한국지능정보시스템학회논문지 제14권 제1호 2008년 3월(pp.117~129)

Cost-Based Directed Scheduling : Part II, An Inter-Job Cost Propagation Algorithm

MIN SOO SUH
현대제철
(mssuh@hyundai-steel.comr)

JAE KYEONG KIM
경희대학교 경영대학
(jaek@khu.ac.kr)

The cost-based scheduling work has been done in both the Operations Research (OR) and Artificial Intelligence (AI) literature. To deal with more realistic problems, AI-based heuristic scheduling approach with non-regular performance measures has been studied. However, there has been little research effort to develop a full inter-job cost propagation algorithm (CPA) for different jobs having multiple downstream and upstream activities. Without such a CPA, decision-making in scheduling heuristics relies upon local, incomplete cost information, resulting in poor schedule performance from the overall cost minimizing objective. For such a purpose, we need two types of CPAs : intra-job CPA and inter-job CPA. Whenever there is a change in cost information of an activity in a job in the process of scheduling, the intra-job CPA updates cost curves of other activities connected through temporal constraints within the same job. The inter-job CPA extends cost propagation into other jobs connected through precedence relationships. By utilizing the cost information provided by CPAs, we propose cost-based scheduling heuristics that attempt to minimize the total schedule cost.

This paper develops inter-job CPAs that create and update cost curves of each activity in each search state, and propagate cost information throughout a whole network of temporal constraints. Also we propose various cost-based scheduling heuristics that attempt to minimize the total schedule cost by utilizing the cost propagation algorithm.

논문접수일 : 2007년 02월 게재확정일 : 2007년 10월 교신저자 : 김재경

1. Introduction

Most of the scheduling literature has focused on single performance measures, referred to as regular measures, that are non-decreasing in job completion times, which include mean flow time, mean lateness, percentage of jobs tardy, and mean tardiness (Baker and Scudder, 1990). However in more complicated scheduling environments in industrial settings, it is required to consider various cost components like the tardiness and inventory holding costs, and to minimize the total cost objective value of a schedule. For instance, in a Just-In-Time(JIT) scheduling environment, an optimal schedule means all jobs are scheduled to finish exactly on their due dates. If a job finishes later than its due date, a tardiness penalty cost occurs, while earlier completed jobs cause additional inventory holding costs. Depending upon the characteristics of the problem considered, a variety of costs including changeover and labor costs need to be incorporated as well.

The cost-based scheduling work has been done in both the Operations Research(OR) and Artificial Intelligence(AI) literature. In OR, due to the excessive complexity inherent in scheduling problems, optimization techniques have been applied to a very restricted and simplified problem. To deal with more realistic problems, heuristic scheduling approach with non-regular performance measures has been studied(Baker and Scudder, 1990; Fox, 1983; Sadeh, 1991; Smith and Cheng, 1993). In the AI literature, the Constraint Satisfaction Problem technique has been applied to solve scheduling problems with more realistic constraints(Beck et al., 1997a; Beck et al., 1997b; Easton and Goodale, 2005; Fox, 1983; Minton et al., 1992; Sadeh, 1991; Zweben et al., 1994). In particular, Sadeh(1991) extended the constraint directed scheduling paradigm to deal with constrained optimization problems using the contention/reliance texture-based heuristic. Ye et al.(1994) applied the Constrained Logic Programming paradigm to the job shop scheduling area. However, from the viewpoint of cost optimization, the literaturenoted above is lacking a method for propagating cost information through temporal constraints, and evaluating a global cost impact incurred by a decision-making in scheduling heuristics. Although Sadeh(1991) presented a *cost propagation algorithm*(CPA) for that purpose, its range for cost propagation was limited within a job restricted to have only multiple downstream activities. Therefore, there has been little research effort to develop a full inter-job CPA for different jobs having multiple downstream and upstream activities. Without such a CPA, decision-making in scheduling heuristics relies upon local, incomplete cost information, resulting in poor schedule performance from the overall cost minimizing objective.

For such a purpose, we need two types of CPAs : *intra-job* CPA and *inter-job* CPA. Whenever there is a change in cost information of an activity in a job in the process of scheduling, the intra-job CPA updates cost curves of other activities connected through temporal constraints within the same job. The inter-job CPA extends cost propagation into other jobs connected through precedence relationships. Our previous research(Kim and Suh, 2007) is developing a heuristic Intra-job CPA. By utilizing the cost information provided by intra-job CPA and inter-job CPA, we propose cost-based scheduling heuristics that attempt to minimize the total schedule cost.

