2005 한국경영과학회/대한산업공학회 춘계공동학술대회
2005 년 5 월 13 일~14 일, 충북대학교

Investment Scheduling of Maximizing Net Present Value of Dividend with Reinvestment Allowed

Chang Sup Sung, Joo Hyung Song and Woosuk Yang
Department of Industrial Engineering, KAIST
Tel : 042-869-3154, Fax: 042-869-3110
cssung@kaist.ac.kr, roalkind@hanmail.net, parani@kaist.ac.kr

Abstract

This paper deals with an investment scheduling problem of maximizing net present value of dividend with reinvestment allowed, where each investment has certain capital requirement and generates deterministic profit. Such deterministic profit is calculated at completion of each investment and then allocated into two parts, including dividend and reinvestment, at each predetermined reinvestment time point. The objective is to make optimal scheduling of investments over a fixed planning horizon which maximizes total sum of the net present values of dividends subject to investment precedence relations and capital limit but with reinvestment allowed. In the analysis, the scheduling problem is transformed to a kind of parallel machine scheduling problem and formulated as an integer programming which is proven to be NP-complete. Thereupon, a depth-first branch-and-bound algorithm is derived. To test the effectiveness and efficiency of the derived algorithm, computational experiments are performed with some numerical instances. The experimental results show that the algorithm solves the problem relatively faster than the commercial software package (CPLEX 8.1), and optimally solves the instances with up to 30 investments within a reasonable time limit.

1. Introduction

1.1. Background

These days, main beneficiaries of companies are investors (shareholders, bondholders) , employees, customers, and business counterparts. In the recent years, the investors have been considered as major sources of capital for companies. Therefore, management relationship between companies and investors, called “Investor Relation” (IR), has become important in management. For example, the subject of IR is a company and the object of IR is an investor. Because IR is one of marketing activities for investors, it relates to corporate finance and communication about business results of companies and their future prospects. In addition, IR aims at improving company values and reducing any associated costs so that it is often concerned with making fund-raising smoothly and cost-effectively.

There have been two ways to raise long-term fund, one way to issue debt and the other to issue stock in general. Any money invested as stock in a company does not accompany with refunding obligation. Rather, it is owned permanently by the company because it has no maturity. Each investor (as a shareholder) has the right to participate in the company management and may get dividend only when the company makes profit. On the other hand, any money invested as debt in a company must be paid back by its maturity date, along with paying some interests regardless of whether the company makes profit or not. Accordingly, companies may prefer shareholders to bondholders because their dividend payment can be adjusted according to their business operation results. Although debt is more reliable than stock from profit point of view, many investors may still invest on stock, because it can provide the investors with the opportunity to get capital gain through stock price volatility. Eventually, shareholders can be considered as the major object of IR.

Shareholders focus on intrinsic values of companies. They may strictly require for companies to do profit sharing activities like dividend payment. If companies are not able to afford such requirement, then they may be faced with financial difficulty. Therefore, for proper evaluation and smooth fund-raising, companies need to notify shareholders about their future business plans and return on investment (ROI) through IR. The future business plan may be composed of various investments which require capital as its execution cost and insure certain expected profit. In addition, some investment must be executed only after some other investment is executed. For instance, a factory expansion plan can not precede the factory construction plan. In such notification, the most important part is concerned with how much the company is going to make return for the shareholders. Their profit is represented by dividend which is usually paid at the end of the company's fiscal year.

From now on, the term "investor" will be used in the same meaning as the term "shareholder".

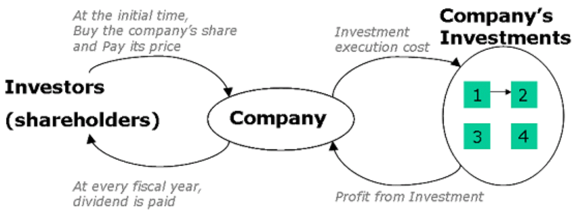

Figure 1-1. Cash flow structure

Figure 1-1 shows the cash flows between the investors and the company and the company's investment. At the initial time, the company issues stock to fund-raise its endowment from the investors. When each investment makes the cash inflow which is composed of amount of cost recovery and additional profit, the profit is accumulated until next fiscal year, and the amount of cost recovery can be used for another investment immediately. At a fiscal year, the accumulated profit is divided into two parts, one part for dividend and the other for reinvestment. The accumulated profit on each fiscal year is differently calculated according to the time schedule of investment. Generally, investors would like to choose today's $1 rather than tomorrow's $1. That is, each dividend as investor's return should be evaluated by criterion (including net present value) with time value considered.

