Fast and Near-Optimum Schedule Optimization for Large-Scale Projects

Wail Menesi, Aff.M.ASCE[1]; Behrooz Golzarpoor[2]; and Tarek Hegazy, M.ASCE[3]

Abstract: Real-life construction projects are large in size and are challenged by many constraints, including strict deadlines and resource limits. In this paper, constraint programming (CP) is used as an advanced mathematical technique that suits schedule optimization problems. A practical CP optimization model has been developed to resolve both deadline and resource constraints simultaneously in large-scale projects. The proposed CP model is much faster than metaheuristic techniques and provides a set of feasible project durations that do not violate resource limits. The paper compares the CP results with several case studies from the literature to prove the practicality and usefulness of the CP approach to both researchers and practitioners. The CP model of this paper could provide solutions within 6.5% deviation from optimum schedules for a large project of 2,000 activities within minutes of processing time. This paper thus contributes to introducing a superior optimization model that is suitable for large-size projects and helps to render schedule optimization a mainstream cost-saving function within commercial scheduling systems. **DOI: 10.1061/(ASCE)CO.1943-7862.0000722.** © *2013 American Society of Civil Engineers.*

CE Database subject headings: Construction management; Scheduling; Costs; Optimization.

Author keywords: Construction management; Scheduling; Constraint programming; Time-cost tradeoff; Resource constrained scheduling; Schedule optimization; Large-scale projects; Cost and schedule.

Introduction

Project deadline and resource limits are practical constraints that coexist in most projects. Whereas heuristic methods for resource-constrained scheduling (RCS) have become mainstream in commercial scheduling software, no commercial software includes any time-cost tradeoff (TCT) heuristic to help meet deadlines, let alone any procedure to resolve both deadline and resource constraints.

In the literature, there is a large body of knowledge related to resolving either the RCS or TCT problem, individually (Fig. 1). 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). Fig. 1 lists example research of the three types, with metaheuristics [e.g., genetic algorithms (GAs)] being the currently prevailing trend. This is because metaheuristics has advantages over exact and heuristic methods when used on small/medium examples (the same as those used in the literature). Metaheuristics uses random search mechanisms that are guaranteed to converge, whereas exact methods can be trapped in no-solutions. In addition, metaheuristics generally provide better solutions than heuristic methods. The few efforts in the literature that focused on the more complex problem of resolving both RCS and TCT simultaneously (Fig. 1), therefore, use metaheuristic methods, mostly GAs. Leu and Yang (1999) proposed one of the earliest models that integrate RCS and TCT using genetic algorithms. Senouci and Eldin (2004) presented another integrated model and used an augmented Lagrangian genetic algorithm for the optimization. Elazouni and Metwally (2007) also used genetic algorithms to expand finance-based scheduling and integrate the techniques of TCT analysis, resource allocation, and resource-leveling. Chen and Weng (2009) presented a GA-based model that segments the optimization into two phases, in which TCT is applied first, followed by another GA model for RCS. Zahraie and Tavakolan (2009) embedded the concepts of TCT, resource-leveling, and RCS in a stochastic multiobjective optimization using a modified GA model. Hegazy (2006) also introduced another GA approach that combines the RCS and TCT using two activity variables (method index and start delay) to resolve a combination of deadline and resource constraints.

To the writers' knowledge, none of the literature efforts of the three solution types in Fig. 1 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 both RCS and TCT constraints simultaneously. Their case study of 360 activities required 32 min 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

[1]Postdoctoral Fellow, Civil and Environmental Engineering Dept., Univ. of Waterloo, Waterloo, ON N2L 3G1, Canada. E-mail: wmenesi@uwaterloo.ca

[2]Graduate Student, Management Sciences Dept., Univ. of Waterloo, Waterloo, ON N2L 3G1, Canada. E-mail: bgolzarpoor@uwaterloo.ca

[3]Professor, Civil and Environmental Engineering Dept., Univ. of Waterloo, Waterloo, ON N2L 3G1, Canada (corresponding author). E-mail: tarek@uwaterloo.ca

Note. This manuscript was submitted on June 8, 2012; approved on March 29, 2013; published online on April 3, 2013. Discussion period open until February 1, 2014; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Construction Engineering and Management***, Vol. 139, No. 9, September 1, 2013. © ASCE, ISSN 0733-9364/(8)/$25.00.