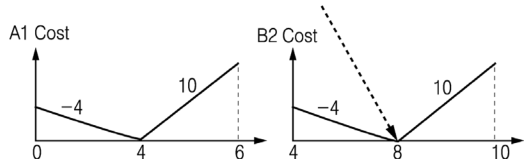
The main aim of this paper is to develop inter-job CPAs that create and update cost curves of each activity in each search state, and propagate cost information throughout a whole network of temporal constraints.

The organization of this paper is as follows. Section 2 describes inter-job CPAs to propagate cost information along temporal constraints. Section 3 describes cost-based scheduling heuristics with their advantages and limitations, followed by concluding remarks and future research directions in Section 4.

2. Inter-Job Cost Propagation Algorithm

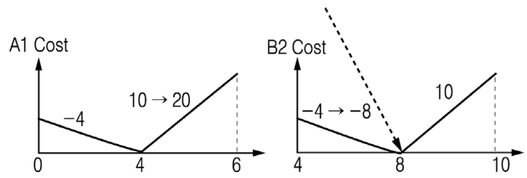
Under the intra-job cost propagation described in our previous research(Kim and Suh, 2007), the CCs of each activity maintain the cost information related to those activities only belonging to the same job. This does not provide global cost information when there are temporally connected activities between different jobs. For example, suppose the two activities A_1 and B_2 in the respective jobs J_1 and J_2 are temporally connected as shown in [Figure 1]. In this case, the intra-job CPA does not affect any CCs since there is no change in the feasible time window of the two activities. However, when the earliest start time of activity A_1 moves to the right over the time point 4, not only the tardiness cost of job J_1 occurs, but also that of job J_2 is accompanied since the earliest start time of activity B_2 moves over its optimal start time due to the temporal propagation. This means that the inter-job cost propagation needs to increase the slope of the tardiness CC of activity A_1 from 10 to 20 considering the tardiness cost of the other job J_2. Likewise, the slope of the inventory CC of activity B_2 should decrease from -4 to -8 by adding the inventory CC slope of activity $A1$. [Figure 2] shows the results when this inter-job cost propagation is applied to the two activities.

In the inter-job cost propagation between different jobs, an initial conflict may arise in their optimal start times as shown in [Figure 3] where the initial optimal start time of activity A_1 goes beyond the initial optimal start time of activity B_2. In such case the inventory cost information of A_1 should be reflected to both the inventory and tardiness CCs of

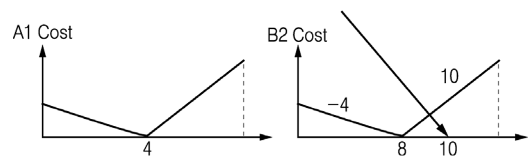


du(A1)=4, du(B2)=4, uic(A1)=4, uic(B2)=4, utc(J1)=10, utc(J2)=10

[Figure 1] When two activities in different jobs are temporally connected, no CC slope change occurs in any activities under the intra-job cost propagation



[Figure 2] Under the inter-job cost propagation, the slope of the tardiness CC for activity A_1 increases from 10 to 20, and the slope of the inventory CC for activity B_2 decreases from −4 to −8



ost(A2)=4, du(A2)=6, ost(B2)=8

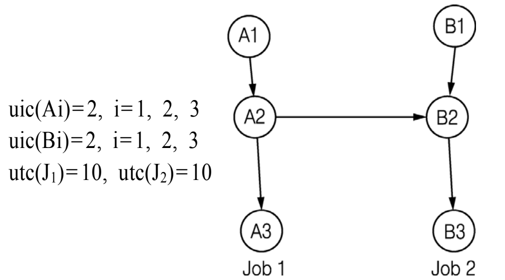
[Figure 3] The initial optimal start time of activity A_1 is greater than that of activity B_2, causing a conflict between the optimal start times of activities, A_1 and B_2

B_2. In this study, we assume there is no initial conflict between two different jobs in their optimal start times, and also that only one temporal constraint is put between two different jobs. We present an inter-job CPA to update CCs of each activity across temporal constraints in different jobs.

Basically, updating CCs in each activity in response to triggering events in the inter-job CPA is similar to that of the intra-job CPA described in Kim and Suh (2007). Main differences arise in the following two cases :

