Hybrid Approach of Constraint Programming and Integer Programming for Solving Resource-constrained Project-scheduling Problems

CHENG Zhao[1], TOMOHIRO MURATA[2]

***Abstract*—Resource –constrained project scheduling problem (RCPSP) is a well known NP hard problem. This paper proposes a hybrid approach of constraint programming (CP) and integer programming (IP) which could solve RCPSP with ideal efficiency and quality. Constraint propagation is key element of constraint programming and is widely used in cumulative resource problem (CRP). CRP can be taken as a relaxed problem of RCPSP. This paper discussed how to eliminate the resource constraint, and transform RCPSP to project scheduling problem (PSP) so that the problem will became easier. Experiment shows this hybrid approach has certain efficient advantage compared with traditional CP method.**

***Index Terms*—constraint programming (CP), cumulative resource problem (CRP), integer programming (IP), project scheduling problem (PSP), resource-constrained project scheduling problems (RCPSP)**

I. INTRODUCTION

RESOURCE-constrained project scheduling problem (RCPSP) is wildly used in real world. Most traditional way of solving RCPSP is integer programming, but because of the complexity of NP-hard, integer programming is very hard to solve more than 200 activities' scale problem. Then many approaches were proposed to deal with this NP-hard problem, e.g., constraint programming or genetic algorithm, but the quality of the solution is still not good enough.

Project scheduling problem (PSP) is the basic problem. In project scheduling problem, a set of activities or tasks with certain priority should be scheduled with the objective of minimal make-span. Based on project scheduling problem, if there is single renewable resource of limited capacity must be considered, the problem turn to be the cumulative resource problem (CRP). Furthermore, if there are more than one limited resources, this would be the resource constrained scheduling problem (RCPSP).

1. CHENG Zhao, graduate student. Graduate School of Information, Production and System, Waseda University, Japan (e-mail: chengzhao@akane.waseda.jp).

2. TOMOHIRO MURATA, professor. Graduate School of Information, Production and System, Waseda University, Japan (e-mail: t-murata@waseda.jp).

For years, researchers developed many ways to solve such kind of scheduling problem. As integer programming could gives the optimal solution with unsatisfactory calculating time, more and more people focus on the constraint programming.

Whereas single approach either exact or heuristic could not balance quality and efficiency very well, people are getting to study hybrid approach with the purpose of fast and accurate solutions of RCPSP.

This paper proposed a constraint propagation algorithm to get the earliest start time and latest end time of activities, and a hybrid approach to solve the problem without resource conflict. Part II describes the problem model and some theorems. Part III shows the detailed procedures of the algorithm. Part IV gives some experiment results. Final part is conclusion.

II. PROBLEM DESCRIPTION AND MATHEMATICAL MODEL

A. *RCPSP problem and mathematical model*

RCPSP could be represented by a directed acyclic graph $G=(V,H)$ and a node set $V=(1,2,\cdots,J)$. Node set $V=(1,2,\cdots,J)$ is the set of activities, and the arc set H is set of activities' precedence relations. Arc $(i,j)\in H$ represents activity j must starts directly after activity i, gives $x_{ij}=\begin{cases}1,\\0,\end{cases}$ denotes the precedence relation between activity i and activity j.

Activity 1 is the only earliest activity and activity J is the only latest activity, represent the start and the end of the project. r_i is the release date of activity i, p_i is processing time and d_i is the due date. Activity i's start time is t_i. In every time period, activity i requires r_{ik} units of resource $k\left(k=1,\cdots,J\right)$. Available resource amount of resource $k\left(k=1,\cdots,J\right)$ is R_k. Objective is finding a minimal make-span solution which satisfies both precedence constraint and resource constraint.

[2] introduced the concept of forbidden set which could be used as guide for introduction for new precedence constraint.

Forbidden set Ω is a set of activities. Given $\forall i, j \in \Omega$, activity i and j are technologically independent, and for the reason of resource constraint, they are not allowed to be scheduled as a whole at any time. If no sub-forbidden set exits in Ω, we call Ω minimal forbidden set.

