A Novel Tabu Search Algorithm for Solving Robust Multiple Resource-Constrained Project Scheduling Problem

Rong-Hwa Huang and Tung-Han Yu

***Abstract*—This research developed an innovative robust project scheduling strategy for efficiently using slack time to reduce interference and derivative cost. Robust scheduling is particularly critical under resource constraints. For an efficient solution to this complex problem, a novel parallel tabu search algorithm was then adopted from project scheduling problem library (PSPLIB). Test results proved that the proposed robust scheduling strategy is effective for moderating the impacts of unpredicted events and that the effectiveness and robustness of the proposed parallel tabu search is superior to that of the conventional tabu search.**

***Index Terms*—Project scheduling, resource constraint, robust scheduling, tabu search.**

I. Introduction

Resource-constrained project scheduling problem (RCPSP) is very complex and frequently applied in the field of manufacturing and optimization. Past literatures of project scheduling focus on the makespan minimization with unlimited resources, which is inconsistence to the reality. With the resource constrained, activities may be forced to postpone. The sequence of activities, derivative result and critical paths are very different. Managers must consider objective factors and resource constraints in order to make optimal decisions. In practice, however, unexpected factors are likely to affect project execution. As [1] observed, project activities are subject to considerable uncertainty that can cause numerous schedule disruptions and delays. These disruptions may result from machine breakdowns, material delays, personnel shortages or changing due dates. Each project includes multiple activities with complex precedent relations, and any unpredicted delay at that activity level can encumber the project by causing late delivery, reduced customer satisfaction, loss of business goodwill, and missed market opportunity. Robust scheduling strategy moderates these potentially enormous losses by effectively and preventively exploiting resources.

Robust scheduling, also known as proactive scheduling, is an approach for developing a baseline schedule that anticipates variability during project execution [2]. Adding buffers at strategic points makes project execution more robust. The underlying concept is that allocating increased resources to the project reduces delays caused by disruptive events. [3], however, discussed the recovery problem, i.e., the problem of efficiently recovering at minimum cost when disruptions occur. The robust resource-constrained project scheduling problem can be abbreviated as RRCPSP. As [4] described, solution-robust project scheduling is a growing research field. Their research provided multiple heuristics to include time buffers in the project scheduling and still respect the due date of the actvites.

Flexible tabu lists memorize the points visited by the algorithm in order to avoid looping. Tabu search has proven efficient for solving NP-hard problems and is now widely used in the field. The resource-constrained project scheduling problem is equivalent to a $FF_c\left|M_j,res\right|w_1\cdot Robust\text{-}w_2\cdot C_{max}$ problem, where M_j is the eligibility constraint and res is the resource constraint. [5] combined tabu search with three particular initial solutions to solve sequence-dependent group scheduling problem in flexible flow shop environment. They find that the impact of different initial solutions is insignificant. Tabu search with the most time-saving initial solution is sufficed to achieve excellent result. A tabu search algorithm proposed by [6] featuring specifically defined moves and structured neighborhoods is proved efficient solving RCPSP. [7] develop a novel ant colony optimization based algorithm to solve no-wait two-stage flow shop scheduling problem and have satisfactory result.

Most literatures attempted to find optimal solutions in terms of makespan minimization with various meta-heuristics algorithms. The importance of available slack time and robust scheduling are often underestimated. Uncertainties in project scheduling may have large impacts and cannot be overlooked. This research solved three resource constrained robust project scheduling problems, and proposed a novel parallel tabu search to achieve balanced available slack time and makespan to produce robust and efficient solutions. In the following sections, we will first define the problem studied in this paper; second, we will introduces the details and procedures of our proposed parallel tabu search algorithm; third, data test result and analysis is summarized; finally, conclusions and some applications are made.

II. Robust Resource-Constrained Project Scheduling Problem

The objective of robust resource-constrained project scheduling problem (RRCPSP) is to obtain a schedule that

Manuscript received July 25, 2012; revised October 5, 2012.
R.-H. Huang is with Department of Business Administration, Fu Jen Catholic University, New Taipei City, Taiwan (e-mail: 0262999@mail.fju.edu.tw)

T.-H. Yu is with Graduate Institute of Business Administration, Fu Jen Catholic University, New Taipei City, Taiwan (e-mail: tunghan@gmail.com)

minimizes the cost of unexpected events with the limitation of restorable resource-constrained, while maintaining an acceptable level of project completion time. The strategy to obtain robust project scheduling is using available slack time to moderate the impact of unexpected events, on the premise to not increase cost or project completion time pointlessly. The problem assumes that all activities cannot be prefabricated and precedence relations are to be obedient. The actual starting time of each activity is dynamic, and resource requirements for activities are known and indivisible.