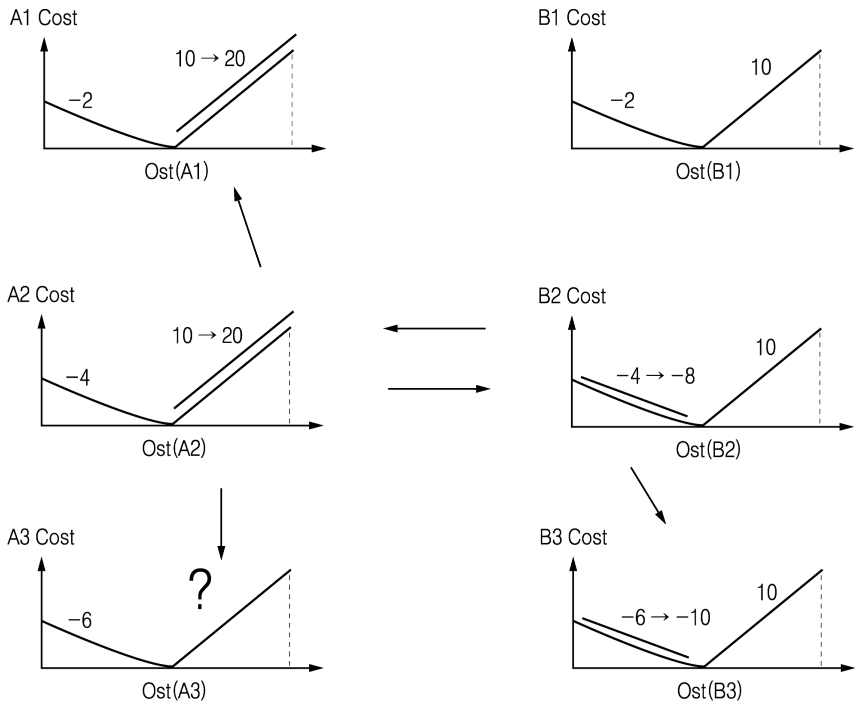
1) when we put a temporal constraint between two jobs, and

2) when the earliest start time of an activity increases.



[Figure 4] A temporal constraint is put between two different jobs

The characteristics of the inter-job CPA are described in detail in the followings.

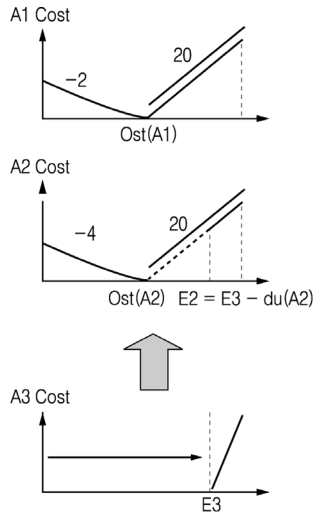


[Figure 5] In the initial inter-job propagation, the tardiness CC slope of the upstream activities of activity A_2 increases by the amount of the unit tardiness cost of activity B_2 while the inventory CC slope of the downstream activities of activity B_2 increases by the amount of the inventory CC slope of activity A_2

2.1 Trade-off between tardiness and inventory costs

Suppose that the two activities A_2 and B_2 in the jobs J_1 and J_2 respectively are temporally connected as shown in [Figure 4].

[Figure 5] shows that an inter-job cost propagation takes place in two ways. First, the slope of the tardiness CC of activity B_2 is propagated into the upstream and downstream activities of activity A_2 to increase their tardiness CC slope. Second, the slope of the inventory CC of activity A_2 is propagated into the downstream activities of activity B_2 to increase their inventory CC slope. This initial cost propagation seems to be straightforward by simply increasing the slope of relevant CCs in affected activities. For instance, in each activity upstream of activity A_2, the tardiness CC slope increases by the amount of 10 (the unit tardiness cost of activity B_2), while the inventory CC slope for each activity downstream of activity B_2 increases by 4 (the cumulative unit inventory cost of activity A_2). The reasoning for this propagation is that when an activity upstream of activity A_2 increases its earliest start time over its optimal start time, it causes not only an intra-job tardiness in job J_1 but also an inter-job tardiness in job J_2 through the temporal constraint between activities A_2 and B_2. The same principle can be applied to the inventory cost propagation. However, a trade-off between inventory and tardiness costs arises when the inter-job cost propagation goes to those activities downstream of activity A_2. We investigate the trade-off relationship between an inter-job tardiness cost and intra-job inventory cost before propagating downwards from activity A_2.



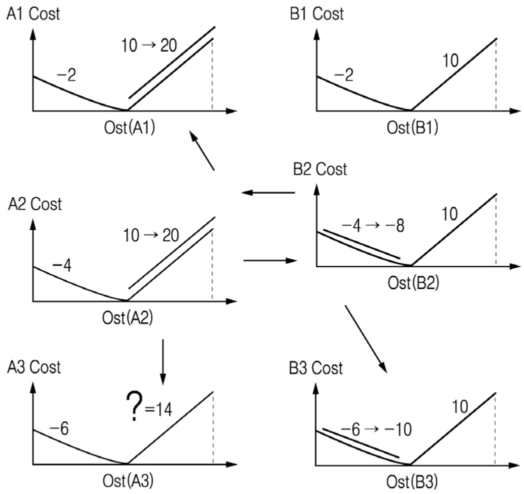
[Figure 6] When the earliest start time of activity A_3 increases to E3, the intra-job tardiness cost of job J_1 was already paid in activity A_3. Then, in activity A_2, two compromising costs remain : inter- job tardiness cost and intra-job inventory cost. Since the unit inter-job tardiness cost 10 is greater than the unit intra-job inventory cost 4, it minimizes the total schedule cost for activity A_2 not to move its optimal start time