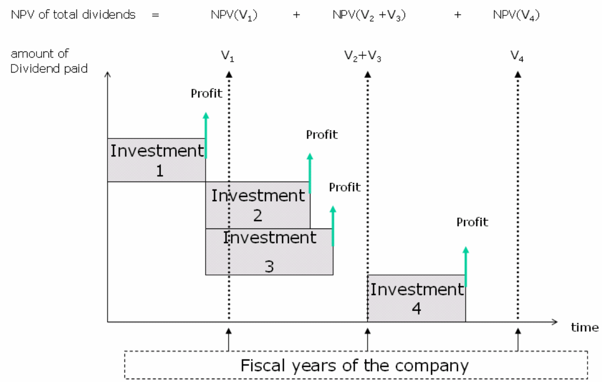

Figure 1-2. Dividend payment structure

Figure 1-2 shows the dividend payment structure for an investment schedule. As shown above, the company's value for the investors can be evaluated by the net present value (NPV) of total dividend. Therefore, this paper deals with the problem whose objective is to optimally schedule investments subject to a fixed deadline (the planning horizon for IR) so that NPV of total dividend is to be maximized subject to the precedence relations of investments and the capital limits with reinvestment allowed.

Based on the above description, the proposed problem can be considered as a kind of capital constrained project scheduling problem (CCPSP); that is, a kind of resource constrained project scheduling problem (RCPSP). Accordingly, the literature review for RCPSP is made in the next section.

1.2. Literature Review

Since the early work of Davis (1973) [16], numerous exact and heuristic algorithms have been developed for RCPSP (including Davis and Patterson, 1975 [17]; Patterson, 1984 [26]; Demeulemeester and Herroelen 1992 [5]). In contrast, a number of studies (Smith-Daniel; and Aquilano, 1987; Baroum and Patterson, 1993 [13]; Padman et al. 1994 [23] ; Padman and Smith-Daniels, 1993 [24]) have found that their heuristic methods developed for RCPSP, which have performed well with respect to duration objective functions, have generated solutions with lower NPV (on average) than their heuristics developed to maximize NPV for resource constrained projects.
A number of previous studies have developed heuristic procedures for scheduling the

RCPSP problem with an NPV objective (RCPSP-NPV). In the first work involving maximization of project NPV, Russell (1970) [27] have treated the unconstrained NPV project scheduling problem to be represented by flows in a network that might be solved as a series of transshipment problems, which illustrated through an example the problem structure and the derivation of dual prices representing the cost of delaying project activities. Grinold (1972) [6] has added a project deadline restriction to the problem and showed that scheduling a project with an NPV objective can be transformed into an equivalent linear program which has the structure of a weighted distribution problem. Russell (1986) [28] has developed heuristics for scheduling the RCPSP-NPV problem which is an extension of the network flow model of Russell (1970) [27]. He has developed several heuristic rules using dual prices from the unconstrained network flow model to establish scheduling priorities in a greedy single-iteration algorithm and tested the performance of their priority rules against some heuristics that were performed well with respect to the objective of minimizing makespan. Russell's study (1986)[28] indicates that for resource tightly-constrained projects, higher NPV schedules can be derived using a priority rule based on target schedule dates from the unconstrained network flow solution and the accompanying dual prices as a tie-breaker. In a subsequent study, Baroum and Patterson (1993) [13] have proposed a heuristic procedure that was based upon a cash flow weight (CFW) rule. They have tested single-pass procedures by using weights (being represented by sum of cumulative future cash flows in the project) and by prioritizing activities for scheduling by giving preference to those activities with the highest cash flow weight. After generating an initial schedule with the cash flow weight rule, multi-pass enhancements were then used to improve project NPV. Baroum and Patterson (1993) [13] have found out through extensive tests that the CFW priority scheduling rule outperformed the minimum slack scheduling rule in all the cases where progress payments were received throughout the project.

Padman et al. (1994) [23] and Padman and Smith-Danieh (1993) [24] have solved the RCPSP-NPV problem by modifying Russell's project scheduling approach in a number of ways. First, they have enhanced the greedy single-iteration scheduling algorithm by updating the unconstrained network model when activities were delayed beyond their optimally scheduled start times in the unconstrained NPV solution. This modification has exploited the capabilities of the dual simplex algorithm for the minimum cost network flow problems (Ali et al., 1989 [12]) that allowed the efficient reoptimization of a partially completed schedule. Second, new priority rules have been developed by use of the revised activity start times and tardiness penalties from the updated network model along with information on resource requirements, cash flows, and activity durations. Third, to estimate the reduction in the project NPV due to activity delays caused by resource constraints, Padman et al. [23] have used a different representation of an activity's tardiness penalty than those used by Russell [28] as discussed below. In an extensive testing on heuristic procedures with resource-constrained projects and cash flows, Padman et al. [23] have found out that their revised optimization-guided heuristics have derived significantly higher NPV schedules than the procedures of Russell (1986) [28] and several other heuristics including the minimum slack and CFW priority rules.

