5th International/11th Construction Specialty Conference
5e International/11e Conférence spécialisée sur la construction

Vancouver, British Columbia
June 8 to June 10, 2015 / *8 juin au 10 juin 2015*

MULTI-OBJECTIVE SCHEDULE OPTIMIZATION USING CONSTRAINT PROGRAMMING

Wail Menesi[1], Zinab Abuwarda[2], Mohamed Abdel-Monem[2], and Tarek Hegazy[2,3]
[1] Project Management Consultant, Revay and Associates Limited, Toronto, Canada.
[2] Dept. of Civil and Environmental Engineering, University of Waterloo, Canada.
[3] tarek@uwaterlooc.ca

Abstract: This paper describes the development of a constraint programming (CP) model for schedule optimization to satisfy both deadline and resource constraints in large-scale construction projects. Unlike many meta-heuristic methods in the literature, the CP model is fast and provides near-optimum solutions to projects with hundreds of activities. The IBM ILOG modeling software and its CPLEX-CP solver engine have been used to develop the proposed CP optimization model for solving the multi-mode resource-constrained project scheduling problem. The model takes many constraints into account, including project deadline, penalty (liquidated damages), incentive, and multiple resource constraints. The paper compares the CP results with the results of other methods, such as heuristic and genetic algorithm optimization, on case studies from the literature to prove the practicality and usefulness of the CP model. The paper also reports the results of CP optimization on larger projects of 1,000 to 2000 activities, which is common in construction practice, and is too large to be solvable by other methods. This research contributes to developing a practical decision support system for resolving real-life constraints in construction projects.

1 INTRODUCTION

Optimizing construction schedules has concerned many researchers in the past decades. In the literature, a broad category of schedule optimization models are referred to as multimode resource constrained problems, when each activity has more than one mode of execution with different resource requirements and costs. In the multimode problems, the main optimization objective has been either:

(a) Minimize the total project duration, while not exceeding resource limits. This problem has been referred to as (Multimode resource constraint scheduling problem, or MRCSP), or

(b) Minimize the total project cost, while not exceeding resource limits or a deadline constraint; which is referred to as (Multimode resource constraint time-cost trade-off, or RC-TCT.

While both cases involve multi-modes and resource constraints, case (b) is more general as it considers both cost and time in its formulation. Despite the extensive research in the literature in both aspects, no commercial software includes any time-cost trade-off (TCT) heuristic to help meet deadlines, let alone any procedure to resolve both deadline and resource constraints. One of the main contributing reasons, perhaps, is that both of these problems are known to be nondeterministic polynomial-time hard (NP-hard), which are very difficult to solve, particularly for large-scale problems.

In the literature, there is a large body of knowledge related to resolving either the RC-TCT or MRCPSP problem. Each problem by itself is considered a large-scale problem that has a large number of possible solutions, even when few activities are considered. In general, however, each of these problems has been addressed by three types of techniques, as follows: (1) traditional mathematical (exact), (2)

heuristic, and (3) metaheuristic (artificial intelligence-based). Good surveys of the exact, heuristic, and metaheuristic procedures for the MRCPSP and the RC-TCT problems are found in Peteghem and Vanhoucke (2010), and in Menesi and Hegazy (2014), respectively.

To the writers' knowledge, none of the literature efforts of the three solution types have addressed large-size problems. Most efforts discussed small demonstration examples of 10 or 20 activities and the few efforts that handled medium-size problems (a few hundred activities) required an unreasonably large time to provide a solution. Kandil and El-Rayes (2005), for example, reported a GA processing time of 55 h for a case study of 360 activities, which was reduced to 9.3 h using a system of parallel computing with 50 processors. One of the latest efforts to handle large-scale problems is the heuristic model of Hegazy and Menesi (2012) for resolving the RC-TCT problem. Their case study of 360 activities required 32 minutes to be solved, which is much faster than GAs, for the same size problem. However, even this heuristic method is expected to take many hours in the case of much larger problems than 360 activities, thus the key drawback of these methods is that finding near optimal schedules for large scale is very time-consuming and solution quality greatly degrades with problem size (Hegazy and Menesi 2012; Menesi et al. 2013; Ling and Fang 2011).The primary objective of this paper, therefore, is to introduce a new model using constraint programming (CP) to provide fast near-optimum solutions to much larger-scale multimode optimization problems, thus providing a practical approach to optimize construction schedules.

