JIOS, VOL. 42, NO. 2 (2018) SUBMITTED 08/17; ACCEPTED 06/18

DOI: https://doi.org/10.31341/jios.42.2.2 UDC 005.8
Open Access Preliminary Communication

Tracking Predictive Gantt Chart for Proactive Rescheduling in Stochastic Resource Constrained Project Scheduling

Mario Brčić *mario.brcic@fer.hr*
Faculty of Electrical Engineering and Computation
University of Zagreb, Zagreb, Croatia

Danijel Mlinarić *danijel.mlinaric@fer.hr*
Faculty of Electrical Engineering and Computation
University of Zagreb, Zagreb, Croatia

Abstract

Proactive-reactive scheduling is important in the situations where the project collaborators need to coordinate their efforts. The coordination is mostly achieved through the combination of the shared baseline schedule and the deviation penalties. In this paper, we present an extension of predictive Gantt chart to the proactive-reactive scheduling needs. It can be used to track the evolution of the relationship between dynamic and static elements through the time. The dynamic elements are evolving probability distributions due to the uncertainty and revealed information. The static elements are time-agreements in the baseline schedule. We demonstrate that in the state-of-the-art proactive-reactive scheduling, the baseline schedule is agnostic to the information received during the project execution. The sources of such inflexibility in the problem model and the scheduling methods are analyzed. The visualization is highlighted as a precursor to developing new methods that proactively change the baseline schedule in accordance with the gained information.

Keywords: Project scheduling, Proactive-reactive scheduling, Gantt chart, Stochastic RCPSP

1. Introduction

Stochastic Resource Constrained Project Scheduling (SRCPSP) deals with scheduling the projects that feature uncertainty in some of its parameters. It has gained the interest of the research community in the last two decades. There are several surveys into the project scheduling under uncertainty [1]–[4]. The developments have taken one of the two main research directions: pure reactive and proactive-reactive scheduling [5].

Pure reactive scheduling optimizes the expected schedule makespan or some other regular measure. The aim is to achieve the compliance with the due dates. The stochastic project is seen as a multistage process, where execution policies deal with uncertainty by creating the schedule in stages [6].

As the projects' complexity increases, projects depend on the growing number of collaborators, and the synchronization between them becomes an important issue. Activities are outsourced, and resources need to be acquired from suppliers during the project execution. The most extreme example of such practices is just-in-time (JIT) manufacturing.

In pure reactive scheduling, the variances of activities' start times tend to be too great, and there are often no evident accumulation points. This makes the synchronization of collaborators' efforts in their pre-activity planning and preparation difficult. It has been pointed out in [2] that the drawback of pure reactive approach is that it does not generate the complete baseline schedule before the initiation of the project.

A baseline schedule contains predictive start times of activities which can be interpreted and used as time-arrangements between the project collaborators. Baseline schedule allows for: allocation of resources to different activities, quoting competitive and reliable due dates, scheduling the activities in accord with all parties within the inbound and outbound supply chain, to agree on time windows for work to be done by subcontractors, to share production schedules with suppliers on a continuous basis using Internet technology, for making cash flow projections, to measure the performance of both management and shop floor personnel, and to project monitoring and control [7].

Proactive-reactive scheduling is the method of choice when, in addition to uncertainty, the project also seeks better means for synchronization between the project collaborators. It uses the baseline schedule, some forms of robustness, and an execution policy. Proactive-reactive scheduling works in the two phases. In the first, proactive phase, a protected baseline is created before the project execution. Such schedule must be as insensitive to the execution variability as dictated by the chosen robustness measure. The most commonly used notion of robustness that focuses on reduction in rescheduling variability is the solution robustness. It penalizes the deviations of the realized scheduled from the predicted baseline schedule. However, using only this form of robustness in the cost function would result in trivially secured schedules that take no risks and have large makespans. For that reason, the quality robustness is added to the objective function to maximize the probability of completing the project before the due date. These two components are competing objectives and their trade-off is most often defined by the appropriately selected numerical scalarization. Proactive phase is related to the safe scheduling [8], [9] and stochastic inventory theory. In the second, reactive phase, the project is executed, and activities are started by the execution policy that deals with the uncertainty realizations in a similar way as do the policies in pure reactive scheduling. The difference here is that the policy tries to start activity executions as close as possible to the baseline schedule times. In that way, it keeps the deviation costs low.