Since the general class of resource constrained scheduling problems is NP-complete (Garey and Johnson, 1979 [11]), as the RCPSP problem, the inherent intractability of the CCPSP problem requires development of heuristic procedures. There have been a few researches devoted to the development of methods that accounted for the special characteristics of the CCPSP problem. Doersch, and Patterson (1977) [1] have proposed the CCPSP problem with cash inflow reinvestment considered and Smith-Daniels, Padman and smith-daniels (1996) [2] have developed the heuristic algorithms about the CCPSP-NPV problem and tested efficiency of the algorithms. To the authors' best knowledge, however, there has not been any research yet to develop an exact algorithm of the CCPSP-NPV problem.

Mario Vanhoucke, Erik Demeulemeester and Willy Herroelen (2001) [3] have developed an exact algorithm of the RCPSP-NPV problem in branch-and-bound approach. Although the RCPSP-NPV problem is different from the CCPSP-NPV problem being considered in this paper such that it does not consider the reinvestment issue and has a different cash flow structure relevant to the objective function, the branch-and-bound schemas of the RCPSP-NPV problem can be adapted in this paper to develop an exact algorithm for the CCPSP-NPV problem.

1.3. Organization

The organization of this paper is briefed as follows. Section 2 gives the problem description. Section 3 analyzes the solution properties and makes some relevant comments, and proposes a branch-and-bound solution procedure for the problem. Section 4 shows numerical results in comparison with the commercial S/W CPLEX and also shows that a number of problem instances are optimally solved within given time limit. Section 5 makes overall conclusions.

2. Problem Description

This paper deals with an investment scheduling problem of maximizing net present value of dividend with reinvestment allowed, where each investment has certain capital requirement and generates deterministic profit. Such deterministic profit is calculated at completion of each investment and then allocated into two parts, including dividend and reinvestment, at each predetermined reinvestment time point. The objective is to make optimal scheduling of investments over a fixed planning horizon which maximizes total sum of the net present values of dividends subject to investment precedence relations and capital limits but with reinvestment allowed.

In the analysis, the proposed problem is transformed to a kind of parallel machine scheduling problem. Thus, the problem can be solved in the approach of scheduling theory. Accordingly, to help readers to make the overall comprehension, this is explained in a scheduling framework. That is, investments are considered as jobs for a parallel machine scheduling problem. In the same manner, the amount of capital required for each investment is considered as the number of machines required for each job. The detailed problem situation is given as follows;

- There are precedence requirements for jobs.
- Each job makes a positive profit (V_i) at its completion time (f_i).
- There are C_B machines at the initial time.
- Processing job i needs c_i machines.
- Additional machines (price=1) can be bought by using a part of the profit (βV_i) earned from job completions at given reinvestment points($T, 2T, .., mT=\delta$).
- All jobs must be scheduled subject to a deadline (δ ; Planning Horizon).
- Time schedule for jobs should be decided (decision variable: job schedule).
- Sum of the remaining profits ($(1-\beta)V_i$) with discount rate (a) is maximized as follows;

  Maximize $\Sigma TV(f_i)(1-\beta)V_i$,

  where $TV(f_i) = e^{-akT}$, if $(k-1)T<f_i\leq kT$.

As seen in the above objective function, it would be better to complete all jobs as fast as possible due to the discount factor.

2.1 Assumption

To derive a mathematical model for the proposed problem, the following assumptions are considered;

- Each job makes a positive profit (not considering any external economical factor such as inflation) at its completion time point.
- Each job has a known, deterministic and integer processing time.
- Reinvestment rate(β) and discount rate(a) are given.
- A part of profit(βV_i) earned from job completions can be reinvested at given reinvestment point.

2.2 Notation

For the mathematical model, the following notation is used;

- Sets

I set of jobs $I = \{0,\cdots,N+1\}$ *

D set of reinvestment points $D = \{T,2T,3T,\cdots,\delta=mT\}$, where δ is deadline.