2 CONSTRAINT PROGRAMMING

Constraint programming (CP) has been successfully used to solve complex combinatorial problems in a wide variety of domains (Chan and Hu 2002; Heipcke 1999) by combining operations research and logic programming techniques. To facilitate the use of CP algorithms in scheduling problems, IBM developed a powerful software library, called IBM ILOG CPLEX Optimization Studio (Beck et al. 2011). It incorporates a CP engine (with IBMILOG CPLEX CP Optimizer) that has specialized syntax for modeling scheduling problems and other combinatorial problems that cannot be easily linearized or solved using traditional mathematical programming methods. The CPLEX-CP optimizer defines the scheduling problem by: a set of time intervals representing activities to be completed; a set of constraints that apply during these intervals or between intervals; and an objective function. Once the model is coded using the modeling language of the IBM ILOG tool (as discussed in the next section), the problem is solved by the CPLEX-CP solver engine using two techniques to find a solution: constructive search and constraint propagation (initial and during search). The initial constraint propagation removes the possible variable values that will not take part in any solution, thus reducing the search space. The constraint propagation during the search, on the other hand, removes all values that violate the constraints. The CP optimizer then uses a constructive search strategy to guide the search for a solution in the remaining part of the search space. The CP optimizer engine continues to search using constructive search and constraint propagation until a solution is found. In the literature, various researchers reported the advantages of CP in scheduling problems. Efforts include Liess and Michelon (2008) for solving the resource-constrained project scheduling problem; Liu and Wang (2008) for solving resource-constrained project scheduling problems considering cash flow; Liu and Shih (2009) for construction rescheduling optimization; Liu and Wang (2011) for optimizing overall profit and satisfying various resource constraints; and Menesi et al. (2013) for time-cost-resource optimization in large-scale projects.

3 CP MODEL FOR SCHEDULE OPTIMIZATION

3.1 General activity representation

Consider a construction project with *n* activities, each activity *i* has a set of associated construction methods (M_i). Each method *k* (from 1 to M_i) represents a construction mode or a way of carrying out activity *i*. Associated with this method (*k*) is a specific duration (d_{ik}), cost (c_{ik}) and resources (r_{ik}). Naturally, these construction methods vary from cheap-and-slow to fast-and-expensive, thus offer ways to speed the activity, if needed. Based on these construction options, the planned duration (D_i) and direct cost (C_i) of each activity *i* can be expressed as a function of which method is used, as follows:

$$D_i = \sum_{k=1}^{M_i} d_{ik} X_{ik} \quad (1)$$

$$C_i = \sum_{k=1}^{M_i} c_{ik} X_{ik} \quad (2)$$

Where, X_{ik} is a binary variable that indicates which method *k* is planned for activity *i.* Thus, if X_{ik} = 1, then method *k* is the one used for activity *i*, while X_{ik} = 0 for all other methods. To make sure only one mode of construction is used for each activity, one constraint is needed for each activity, where the sum of the method indices X_{ik} s should equal to 1.

$$\sum_{k=1}^{mi} X_{ik} = 1 \qquad i = 1, 2, \ldots, n \quad (3)$$

Network Logic Constraints: The network logic is defined in a set of hard constraints. The logical relationship between any activity (*i*) and its immediate predecessor (*p*), is expressed as:

$$SS_i - SF_p \geq 0 \qquad p = 1, 2, \ldots, NP \quad (4)$$

Where SS_i is the scheduled start time of activity *i*; SF_p is the scheduled finish time of the predecessor activity; and NP is the number of immediate predecessors for activity *i*.

Resource Constraints: The project resource constraints are expressed in the model as follows:

$$\sum_{i \in S_t} r_{ikl} \leq R_{lt}, \qquad l = 1, \ldots, L; \qquad t = 1, \ldots, T \quad (5)$$

Where, S_t is the set of eligible activities in time period (*t*), r_{ikl} is amount of resource *l* required by construction method *k* of activity *i*, R_{lt} is the amount of resource *l* available in time period *t*, *L* is number of resource types, *t* is the time period (t = 1,…, *T*), and *T* is project completion time.