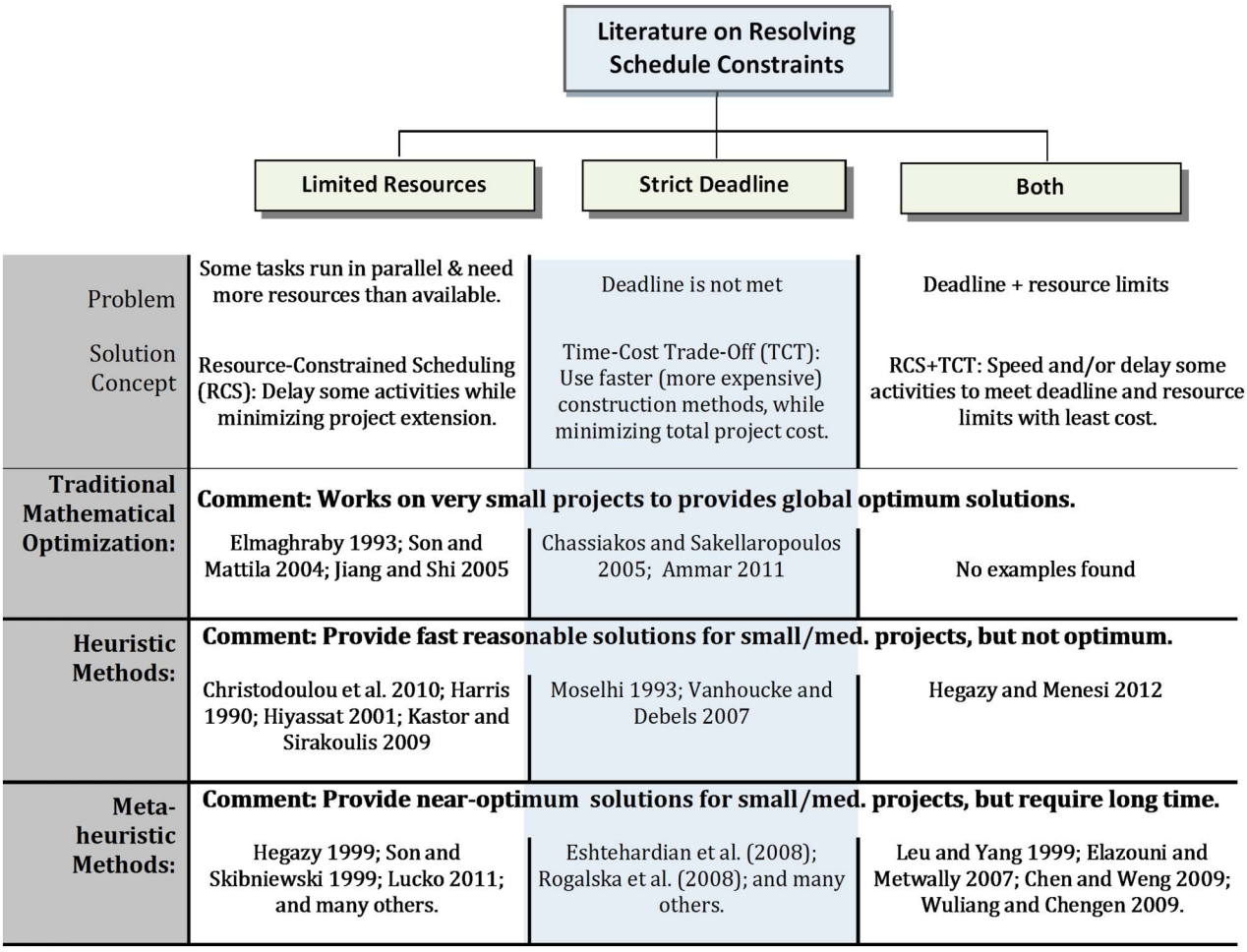

Fig. 1. Sample literature on resolving schedule constraints

rendering schedule optimization a difficult objective to achieve for large-scale projects. The primary objective of this paper, therefore, is to introduce a new approach using constraint programming (CP) to provide fast near-optimum solutions to much larger-scale RCS and TCT problems, thus providing a practical approach to optimize construction schedules.

Constraint Programming

Combinatorial optimization problems such as RCS and TCT have recently been addressed by an advanced computational method termed CP (Gorman and Kanet 2010; Brailsford et al. 1999). CP combines logic programming and operations research techniques, has been successfully used to solve complex combinatorial problems in a wide variety of domains, and has particular advantages in scheduling problems (Chan and Hu 2002; Heipcke 1999), including the following: (1) its efficient solution search mechanism, (2) flexibility to consider a variety of constraint types, and (3) convenience of model formulation. CP exploits the relationships between decision variables and model parameters to determine alternative feasible solutions (Chan and Zeng 2003), which is a unique characteristic of CP. To facilitate the use of CP algorithms in scheduling problems, IBM developed a powerful optimization package, termed *IBM ILOG CPLEX Optimization Studio* (Beck et al. 2011), incorporating a CP optimizer engine that offers features specially adapted to solving scheduling problems.

ILOG CPLEX Optimization Studio provides a programming language for describing a problem in terms of variables, objective function, constraints, and all of the internal computations, similarly to other optimization tools. In addition, it allows the use of specific constraints that are specialized to constraint programming and scheduling problems (e.g., a start-to-start logical relationship between two activities). Once the model is coded in the specific programming language of the software (as discussed in the next section), the problem is solved by the CPLEX-CP solver engine. According to the CP optimizer documentation (IBM 2012), it uses two techniques to find a solution, as follows: (1) constructive search, and (2) 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 search removes all of the 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 during the search until a solution is found.