L set of time $L = \{0,1,\cdots,\delta\}$

H set of precedence relations $(i,j)\in H$ $(i,j \in I)$

* 0 and $N+1$ denote dummy jobs, where 0 denotes the job which precedes all the rest of the jobs, indicating the start time of entire schedule, and $N+1$ denotes the job which follows all the rest of the jobs, indicating the completion time of an entire schedule.

- Constants

c_i required number of machines for job i $(i \in I)$

p_i processing time for job i $(i \in I)$

V_i profit earned from job i $(i \in I)$; $(V_i > 0)$

C_B initial number of machines

α discount rate

β reinvestment rate

- Decision Variables

f_i completion time of job i $(i \in I)$

- Variable sets

S_t set of jobs starting in the time interval $[0,t]$ $(t\in L)$

E_t set of jobs completing in the time interval $[0,t]$ $(t\in L)$

The variable sets depend on the decision variables. In other words, the set of jobs at a time point is determined by using job starting (or completion) time information to know whether a job starts (or completes) before the time point or not. S_t and E_t are used to identify started jobs and completed jobs at a time point, respectively.

2.3 Formulation

The proposed problem can be formulated in IP(integer programming) as in the problem, MP.

<u>Problem MP</u>

$$\text{Maximize} \quad \sum_{t\in D}\Big(\sum_{m=t-T+1}^{t}\sum_{i}(1-\beta)V_i X_{im}\Big)e^{-\alpha t} \qquad (1)$$

$$\sum_{t=0}^{\delta} X_{it} = 1 \qquad \forall i \in I \qquad (2)$$

$$-\sum_{t=0}^{\delta} tX_{jt} + \sum_{t=0}^{\delta} tX_{it} \le -p_j \qquad (i,j)\in H \qquad (3)$$

$$\sum_{t=0}^{\delta} tX_{n+1t} \le \delta\,,\quad X_{00} = 1 \qquad (4)$$

$$\sum_{i=1}^{n} c_i \sum_{m=t+1}^{t+p_i} X_{im} \le C_B + \sum_{\substack{m\in D\\ m\le t}}\Big(\sum_{c=m-T}^{m}\sum_{i}\beta V_i X_{ic}\Big) \forall t \in L \qquad (5)$$

$$X_{it} \in \{0,1\} \qquad (6)$$

The objective function (1) is to maximize sum of the remaining profits after profit reinvestment. Constraint (2) represents all jobs must be scheduled only once. Constraint (3) represents precedence relations between the associated jobs. Constraint (4) represents deadline limit with last dummy node and start time setting with initial dummy node, respectively. Constraint (5) ensures that the number of machines required for the jobs can not exceed the available number of machines at each time point. The available number of machines at a time point is the initial number of machines plus the number of machines to be obtained from reinvestment by the time point. The decision variable X_{it} has the following meaning.

$$X_{it} = \begin{cases} 1 & \text{if job } i \text{ completes at time } t \\ 0 & \text{otherwise.} \end{cases}$$

3. Solution Approach

The resource constrained scheduling problem with even one resource type, two jobs and precedence relations is NP-complete in strong sense, referring to Garey and Johnson [11]. Therefore, the proposed problem which has n jobs, one resource type and precedence relations is also NP-complete in strong sense, so that the branch-and-bound approach is used in this paper to derive the optimal solution procedure. The foundation of the branch-and-bound approach and the successful applications to some other similar problems (especially, resource constrained scheduling problem) are summarized in Mario Vanhoucke, Erik

Demeulemeester, Willy Herroelen [3], Arno Sprecher, Andreas Drexl [4], Erik Demeulemeester and Willy Herroelen [5].

3.1 Solution Domain Analysis

To search the optimal solution efficiently, the solution domain will be analyzed first.

Comment 1
In the proposed problem, initial time point, job completion time points and reinvestment points can be considered as "dispatching points" in optimal solution search.

Comment 2
At a dispatching point, some job, which is delayed in spite of being able to start with the available number of machines, may exist.

Comment 1 enables us to restrict the possible search domain for the optimal solution and Comment 2 shows that the time interval for any job not to be executed may exist in a feasible solution of the proposed problem.

3.2 Solution Bound Analysis

The branch-and-bound approach needs to get the effective solution bound as well as the branching strategy. This paper presents two properties to calculate the upper bound for the proposed problem.

Property 1
The proposed problem can be solved in polynomial time if the number of machines is unlimited and the associated objective function value denoted by UB_0 is an upper bound on the optimal objective value of the original problem.