Available slack is the difference between latest finish time and the actual finish time for each activity. It allows extensive project execution time resulting from unexpected disruptions and increases the available slack for each activity. No extra resources are required at this stage since available slack could be created from proper scheduling. The buffer prevents unexpected delay and the following recovery problem. Available slack has a decreasing utility on the same activity in the objective function to prevent stacking of available slack.

Besides robustness, efficiency is also essential for minimizing project completion time. Hence a weighted objective function consists of makespan and available slack is highly flexible. Under stable circumstances, higher makespan weight helps managers to finish the project in short time; while under uncertainty, higher available slack weight prevent unpredicted events from jeopardizing the whole project. Three restorable resource constraints considered in this study are machine hours, labor hours and site hours. At any time the resources in use do not exceed their limitations.

III. Parallel Tabu Search Algorithm

Tabu search has better effectiveness to avoid repeat solution compares to other heuristics. One of the disadvantages of tabu search is the tendency to fall into local optimization. The research proposes a parallel tabu search (PTS) algorithm to simulate multiple processors. Each processor randomly generates an initial solution simultaneously and begins search in the neighborhoods. It then moves to the best solution in neighborhood or a solution that satisfies aspiration criterion. Route is memorized in a shared tabu list to avoid repeat search across processors.

The procedures of parallel tabu search (PTS) proposed in this study are described as below.

Step 1: Simulates multiple CPUs and each generates a start solution S_i.

Step 2: Search for best solution S_k in the neighborhood.

Step 3: Confirm that if S_k is not searched by other CPUs. If searched, go to *Step 8*, otherwise go to *Step 4*.

Step 4: Confirm that if S_k is in tabu list. If yes, go to *Step 5*, otherwise go to *Step 6*.

Step 5: Confirm that if S_k satisfies aspiration criterion. If yes, go to *Step 6*, otherwise go to *Step 8*.

Step 6: Check if S_k is better than the global best solution S^*.If yes, go to *Step 7*, otherwise go to *Step 8*.

Step 7: Movement confirmed. Update tabu list and let $S_k=S^*$.

Step 8: Check if it reaches stop criterion. If yes, end project, else go to *Step 1*.

IV. Data Test and Analysis

A. Test Data

To verify algorithm performance, 30 test data sets for each experimental combination were drew from PSPLIB for four project scale of 30, 60, 90, 120 activities and 17 types of simulated CPUs. There are 30x4x17, of totally 2040 instances tested in our research. Each test data set has a tabu list length and three recoverable resources labor (R1), machine hour (R2) and site hour (R3). Table 1 illustrates test data types. Test program is complied with C++ language and executed on a 2.21GHz AMD Processor, 1.00GB RAM and Linux OS PC.

TABLE I: Test Data Types

Numbers of work (*n*)	R_1	R_2	R_3	Tabu List
30	20	20	20	6
60	20	20	20	8
90	20	20	20	10
120	30	30	30	12

B. Algorithm Effectiveness

This research choose to simulate 17 types of CPU numbers 1, 5, 10, 15, 20, 25, …, 80 for PTS. Users can choose any appropriate number of CPUs for their problems. The robust project scheduling is given the weight $(w_1,w_2) = (0.5,0.5)$ to available slack and project completion time. The makespan minimization project scheduling is given the weight of $(w_1,w_2) = (0,1)$. The weights are also adjustable depending on the situations project managers meet. A higher weight of available slacks emphasizes project robustness against disruptions, while a higher makespan weight provides lean project schedule. The effect of CPU numbers to optimal solutions and computation time is listed in Table 2. The effectiveness measurement is the difference of project completion times when disruptions occurred between normal makespan minimization schedule and robust schedule. The larger the measurement values, the better of the robust schedule when disruptions occur.

TABLE II: Effectiveness Test Results

n	**30**		**60**		**90**		**120**	
Amount of CPU	Eff.	Exec. Time (sec)	Eff.	Exec. Time (sec)	Eff.	Exec. Time (sec)	Eff.	Exec. Time (sec)
1	9.18	0.78	32.99	4.57	39.57	8.91	136.95	11.74
5	15.28	1.85	51.91	7.24	60.02	21.02	276.09	39.66
10	23.63	2.45	52.02	10.63	76.05	39.86	337.47	49.31
15	30.94	2.98	54.87	17.45	87.35	54.83	334.66	71.26
20	28.51	3.55	57.43	21.91	117.74	76.09	379.95	117.13
25	50.39	3.84	56.39	28.12	89.72	88.64	394.99	136.25
30	47.79	4.27	57.47	32.83	119.28	108.51	414.32	178.33
35	49.96	4.69	58.02	38.54	119.40	123.57	402.05	193.26
40	43.18	5.31	57.16	45.23	116.41	137.19	419.74	223.99
45	45.15	5.79	59.18	56.56	113.88	152.96	428.49	265.71
50	53.03	6.09	58.04	64.17	123.89	174.24	412.22	298.12
55	49.16	7.24	57.40	73.25	120.44	198.69	410.92	326.76
60	54.51	7.74	59.16	84.77	137.53	206.51	398.77	360.48
65	55.23	8.38	58.17	91.29	135.36	223.37	429.72	379.75
70	53.76	9.47	59.06	98.51	139.33	239.95	432.96	397.51
75	62.21	11.52	59.76	106.49	142.50	253.31	415.1	425.06
80	52.96	13.53	60.87	113.31	114.13	275.73	431.48	434.77