Among others, in [10] it has been pointed out that there is a wide gap between the project management discipline and the research on project management. Part of the reason is that practitioners' have no knowledge of the state-of-the-art algorithmic developments. On the other hand, current algorithmic procedures cannot deal with the full complexity and generality of the real-world projects.

Gantt chart is an important tool in the project management. It enables a visual representation of the project's progress. Predictive extensions of Gantt chart have been used in the scheduling of stochastic problems [8],[11] for illustrating and emphasizing different aspects of solutions. We aim to investigate the effect of the gradually-incoming information on the relationship between the baseline schedule and the anticipated behavior of the project executing system. The Gantt chart is extended with the necessary elements to enable the aforementioned inquiry.

The contributions of this paper are:

- An extension of predictive Gantt chart suitable for proactive-reactive scheduling
- The behavior of state-of-the-art proactive-reactive method from [11] is visually tracked using the extended chart and the discrepancy is shown between the behaviour of the model and our intuitive notion of real-world projects's features
- Analysis of current models and procedures in order to pinpoint the responsibility for such discrepancy

This paper is organized as follows. The related work in the visualization of project progress using Gantt chart is listed in Section 2. The example problem and state-of-the-art scheduling method are given in Section 3. We present our extension of predictive Gantt chart in Section 4. Also, in Section 4 state-of-the-art method for proactive-reactive scheduling is explained and its results are put into the analysis using the new chart. Future research directions are drafted in Section 5, and we offer conclusion in Section 6.

2. Related Work

In this section, we shall cover the related Gantt chart-based visualizations. Initially, the progress on projects was kept in written or tabular form [12]. However, it was hard to get an intuitive grasp of the project's progress from such formats. The important stepping stone in the history of project management was the creation of Gantt charts by Henry Gantt at the beginning of the 20th century [13]. It is a visualization tool for deterministic projects that shows planned and actual progress against the horizontal time scale. Each project activity gets its horizontal swim-lane, and a horizontal bar is placed to mark the time interval when the activity execution should take place. The original Gantt chart deals only with the deterministic durations and does not offer solutions for unforeseen events. Also, it assumes there are no resource constraints, i.e. that we always have enough of resources to execute tasks as soon as they are feasible. With time, many extensions were made to incorporate the needs of project management, especially with the advent of computers.

Predictive Gantt chart (shown in Figure 1) was first presented in [8]. It is used as a mean of visualizing scheduling decisions in stochastic environments. Similar to the plain Gantt chart, horizontal bars are used for each activity. Horizontal bars present spaces where cumulative distribution functions (CDF) are displayed for activity start and finish time. The bottom and the top of each bar represent probabilities of zero and one, respectively. The leftmost curve in each horizontal swim-lane is the CDF for activity's start time (for example activity A has a step function at a time-point 9), and the rightmost curve is the CDF for activity's finish. The horizontal bar at the bottom holds the CDF for the project finish, and marks the due date as a vertical line (in this example at the time-point 60).

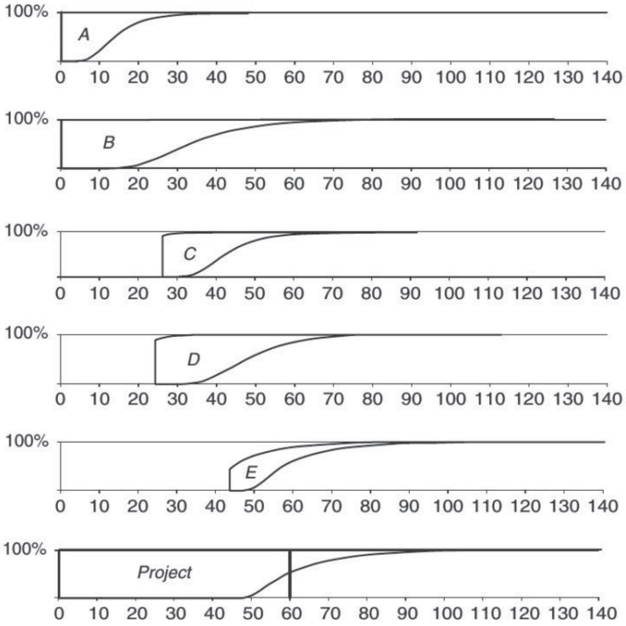

Figure 1. Predictive Gantt chart from [8]