Proof) The objective function of the proposed problem is to maximize total sum of the remaining profits with discount rate. As mentioned before, it would be better to complete all jobs as fast as possible, due to the discount factor. Therefore, in the situation with only the precedence relations considered, the objective value can be maximized by the forward algorithm which finds earliest completion time for each job. It is sure that the forward algorithm is solvable in polynomial time. Obviously, UB_0 is an upper bound on the optimal objective value of the proposed problem. □

The following notations will be used from now on. At a "dispatching point" m,

C_m = available number of machines
A_m = set of schedulable jobs
r_i = negative influence on the objective function when job i is not selected to start at m

Using those notations, UB_0 can be obtained by use of the forward algorithm to find the associated earliest completion times(EF_i) with each unscheduled job at dispatching point m under only given precedence relations like the following;

$$UB_0 = \sum_i TV(f_i)\cdot(1-\beta)V_i \ ,$$

$$where \ \ TV(f_i) \ = \ e^{-\alpha kT}, \ if\,(k-1)T < f_i \le kT$$

$$f_i = EF_i \ \text{ for } \forall i \notin S_m$$

This upper bound may be too large to use as a solution bound, because it does not consider any machine conflicts. To strengthen the upper bound, the available number of machines (C_m) can be considered for the schedulable jobs at dispatching point m so that ***Property 2*** is derived.

Property 2
An upper bound denoted by UB_1 ($\le UB_0$) can be obtained in pseudo-polynomial time at each "dispatching point".

Proof) Let's consider the following problem.

$$UB_1 = Max\ UB_0 - \sum_i r_i(1-X_i)$$

$$\sum_i c_i X_i \le C_m$$

$$where \ \ TV(f_i) \ = \ e^{-\alpha kT}, \ if\,(k-1)T < f_i \le kT$$

$$f_i = EF_i \ , \forall i \notin S_m$$

$$r_i = \{TV(f_i) - TV(f_i + \pi_i)\}\cdot(1-\beta)V_i \ , \forall i \in A_m$$

$$\pi_i = \underset{j \ne i}{Min}\{p_j, T([m/T]+1) - m\} \ \ \forall j \in A_m$$

π_i represents the next dispatching. r_i represents the decreasing amount of the objective value when job i delays until the next dispatching point because of limitation on the number of available machines. The above problem considers machine availability only at the current dispatching point. UB_1 is an upper bound on the optimal objective value of the original problem and is obviously smaller than UB_0. Also, the above problem is a 0-1 knapsack problem so that it can be solved in pseudo-polynomial time (Bertsimas, Dimitris, Tsitsiklis, John N.[9]) . □

3.3 Branch-and-Bound algorithm

In this section, a branch-and-bound procedure for the proposed problem is derived, based on the above analyses.

The basic idea of the algorithm is to use "dispatching points" as branch-nodes and a strengthened solution bound at each "dispatching point" as the upper-bound at the associated node, respectively. The proposed Branch-and-Bound algorithm is given as follows.

Step 1:

- Make all possible combinations (including empty case; by comment 2) of schedulable jobs (whose all predecessors have already finished) at the current node with the available number of machines considered. Each combination is assigned to a branched node (next level).

Step 2:

- For each branch-node, calculate the next dispatching point m and the number of machines that will be available at m and Upper Bound.

Step 3: Node selecting (depth-first)

- Select the node with the maximum Upper Bound from among all the nodes at the current level.

Step 4: Node fathoming

- Go to Step 5 if the current node is fathomed by the following criteria. Otherwise, go to Step 6.

The current node is fathomed if the maximum value mong the earliest completion times of jobs is larger than Deadline. The current node is fathomed if the best feasible solution value found so far is equal to or larger than UB.

Step 5: Backtracking

- Backtrack to the previous level of the branch-and-bound tree and go to Step 3 if the level is not zero. Otherwise, finish the procedure with the best feasible solution which is set as optimal solution.

Step 6:

- Go to Step 1 if the current node is not a leaf node. Otherwise (that is, leaf node), update the best feasible solution and go to Step 5.

3.4 Heuristic Procedures

In this section, a heuristic procedure is derived to find an initial good feasible solution for the proposed branch-and-bound algorithm. To reduce the number of branched nodes, the initial good feasible solution needs to be taken as early as possible. In this paper, a heuristic procedure with the following three kinds of priority rules is proposed.

Rule 1 → Larger weight (= profit) job starts earlier

Considering the time value of money, the profit generated earlier is preferred to the one later. If all the other conditions are same, the larger profit should be generated earlier than the smaller one.

Rule 2 → Smaller processing time job starts earlier (SPT rule)

If all the other conditions are same, the smaller processing time job can generate its profit earlier than the larger one.

