Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Discrete Mathematics 36 (2010) 407–414

Electronic Notes in DISCRETE MATHEMATICS

www.elsevier.com/locate/endm

Large-scale multi-period precedence constrained knapsack problem: A mining application [1]

Eduardo Moreno [a,2] Daniel Espinoza [b,3] Marcos Goycoolea [c,4]

[a] *Faculty of Engineering and Science, Universidad Adolfo Ibañez, Santiago, Chile*

[b] *Department of Industrial Engineering, Universidad de Chile, Santiago, Chile*

[c] *School of Business, Universidad Adolfo Ibañez, Santiago, Chile*

Abstract

We study an extension of the precedence constrained knapsack problem where the knapsack can be filled in multiple periods. This problem is known in the mining industry as the open-pit mine production scheduling problem. We present a new algorithm for solving the LP relaxation of this problem and an LP-based heuristic to obtain feasible solutions. Computational experiments show that we can solve real mining instances with millions of items in minutes, obtaining solutions within 6% of optimality.

Keywords: Precedence Constrained Knapsack Problem, Open-pit Mining, Integer Programming

[1] Authors acknowledge the support of Chilean Government through Basal project CMM and Millennium Institute on Complex Systems, both hosted at Universidad de Chile.

[2] Email: `eduardo.moreno@uai.cl`

[3] Email: `daespino@dii.uchile.cl`

[4] Email: `marcos.goycoolea@uai.cl`

1571-0653/$ – see front matter
doi:10.1016/j.endm.2010.05.052

1 Introduction

The Precedence-Constrained Knapsack Problem (PCKP) [8] is a generalization of the classic knapsack problem. Each item i has a profit p_i, a weight q_i and a set of precedences $\mathcal{P}_i$. The problem consists in selecting a subset of items in such a way as to maximize profits, and so that (1) the added weight of the selected items is no greater than c, and (2) so that the precedence constraints are satisfied; that is, in such a way that if item i is selected, then so must every item in $\mathcal{P}_i$. Many exact and heuristic methods have been proposed for solving this problem (see for example, [12,14]), including a number of techniques based on integer programming (see [4,10,13]). In this paper, we study a generalization of this problem called the Multi-period Precedence-Constrained Knapsack Problem (MPPCKP). In this problem, the knapsack can be filled in different periods, where every item has a (usually decreasing) different profit on each period. Our motivation for studying this problem comes from a recognized problem in the mining industry known as *open pit mine production scheduling*. In this application, items are blocks that should be scheduled for extraction in time. The knapsack constraint represents production capacity constraints, and precedence restrictions represent operational geometrical considerations (blocks can be extracted only if all blocks above and within a prescribed cone have been extracted before in time). A first optimization model for this problem was proposed by Johnson [9], and several authors have developed linear and integer programming techniques for solving this problem (see for example [6,11,3,2]).

Formally, the precedence relationships will be represented in terms of a digraph $G = (\mathcal{V}, \mathcal{A})$ where $(u, v) \in \mathcal{A}$ represents that item u must be included before or at the same period that item v. We assume that digraph G only contains immediate precedence relationships. That is, if $u, v, w \in \mathcal{V}$ are such that $(u, v) \in \mathcal{A}$ and $(v, w) \in \mathcal{A}$ then $(u, w) \notin \mathcal{A}$.

Let T be the number of periods, c_t represent the capacity available at period t, q_v the weight of item v and $p_v(t)$ the profit of item v if it is included at period t. Define binary variables x_v^t indicating if item v is included in the knapsack by time t. A formulation to solve this problem is presented in Figure 1.

In our context, we assume that the profit $p_v(t)$ of an item v in time t is proportional to the profit of the previous period. Hence, we can assume that $p_v(t) = \alpha^{t-1} p_v$. This is a natural assumption in the context of net-present-value optimization where profits are of the form $\left(\frac{1}{1+r}\right)^t p_v$, for an undiscounted profit p_v, a discount rate r and discrete time periods $t = 1, \ldots, T$.

$$
\begin{aligned}
&\max \sum_{v \in \mathcal{V}} \sum_{t=1}^{T} p_v(t) \cdot (x_v^t - x_v^{t-1}) \\
&\text{s.t.} \\
&\quad \sum_{v \in \mathcal{V}} q_v (x_v^t - x_v^{t-1}) \le c_t && \forall t \in 1 \ldots T \\
&\quad x_v^t \le x_u^t && \forall (u,v) \in \mathcal{A},\ \forall t \in 1 \ldots T \\
&\quad x_v^t \le x_v^{t+1} && \forall v \in \mathcal{V},\ \forall t \in 1 \ldots T-1 \\
&\quad x_v^t \in \{0,1\} && \forall v \in \mathcal{V},\ \forall t \in 1 \ldots T \\
&\quad x_v^0 = 0 && \forall v \in \mathcal{V}
\end{aligned}
$$

Fig. 1. The MPPCKP Formulation

2 Computing feasible solutions: Toposort Heuristics

In this section we present a family of heuristics with which to obtain feasible solutions for MPPCKP. Our main result in this section is heuristic which takes as input the LP-relaxation solution of MPPCKP and yields (in our computational tests) very good feasible solutions of MPPCKP.