Suppose activity A_3 increases its earliest start time to the time point E3 as shown in [Figure 6], and causes an intra-job tardiness cost with the unit tardiness cost 10 of job J_1. Responding the earliest start time increase of activity A_3, activity A_2 can choose two alternative actions : (a) it moves its optimal start time to E2, E2 = E3 − du(A2), with additional unit inter-job tardiness cost 10; (b) it does *not* move its optimal start time at the expense of a unit intra-job inventory cost 4. In this example, the inter-job tardiness cost dominates over the intra-job inventory cost, and then it is a better choice from the global cost minimizing objective for activity A_2 not to move its optimal start time at the expense of its intra-job inventory cost. By this trade-off analysis, the inventory CC slope of activity A_2 propagates down to its downstream activities to increase their tardiness CC slope.

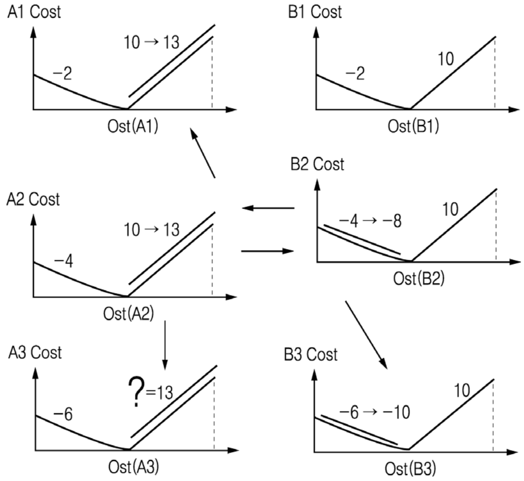
[Figure 7] shows the final result of the initial cost propagation for the problem shown in [Figure 5]. Contrarily, in such a case that the inventory cost dominates over the inter-job tardiness cost, the tardiness CC slope of activity A_3 increases by the amount of the tardiness CC slope of activity B_2. Suppose that job J_2 has a unit tardiness cost 3. Then in activity A_2, the unit inter-job tardiness cost 3 becomes less than unit intra-job inventory cost 4, making activity A_2 to move its optimal start time in case of the earliest start time increase of activity A_3. Therefore, the slope of the tardiness CC in activity A_3 increases by adding the unit inter-job tardiness cost of activity B_2 to the unit intra-job tardiness cost of activity A_2 as shown in [Figure 8]. This trade-off relationship between tardiness and inventory costs is a major breakthrough to expand the intra-job CPA into inter-job CPA.

2.2 Inventory cost recovery

The intra-job CPA guarantees that all the CC slopes of each activity always increase irrespective of the events occurring. This is not any more applicable in the case of the inter-job CPA, especially in the event of increasing the earliest start time of an activity due to the tardiness dominion discussed earlier. For instance, suppose that the earliest start time of activity *As* in [Figure 7] was actually increased to E3 as shown in [Figure 9]. Since activity

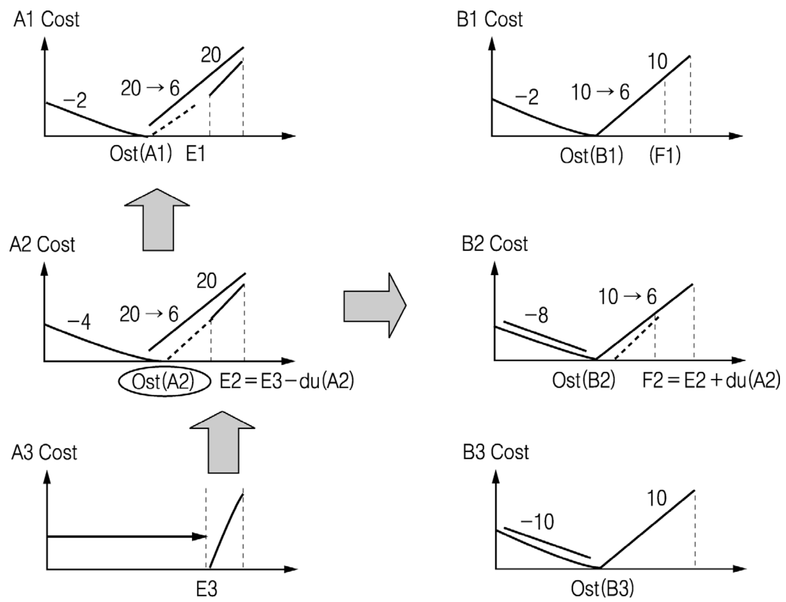


[Figure 7] The slope of the tardiness CC in activity A_3 has a value of 14 by adding the cumulative unit inventory cost of activity A_2, 4, to its unit intra-job tardiness cost 10