Rule 3 → Larger weighted processing time job starts earlier (WSPT rule)

This priority rule is the hybrid rule of the above two rules.

Although each priority rule above has the reasons to get a good feasible solution, the reasons do not provide the optimality condition. Therefore, all three priority rules were used together to take the better feasible solution. Assuming that the objective values derived by using each priority rule are H(Rule 1), H(Rule 2) and H(Rule 3), respectively, the heuristic procedure is given as follows.

Step 0: Initialization

Set variable k=1.

Step 1:

If k>3, an initial feasible solution is found from Max {H(Rule 1), H(Rule 2), H(Rule 3)} and terminate the procedure. Otherwise, set time m=0 and go to Step 2.

Step 2:

Find the schedulable jobs ($\in A_m$) at time m and sequence them by priority rule. If there is no more schedulable job at time m, go to Step 4. Otherwise, go to Step 3.

Step 3: Move to the next dispatching point
According to the sequence, the schedulable jobs can be scheduled to start at time m until there is no machine conflict. If the machine conflict happens or all the schedulable jobs are scheduled without any machine conflict, go to Step 2.
Step 4:
Calculate H(Rule k) and then k = k + 1. Go to Step 1.

3.5 Overall Procedures

In this section, the overall procedure is presented by combining the branch-and-bound procedure and the heuristic procedure as shown below.

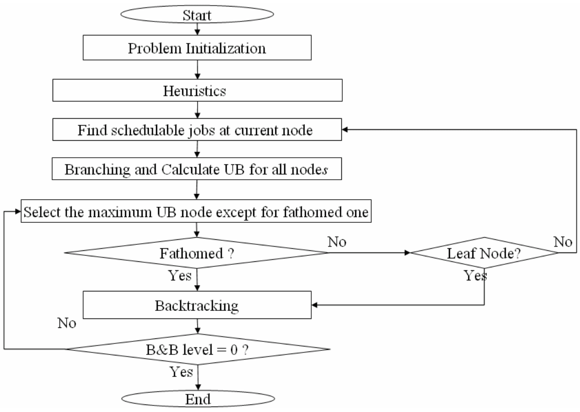

Figure 3-2. Overall Procedure

4. Computational Experiments

To evaluate the performance of the branch-and-bound algorithm presented in Section 3, computational experiments are carried out with some numerical instances. The algorithm has been coded in Visual C++ (Version 6.0) under Windows XP on a personal computer (Pentium 2.4 GHz processor).

Two kinds of experiments are tested with parameter settings as in Table 4-1.

Precedence relations	set precedence relations with 10% probability
Capital required for each investment	randomly selected from the interval [1, 50]
Profit from each investment	randomly selected from the interval [1, 15]
Investing duration	randomly selected from the interval [1, 15]

Table 4-1. Parameter settings used to generate the test instances

In the first experiment, to compare the efficiency of the proposed algorithm with that of CPLEX 8.1 (the commercial software package for mixed integer programming problems), ten problems are tested, each problem being represented by a set of the selected combinations of the number of investments(N), planning horizon(δ), reinvestment period (interval between two adjacent reinvestment points, T) and reinvestment rate (β) under given discount rate(α) and the initial capital(C_B) as shown in Table 4-2.

α = 1.5%, C_B = 50				Elapsed time (sec)		
N	δ	T	β	B&B (with UB)	B&B (without UB)	CPLEX
3	20	4	0.7	0.014	0.015	0.500
3	20	4	0.3	0.016	0.015	0.422
6	40	5	0.7	0.015	0.016	0.390
6	40	5	0.3	0.016	0.016	0.437
10	60	6	0.7	0.515	0.781	3.578
10	60	6	0.3	1.471	1.687	4.531
15	80	8	0.7	8.846	10.468	238.703
15	80	8	0.3	35.152	41.719	334.531
20	100	10	0.7	15.780	21.406	11139.406
20	100	10	0.3	77.307	87.360	17020.109
		Ave.		13.9138	16.3483	2874.2607

Table 4-2. Performance of the proposed branch-and-bound comparison with CPLEX

Table 4-2 represents the optimal solution search time for the branch-and-bound algorithm with upper bound strengthening, without strengthening, and CPLEX, respectively. As shown in Table 4-2, the average computation time of the branch-and-bound algorithm is much larger than the average computation time of CPLEX. In addition, the upper bound strengthening process can reduce computation time of the proposed problem by about 15%.