CP Model for Schedule Optimization

Consider a construction project with n activities; each activity i has a set of associated construction methods (M_i). Every method k (from 1–M_i) represents a construction mode or a method of

J. Constr. Eng. Manage. 2013.139:1117-1124.

performing activity i. Associated with this method (k) are a specific duration (d_{ik}), cost (c_{ik}), and resources (r_{ik}). These construction methods vary from cheap (slow) to fast (expensive), and thus offer methods 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:

$$D_i = \sum_{k=1}^{M_i} d_{ik} X_{ik} \tag{1}$$

$$C_i = \sum_{k=1}^{M_i} c_{ik} X_{ik} \tag{2}$$

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, and $X_{ik} = 0$ for all of the other methods. To ensure only one mode of construction is used for each activity, one constraint is needed for each activity, in which the sum of the method indices X_{ik}s should be equal to 1:

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

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

$$TPC = PDC + PIC \tag{4}$$

where the project direct cost is the summation of the activities' direct costs in Eq. (2):

$$PDC = \sum_{i=1}^{n} C_i = \sum_{i=1}^{n} \sum_{k=1}^{M_i} c_{ik} X_{ik} \tag{5}$$

The project indirect costs 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

$$PIC = IC_0 + IC \times T \tag{6}$$

where IC_0 = fixed indirect costs (e.g., permits, mobilization cost, temporary hookups, temporary facilities, and purchase advances); IC = indirect cost per time period (i.e., daily expenditures); and T = 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

$$C_{pd} = Y \times C_d \times (T - \text{deadline duration}) \tag{7}$$

where Y is a zero-one variable representing if a project delay occurred (i.e., $Y = 1$ if the project duration $T >$ deadline duration); and C_d = cost of project delay per day (i.e., delay penalty). However, if the project is scheduled to be completed before the deadline, the incentive (IN) for early completion is expressed as

$$IN = Z \times B \times (\text{deadline duration} - T) \tag{8}$$

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

Objective Function

The objective of the optimization model is to minimize the sum of the costs described previously, formulated as

$$\text{Minimize } C = TPC + C_{pd} + IN \tag{9}$$

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

$$SS_i - SF_p \geq 0 \qquad p = 1, 2, \ldots, NP \tag{10}$$

where SS_i = scheduled start time of activity i; SF_p = scheduled finish time of the predecessor activity; and NP = number of immediate predecessors for activity i.

Project Completion Constraint

The project completion constraint is expressed as

$$SF_E \leq \text{deadline duration} \qquad E = 1, 2, \ldots, NE \tag{11}$$

where SF_E = finish time of ending activities; E = ending activity; and NE = number of ending activities.

Resource Constraints

The project resource constraints are expressed in the model as

$$\sum_{i \epsilon S_t} r_{ikl} \leq R_{lt}, \qquad l = 1, \ldots, L; \qquad t = 1, \ldots, T \tag{12}$$

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

Results of Experiments

Experiments on Small Projects

The previous formulation has been modeled in *IBM ILOG CPLEX Optimization Studio*, as shown in Fig. 2, with lines of code to specify the problem parameters, optimization variables, objective function, and constraints. As shown in Fig. 2, applying the model to optimize the schedule for a specific project requires arranging the project data (activities, relationships, and modes) into a specific format that is readable by the optimization model. For experimentation purposes, the developed model was applied on two literature case studies (Leu and Yang 1999; Chen and Weng 2009), with nine and 10 activities, respectively. The data file in Fig. 2 relates to the second case study (Chen and Weng 2009) and shows how the case study data in Fig. 3 are easily arranged to suit the model. For comparison purposes, the same two case studies were also solved by the heuristic approach of Hegazy and Menesi (2012) that combines RCS and TCT. As such, the results of CP optimization will be compared with the heuristic approach and the GA results reported in the cited publications of the two case studies.

For the two case studies, Table 1 shows a comparison of the results, showing the superior performance of the CP model in terms of both the solution quality and processing time. Fig. 4 shows a detailed CP solution to this 10-activity case.