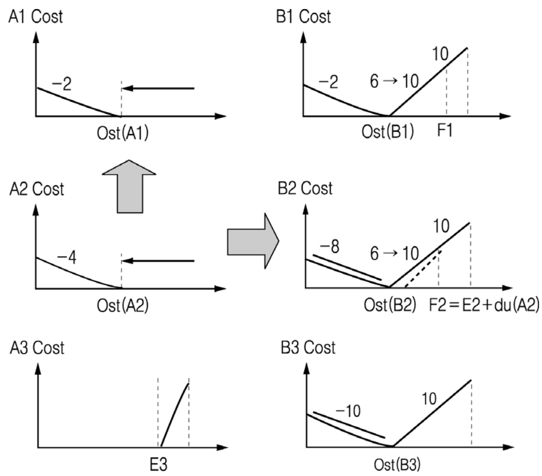
uic(Ai)=2, i=1, 2, 3
uic(Bi)=2, i=1, 2, 3
utc(J_1)=10, utc(J_2)=3S

[Figure 8] The slope of the tardiness CC in activity A_3 has a value of 13 by adding the unit inter-job tardiness cost of activity B_2, 3, to its unit intra-job tardiness cost 10



[Figure 9] When the earliest start time of activity A_3 increases to E3, the tardiness CC slope of activity A_2 decreases by subtracting both its unit intra-job tardiness cost 10 and cumulative unit inventory cost 4. This applies in the same manner to activity A_1. In the inter-job activities B_1, B_2, their tardiness CC slope for the time interval affected by activity A_2 is subtracted by the cumulative unit cost of activity A_2

A_2 is dominated by the inter-job tardiness cost of B_2, the optimal start time of activity A_2 does not move at the expense of a resulting intra-job inventory cost for the time interval [$ost(A_2)$, E2] where E2 is E3 − du(A_2). However, if the optimal start time of activity A_2 should increase later by itself or other activities, then that intra-job inventory cost can be recovered. For this reason, the tardiness CC slope of activity A_2 for [$ost(A_2)$, E2] changes from 20 to 6 by subtracting both the unit intra-job tardiness cost 10 of activity B_2, and cumulative unit inventory cost 4 of activity A_2. Likewise, the tardiness CC slope of activity B_2 and its upstream activities decreases by subtracting the cumulative unit inventory cost of activity A_2. [Figure 10] shows an inventory cost recovery where the latest start time of activity A_2 decreases to its opti-



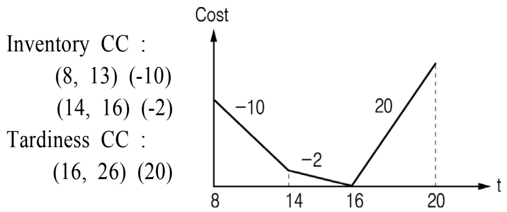
[Figure 10] When the latest start time of activity A_2 decreases to its optimal start time, the tardiness CC slope of activity B_2 recovers the cumulative unit inventory cost of activity A_2 since their temporal relationship in the tardiness CC becomes cut off mal start time, increasing the tardiness CC slope of activities B_1, B_2 for their relevant time interval by the amount of the unit cumulative inventory cost of activity A_2.

The worst-case time complexity to update inventory and tardiness CCs at a search state is due to the inventory recovery process whenever there is a change in the slope of tardiness CCs in each activity. This process has complexity of $O(kn)$ where n is the number of activities, and k is an upper bound on the number of activities residing on the non-critical path. The overall time complexity for the inter-job CPA is $O(kn^2)$.

2.3 Estimation of Cost Impacts

By updating and maintaining the tardiness and inventory CCs for each activity in constructing schedules, we can estimate a global *cost impact* of a scheduling decision-making before it is done implicitly by a propagator or explicitly by a scheduling heuristic. Therefore, this global cost propagation by a CPA helps cost-based scheduling heuristics to go in the right direction from the cost perspective. For instance, suppose a scheduling heuristic tries to put a sequence constraint between the two activities A, B : $A \rightarrow B$ or $B \rightarrow A$. By calculating the cost impacts caused by each decision, the heuristic can choose one with a minimum cost impact. Even when there is no schedule cost occurring, we can still measure the increase of the CC slopes, and use it for the same purpose.