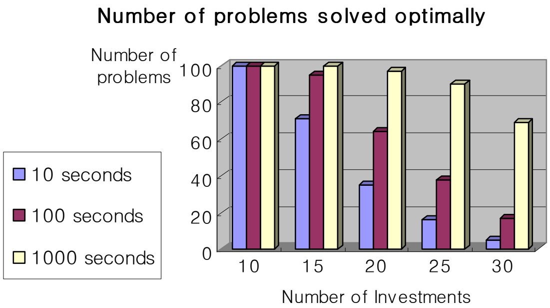

Figure 4-1. Effect of the Number of Investments on the Number of Problems Solved Optimally

In the second experiment, to test how large size of the proposed problem can be solved optimally in a reasonable time limit, one hundred problems are tested for a different number of investments and the associated CPU time limit with the test instance where
$\delta = 10 * N$, $T = 5$, $\beta = 0.6$, $\alpha = 0.015$, $C_B = 60$.

Figure 4-1 displays the number of problems (out of 100) solved optimally for a different number of investments and the associated CPU time limit. Clearly, the number of investments has a significant effect on the average CPU time and on the number of problems solved optimally. Almost all problems with 10 investments can be solved optimally within 10 seconds of CPU time. For problems containing 20 investments, about 33% of the problems can be solved optimally when the allowed CPU time is 10 seconds, whereas about 97% of the problems can be optimally solved when the time limit is 1000 seconds. For problems with 30 investments, only about 7% of the problems can be solved within 10 seconds of CPU time, whereas about 65% of the problems can be solved to optimally when the allowed CPU time is 1000 seconds.

5. Concluding Remarks

This paper deals with an investment scheduling problem subject to capital limit to maximize net present value of dividend over a given planning horizon which is commonly required for investor relation.

The investment scheduling problem can be transformed to a kind of parallel machine scheduling problem, and formulated as an integer programming with the objective of maximizing total sum of remaining profits and the constraints concerned with fixed deadline, precedence relations, and number of machines with reinvestment allowed to buy more machines.

As a solution approach, a depth-first branch-and-bound algorithm is derived in this paper. To solve the proposed problem more efficiently, some solution properties are characterized to derive solution bounds. Based on the properties, a heuristic algorithm is proposed to find an initial good feasible solution. In order to evaluate effectiveness and efficiency of the proposed algorithm, computational experiments are performed with some numerical instances. The experiment results show that the proposed algorithm solves the instances more efficiently than CPLEX 8.1, and the solution bounds are very helpful in the associated solution search. Moreover, the results indicate that the branch-and-bound algorithm is able to optimally solve the problem instances of up to 30 investments in a reasonable time limit.

As a further study, an extended problem with the reinvestment rate to be determined at each reinvestment point as a decision variable may be interesting. Considering money lending activity in the problem may also be interesting. Moreover, the issue of integrating investment selection and scheduling may also be interesting. In practical situations, each profit from the associated investment may depend on investment timing, so that it would be better to treat in stochastic approaches. In other words, it would be better to consider such profit as random variable in further study.

References

[1] Doersch, R.H. and J.H. Patterson, “Scheduling a Project to Maximize its Present Value: A