3.2 Problem-specific formulation

Case of MRCPSP problems: Objective Function is to minimize the project duration (T), as follows:

$$Minimize\ T \quad (6)$$

Where T = Maximum scheduled finish time SF_p among all activities

Case of RC-TCT problems: At the project level, the direct cost of the project is noted as PDC and the indirect cost as PIC, thus, the total project cost (TPC) can be expressed as:

$$TPC = PDC + PIC \quad (7)$$

Where, the project direct cost is the summation of the activities' direct costs in Eq. 2, as follows:

$$PDC = \sum_{i=1}^{n} C_i = \sum_{i=1}^{n} \sum_{k=1}^{M_i} c_{ik} X_{ik} \quad (8)$$

The project indirect costs, on the other hand, are project-duration dependent; the longer the project duration, the more indirect costs are incurred and vice versa. The relationship between PIC and the project duration (T) can be expressed as:

$$PIC = IC_0 + IC * T \quad (9)$$

Where IC_0 is the fixed indirect costs (e.g., permits, mobilization cost, temporary hookups, temporary facilities, purchase advances), and IC is the indirect cost per time period (i.e., daily expenditures), and T is the total project duration. To consider practical scheduling situations, the model adjusts the total project

cost considering penalty and incentives amounts. If the project schedule is delayed beyond a defined deadline, the delay cost (C_{pd}) is expressed as:

$$C_{pd} = Y * C_d * (T - \text{deadline duration}) \quad (10)$$

Where: Y is a zero-one variable representing if a project delay occurred (i.e., Y = 1 if the project duration T is greater than the deadline duration), and C_d is the cost of project delay per day (i.e., delay penalty). On the other hand, if the project is scheduled to finish before the deadline, the incentive for early completion is expressed as:

$$IN = Z * B * (deadline\ duration - T) \quad (11)$$

Where Z is a zero-one variable representing if the schedule is completed earlier than the deadline (i.e., Z = 1 if project duration T is less than the deadline duration), and B is the incentive per day saved.

Project Completion Constraint: The project completion constraint is expressed as:

$$SF_E \leq Deadline\ Duration \qquad E = 1, 2, \ldots, NE \quad (12)$$

Where: SF_E is the finish time of ending activities, NE is the number of ending activities.
Objective Function is to minimize the sum of the costs described above, formulated as follows:

$$\text{Minimize C} = TPC + C_{pd} - IN \quad (13)$$

3.3 Multi-objective CP Model

In addition to the above single-objective multimode model in which the objective was to minimize the total duration or cost, another bi-objective multimode model was developed to include a secondary objective to minimize the fluctuations in the resource usage (resource leveling). The resource leveling problem itself is also a NP-hard problem and aims at achieving the most efficient resource utilization by reducing the peaks and valleys in the resource usage profile, without increasing project duration (Ponz-Tienda et al. 2013). One of the benefits of CPLEX-CP is that it has an internal function staticLex that can define an objective function with multiple objectives, with the first objective being most important, and so on. Utilizing this function, a multi-objective version of the CP model was developed with two objectives: (1) minimize project duration for MRCPSP problems or minimize project total cost for RC-TCT problems; and (2) minimize the peak resource demand. This is based on the fact that reducing the peak use of resources results in a more levelled resource profile.

4 IMPLEMENTATION AND RESULTS

Both of the proposed single-objective and multiple-objective CP formulations have been modeled in IBM ILOG CPLEX Optimization Studio, as shown in Figure 1. The figure shows a portion of the CP code and the ILOG modeling environment. Once developed, the CP models were applied to literature case studies.

4.1 Experiments on the MRCPSC problem

The developed CP models for MRCPSC was applied to a 10-activity project reported in Zhang (2012) who compared the solutions among three metaheuristic methods, and thus can be readily compared with those of the CP model of this paper. Each activity in the project has up to three modes of execution and one type of renewable resource. The project network and the three modes for each activity are shown in Figure 2. The available amount of resources is limited to 5 resource units per day.

4.2 Single-Objective Experiments on Small Projects:

The results of single-objective CP optimization were compared with the results of three metaheuristic methods for the case study reported in Zhang (2012): ant colony optimization (ACO); particle swarm optimization (PSO); and genetic algorithm (GA). The comparison of results shows the superior performance of the CP model in terms of solution quality and processing speed. Repeating the

experiment several times for this 10-activity case, the ACO algorithm could achieve the best result (14 days) only 81% of the time, which is better performance than PSO and GA. The CP solution, on the other hand, achieved the 14-day optimum solution in all experiments. To verify the CP solution, the resulting activities' start and finish times were entered into Microsoft Project Software, as shown in Figure 3a. The figure verifies that project duration is shortest, non-critical activities are not crashed, and that the resource-loaded schedule does not exceed the resource availability limit.