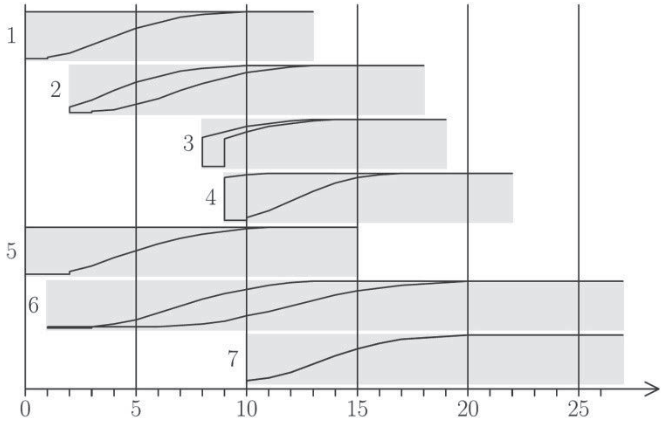

Figure 2. Predictive Gantt chart from [11]

Deblaere et al. [11] used a predictive Gantt chart to visualize and compare the functioning and performance of three proactive-reactive execution policies. The subtle change in the chart in Figure 2 is that the horizontal bars also hold the information about probability distribution support. The left and right edges of each bar represent the minimum start and maximum finish time, respectively. We can notice that the both predictive Gantt charts in Figure 1 and Figure 2 have been created at the time-point *t=0*, hence using only the probabilistic information available at that time. The information available at the time-point *t=0* sufficed for the theoretical and algorithmic assumptions of the methods presented there. The predictive Gantt chart in Figure 2 is taken as the basis for our extensions in Section 4. The extension is then applied to the proactive-reactive scheduling to uncover shortcomings in the existing scheduling approaches.

	Remaining Effort	Finish [E]
Project	40.29h - 65.71h	**06/12/18**
1	5.76h - 14.24h	**29/11/18**
2	5.22h - 20.78h	**30/11/18**
3	3.22h - 18.78h	**04/12/18**
4	7.76h - 16.24h	**05/12/18**
5	4.88h - 9.12h	**06/12/18**

Figure 3. LiquidPlanner's Timeline View

LiquidPlanner [14] is a priority-based predictive project management solution used in practice. It uses ranges of estimates to infer probability distributions of activity durations. The simulation is employed to create timeline view, a form of a predictive Gantt chart that is used to track the progress of activities. Scheduling decisions are not optimized but are delegated to the users by specifying task priorities on top of which the system creates the current schedule in purely-reactive fashion. Also, the user has to specify the allocation of specific resources to the tasks, although this is also a part of the optimization problem. Collaborative synchronization is not explicitly dealt with but is being delegated to the user to achieve manually. The LiquidPlanner's timeline view is given in Figure 3. The difference here is that the chart is updated in time with the newly available information, hence tracking the project evolution. The horizontal bars spread between the minimum start and maximum finish times. However, CDFs are not displayed in the chart, only the expected values and the inter-percentile bars for finish times. By not optimizing many of the aspects of choices on projects, this tool shows the gap between the practical use and the research results.

3. Example

In this section, we shall define example project and the state-of-the-art method from the literature on which we shall explain and apply extended chart to analyze behavior.

3.1. The Input Project

We shall use a project defined by the project network in Figure 4 as an example through the rest of this paper. SRCPSP is defined as a tuple *(V, E, d, R, B, D, δ, c)*.