As the definition shown above, we can see, given any minimal forbidden set Ω, with any $k \in \{1, \cdots, K\}$, $\sum_{j \in \Omega} r_{jk} \succ R_k$, as while as $\sum_{j \in \Omega - \{i\}} r_{jk} \prec R_k$ with any $i \in \Omega$. Obviously, if we add a new precedence relation of any two activities in any minimal forbidden set, it will eliminate the resource conflict. If all the minimal forbidden sets could be eliminated, then we could get a feasible solution of RCPSP.

Ω_p is the minimal forbidden set of only two activities, and Ω_n has more than three activities, $p \in (1, \cdots, P)$, $n \in (1, \cdots, N)$. M is bigger than any other element of the problem. H^* is transitive precedence.

RCPSP mathematical model is:

$$\min t_j, \tag{1}$$

$$\text{s.t.}: x_{ij} = 1, x_{ji} = 0, \forall (i, j) \in H^*, \tag{2}$$

$$0 \le x_{ij} + x_{ji} \le 1, \forall i, j \in V, \tag{3}$$

$$x_{ik} \ge x_{ij} + x_{jk} - 1, \forall i, j, k \in V, \tag{4}$$

$$\textbf{P:}\ x_{ij} + x_{ji} = 1, i, j \in \Omega_p, p = 1, \cdots, P, \tag{5}$$

$$\sum_{i, j \in \Omega_n} x_{ij} \ge 1, n = 1, \cdots, N, \tag{6}$$

$$t_j \ge t_i + (p_i + M) x_{ij} - M, \forall i, j \in V, \tag{7}$$

$$x_{ij} \in \{0, 1\}, \forall i, j \in V, \tag{8}$$

$$r_i \le t_i \le d_i - p_i, \forall i \in V. \tag{9}$$

The objective function (1) is to minimize the total completion time. Equations (2) reflect the precedence constraints. Constraints (3) are the transitivity constraints. (3) and (4) jointly eliminate cycles. The constraints (5) take exactly on arc from each incompatible pair and (6) at least on arc form each incompatible set. Constraints (7) relate t_i to t_j if there is an arc from I to j. If there is not, the value of M leaves them unrelated. Finally, (8) and (9) are the integrality constraints.

B. Constraint propagation of CRP

Constraint propagation is one of the key elements of constraint programming. When cumulative resource problem (CRP) constraint propagation algorithm executes the propagation, it usually transforms the precedence constraint to the available time zone, i.e. release date and due date.

Here gives the definition of CRP. CRP can be represented by a set of activities $V = (1, 2, \cdots, J)$ and a renewable resource. Resource limit $C \succ 1$, gives total energy needs $e_i = r_{i1} \times p_i$ by activity i. Its objective is to minimize the make-span, while the resource demand doesn't exceed the limit. CRP can be seen as a relaxation problem from RCPSP. To deal with RCPSP, we can pick one resource constraint, relaxing others, to get a CRP instance.

TABLE I
PARAMETERS OF MODEL

Symbol	Meaning
V	Set includes all activities
G	Direct acyclic graph
H	Arc
Ω	Forbidden set
r_i	Release date
r_{ik}	Resource capacity demand
R_k	Resource capacity
p_i	Processing time
M	A large number than any other element
Ω_p	Minimal forbidden set
e_i	Energy demand
$E(t_1, t_2)$	Energy consumption
$E(i, t_1, t_2)$	Energy consumption by activity i
t_i	Start time

Proposition 1[3]

Given activity i and time zone $[t_1, t_2]$, let $E(i, t_1, t_2)$ be the "left-shift/right-shift" required energy consumption of activity i over $[t_1, t_2]$.

$$E(i, t_1, t_2) = r_{i1} \times \min\left(t_2 - t_1, p_i^+(t_1), p_i^-(t_2)\right), \tag{10}$$

$$p_i^+(t_1) = \max\left(0, p_i - \max(0, t_1 - r_i)\right), \tag{11}$$

$$p_i^-(t_2) = \max\left(0, p_i - \max(0, d_i - t_2)\right). \tag{12}$$

$E(t_1, t_2)$ is all the activities' energy consumption over $[t_1, t_2]$. $E(t_1, t_2) = \sum_{i \in V} E(i, t_1, t_2)$.