Zero-One Programming Approach, ” Management Science, Vol. 23, No. 8 (April 1977), pp. 882-889.
[2] D.E. Smith-Daniels, R. Padman, V.L. smith-daniels, “ Heuristic scheduling of capital constrained projects ” Journal of Operations Management, Vol. 14, pp.241-254, 1996.
[3] Mario Vanhoucke, Erik Demeulemeester, Willy Herroelen, “ On maximizing the net present value of a project under renewable resource constraints ” Management Science, Vol. 47 No.8, pp. 1113-1121, 2001.
[4] Arno Sprecher, Andreas Drexl, “Multi-mode resource-constrained project scheduling by a simple, general and powerful sequencing algorithm ” European Journal of Operational Research, Vol. 107, pp. 431-450, 1998.
[5] Erik Demeulemeester, Willy Herroelen, “ A branch and bound procedure for the multiple resource constrained project scheduling problem” Management Science, Vol. 38 No.12, pp. 1803-1818, 1992.
[6] Grinold, R. C. 1972. “The payment scheduling problem”. Naval Res. Logistics Quart. 19 123-136.
[7] G. Ulusoy, F. Sivrikaya-Serifoglu, S. Sahin, “ Four Payment Models for the Multi-Mode Resource Constrained Project Scheduling Problem with Discounted Cash Flows” Annals of Operational Research, Vol. 102, pp. 237-261, 2001.
[8] L. Ozdamar, H. Dundar, “A flexible heuristic for a multi-mode capital constrained project scheduling problem with probabilistic cash inflows ” , Computers & operations research, v.24 no.12, 1997, pp.1187-1200
[9] Bertsimas, Dimitris, Tsitsiklis, John N. “Introduction to Linear Optimization”, Athena scientific, 1997.
[10] Smith-Daniels, D.E., and Smith-Daniels, V.L., "Maximizing the net present value of a project subject to material and capital constraints", Journal of Operations Management 7/1 (1987) 33-45.
[11] Michael R. Garey and David S. Johnson, “Computer and intractability; A guide to the theory of NP-completeness”, Bell Laboratories, Murray Hill, New Jersey,1979
[12] Ali, A.I., R. Padman and H. Thiagarajan, 1989. “Dual algorithms for pure network problems”, Operations Research, vol. 37, no.1, pp. 159-171.
[13] Baroum, S. and J.H. Patterson, 1993. A comparative evaluation of cash flow weight heuristics for maximizing the net present value of a project, Working paper, Indiana University.
[14] Bey, R.B., R.H. Doersch and J.H. Patterson, 1981. “ The net present value criterion: Its impact on project scheduling ” , Project Management Quarterly, vol. 12. no. 2, June, pp.35-42.
[15] Bowen. H.K. K.B. Clark. CA. Holloway and SC. Wheelwright, 1994. The Perpetual Enterprise Machine, Oxford University Press, New York.
[16] Davis, E.W. 1973. “Project scheduling under resource constraints - Historical review and categorization of procedures ” , ABE Transactions, vol. 5, no. 4, pp. 297-313.
[17] Davis, E.W. and 3.H. Patterson, 1975. “ A comparison of heuristic and optimum solutions in resource-constrained project scheduling”, Management Science, vol. 21, no. 8. April, pp-944-955.
[18] Duncan, D.B.. 1955. “ Multiple range and multiple F tests”, Biometrics, vol. 11. pp. 1-42.
[19] Elmaghraby. SE. and W. Herroelen, 1990. “The scheduling of activities to maximize the net present value of projects ” , European Journal of Operational Research, vol. 49, pp. 35-49.
[20] G. Ulusoy , S. Cebelli, 2000. “An equitable approach to the payment scheduling problem in project management ” , European Journal of Operational Research, vol. 127 pp. 262-278.
[21] Kurtulus, I.S. and E.W. Davis, 1977. “Multi-project scheduling: Categorization of heuristic rules performance”, Management Science, vol. 23, no. 8, pp~ 882-889.
[22] Orlin, J.B., 1984. Genuinely polynomial simplex and non-simplex algorithms for the minimum cost network flow problem, Sloan working paper No. 1615-84. Sloan School of Management, Massachusetts Institute of Technology, December.
[23] Padman, R.. DE. Smith-Daniels and V.L. Smith-Daniels. 1994. Heuristic scheduling of resource-constrained projects with cash flows: An optimization-guided approach, Working paper, School of Urban and Public Affairs, Carnegie-Mellon University.
[24] Padman. R. and DE. Smith-Daniels, 1993. “ Early-tardy cost trade-off in resource constrained projects with cash flows: An optimization-guided heuristic approach ” , European Journal of Operational Research, vol. 64. pp. 295-311.
[25] Pascoe. T., 1965. An experimental comparison of heuristic methods for allocating resources, Unpublished Ph.D. Paper, Cambridge University.
[26] Patterson. J.H., 1984. “ A comparison of

exact approaches for solving the multiple constrained resource project scheduling problem", Management Science, vol. 30. no. 7, pp. 854-867.

[27] Russell, A.H., 1970. " Cash flows in networks", Management Science, vol. 16, no. 5, pp. 357-373.

[28] Russell, R.A., 1986. " A comparison of heuristics for scheduling projects with cash flows and resource restrictions", Management Science, vol. 32, no. 10. pp. 129l- 1300.

[29] Smith-Daniels, D.E. and NJ. Aquilano, 1987. " Using a late-start resource-constrained project schedule to improve project net present value", Decision Sciences, vol. 18, pp. 369-387.

[30] Stewart, T.A., 1995b. "The corporate jungle spawns a new species: The project manager", Fortune, vol. 131. no. 5, pp. 179- 180.

[31] Sung, C. S. and S. K. Lim, "A Project Activity Scheduling Problem with Net Present Value Measure", International Journal of Production Economics, Vol. 37, P. 177 - 187, 1994

[32] Sung, C. S. and S. K. Lim, "A Scheduling Procedure for a General Class of Resource-Constrained Projects", Computers and Industrial Engineering, Vol. 32, No. 1, P. 9 - 17, 1997