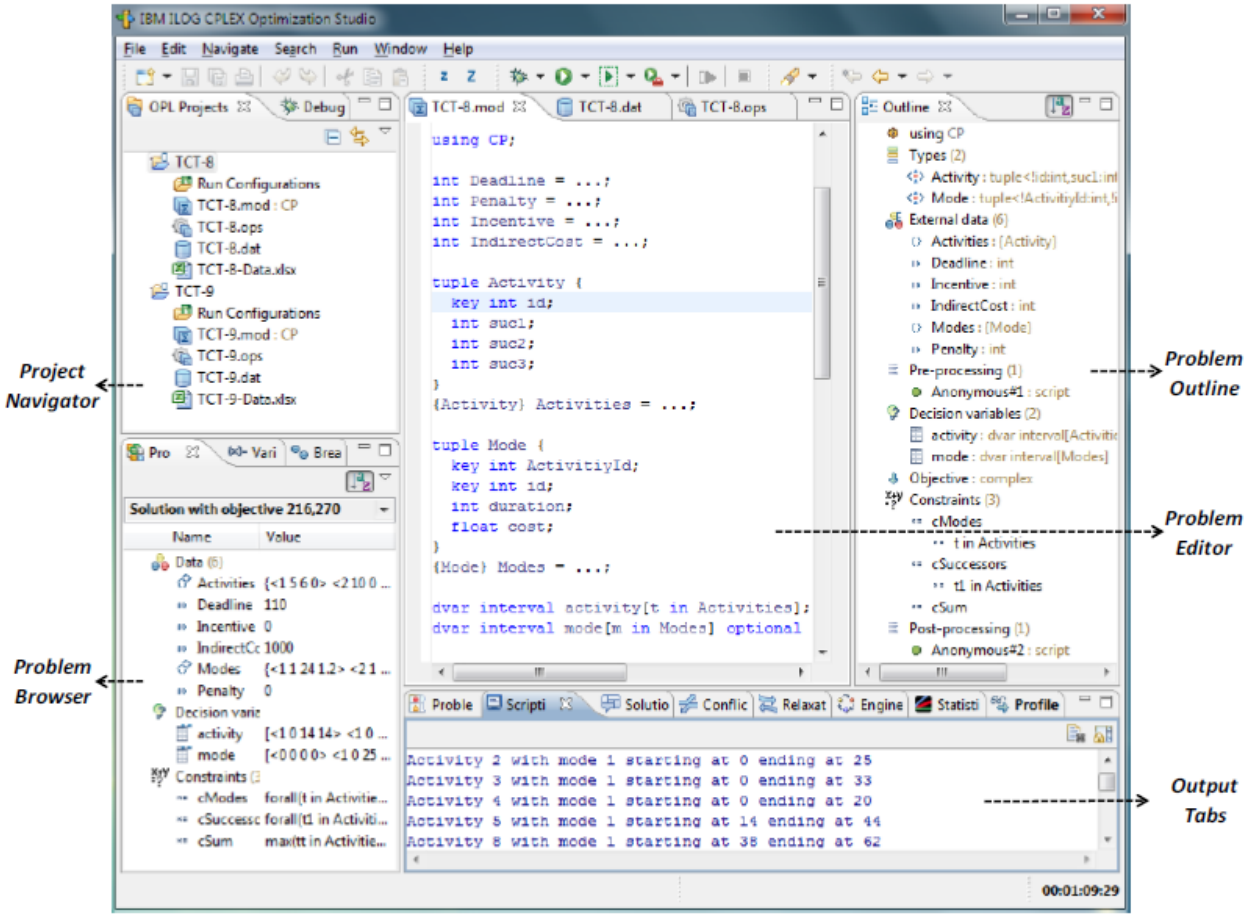


Figure 1: ILOG CPLEX Optimization Studio

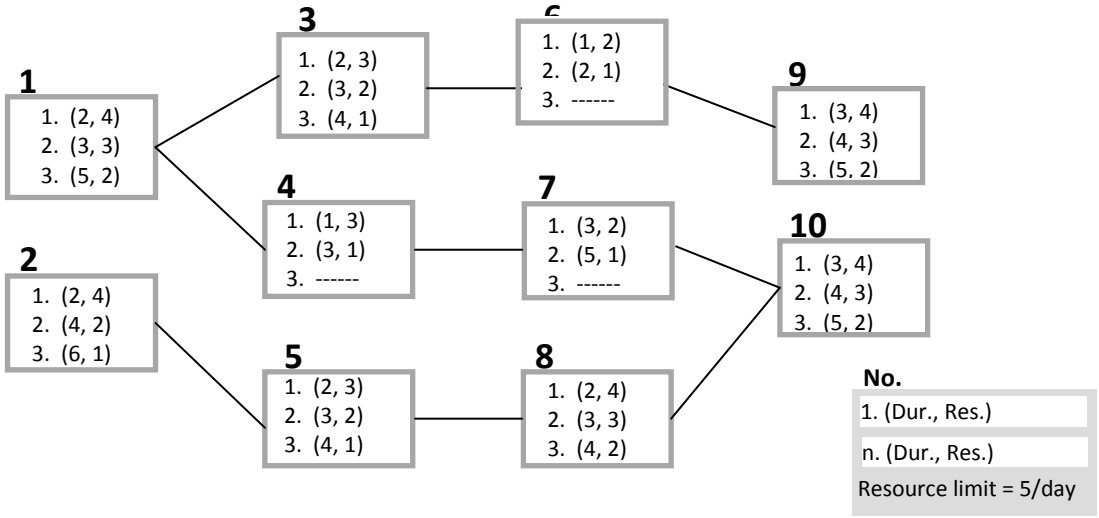
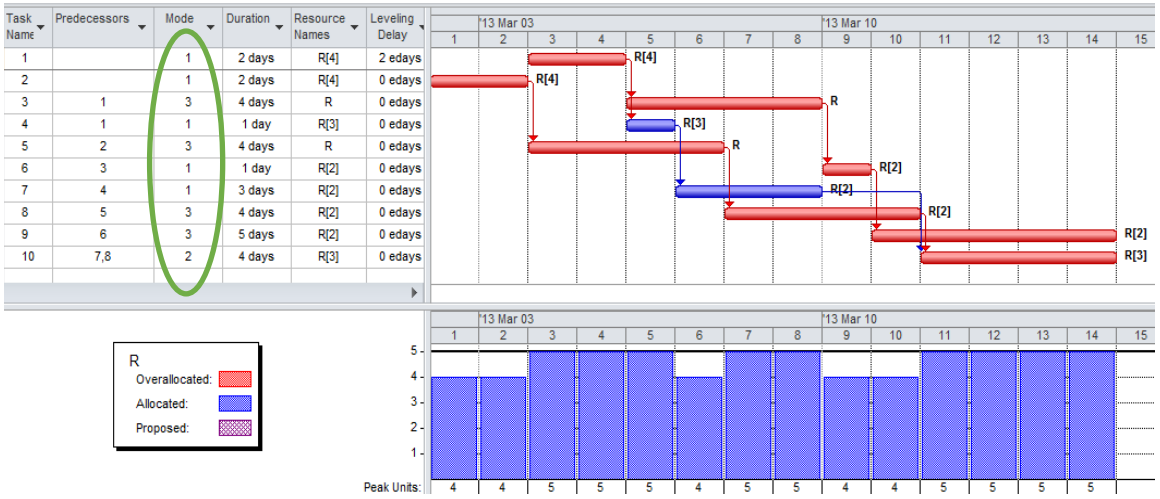


Figure 2: Project network and activity modes for a 10-activity case study project

(a) Decisions of the single-objective model.

Task Name	Predecessors	Mode	Duration	Resource Names	Leveling Delay
1		1	2 days	R[4]	2 edays
2		1	2 days	R[4]	0 edays
3	1	3	4 days	R	0 edays
4	1	1	1 day	R[3]	0 edays
5	2	3	4 days	R	0 edays
6	3	1	1 day	R[2]	0 edays
7	4	1	3 days	R[2]	0 edays
8	5	3	4 days	R[2]	0 edays
9	6	3	5 days	R[2]	0 edays
10	7,8	2	4 days	R[3]	0 edays



(b) Decisions of the multi-objective model.

Task Name	Predecessors	Mode	Duration	Resource Names	Leveling Delay
1		2	3 days	R[3]	0 edays
2		2	4 days	R[2]	0 edays
3	1	2	3 days	R[2]	1 eday
4	1	1	1 day	R[3]	0 edays
5	2	2	3 days	R[2]	0 edays
6	3	2	2 days	R	0 edays
7	4	2	5 days	R	0 edays
8	5	2	3 days	R[3]	0 edays
9	6	3	5 days	R[2]	0 edays
10	7,8	2	4 days	R[3]	0 edays

