.0% |
| Not effective | 0.0% |

| Ability to view optimized alternatives | % |
|---|---|
| Strongly effective | 26.7% |
| Seems effective | 66.7% |
| Not sure | 6.7% |
| Not effective | 0.0% |