Suppose that an activity A has its CCs as shown in [Figure 11]. If an event increases the earliest start time of activity A to 18, its cost impact is 40(=20×2). If the latest start time of activity A decreases to 10, the cost impact is 36(=10×3+2×3).



[Figure 11] CCs used for estimating cost impacts of a scheduling decision

3. Cost-based Constraint-Directed Scheduling Heuristics

In the constraint directed scheduling literature, some heuristic search techniques(Beck et al., 1997a; Beck et al., 1997b; Zweben, 1994) have been introduced to minimize the maximum finish time of all jobs. The main effort of those heuristics is devoted to find the most critical constraint in the analysis of each search state, and make a heuristic decision to reduce this criticality. The intuition is that by focusing on the most critical resources and activities, they attempt to make a decision that reduces the likelihood of reaching a search state where the constraint is broken(Beck et al., 1997b). We call this kind of heuristics as being *feasibility-based* since their criticality estimates are based upon the feasibility perspective not to break the resource constraint.

For the cost-based optimization problem, we need to have the cost perspective where the criticality is estimated to identify the resource and activity with the highest impact on the schedule cost. The *cost-based* heuristics make a scheduling decision based on the estimate for the cost criticality to minimize the total schedule cost.

In this paper, we present three scheduling heuristics to deal with the cost optimization problem. Firstly, we adopt a texture-based heuristic that finds the most critical resource and activity from the highest contending area, and applies the cost-based sequencing heuristic. The second heuristic utilizes the optimal start times of each activity to identify the most critical resource and activity. The last heuristic compares every pair of activities to find the cost critical activity.

3.1 The Cost-based SumHeight Heuristic

The first heuristic is a texture based one where contention and reliance textures are estimated to determine the most critical resource and activities. In scheduling, a texture is a measurement of a fundamental, problem-invariant property of a constraint graph, and thus provides information which a scheduling heuristic canuse to guide search directions. Contention is the extent to which two activities compete for the same resource at the same time while reliance is the extent to which an activity must be executing on a specific resource at a particular time point in order to find a schedule. For the estimation of contention and reliance textures, we adopt the SumHeight heuristic (Beck et al., 1997b)that makes a commitment on the activities most reliant on the resource for which there is the highest contention. At a high level, SumHeight does the following:

1) Identify the resource and time point with the maximum contention.
2) Identify the two activities, A and B, which rely most on that resource and time point, and that are not already connected by a path of temporal constraints.
3) Analyze the consequences of each sequence possibility (A→B and B→A) and choose the one that appears to be superior.

The maximum contention is obtained by aggregating the individual demands of each activity for each resource. To calculate an activity's demand for a particular resource, a uniform probability distribution[1) over the possible start times for an activity is assumed : each start-time has probability 1/|STD|. The individual demand, *ID*(*A*, *R*, *t*), is the amount of resource *R*, required by activity *A*, at time *t*. It is calculated as follows, for all $est_A \le t < lft_A$:

$$ID(A,R,t) = \frac{\min(t, lst_A) - \max(t - du_A + 1, est_A)}{|STD_A|}$$

To escape scaling with the scheduling horizon, SumHeight uses an event-based representation and a piece-wise linear approximation of the *ID* curve of activity. The individual activity demand is represented by four (*t*, *ID*) pairs :

Case 1) $du_A >= |STD_A|$:

$(est_A, \frac{1}{|STD_A|})$,

1) In the cost optimization problem, the schedule cost of an activity differs at different start time points, therefore, the probability distribution over the possible start times for the activity might be biased accordingly. For instance, by assigning a larger probability to start times with lower cost values, the start time probability distribution of an activity becomes subject to a cost preference. The MicroBOSS system [11] has used this subjective probability distribution to build an individual demand curve of each activity for a particular resource.

$(lst_A,\ 1)$,
$(eft_A,\ 1)$,
$(lft_A,\ 0)$

Case 2) $du_A < |STD_A|$:

$(est_A,\ \frac{1}{|STD_A|})$,
$(eft_A,\ \frac{du_A}{|STD_A|})$,
$(lst_A,\ \frac{du_A}{|STD_A|})$,
$(lft_A,\ 0)$.

In Case 1), the possible perturbation in the start time probability distribution of an activity takes place only during the time intervals, $(est_A,\ lst_A)$, and $(eft_A,\ lft_A)$. Therefore, in such case that every activity has its duration greater than the size of its *STD*, the SumHeight heuristic with a uniform start time probability distribution produces the approximately same *ID* curve of the MicroBOSS system with a cost-biased start time probability distribution.