Proposition 2[5]

If a instance of CuSP satisfies

$$E(t_1, t_2) \le R_1 \times (t_2 - t_1), \forall [t_1, t_2], t_1 \prec t_2 \prec UB, \tag{13}$$

UB is upper bound.

then this instance's energy is available.

Reference [7] proposed a constraint propagation theorem of CRP.

Theorem 1 P_i is a set of activities before activity i. Earliest start time of i will be r_i'.

$$r_i' = \max(r_i, r_{P_i} + \frac{1}{R_1} \sum_{j \in P_i} r_{j1} \times p_j)$$

Reference [6] proposed another constraint propagation theorem.

Theorem 2 V' is a subset of V, given activity $i \notin V'$. If $e_{V'\cup\{i\}} \succ C \times (d_{V'} - r_{V'\cup\{i\}})$, then V' should end before i.

If $e_{V'\cup\{i\}} \succ C \times (d_{V'\cup\{i\}} - r_{V'})$, then V' should start after i.

Theorem 1 only considers the pre-order, and ignores the activities after i.

Theorem 2 only considers the available time zone, and ignores the precedence constraint.

Theorem 2 can eliminate the precedence constraint. But its available time zone is too large, that cause low efficiency. So here bring in the *theorem 1* to strengthen the constraint converting.

Here gives a CRP constraint propagation algorithm:

i. Calculating all the $e_{V'}$, and using *theorem 2* get the new release date r_i' and due date d_i'.

ii. Using theorem 1 renew the release date r_i' and due date d_i'.

iii. If new release date and due date was generated, go to step i, otherwise, output the release date r_i' and due date d_i'.

By these procedures, we can get a rough P model, which has removed some of the possible resource conflicts.

III. Algorithm design

A. Preprocessing

The heart of this hybrid approach is problem transformation and constraint converting. These are preprocessing of the problem.

First, initial release date and due date could be calculate out by critical path method. Second, select one resource as the only renewable resource, in order to relax the RCPSP to CRP. Use CRP propagation algorithm to renew the release date and the due date so that the decision variable's domain could be greatly reduced. Then execute this algorithm to other resource one by one. Finally, output the new release date and the due date. Till now, constraint propagation preprocessing is over.

New release date and due date could limit domain and reduce the resource conflict, but it still won't eliminate the conflict. So it needs another processing to completely remove the resource constraint by generating new precedence constraint instead of resource constraint. Detailed procedures will be introduced after.

After the preprocessing of RCPSP, we can get new problem model **P**. In this model, there is no resource constraint, and also, some decision variables' domains are cut, which makes the calculation much easier.

B. Algorithm design

The main idea of hybrid algorithm for RCPSP is relaxing RCPSP, lower the complexity.

First step, CRP propagation algorithm reduce the single resources constraint and generate the available time zone for each activity, i.e. renew the release date and due date. This step will repeat till all the resources are included.

Second step, we find all possible resource conflicts, and generate new precedence constraint in order to eliminate these resource conflicts.

Finally, after these two steps above, new release date and new date and new precedence are generated. After these procedures, original resource-constrained project scheduling problem is simplified to project scheduling problem. With help of domain of starting time, computation becomes much easier.

Complete procedures are shown as below:

i. Using CRP propagation algorithm renew every activities' r_i and d_i, new release date and due date are r_i' and d_i'.

ii. If $(i,j) \in H^*$, $r_i' \geq d_i'$, then add new arc (j,i) into the graph.

iii. Generate all the minimal forbidden set Ω_p.

iv. If $i, j \in \Omega_p$, $r_i' + p_i + p_j \geq d_i$, then add new arc (j,i) into the graph.

v. If $i, j \in \Omega_p$, and new arc (i,j) will cause critical path length increases, then add new arc (j,i) into the graph.

vi. If $i, j \in \Omega_p$, and new arc (i,j) will cause no energy feasible solution, then add new arc (j,i) into the graph.

vii. If new arc was generated from ii to vi, turn to i. Otherwise go to viii.

viii. Generate new graph and output every activities' r_i' and d_i'.

ix. Generate time window for t_i,

$$r_i' \leq t_i \leq d_i' - p_i, i = 1, \cdots, n.$$

x. Preprocess variable $x_{ij}, \forall (i,j) \notin H^*$.