J. Constr. Eng. Manage. 2013.139:1117-1124.

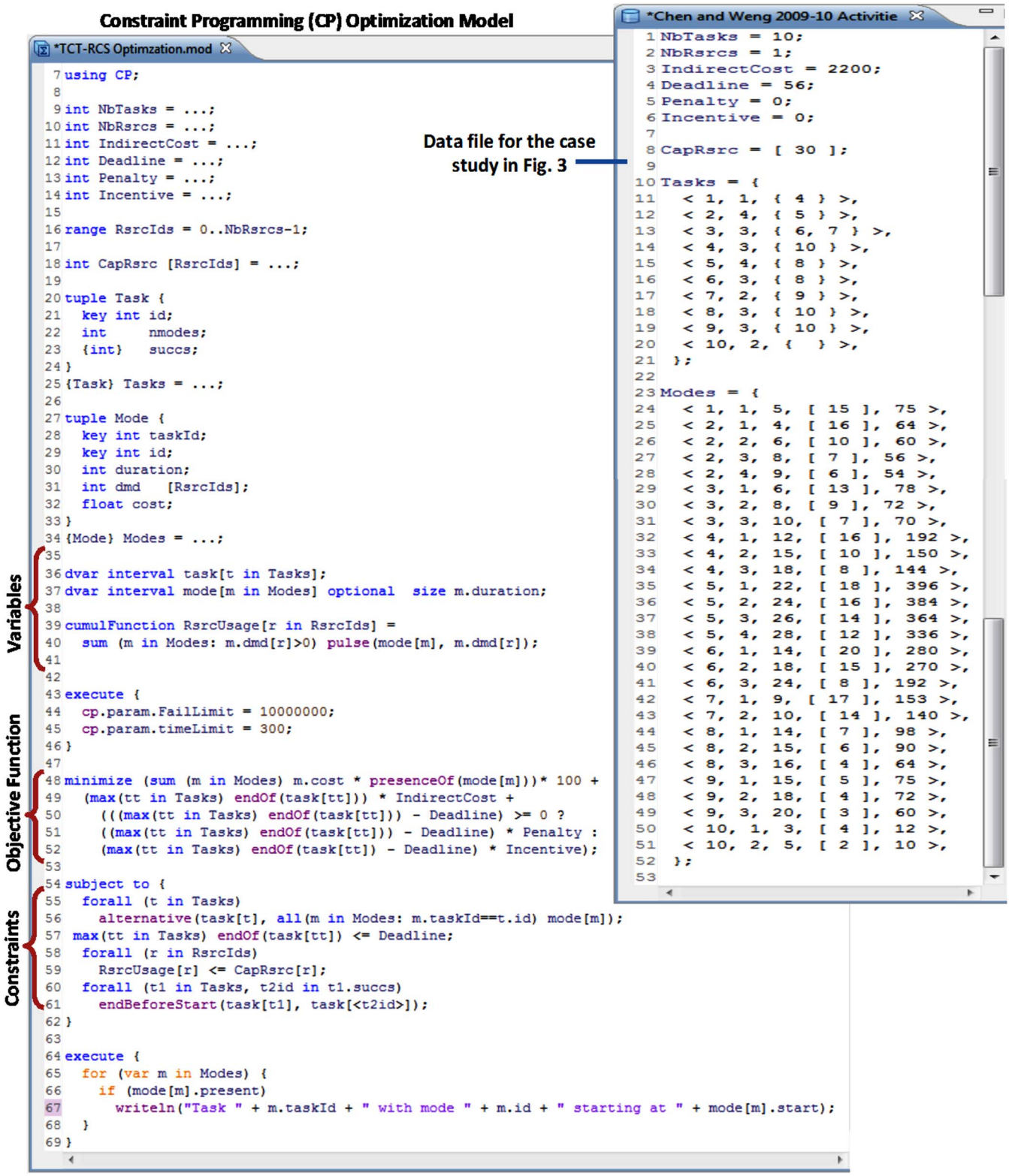

Fig. 2. CP model for resolving resource and deadline constraints

Experiments on Medium- and Large-Scale Projects

The case study of Chen and Weng (2009) was used as the basis for creating medium-size (100 and 300 activities) and larger-size (1,000 and 2,000 activities) projects. The base case study project consists of 10 activities having up to four discrete options that use varying amounts of one limited resource, as shown in Fig. 3. For example, the 100-activity project was created by copying the base project 10× in a serial manner. This approach of creating multiple copies has the benefit of knowing the expected optimum solution for all of the cases (multiples of the optimum solution in the 10-activity case), to be used for testing the quality of the solutions obtained using CP.

The experiments using the CP model and heuristic method were performed for the base case of 10 activities and for the projects of 100, 300, 1,000, and 2,000 activities. Table 2 presents the