After calculating the aggregate demand curve on each resource, SumHeight identifies the resource and time point for which there is the highest contention. The two activities mostly relying on that critical time point are identified, and sequenced using the MinimizeMax heuristic[2)] that attempts to reduce the resulting cost impacts of the sequencing decision.

MinimizeMax Sequencing Heuristic MinimizeMax(MM) identifies the sequence which satisfies the following :

2) This is a variant of SumHeight's original MinimizeMax sequencing heuristic(Beck et al., 1997b).

$$\mathrm{MM} = \min(p\text{-}cost(A \rightarrow B),\ p\text{-}cost(B \rightarrow A))$$

$p\text{-}cost(A \rightarrow B)$ is an estimate of the cost impacts incurred by the new ordering decision, which can be provided using CCs as described in Section 2.3 If both $p\text{-}cost(A \rightarrow B)$ and $p\text{-}cost(B \rightarrow A)$ are zero, then *p-cost* is recalculated to obtain the maximum increase of the CC slopes caused by the inter-job CPA.

The MM heuristic chooses the lowest maximum cost impact. The intuition is that since we are trying to reduce the schedule cost, we estimate the worst case increase and then make the commitment that avoids it.

The SumHeight heuristic has been reported to produce good results in the job shop scheduling problem to minimize the makespan. However, since the criticality of resource and activity is determined from the individual demand curve of an activity, it is mainly affected by the possibility of breaking the resource constraint. The smaller the domain size of an activity is, the more it becomes critical irrelevant of its cost values. Therefore, we expect the performance of SumHeight in the cost optimization problem might be good in such case that critical activities have also large cost values. In contrast, when there is an activity with a small domain but having lower cost value, SumHeight makes it critical on a contending resource, and then, there is a risk of having a bad impact on the cost critical activities with higher cost values.

3.2 The Cost-based SumHeight-II Heuristic

The SumHeight-II heuristic is a variant of SumHeight in that it uses a different individual demand curve to estimate the criticality. To focus on the cost criticality resource and activity in each search state, SumHeight-II uses the optimal time start and finish times of an activity for making the individual activity demand curve. The individual demand curve of the activity is represented by two time points:

$$(ost,\ 1),\ (ost+du,\ 1)$$

Based on the new individual demand curve, we make the aggregate demand curve for each resource, and identify the critical resource and activities in the same way as SumHeight. The MM heuristic is applied to sequence the two most critical activities. By using new individual activity demand curve, SumHeight-II is expected to effectively identify the cost critical resource. However, the cost impacts of activities competing for the critical resource are not considered in making the individual demand curve of activities. That means even if a resource is severely contended for, its impact on the total schedule cost might be low if the competing activities have lower cost values.

3.3 The P-Cost Heuristic

The SumHeight and SumHeight-II heuristics estimate the criticality of a resource constraint based on the individual demand curve of an activity. However, the individual demand curve alone can neither represent the sequencedependent cost like the changeover costs, nor consider cost impacts between activities. To identify the critical activities with the highest cost impact, we calculate the cost impacts of a scheduling decision (here, putting a sequence constraint) for every pair of activities and select the two activities with the maximum value. In more detail, the P-Cost heuristic operates as follows:

Step 1 : Every pair of activities which are not connected to each other by a path of temporal constraints is chosen. In the pair of the two activities, A and B, we calculate the cost impact resulting from each sequence possibility($A \rightarrow B,\ B \rightarrow A$) :

$$p\text{-}max = \max(p\text{-}cost(A \rightarrow B),\ p\text{-}cost(B \rightarrow A))$$

Step 2 : A pair of activities with the maximal *p-max* is identified, and sequenced by using the MinimizeMax sequencing heuristic.

From the cost minimizing objective, we expect this heuristic outperform other heuristics since it's heuristic decision-making can fully utilize global cost information provided by CPAs.

4. Concluding Remarks

The central aim of this paper is to present CPAs for propagating cost information globally across temporal constraints. Especially, the inter-job CPA is the first work of which we are aware in the constraint directed scheduling literature. A more refined inter-job CPA needs to be developed for such a case where a problem has multiple temporal constraints between two different jobs with a potential conflict in their optimal start times. We believe that kind of extension can be easily done by exploiting the current characteristics of the inter-job CPA such as trade-off and cost recovery.

Based on CPAs, we have proposed cost-based heuristics to guide the search to minimize the total schedule cost. As a next step for further research, we plan to perform a series of experimentation to compare the performance of different heuristics. Afterwards, the current cost model having tardiness and inventory cost components can be extended to include a variety of cost components such as change-over, labor costs.