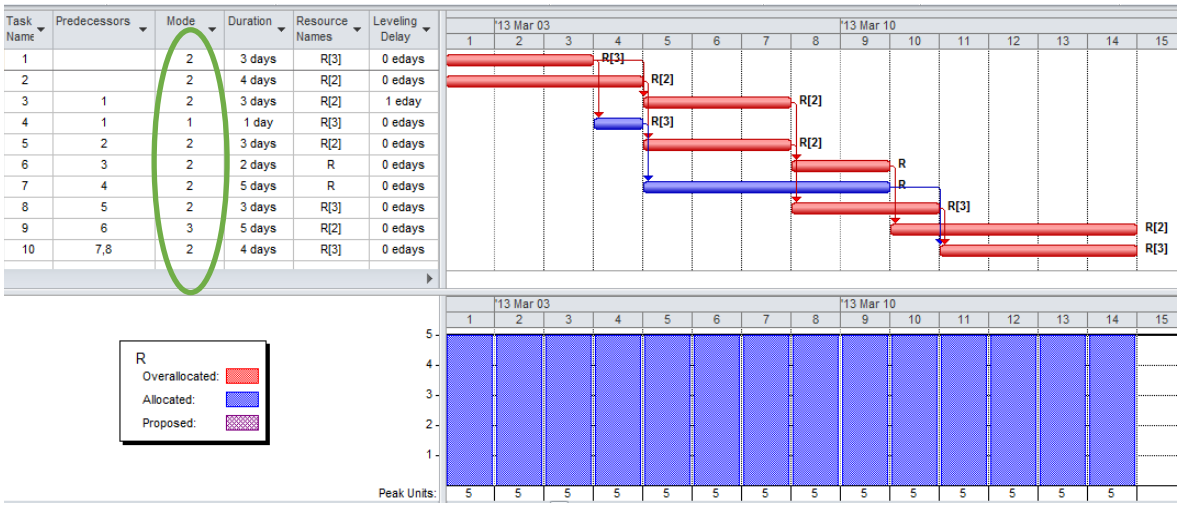


Figure 3: CP Solutions for the 10-activity case study

Single-Objective Experiments on Medium and Large Projects: Experiments on medium and large-scale projects were carried out. The 10-activity project of Zhang (2012) was used as the basis for creating larger projects with 100, 500, 1,000, 1,500, and 2,000 activities. The results of three experiments for the case of 100 activities experienced processing time of 1 s, 1 min, and 3 min, respectively. In the 3-min case, CP reached the optimal solution and, thus, its deviation from the optimal solution is zero. Also, for a 500-activity project, the CP model was able to reach a solution with a duration deviation of only 4.6% within 10 min. The quality of the CP solution for the same 500- activity project was then greatly improved when the processing time was extended to 30 min, reaching a deviation of 2.57% from the optimal solution. Therefore, for medium-size projects, the CP model can reach within 3% from the optimal duration, without violating resource limits, within a reasonable processing time (can be much faster if experiments were done on a desktop machine). For the 1,000-activity case, increasing the processing time to 2 h reduced the deviation from 18.9 to less than 5%, whereas in the 1,500-activity case; the deviation was greatly reduced from 21.0 to 7.3% when the processing time was increased from 1 min to 2 h. For the 2000-activity case, which is extremely large, the deviation remained at about 9.6% even with a processing time of 3 h. This result is considered practically reasonable. A deviation of 4.78 or 9.61% from optimum in the cases of 1,000 and 2,000 activities are not high. Without optimization, solutions are expected to be much worse or cannot be achieved. Figure 4 summarizes CP solutions for different problem sizes. Based on these results, the CP model proved to provide good solution quality with

reasonable processing time (20 min), thus, proving the practicality of the CP model in handling large-scale projects.