J. Constr. Eng. Manage. 2013.139:1117-1124.

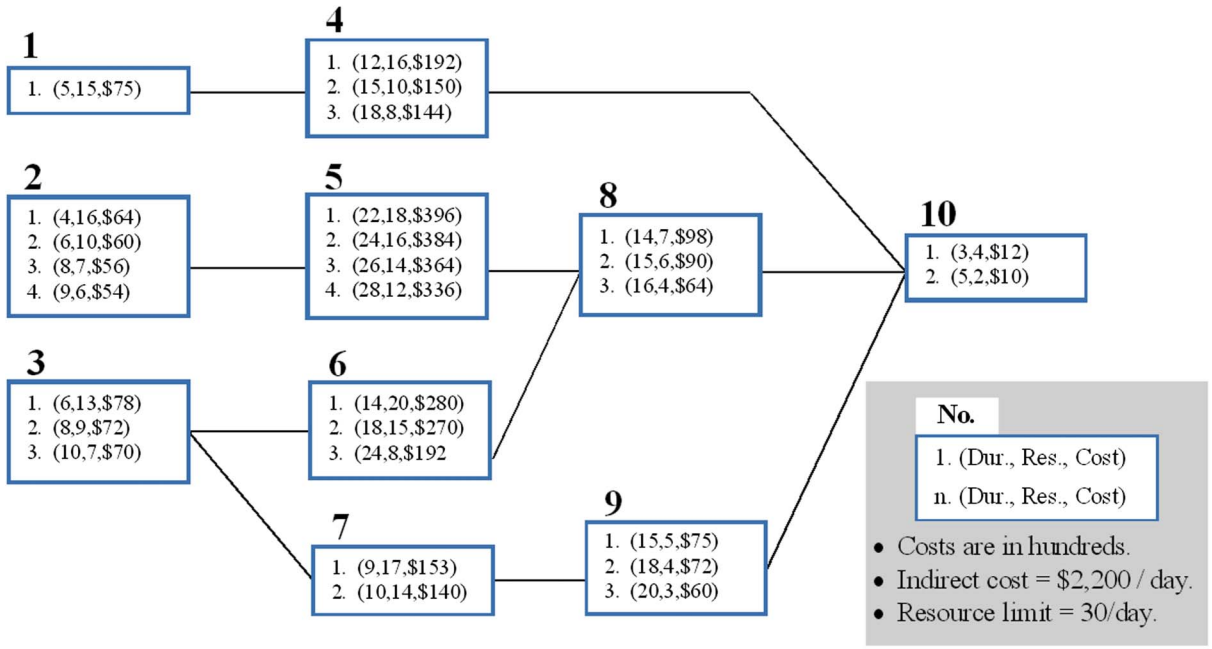

Fig. 3. Network and activity options for the example project of Chen and Weng (2009)

Table 1. Comparison among CP, GA, and Heuristic Methods on Small-Size Projects

Item	Leu and Yang (1999)	Chen and Weng (2009)
Number of activities	9	10
Case study	Each activity has up to five discrete options that use varying amounts of three limited resources	Each activity has up to four discrete options that use varying amounts of one limited resource (Fig. 3)
GA optimization	Project cost = US$7,400, project duration = 64 days, processing time = not noted	Project cost = US$244,000, project duration = 56 days, processing time = 8 min
Heuristic approach	Project cost = US$7,400, project duration = 64 days, processing time = 2 s	Project cost = US$245,900, project duration = 59 days, processing time = 2 s
Proposed CP	Project cost = US$7,400, project duration = 64 days, processing time = 1 s	Project cost = US$244,000, project duration = 56 days, processing time = 1 s
Comments	Identical results	CP produced the optimal solution in 1 s as compared with 8 min in GA; the heuristic approach produced a solution in 2 s, but is less optimal