First, when the number of simulated CPU is 1, classical tabu search algorithm is used. As found in all 4 types of numbers of works, the robust project scheduling is better than the makespan minimization project scheduling if disruptions occur. The degree of effectiveness is highlighted as the project scales grow. Second, the degree of effectiveness significantly improved with multiple simulated CPUs. The efficacy of our proposed parallel tabu search (PTS) algorithm is greatly support by test results.

V. Conclusion

This research proposed a parallel tabu search (PTS) algorithm to solve robust RCPSP. The PTS algorithm incorporates the parallel neighborhood search in the algorithm. Each simulated CPU in PTS continues to search more deeply and widely in the solution space. The stopping criteria of simulated CPUs include the situations of repeated solutions to other CPUs and unimproved solutions. Simulated CPUs cooperatively and competitively search for better solutions. Inefficient CPUs are eliminated along the process. This paper proposes a weight perspective in which practitioners can conveniently adjust the weight base on the uncertainty of the project.

Most studies of RCPSP focused on minimizing makespan and cost. While the makespan of each project closely line up in a tight stream of schedule, the tightness of the schedule is not appreciated under dynamic or uncertain circumstances. Any element of unexpected disruptions could vastly change the progress of the project and worsen up the results. A robust scheduling strategy is needed to minimize the unpredicted impact. This research probes the moderating effect of available slack on unexpected delay in the overall project by considering RCPSPs and three restorable resources: machine hours, labor hours and site usage hour. Properly distributing available slack between activities reduces time delays caused by precedence activities and is thus increases robustness. Effective project scheduling is needed not only to complete the project with minimal makespan and cost, but also to enhance robustness and to respond to unexpected events immediately without affecting the overall project. This study built a mathematical model that considered both available slack and makespan to avoid unlimitedly extending the project makespan while at the same time maintaining proper robustness. A higher objective value indicates a more robust schedule. A quite promising application of our proposed algorithm would be to utilize artificial CPUs of several tabu search algorithm or other meta-heuristics. This way, our proposed PTS algorithm can combine the strength of other meta-heuristics and improve the solution quality further.

References

[1] W. Herroelen and R. Leus, "Identification and illumination of popular misconceptions about project scheduling and time buffering in a resource-constrained environment," *Journal of the Operational Research Society*, vol. 56, pp. 102-109, 2005

[2] W. Herroelen and R. Leus, "Project scheduling under uncertainty—survey and research potentials," *European Journal of Operational Research*, vol. 165, pp. 289-306, 2005

[3] G. Zhu, J. F. Bard, and G. Yu, "Disruption management for resource-constrained project scheduling," *Journal of the Operational Research Society*, vol. 56, pp. 365-381, 2005

[4] S. V. D. Vonder, E. Demeulemeester, W. Herroelen, "Proactive heuristic procedures for robust project scheduling: An experimental analysis," *European Journal of Operational Research*, vol. 189, pp.723-733, 2008

[5] R. Logendran, P. De Szoeke, and F. Barnard, "Sequence-dependent group scheduling problems in flexible flow shops," *International Journal of Production Economics*, vol. 102, pp. 66-86, 2006

[6] P. R. Thomas and S. Salhi, "A tabu search approach for the resource constrained project scheduling problem," *Journal of Heuristics*, vol. 4, pp. 123-139, 1998.

[7] R. H. Huang, C. L. Yang, and Y. C. Huang, "No-wait two-stage multiprocessor flow shop scheduling with unit setup," *International Journal of Advanced Manufacturing Technology*, vol. 44, pp. 921-927, 2009.

Rong-Hwa Huang was born in Taiwan in 1958, and has a PhD degree in industrial management.. He is currently a professor of Department of Business Administration, Fu Jen Catholic University. Dr. Huang specializes in the study of production and operational management, operation research, scheduling theory and application, system performance evaluation, background music and work performance.

Tung-Han Yu was born in Taiwan in 1983. He is currently a PhD candidate in business administration, of Graduate Institute of Business Administration, Fu Jen Catholic University. His research interests includes: scheduling theory and application, operation research.