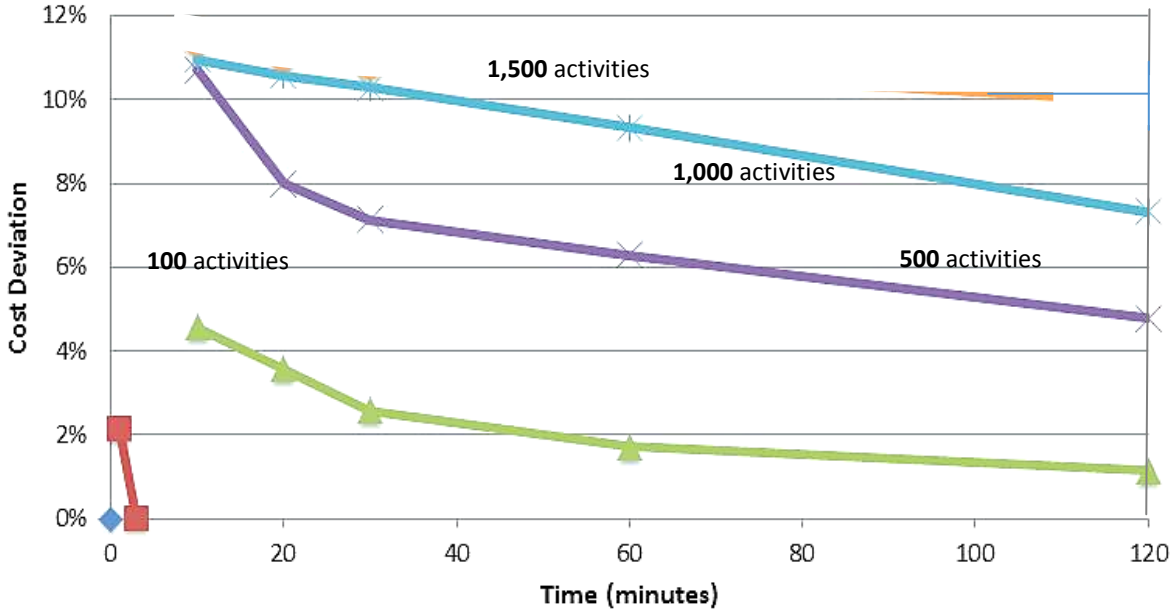


Figure 4: Solution quality and processing time for different sizes of MRCPSC problems

Multiple-Objective Experiments: The bi-objective model was then tested on the 10-activity case study and was found to work extremely well and does not require additional processing time. The resulting resource profile of the bi-objective optimization for the 10-activity project (Figure 3b) shows a much better resource profile than the single-objective solution in Figure 3a. The bi-objective model was also tested on the 1,000-activity case, with a processing time of two hours. The optimization reduced the solution deviation to 5%, with a standard deviation of daily resource usage being 1.018 (as compared with the single objective case that produced a 4.78% solution deviation and 1.030 standard deviation of resource usage). This shows the ability of the CP model to correctly trade-off among the objectives used.

4.1 Experiments on RC-TCT problem

Single-Objective Experiments on Medium and Large-Scale Projects: The case study of Chen and Weng (2009) was used as the basis for creating larger-size projects (1000, and 2000 activities). The base case study project consists of ten activities having up to 4 discrete options that use varying amounts of one limited resource. The experiments using the CP model and the heuristic method were performed for the base case of 10 activities and for the projects of 300, and 2000 activities. Table 1 presents the optimization results for each experiment, including the solution's total cost, project duration, processing time, and deviation from optimum solution. As shown in Table 1, the CP solution for the 10-activity project (involving 38 variables and 51 constraints) reached the optimal solution in one second. For large-size projects with 2000 activities, which is extremely large, the deviation remained at about 6.5% even with longer processing time (up to 2 hours). This result is considered practically reasonable and the CP model proved to provide good solution quality with reasonable processing time (30 minutes), thus, proving the practicality of the CP model in resolving the combined schedule constraints in large-scale projects. This result represents a benchmark for further research to improve upon.

Table 1: Results of the CP model on different project sizes

Project Size	Expected Optimum Solution	CP Solution		CP Deviation *
		Duration & Cost ($)	Processing Time	
10 Activities 38 Variables 51 Constraints	Duration = 56 Cost = $244,000	Dur. = 56 $ = $244,000	1 Sec	0%
300 Activities 1140 Variables 1530 Constraints	Duration = 1680 Cost = $7,320,000	Dur. = 1820 $ = $ 7,751,700	15 Sec	5.90%
		Dur. = 1698 $= $7,426,900	10 min	1.46%
		Dur. = 1686 $ = $7,384,900	20 min	0.88%
2000 Activities 7,600 Variables 10,200 Constraints	Duration = 11200 Cost = $48,800,000	Dur. = 12353 $ = $52,053,100	40 Sec	6.67%
		Dur. = 12298 $ = $51,969,200	30 min	6.50%
		Dur. = 12274 $ = $51,916,400	120 min	6.39%

***** Deviation from optimal = (CP cost in column 3 – Optimum cost in column 2) / Optimum cost in column 2

5 CONCLUDING REMARKS

Project managers usually have different options to complete activities with different resources and different production rates. Widely used scheduling software, such as Primavera P6 and Microsoft Project, do not have an option to enter more than one construction method or mode for an activity. In other words, using these software systems, project managers have to manually enter one option at a time in order to see the impact of using different activity modes on the project completion date. Currently, more owners are asking the contractors to submit and use resource-loaded schedules to manage their projects. While resource loading schedule may be a difficult exercise, it enables management to assess the amount of resources available and avoid risks early in project planning. If the resource-loaded schedule alerts decision makers that the available resources will not suffice to execute the work on time as planned, management can begin negotiating for additional resources early in the project.