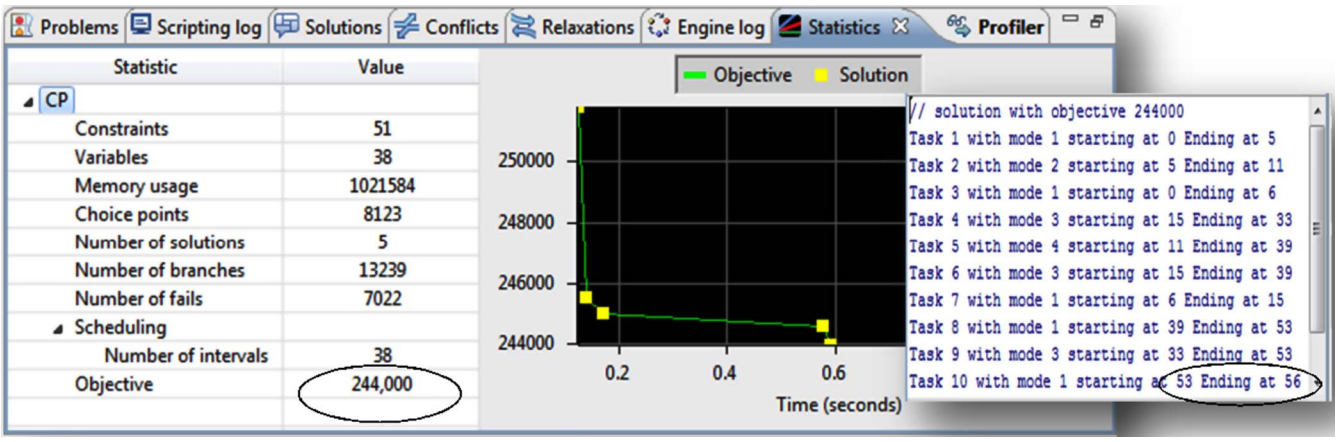

Fig. 4. CP solution for the 10-activity case

J. Constr. Eng. Manage. 2013.139:1117-1124.

Table 2. Results of the CP Model on Different Project Sizes

Project size	Expected optimum solution	CP solution: Duration (Days) and cost (US$)	CP solution: Processing time	CP deviation (%)[a]	Heuristic solution[b]
10 activities 38 variables 51 constraints	Duration = 56, cost = US$244,000	Duration = 56, cost = 244,000	1 s	0	Duration = 59, cost = US$245,900, proc. time = 3 s
100 activities 380 variables 510 constraints	Duration = 560, cost = US$2.44 million	Duration = 566, cost = 2.497 million	15 s	2.34	Duration = 590, cost = US$2.459 million, proc. time = 1 min, deviation = 0.78%
		Duration = 555, cost = 2.4529 million	5 min	0.53	
		Duration = 556, cost = 2.4479 million	10 min	0.32	
300 activities 1,140 variables 1,530 constraints	Duration = 1,680, cost = US$7.32 million	Duration = 1,820, cost = 7.7517 million	15 s	5.90	Duration = 1,770, cost = US$7.377 million, proc. time = 21 min, deviation = 0.78%
		Duration = 1,735, cost = 7.4794 million	5 min	2.18	
		Duration = 1,698, cost = 7.4296 million	10 min	1.46	
		Duration = 1,686 cost = 7.3489 million	20 min	0.88	
1,000 activities 3,800 variables 5,100 constraints	Duration = 5,600, cost = US$24.4 million	Duration = 6,121, cost = 25.9238 million	15 s	6.24	Heuristic method not suitable for this problem size
		Duration = 6,128, cost = 25.8827 million	5 min	6.07	
		Duration = 5,986, cost = 25.5717 million	20 min	4.80	
		Duration = 5,929, cost = 25.4197 million	120 min	4.18	
2,000 activities 7,600 variables 10,200 constraints	Duration = 11,200, cost = US$48.8 million	Duration = 12,353, cost = 52.0531 million	40 s	6.67	Heuristic method not suitable for this problem size
		Duration = 12,313, cost = 52.0022 million	10 min	6.56	
		Duration = 12,298, cost = 51.9692 million	30 min	6.50	
		Duration = 12,274, cost = 51.9164 million	120 min	6.39	

[a]Deviation from optimal = (CP cost in column 3 − optimum cost in column 2)/optimum cost in column 2.
[b]Proc. = processing.

optimization results for each experiment, including the solution's total cost, project duration, processing time, and deviation from the optimum solution. The deviation is calculated as

$$\text{Deviation}(\%) = [(\text{solution cost} - \text{optimum cost}) \times 100]/\text{optimum cost} \quad (13)$$

As shown in Table 2, the CP solution for the 10-activity project (involving 38 variables and 51 constraints) reached the optimal solution and thus its deviation from the optimal solution is zero. The solution required only 1 s, which is a very fast performance. As opposed to this solution, the heuristic method required only 3 s but could not reach the optimum solution.

For medium-size projects (100 and 300 activities), both the heuristic approach and the CP model were applied, and Table 2 summarizes the results. All of the experiments were performed on a laptop with a 2.4-GHz CPU and 3 GB of random-access memory (RAM). In terms of processing time, the heuristic approach required about 1 and 21 min to process the case studies of 100 and 300 activities, respectively. The CP model, however, shows improved results as the processing time increases. Table 2 shows the results of three experiments for the case of 100 activities, with processing times of 15 s, 5 min, and 10 min, respectively. In the latter two cases, the deviation from optimal was reduced to less than 1%. For a 300-activity project, the CP model was able to reach a solution with a cost deviation of 5.9% within 15 s. The quality of the CP solution for the same 300-activity project was then greatly improved when the processing time was extended to 20 min, attaining a deviation of less than 1% from the optimal solution. In both the 100 and 300 activity cases, the CP solution is much better in terms of project duration than the heuristic method, which also provided less than 1% cost deviation from optimum. Therefore, for medium-size projects, the CP model can reach within 1% from the optimal cost, with a duration that is close to optimum without violating resource limits, within a reasonable processing time (and can be much faster when using a more powerful machine).

For large-size projects with 1,000 and 2,000 activities, only the CP model was applied because the heuristic approach would not be practical; it would have required a very long processing time (one experiment on 1,000 activities required the heuristic method more than 16 h without producing a feasible result). Fig. 5 presents a summary of CP solutions for different problem sizes. For the 1,000-activity case, increased processing time could reduce the deviation from 6 to 4%, whereas in the 2,000 activity case, which is extremely large, the deviation remained at about 6.5% even with

J. Constr. Eng. Manage. 2013.139:1117-1124.

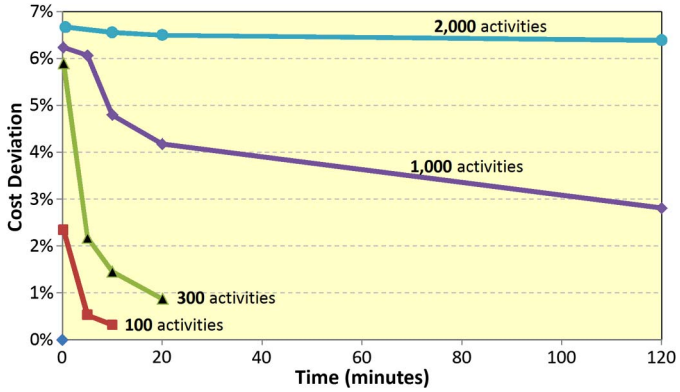

Fig. 5. CP solution quality versus processing time for different problem sizes

a longer processing time (up to 2 h). This result is considered practically reasonable. A deviation of 6.5% from the optimum in the case of 2,000 activities is not high. Based on these results, the CP model proved to provide good solution quality with a reasonable processing time (30 min), 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 upon which to improve.