Recall that $G = (\mathcal{V}, \mathcal{A})$ defines an acyclic directed graph. It is known that G admits a *topological ordering* of its nodes; that is, an ordering $\{v_1, v_2, \ldots,\}$ such that if $(v_i, v_j) \in \mathcal{A}$ then $i < j$. Given a weight $w(v)$ for each item $v \in \mathcal{V}$, we say that ordering $\{v_1, v_2, \ldots\}$ is topologically sorted with respect to w if it is a topological ordering, and in addition, $(|i-j| = 1) \wedge (v_i \nrightarrow v_j) \wedge (v_i \nleftarrow v_j) \wedge (w_i < w_j)$ implies $(i < j)$. Given a topological ordering of $\mathcal{V}$, it is easy to obtain a feasible solution for MPPCKP. In fact, it is simply a matter of sequentially including all items in the order prescribed by the topological ordering, scheduling each item as early as possible, and updating the available capacity of the time period in which items are schedulued. Because items are scheduled in topological order, precedences will not be an issue. Hence, it is simply a matter of determining the earliest time period in which there is capacity available in order to schedule each item. We refer to this algorithm as *TopoSort* heuristic.

By defining weights in different ways this scheme can lead to a whole class of TopoSort heuristics. For example, a Greedy TopoSort heuristic is obtained by assigning as weight the profit of each item ($w(v) = p_v$). It is interesting to note that this scheme generalizes the heuristic of Gershon [7], commonly used in the context of open-pit mining, in which the weight of each item is defined as the sum of the profit of all items in the inverse-precedence set (that is, for a given item i, the set of all items j for which i is a precedence).

We propose a weight function based on the LP relaxation $\bar{x}$ of MPPCKP. For each time period t and each item v define $y_v^t = \bar{x}_v^t - \bar{x}_v^{t-1}$. Additionally, define $y_v^{T+1} = 1 - x_v^T$. Observe that $\sum_{t=1}^{T+1} y_v^t = 1$ for each v. Thus, for $t < T$ we can imagine that y_v^t represents the probability that item v is included in period t, and that y_v^{T+1} represents the probability that item v is not included in the knapsack in any period. This suggests defining $w(v)$ as the expected value $w(v) = \sum_{t=1}^{T+1} t y_v^t$. Note that if $(u, v) \in \mathcal{A}$, then $w(v) \leq w(u)$. Moreover, if $\bar{x}$ is an optimal (integer) solution of MPPCKP, then $w(v)$ is either equal to the time of inclusion of item v, or equal to $T + 1$ if v is not included. Hence, using Toposort with this weight function will lead to an optimal solution. We call this variant the *Expected-Time TopoSort* heuristic.

3 Solving the LP-Relaxation of MPPCKP

In order to use the Expected TopoSort heuristic or to solve the IP formulation of MPPCKP directly it is necessary to first solve the LP relaxation. However, as we will see in the computational results, this can be very difficult for large problem instances. In this section we describe a new algorithm, which we call the *Critical Multiplier Algorithm*, for solving the LP relaxation of MPPCKP. The algorithm is based on two observations. The first is that in order to solve a (single-time period) PCKP instance, it suffices to solve two single-time period maximum closure problems and to take a convex combination of the solutions. The second observation is that in order to solve a (multiple-time period) instance of MPPCKP, it suffices to solve a sequence of single-time period problems and put together the solutions in a correct way.

3.1 The single time period case.

Define the linear relaxation of the PCKP with capacity κ as following:

$$
\begin{aligned}
CP(\kappa) = \max\ & px \\
\text{st}\quad & qx \leq \kappa \\
& x_i \leq x_j \qquad \forall (i,j) \in \mathcal{A} \\
& 0 \leq x_i \leq 1 \quad \forall i \in \mathcal{V}
\end{aligned}
$$

where we assume that $q \in \mathbb{R}_+^{|\mathcal{V}|}$ and $\kappa \in \mathbb{R}_+$. In this section we are concerned with efficiently solving a problem of the form $CP(\kappa)$. Observe that this corresponds to solving the linear relaxation of PCKP. Let us define the following problem:

$$\begin{aligned} UP(\lambda) = \max \ & (p - \lambda q)x \\ \text{st} \quad & x_i \leq x_j \qquad \forall (i,j) \in \mathcal{A} \\ & 0 \leq x_i \leq 1 \quad \forall i \in \mathcal{V} \end{aligned}$$

Note that solutions to $UP(\lambda)$ are integral by total-unimodularity of the constraint matrix. Consider two feasible solutions x, y of $UP(\lambda)$. We say that x *dominates* y (and write $y \prec x$) if $x \neq y$ and $y \leq x$. It is easy to see that there exists a maximal non-dominated optimal solution of $UP(\lambda)$. We henceforth denote this unique solution $x(\lambda)$.