If $r_i' + p_i + p_j \geq d_j'$, set $x_{ij} = 0$;

If new arc (i,j) cause critical path length larger than UB, set $x_{ij} = 0$;

If new arc (i,j) will cause no energy feasible solution, set $x_{ij} = 0$.

xi. Output mode **P** and calculate out solutions.

C. Effect of algorithm

This preprocessing has two effects: ① Determine the precedence of activities of some minimal forbidden set, eliminate the precedence constraint. ② Reduce the domain of

ISBN: 978-988-19251-9-0
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

decision variable x_{ij}, cut down the searching space of t_1.

IV. Experiment Design and Result

In experiment, we focus on two behaviors, computing efficiency and solution's optimality. Efficiency could be represented by quantity of solutions calculated in unit of time.

TABLE II
THE COMPARISON OF PERFORMANCES OF TWO RUNS

	Hybrid approach	Traditional CP
Feasible solutions found	15	6
Best objective	111	119
Solution optimality	98.2%	91.6%
CPU/s	3600	3600

As comparison, I chose pure constraint programming algorithm.

I use standard instance of PSPLIB as test case. The instance is J120. It has 120 activities plus a start activity and an end activity. It is large scale and very difficult to schedule. Implementation platform is ILOG and C++ library.

For some instances of J120, PSPLIB already gives the upper bound and lower bound. If the upper bound and the lower bound are same, it means the optimal value is determined. Take J120_1_2 as example, the upper bound and the lower bound are both 109, i.e., its optimal objective value is 109.

TABLE III
THE COMPARISON OF AVERAGE PERFORMANCES OF TWO RUNS

	Hybrid approach	Traditional CP
Avg Solution optimality	97.9%	93.4%
CPU/s	3600	3600

In the experiment, the upper bound was set at 150, and the time limit was set at 3600s. The results of J120_1_2 are shown as TABLE II.

As we can see from TABALE II, Hybrid approach of CP and IP shows great superiority in efficiency. During 3600 CPU seconds, hybrid approach has found 15 solutions, but pure CP approach has only found out 6 solutions.

Then more experiments are implemented, i.e., J120_1_3, J120_1_7, J120_1_9, J120_3_2, J120_3_6, J120_4_4, J120_5_9, J120_8_1 and J120_14_3. Seen as TABLE III, average results prove the hybrid approach's merit in general cases.

V. Conclusions

This paper proposed a hybrid approach of CP and IP for RCPSP. The experiment shows constraint propagation algorithm's great performance in preprocessing and simplifying large-scale problem. After the transformation from RCPSP to time-window constraint PSP, the problem is much easier to computing, and the efficiency is greater than the traditional CP progress. Also in an hour's computing time, both hybrid approaches traditional CP failed to find optimal solution of J120_1_2 (which its make-span is 109), but still hybrid approach gave the better solution than traditional CP approach does.

References

[1] R. H. Mohring, Andreas S. Schulz, F. Stork, M. Uetz, Solving Project Scheduling Problems by Minimum Cut Computations. *Management Science © 2003 INFORMS*, Vol. 49, No.3, March 2003 pp. 330-350. .

[2] C. Artigues, S. Demassey, E. Neron, Resource-Constrained Project Scheduling: Models, Algorithms, Extensions and Applications. 1993, pp. 49–135.

[3] Baptiste P., Le Pape C, Nuijten W. *Constrained Based Scheduling.* Amsterdam: Kluwer, 2001.

[4] R. Alvarez-Valdés, J. M. Tamarit. "The project scheduling polyhedron: dimension, facets and lifting theorems [J]", *European Journal of Operational Research,* 1993, 67(2), pp. 204-220.

[5] Mercier L, Hentenryck P V. Edge finding for cumulative scheduling. *Informs Journal on Computing*, 2008, 20(1), pp. 143-153.

[6] Nuijten W. Time and Resource Constrained Scheduling: A Constraint Satisfaction Approach [Ph.D. dissertation], Eindhoven University of Technology, Eindhoven, Netherlands, 1994.

[7] Laborie P. Algorithms for propagating resource constraints in A.I. planning and scheduling: existing approaches and new results. *Artificial Intelligence*, 2003, 143(2), pp. 151-188.