Experiments on Multimode Resource-Constrained Project Scheduling

Other experiments of the CP model involved applying it to a different schedule optimization problem known as the multimode resource-constrained project scheduling problem (MRCPSP). Similarly to the TCT and RCS problem, the MRCPSP deals with projects in which the activities have more than one mode of execution with different resource requirements. In this case, the objective is to minimize the total duration (not the cost) of the project and not exceed the resource limits.

A good survey of the exact, heuristic, and metaheuristic procedures that have been proposed in the literature for MRCPSP problems can be found in Peteghem and Vanhoucke (2010). Among the most recent efforts, Zhang (2012) used an ant colony optimization (ACO) method to solve MRCPSP problems. Zhang (2012) used a 10-activity project with three modes and one type of renewable resource. Using a modified version of the constraint programming model of this paper, the CP model was able to determine the optimal schedule in less than 1 s.

Discussion of CP Performance

Based on the many experiments performed using the CP model, several observations can be made, as follows:

- *IBM ILOG CPLEX Optimization Studio* proved to be a useful tool for constraint programming. It is available freely for academic use and its focus on scheduling renders it very useful for construction projects. This tool is easily programmable and has familiar terminology such as successors, modes, resources, and so on.
- Based on many experiments, the results of the CPLEX-CP solver are sensitive to the deadline constraint and processing time. Setting a relaxed deadline duration (i.e., longer than needed) and a reasonable amount of processing time can reach a better solution than using a strict deadline (this may be because the software quickly eliminates many solution branches that initially violate the deadline constraint, even though some of them can provide better solutions after several cycles of processing).
- It is possible to integrate the proposed CP model as an add-on to existing project management software to incorporate powerful schedule optimization capabilities.
- In general, future work in this research area is suggested to focus on introducing better results on larger-size problems. Introducing new metaheuristics or tweaking existing ones to achieve minor performance improvement on small-size problems (10 or 20 activities) has no practical value.
- Ongoing research by the writers includes experimenting with the proposed CP model on actual projects and modifying the model to suit multiple baseline updates and schedule optimization during construction.
- The results reported in this paper can be used as a benchmark for the writers and other researchers to compare and improve. The challenge is to attain 1% deviation from optimum, in terms of both duration and cost, within a few minutes of processing time on a personal computer, for a network of 1,000 activities or more.

Summary and Conclusions

Resource-constrained scheduling and time-cost tradeoff problems have attracted many researchers in the past few decades. This paper first highlighted research efforts that introduced RCS models to fulfill resource constraints, TCT models to meeting project deadlines, and other combined models that try to satisfy both deadlines and resource limits in a single procedure. None of these literature efforts have addressed practical-size problems with more than 1,000 activities. This paper therefore proposed a constraint programming approach that combines RCS and TCT analyses to resolve both time and resource constraints simultaneously for large-scale projects. The proposed CP approach has been programmed in *IBM ILOG CPLEX Optimization Studio* software. Experimenting on several cases proved the ability of the proposed CP model to produce near-optimum solutions to large-scale projects consisting of 1,000 and 2,000 activities within a few minutes of processing time, which is a performance unmatched by metaheuristic models such as genetic algorithms.

This research contributes to developing a practical decision support system for resolving multiple constraints in real-size construction projects. The full CPLEX-CP code provided in the paper represents a well-structured and easy-to-follow model that readers can readily use and upon which to improve. The CP approach can be used not only to determine a feasible construction schedule but, with continued improvements, can also be applied throughout the construction process to keep projects on track. Researchers are challenged to improve upon the results of this paper to attain 1% deviation from the optimum time and cost within a few minutes of processing time on a personal computer for a network of 1,000 activities or more. Addressing this challenge will motivate the adoption of schedule optimization features within commercial scheduling systems.

References

Ammar, M. (2011). "Optimization of project time-cost trade-off problem with discounted cash flows." *J. Constr. Eng. Manage.*, 137(1), 65–71.

Beck, J. C., Feng, T. K., and Watson, J. (2011). "Combining constraint programming and local search for job-shop scheduling." *INFORMS J. Comput.*, 23(1), 1–14.