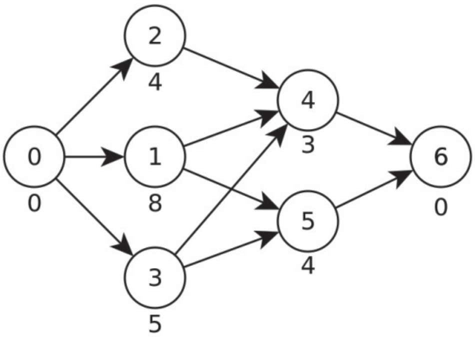

Figure 4. Project network for the example project

Labeled nodes represent non-preemptive activities in the set *V*. There are *n=5* non-dummy activities, and two dummy activities labeled *0* and *n+1* that are used to mark the beginning and the finish of the project. The edges represent asymmetric zero-lag finish-start precedence relationships in the set *E*. We have one resource in the set of renewable resources *R* with the availability of 15 units defined in the matrix *B*. The demands from the matrix *D* of each activity on that resource are given below each activity's node in Figure 4. The project's due date is *δ =14*. A random vector *d* represents uncertain activity durations. All the non-dummy activities in the example have independently uncertain duration, each following discretized beta distribution with shape parameters *α=2* and *β=5*. For each activity, beta distribution with support *[0,1]* was scaled to the supports given in Table 1. The discretization is done with the ceiling function so that all events in the real interval *(t,t+1]* are assigned to the discrete event *t+1*. Dummy activities have a deterministic duration

Activity label	Lower bound	Upper bound
1	2	8
2	1	12
3	0	11
4	3	9
5	2	5

Table 1. Scaling ranges for beta distributions

of zero. The objective function c is given in Equation (1). The inputs to c are the used execution policy Π and the baseline schedule $\vec{\mathfrak{s}}$. The remaining element in the formula is the realized schedule s. Both solution and quality robustness are included.

$$\mathrm{c}(\Pi, \vec{\mathfrak{s}}) = \sum_{i=1}^{n+1} \{\mathbb{E}[c_i^{+} \cdot max(s_i - \mathfrak{s}_i, 0) + c_i^{-} \cdot max(\mathfrak{s}_i - s_i, 0)]\} \qquad (1)$$

The solution robustness in Equation (1) is comprised of summands for $i=1$ to $i=n$. The remaining summand is for the last, dummy, activity and denotes the quality robustness because the $\mathfrak{s}_{n+1}$ is set to the due date δ. Both robustness measures penalize separately overrun (unit price c_i^{+}) and under-run (unit price c_i^{-}) of the baseline time. The objective function of the form given in the Equation (1) was first proposed in [11]. We shall refer to the used solution robustness as the *asymmetric stability measure*. If $\forall i \in V \backslash \{0, n+1\}(c_i^{-} = c_i^{+})$, then we get a *symmetric stability measure* from [2]. The majority of proactive scheduling procedures described in the literature commonly measure solution robustness using symmetric stability measure [7].

Activities 3 and 4 are inflexible, meaning that they incur costs for missing the predicted start time. Activity 3 has under-run and overrun unit costs of $c_3^{-} = 5$ and $c_3^{+} = 2$, respectively, for the actual start time. Activity 4 has under-run and overrun unit costs of $c_4^{-} = 1$ and $c_4^{+} = 6$. All the activities except 3 and 4 are completely flexible, meaning that they incur no costs for missing the predicted start time. The project has a penalty for exceeding the due date $c_6^{+} = 3$, and bonus for early project completion of $c_6^{-} = -19$ (negative cost is a bonus).

3.2. Example Scheduling Method

In our example, we shall use Resource-based Policies with Release Times (RPRT) optimized with Stochastic-Based Descent (SBD) from [11]. It is state-of-the-art, top performing method from the literature. As such, it is the most appropriate for showing the deficiencies of current approaches. Without going into the details of SBD optimization procedure, which is irrelevant in our exposition, we shall explain RPRT policy family. RPRT policies are parameterized by the vector of priorities π and the vector of release times τ for non-dummy activities. The RPRT policy at each time-point t uses the parallel schedule generation scheme to start activities with release times greater than or equal to t. Activity starting is done in order of priorities π. For each RPRT policy, an optimal baseline schedule can be calculated from the simulation traces. This is done using the critical fractile based solution for the newsvendor problem. Newsvendor problem balances the costs of overrun and underrun to find the point that generates the least expected deviation cost.

We notice that the execution policy, which controls the execution of the project, only starts activities at the selected times. However, it does not change the baseline schedule in any other way. This constraint is inherent in the definition of the RPRT policy. We shall see, using the visualization, what does such constraint entail.