References

[1] Baker, K. R. and G. D. Scudder, "Sequencing with earliness and tardiness penalties : a review", *Operations Research*, Vol.38, No.1 (1990), 22～36.

[2] Beck, J. C., A. J. Davenport, E.M. Sitarski, and M. S. Fox, "Texture-based heuristics for scheduling revisited", Proceedings of AAAI-97, 1997a.

[3] Beck, J. C., A. J. Davenport, E. M. Sitarski, E.M., and M. S. Fox, "Beyond contention : extending texture-based scheduling heuristics", Proceedings of AAAI-97, 1997b.

[4] Easton, F. and J. Goodale, "Schedule Recovery : Unplanned Absences in Service Operations", *Decision Sciences*, Vol.36, No.3 (2005), 459～488.

[5] Faaland, B. and T. Schmitt, "Cost-based scheduling of workers and equipment in a fabrication and assembly shop", *Operations Research*, Vol.41, No.2(1993), 253～268.

[6] Fox, M. S, Constraint-directed search : a case study of job-shop scheduling, Ph.D. thesis, Carnegie Mellon University, 1983.

[7] Kim, J. K. and M. S. Suh, "Cost-Based Directed Scheduling : Part I, An Intra-Job Cost Propagation Algorithm", *Journal of Intelligent Information Systems*, Vol.13, No.4(2007), 121～135.

[8] Minton, S., M. Johnston, A. Philips and P. Laird, "Minimizing conflicts : a heuristic repair method for constraint satisfaction and scheduling problems", *Artificial Intelligence*, Vol.58(1992), 161～205.

[9] Morton, T. E., S. R. Lawrence, Rajagopolan and S. Kerre, "SCHED-STAR : a price-based shop scheduling module", *Journal of Manufacturing and Operations Management*, Vol.1, No.2(1988), 131～181.

[10] Ow, P. S. and T. E. Morton, "The single machine early/tardy problem", *Management Science*, Vol.35, No.2(1989), 177～191.

[11] Sadeh, N, Looking-ahead techniques for micro-opportunistic job shop scheduling. Ph.D. thesis, Carnegie Mellon University, 1991.

[12] Sadeh, N. and M. S. Fox, "Variable and value ordering heuristics for the job shop scheduling constraint satisfaction problem", *Artificial Intelligence*, Vol.86, No.1 (1996).

[13] Smith, S. F. and C. C. Cheng, Slack-based heuristics for constraint satisfaction scheduling, Proceedings AAAI-93, 1993, 139～144.

[14] Ye, P., D. Glass, M. McTear, and J. G. Hughes, Job cost and constraint relaxation for scheduling problem solving in the CLP paradigm, 11th European Conference on Artificial Intelligence, 1994, 540～644.

[15] Zweben, M., B. Daun, E. Davis, and M. Deale, M, Scheduling and rescheduling with iterative repair, Intelligent Scheduling (Eds., Zweben, M. and Fox, M.), 1994, 241～256.

Abstract

비용기반 스케줄링 : **Part II**, 작업간 비용 전파 알고리즘

서민수* · 김재경**

현실세계의 복잡한 스케줄링 문제를 해결하기 위하여 AI기반의 비용기반 휴리스틱 방법들이 많이 제시되어 왔다. 하지만 다양한 작업(job)을 대상으로 하는 작업간 비용 전파 알고리즘(CPA)에 관한 연구는 부족한 상황이다. 그러한 CPA없이 스케줄링을 한다는 것은 지역적이고 불충분한 정보에 기반하므로 전체 비용을 최소화 하는 목적을 달성하는데 많은 어려움이 있었다. 전체 비용을 최소화하기 위하여는 작업내 CPA와 작업간 CPA, 두 종류의 CPA가 필요하다. 작업내에서 변화가 생긴 비용에 관한 정보는 작업간 CPA를 통하여 연결된 이웃 작업으로 전파된다. 작업내 CPA는 이전 연구 [7] 주제이고, 이번 연구에서는 작업간 CPA와 이러한 비용 정보를 기반으로 전체 비용을 최소화하는 비용기반 휴리스틱 스케줄링 기법을 제안한다. 즉, 이번 연구에서는 탐색 과정에서 각 activity의 비용 함수를 만들고 개선하는 작업간 CPA를 개발하고, 비용 정보를 일시적인 제약조건하의 전체 네트워크에 전파하는 방법을 개발하였다. 이러한 비용 전파 알고리듬을 이용함으로써 전체 스케줄링 비용을 최소화하는 다양한 비용기반 휴리스틱 기법들을 제시하였다.

* 현대제철
** 경희대학교 경영대학