J. Constr. Eng. Manage. 2013.139:1117-1124.

Brailsford, S. C., Potts, C. N., and Smith, B. M. (1999). "Constraint satisfaction problems: Algorithms and applications." *Eur. J. Oper. Res.*, 119(3), 557–581.
Chan, W. T., and Hu, H. (2002). "Constraint programming approach to precast production scheduling." *J. Constr. Eng. Manage.*, 128(6), 513–521.
Chan, W. T., and Zeng, Z. (2003). "Coordinated production scheduling of prefabricated building components." *Proc., Construction Research Congress*, ASCE, Reston, VA, 1–8.
Chassiakos, A. P., and Sakellaropoulos, S. P. (2005). "Time-cost optimization of construction projects with generalized activity constraints." *J. Constr. Eng. Manage.*, 131(10), 1115–1124.
Chen, P. H., and Weng, H. (2009). "A two-phase GA model for resource constrained project scheduling." *Automat. Constr.*, 18(4), 485–498.
Christodoulou, S. E., Ellinas, G., and Michaelidou-Kamenou, A. (2010). "Minimum moment method for resource leveling using entropy maximization." *J. Constr. Eng. Manage.*, 136(5), 518–527.
Elazouni, A., and Metwally, F. (2007). "Expanding finance-based scheduling to devise overall-optimized project schedules." *J. Constr. Eng. Manage.*, 133(1), 86–90.
Elmaghraby, S. (1993). "Resource allocation via dynamic programming in activity networks." *Eur. J. Oper. Res.*, 64(2), 199–215.
Eshtehardian, E., Afshr, A., and Abbasnia, R. (2008). "Time–cost optimization: Using GA and fuzzy sets theory for uncertainties in cost." *Constr. Manage. Econ.*, 26(7), 679–691.
Gorman, M., and Kanet, J. (2010). "Formulation and solution approaches to the rail maintenance production gang scheduling problem." *J. Transp. Eng.*, 136(8), 701–708.
Harris, R. B. (1990). "Packing method for resource leveling (pack)." *J. Constr. Eng. Manage.*, 116(2), 331–350.
Hegazy, T. (1999). "Optimization of resource allocation and leveling using genetic algorithms." *J. Constr. Eng. Manage.*, 125(3), 167–175.
Hegazy, T. (2006). "Simplified project management for construction practitioners." *Cost Eng. J.*, 48(11), 20–28.
Hegazy, T., and Menesi, W. (2012). "Heuristic method for satisfying both deadlines and resource constraints." *J. Constr. Eng. Manage.*, 138(6), 1–9.
Heipcke, S. (1999). "Comparing constraint programming and mathematical programming approaches to discrete optimization–The change problem." *J. Oper. Res. Soc. Jpn.*, 50(6), 581–595.
Hiyassat, M. A. S. (2001). "Applying modified minimum moment method to multiple resource leveling." *J. Constr. Eng. Manage.*, 127(3), 192–198.
IBM ILOG CPLEX Optimization Studio V12.3. (2012). [Computer software]. IBM, Armonk, NY.
Jiang, G., and Shi, J. (2005). "Exact algorithm for solving project scheduling problems under multiple resource constraints." *J. Constr. Eng. Manage.*, 131(9), 986–992.
Kandil, A., and El-Rayes, K. (2005). "Parallel computing framework for optimizing construction planning in large-scale projects." *J. Comput. Civ. Eng.*, 19(3), 304–312.
Kastor, A., and Sirakoulis, K. (2009). "The effectiveness of resource levelling tools for resource constraint project scheduling problem." *Int. J. Proj. Manage.*, 27(5), 493–500.
Leu, S., and Yang, C. (1999). "GA-based multicriteria optimal model for construction scheduling." *J. Constr. Eng. Manage.*, 125(6), 420–427.
Lucko, G. (2011). "Integrating efficient resource optimization and linear schedule analysis with singularity functions." *J. Constr. Eng. Manage.*, 137(1), 45–55.
Moselhi, O. (1993). "Schedule compression using the direct stiffness method." *Can. J. Civ. Eng.*, 20(1), 65–72.
Peteghem, V. V., and Vanhoucke, M. (2010). "A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem." *Eur. J. Oper. Res.*, 201(2), 409–418.
Rogalska, M., Bożejko, W., and Hejducki, Z. (2008). "Time/cost optimization using hybrid evolutionary algorithm in construction project scheduling." *Automat. Constr.*, 18(1), 24–31.
Senouci, A., and Eldin, N. (2004). "Use of genetic algorithms in resource scheduling of construction projects." *J. Constr. Eng. Manage.*, 130(6), 869–877.
Son, J., and Mattila, K. (2004). "Binary resource leveling model: Activity splitting allowed." *J. Constr. Eng. Manage.*, 130(6), 887–894.
Son, J., and Skibniewski, M. (1999). "Multiheuristic approach for resource leveling problem in construction engineering: Hybrid approach." *J. Constr. Eng. Manage.*, 125(1), 23–31.
Vanhoucke, M., and Debels, D. (2007). "The discrete time/cost trade-off problem: Extensions and heuristic procedures." *J. Schedul.*, 10(4–5), 311–326.
Wuliang, P., and Chengen, W. (2009). "A multi-mode resource-constrained discrete time–cost tradeoff problem and its genetic algorithm based solution." *Int. J. Proj. Manage.*, 27(6), 600–609.
Zahraie, B., and Tavakolan, M. (2009). "Stochastic time-cost-resource utilization optimization using nondominated sorting genetic algorithm and discrete fuzzy sets." *J. Constr. Eng. Manage.*, 135(11), 1162–1171.
Zhang, H. (2012). "Ant colony optimization for multimode resource-constrained project scheduling." *J. Manage. Eng.*, 28(2), 150–159.

J. Constr. Eng. Manage. 2013.139:1117-1124.