4. Tracking Predictive Gantt Chart

We have created the code for drawing tracking predictive Gantt charts from the simulation trace data. It is written in Python, using matplotlib library for graphing. The code is available at the public repository[1].

In extending the chart in Figure 2, from [11], several elements are added:

- **time tracking** – the chart is taking in the new information that is revealed with the passage of time and creates conditional cumulative distribution functions that correspond to the new situation.
- **baseline schedule times** – for each activity
- **ideal baseline-rescheduling times** – for each unstarted activity. The same method-specific procedure used to find the baseline schedule before the project start is used for calculating the ideal rescheduling times. These times are the ideal to renegotiate between the project collaborators in the light of new data. The aim is to renegotiate as close as possible to these times by finding the best interpolation when considering the rescheduling costs.

The extended chart is demonstrated on tracking the execution of state-of-the-art proactive-reactive scheduling method described in Subsection 3.2 on the problem defined in Subsection 3.1. We have used the simulation library from [15] to calculate the RPRT policy μ for this project using the SBD procedure. The simulation traces, necessary for the graphing of the current state of the project, were generated for one scenario through all the execution time-points. We intentionally show the situation with an opportunity for improvement, which the method in question does not utilize.

In Figure 5, the predictive Gantt chart generated by μ at time-point $t=0$ is given. It is similar to the chart in Figure 2, apart from the fact that they are created for different projects. Both are created at the time-point $t=0$, using only the information available before the start of project execution. On this chart, the baseline times for activities are also marked, using dashed vertical lines positioned at the corresponding time. The variance in start times is reduced by using release times in μ. For example, inflexible activity 4 has start time CDF distorted by the release time $\tau_4=7$ to have the initial step in CDF. Without release time, the CDF's initial shape would be smoothed towards probability 0 like it is in the case of activity 5.

The predictive Gantt chart is based on computational scheduling simulations, so it carries the information not only of the precedence but also of the resource constraints. In the Gantt chart, horizontal bars are non-overlapping in time for the pairs of activities constrained by precedence or resource availability. In contrast, in predictive Gantt which is the accumulation of Gantt charts for many simulation scenarios, such strict non-overlapping becomes fuzzy in the probabilistic representation. For example, activity 1 precedes both activities 3 and 4, but their horizontal bars overlap in time (see Figure 5).

In Figure 6, the chart at the time-point $t=4$ is depicted. The vertical dashed line across the chart marks the current time. The chart is divided into two parts: the

---

[1] https://github.com/mbrcic/tp_gantt

predictive stochastic part for $t>4$ and realized deterministic part for $t\leq4$. The realized part shows the additional information we have received relative to $t=0$: activities 2 and 3 still have not finished with execution, meaning that their duration is greater than 4. No other activity started with the execution due to the resource and precedence constraints. The gained information affected the predictive part. The cumulative distribution functions, as dynamic elements, changed in response to the