This paper presents a fast and powerful tool for project managers based on constraint programming to perform what-if analysis for large-size projects and enables them to efficiently utilize their resources. . Although it is possible to load the activities modes automatically into the CP process, a full automated process is not yet developed. This process automation is currently an ongoing task by the authors to fully link the CP model with Primavera P6 or Microsoft Project software, which are the industry most-used scheduling software systems. Another limitation of the current CP model is the fact that it is designed to optimize the schedules before construction starts. Serious modifications are therefore needed to modify the model so that it accepts actual progress information and then produces an optimized corrective action plan that does little disturbance to the already selected modes for the activities. This is in addition to considering other constraints related to cash flow and other requirements, during the optimization. One final note is that despite the fact that the CP modeling tool is easily programmable, requiring the scheduling expert to have computer programming skills is a serious obstacle to implementing CP models, until full automation is achieved. The encouraging results of this paper, therefore, may motivate the producers of industry standard software systems to incorporate some optimization features into their tools, so that optimization becomes a main stream tool in the industry. The focus of future work in this area should then be on introducing efficient methods to handle practical size problems. Tweaking existing methods to achieve minor performance improvement on small problems has less practical value.

References

Beck, J. C., Feng, T. K., and Watson, J. (2011). “Combining Constraint Programming and Local Search for Job-Shop Scheduling.” *INFORMS Journal on Computing*, 23(1), 1–14,

Chan, W. T., and Hu, H. (2002). “Constraint Programming Approach to Precast Production Scheduling." *J. Constr. Eng. Manage., ASCE,* 128(6), 513-521.

Chen, P. H. and Weng, H. (2009) “A two-phase GA model for resource constrained project scheduling,” *Automation in Construction*, 18(4), 485-498.
Hegazy, T. and Menesi, W. (2012). “Heuristic Method for Satisfying Both Deadlines and Resource Constraints.” *J. Constr. Eng. Manage., ASCE*, 138 (6), 1–9.
Heipcke, S. (1999). “Comparing constraint programming and mathematical programming approaches to discrete optimization - The change problem.” *Jr. of the Operations Res. Soc. of Japan*, 50, 581–595.
IBM ILOG CPLEX Optimization Studio V12.3. (2012). CP Optimizer User’s Manual. International Business Machines Corporation, Armonk, New York.
Kandil, A, and El-Rayes, K. (2005) “Parallel Computing Framework for Optimizing Construction Planning in Large Scale Projects,” *Journal of Computing in Civil Eng., ASCE*, 19 (3), 304-312.
Liess, O., and Michelon, P. (2008). “A constraint programming approach for the resource-constrained project scheduling problem.” Ann. Oper. Res., 157(1), 25–36.
Liu, S., and Shih, K. (2009). “Construction rescheduling based on a manufacturing rescheduling framework.” *Autom. Constr.*, 18(6), 715–723.
Liu, S., and Wang, C. (2008). “Resource-constrained construction project scheduling model for profit maximization considering cash flow.” *Automation in Construction*, 17(8), 966–974.
Liu, S., and Wang, C. (2011). “Optimizing project selection and scheduling problems with time-dependent resource constraints.” *Automation in Construction*, 20(8), 1110–1119.
Menesi, W., Golzarpoor, B., and Hegazy, T. (2013). “Fast and near optimum schedule optimization for large-scale projects.” *J. Constr. Eng. Manage*., ASCE, 139(9), 1117–1124.
Menesi, W., and Hegazy, T. (2014). “Multimode Resource-Constrained Scheduling and Leveling for Practical-Size Projects.” *J. Manage. in Eng., ASCE,* ISSN 0742-597X/04014092(7).
Peteghem, V. V., and Vanhoucke, M. (2010). “A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem.” *European Journal of Operational Research*, 201(2), 409–418.
Ponz-Tienda, J., Yepes, V., Pellicer, E., and Moreno-Flores, J. (2013). “The resource leveling problem with multiple resources using an adaptive genetic algorithm.” *Automation in Construction*, 29, 161–172.
Zhang, H. (2012). “Ant Colony Optimization for Multimode Resource-Constrained Project Scheduling.” *J. Constr. Eng. Manage.*, ASCE, 28(2), 150–159.