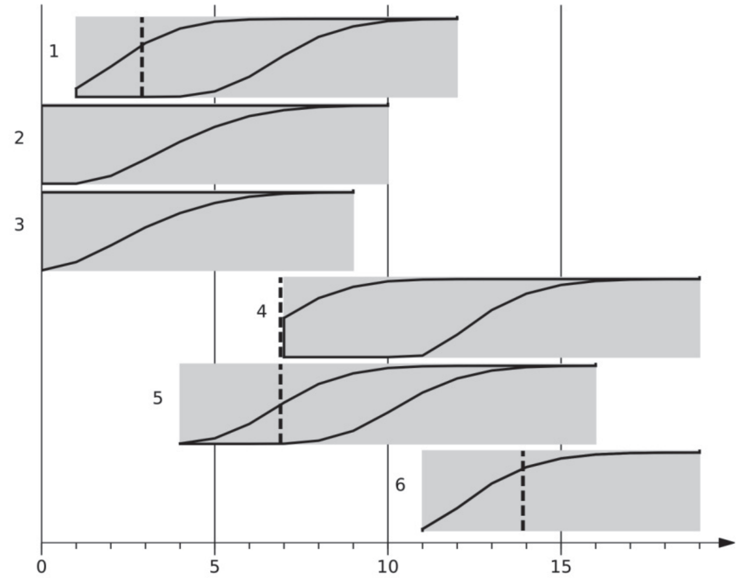

Figure 5. Tracking predictive Gantt chart at timepoint t=0

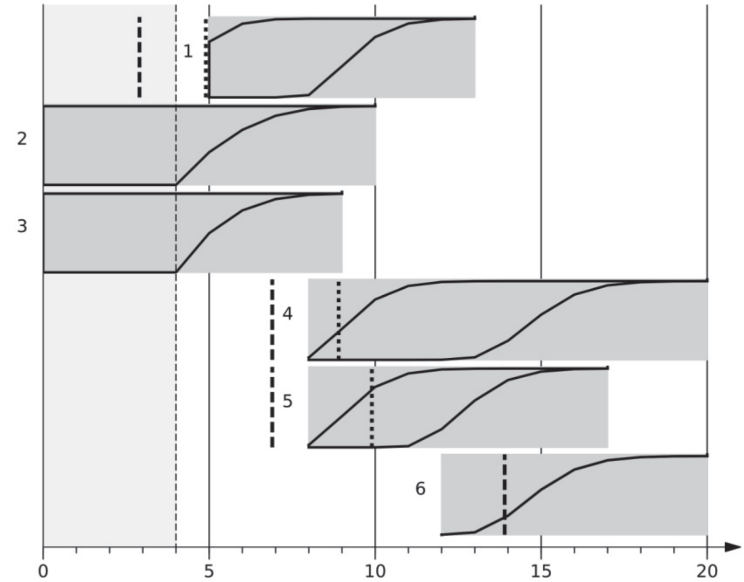

Figure 6. Tracking predictive Gantt chart at timepoint t=4

new information. The baseline schedule remained unaltered, as RPRT policies do not change the baseline. It is evident that the current baseline times for activities 1,4, and 5 are not optimal. In fact, they are not even feasible as the start time CDFs have drifted to the right. The dotted vertical lines show the ideal baseline-rescheduling times. The identical procedure for the news-vendor problem was employed to calculate the new times, aiming at the same critical fractiles in CDF as in the initial solution.

4.1. Analysis

The extended chart revealed that baseline times could become not just suboptimal, but even infeasible. The information of infeasibility or suboptimality can even become evident many time-steps in advance, so there is a potential for proactive recourse. Maybe the baseline schedule times can be renegotiated. RPRT or other methods from the literature do not proactively change the baseline times. In that way, execution incurs greater rescheduling costs due to under-run and overrun which reflect the increased pressure for effort on the project executing environment. It is important to identify potential sources of inflexibility in models and methods that are responsible for such myopic behavior.

Regarding potential sources of inflexibility in scheduling methods, we can examine RPRT. As already mentioned in Subsection 3.2, RPRT policy family does not allow for proactive baseline rescheduling. Its definition allows only for manipulation with the activities' starts. Also, RPRT parameters are adjusted by simulation to perform well on projects starting from time-point *t=0*, which means that they are optimized on CDFs that have access only to the limited amount of information. Such, static RPRT policy, must be well-performing on a great number of scenarios. However, with the passage of the time, the number of relevant scenarios reduces, due to stochastic filtering. Furthermore, RPRT policy may have suboptimal performance on the set of relevant scenarios that can have different characteristics from the previous sets. For example, we can see in Figure 5 that the release time τ_4 reduces the start time variance of activity 4 by creating a small initial step in CDF. In Figure 6, the CDF has drifted to the right, and its initial part has a smooth initial transition from 0. Potentially, the release time could be adjusted to reduce the variability in such and similar situations. As the current proactive-reactive methods only manipulate actual moments that activities are started, it is evident that proactive rescheduling significantly increases the complexity of the optimization problem. At each time-step, any unscheduled part of the baseline schedule might proactively change. This makes for the much greater complexity of the control space and for the increase in the computational cost of optimizing in that space. New algorithms might improve performance in two ways:

- act only from the static aspect; adjust the baseline schedule to the new information
- change both the dynamic and static elements; try to reassemble the new tracking predictive Gantt chart in accordance with the new data to get better overall performance. The reassembling is performed by changing the

scheduling process that created the chart and then the baseline schedule is proactively adjusted to the new chart.

Regarding the inflexibility responsible for this constrained behavior in the presented model, we must check the used robustness measures in the objective functions. We shall focus on the solution robustness as it deals with the rescheduling of activities. Solution robustness is defined as the difference between the baseline schedule and the realized schedule during the project execution [7]. Such definition does not measure any intermediate changes to the baseline. Proactive changes to the baseline are done in intermediate steps, which means that even if we did proactive rescheduling, we could not reduce the execution cost. That makes proactive rescheduling useless and ineffective. We could try to extend existing measures to enable intermediate changes. However, such proactive changes should cost less than their equivalent decisions of starting activities. Cost-based flexibility (CBF) family of measures was proposed in [16]. In the CBF, the intermediate changes are accounted for, and the rescheduling cost depends on the size of the change as well as *the temporal distance of the change*. The way to make proactive changes effective is by making the cost of changes decreasing with the temporal distance of the changes. Such notion is aligned with the intuition of real-world projects and the idea of proactive rescheduling. Furthermore, it was found in [16] that symmetric stability measure, extended under the framework of the CBF family to account for intermediate changes, still does not make proactive changes effective. The reason is that all the changes have the same cost. In the case of asymmetric stability measure, several extensions are possible. Some rule out proactive changes as ineffective, while others allow for limited form due to the asymmetry in pricing. When deciding on the form of a measure based on CBF, we must be careful to align the model's notion of rescheduling flexibility with actual flexibility in our project executing environment. Failure to do so would either over-constrain or under-constrain the system. In the case of over-constraining, we would get a situation similar to that of symmetric stability measure where we have a reduced utility of proactive changes. In the case of under-constrain, we would erode the utility of the baseline schedule as the behavior becomes close to the pure reactive scheduling. It is evident that the proactive rescheduling depends on striking a balance between the proactivity of the baseline schedule in current proactive-reactive procedures and the flexibility of pure reactive scheduling. Ideally, we would like to have a gradual transition from the region of fixed baseline in the near future towards the more flexible region in a distant future where the baseline is open to changes under reasonable rescheduling costs.

5. Future Developments

We have highlighted the tracking predictive Gantt chart as a precursor to the theoretical and algorithmic developments in the area of proactive rescheduling and proactive-reactive scheduling in general. In correspondence with the different elements added into the theoretical or algorithmic consideration, new elements can be easily added to the graph to help with the tracking, intuition regarding consequences and repercussions. Using this graph, we can drill into the questions of negative bias of defined policy families or search algorithms. This means finding inefficiencies in controlling or searching procedures to inspire the creation of new algorithms. Different execution policies can be compared beyond the initial point to find how they maintain the relationship between static and dynamic elements. Also, we can check the effect and structural reasonableness of model elements, such as used robustness measures. We can check each in comparison to common sense and expected behavior in real-world problems to see if our model covers the assumptions well, or if it adds too many fictive assumptions that complicate or disable attempts of applications.

The graphing can be implemented in C++ and embedded into the existing scheduling library from [15] to get more responsive and faster system that can cover the visualization of many scenarios, and the creation of interactive sequential animations. This can enable researchers to cover more ground when contrasting many different aspects to find algorithmic blind-spots and identify inefficiencies.

In [16] authors have proposed another family of functions, CBF, that enable intermediate changes. The emphasis in that family is on reducing the baseline rescheduling costs with the temporal distance of change. Near-future changes incur greater rescheduling cost than changes of the similar size set in the further future. Such view is aligned with the intuitive notion that adjustments to changes take time. For near-future changes, there is less time for adjustments, while for further future there is plenty. Also, farther-future plans are viewed less strictly due to the involved uncertainty, and it would be illusory to think that everything can be predicted with perfect precision. Events in far future are preceded by long sequences of uncertain realizations and hence have greater variance.

6. Conclusion

We have presented the introduction into the issues with modern projects; dealing with uncertainty and synchronization between the project collaborators. Proactive-reactive scheduling is the research direction that deals with such requirements. We extended the predictive Gantt chart from [8] to track the execution of the stochastic project through the time. On the same chart, we superimposed dynamic elements (due to evolving uncertainty) and static elements (due to time-agreements in the baseline schedule) in order to track the evolution of their relationship. The presented extension is conceptually basic and allows for extensions according to the future research directions. We have demonstrated, through the example, the situation where the behavior of current methods does not match the real-world intuition. Namely, baseline times can become suboptimal or even infeasible and there are no attempts at adjusting them proactively. It shows that the theoretical and algorithmic inquiry should be taken to the issue of proactive rescheduling. The analysis was conducted into the sources of such inflexibility in scheduling methods and problem model. Current scheduling methods by definition do not attempt to change the baseline schedule proactively. Commonly used solution robustness does not account for intermediate changes, and that makes the proactive rescheduling ineffective. The CBF family of measures from [16] constitutes a promising research direction as it makes the rescheduling cost dependent on the temporal distance of the change. Such dependency can make proactive rescheduling an effective scheduling behavior.

References

[1] A. J. Davenport and J. C. Beck, “A survey of techniques for scheduling with uncertainty,” Toronto, Canada, unpublished manuscript, 2002.

[2] W. Herroelen and R. Leus, “Robust and reactive project scheduling: a review and classification of procedures,” *International Journal of Production Research*, vol. 42, no. 8, pp. 1599–1620, 2004.

[3] W. Herroelen and R. Leus, “Project scheduling under uncertainty: survey and research potentials,” *European Journal of Operational Research*, vol. 165, no. 2, pp. 289–306, 2005.

[4] M. Brcic, D. Kalpic, and K. Fertalj, “Resource constrained project scheduling under uncertainty: a survey,” in Proceedings of the 23rd Central European Conference on Information and Intelligent Systems, Varazdin, Croatia, 2012, pp. 401–409.

[5] P. Lamas and E. Demeulemeester, “A purely proactive scheduling procedure for the resource-constrained project scheduling problem with stochastic activity durations,” *Journal of Scheduling*, vol. 19, no. 4, pp. 409–428, Aug. 2016.

[6] R. H. Möhring, F. J. Radermacher, and G. Weiss, “Stochastic scheduling problems I — General strategies,” *Zeitschrift für Operations Research.*, vol. 28, no. 7, pp. 193–260, 1984. doi: 10.1007/BF01919323

[7] E. Demeulemeester and W. Herroelen, *Robust Project Scheduling*. Now Publishers Inc, 2011.

[8] K. R. Baker and D. Trietsch, *Principles of Sequencing and Scheduling*, 1 edition. Hoboken, N.J: Wiley, 2009.

[9] K. R. Baker and D. Trietsch, “Safe scheduling: setting due dates in single-machine problems,” *European Journal of Operational Research*, vol. 196, no. 1, pp. 69–77, Jul. 2009.

[10] M. Vanhoucke, *Project Management with Dynamic Scheduling: Baseline Scheduling, Risk Analysis and Project Control*. Springer Science & Business Media, 2013.

[11] F. Deblaere, E. Demeulemeester, and W. Herroelen, “Proactive policies for the stochastic resource-constrained project scheduling problem,” *European Journal of Operational Research*, vol. 214, no. 2, pp. 308–316, Oct. 2011.

[12] M. Kozak-Holland, *The History of Project Management*. Multi-Media Publications, 2011.

[13] W. Clark and H. L. Gantt, *The Gantt chart, a working tool of management;* New York, Ronald Press, 1923.

[14] "LiquidPlanner - Online Project Management Software for Manufacturing Teams," *LiquidPlanner*. [Online]. Available: https://www.liquidplanner.com/. [Accessed: 30- Jul- 2017].

[15] M. Brčić and N. Hlupić, "Simulation library for resource constrained project scheduling with uncertain activity durations," in Proceedings of the 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014, pp. 1041–1046. doi: 10.1109/MIPRO.2014.6859722

[16] M. Brčić, D. Kalpić, and M. Katić, "Proactive reactive scheduling in resource constrained projects with flexibility and quality robustness requirements," *Lecture Notes in Computer Science,* Vol. 8596, pp. 112–124, 2014. doi: 10.1007/978-3-319-09174-7_10