It is also known that for $\mu^1, \mu^2 \in \mathbb{R}$ and the optimal non-dominated solutions $x(\mu^1), x(\mu^2)$ of $UP(\mu^1)$ and $UP(\mu^2)$ respectively, if $\mu^2 > \mu^1 \geq 0$ then $x(\mu^2) \leq x(\mu^1)$. Due to this property, we say that λ is a *critical multiplier of* UP if $x(\lambda + \varepsilon) \prec x(\lambda)$ for all $\varepsilon > 0$. Observe that if μ and ν are distinct critical multipliers of UP, by definition, either $x(\mu) \prec x(\nu)$ or $x(\nu) \prec x(\mu)$. This means there is only a finite set of critical multipliers. Let $\Lambda = \{\lambda^1, \lambda^2, \ldots, \lambda^m\}$ represents the set of all critical multipliers, sorted in decreasing order. Note that if $\mu^2 > \mu^1$ then $qx(\mu^2) \leq qx(\mu^1)$. Hence, for $\kappa > 0$, we can define $\lambda^u = \max\{\lambda \in \Lambda : qx(\lambda) \geq \kappa\}$ and $\lambda^l = \min\{\lambda \in \Lambda : qx(\lambda) \leq \kappa\}$. If $\lambda^u = \lambda^l$ then $x(\lambda^u) = x(\lambda^l)$ is an optimal solution of $CP(\kappa)$. On the contrary, if $\lambda^u < \lambda^l$, then the solution of $CP(\kappa)$ is a convex combination of $x(\lambda^u)$ and $x(\lambda^l)$.

Theorem 3.1 *Assume $\lambda^u < \lambda^l$, and let $b^u = qx(\lambda^u), b^l = qx(\lambda^l)$. Define $\alpha = \frac{b^u - \kappa}{b^u - b^l}$. Then $\bar{x} = \alpha x(\lambda^l) + (1-\alpha)x(\lambda^u)$ is optimal for $CP(\kappa)$.*

Proof. (Sketch) First, note that $\bar{x}$ is feasible of $CP(\kappa)$. In fact, the precedence constraints hold by convexity, and the knapsack condition holds since $q\bar{x} = \alpha qx(\lambda^l) + (1-\alpha)qx(\lambda^u) = \alpha b^l + (1-\alpha)b^u = \kappa$. Secondly, it is possible to construct, from the solution of the dual of $UP(\lambda^u)$, an optimal solution of the dual of $CP(\kappa)$ with the same objective value, concluding that $\bar{x}$ is optimal for $CP(\kappa)$. □

3.2 Extension to the multi-time period case

Let $U_t = \sum_{k=1}^{t} c_k$ be the accumulated capacity at time t. We begin by noting that an optimal solution of the linear relaxation of MPPCKP is also an optimal solution of the following problem:

$$
\begin{aligned}
MP = \max \ & \textstyle\sum_{t=1}^{T} p(t) \cdot (x^t - x^{t-1}) & & \\
\text{st} \quad & x_i^t \leq x_j^t & & \forall t = 1, \ldots, T \quad \forall (i,j) \in \mathcal{A} \\
& x^t \leq x^{t+1} & & \forall t = 1, \ldots, T-1 \\
& q \cdot x^t \leq U_t & & \forall t = 1, \ldots, T \\
& 0 \leq x^t \leq 1 & & \forall t = 1, \ldots, T \\
& x^0 = 0 & &
\end{aligned}
$$

The key property of this formulation is that we can solve each time period separately, and to merge all solutions to construct an optimal solution of MP.

Theorem 3.2 *Let $\bar{x}^t$ be an optimal solution of CP(U_t) for all $t = 1 \ldots T$. Then, vector $\bar{x} = (\bar{x}^1, \bar{x}^2, \ldots, \bar{x}^T)$ is optimal for MP, and therefore is optimal for the linear relaxation of MPPCKP.*

Proof. (Sketch) First, observe that if $U_t < U_{t+1}$ then $\bar{x}^t \leq \bar{x}^{t+1}$. Therefore, $\bar{x}$ is a feasible solution for MP. Second, it is possible to prove that if $\bar{y}$ is a feasible solution for MP, then the vector of variables corresponding to time t is feasible for $CP(U_t)$, and therefore its objective value is at most the corresponding objective value of $\bar{x}$. This proves that $\bar{x}$ is optimal for the LP relaxation of MPPCKP. □

This naturally leads to an algorithm to construct the linear relaxation of the MPPCKP. We call this algorithm the CRITICAL MULTIPLIER ALGORITHM (see Algorithm 1).

Algorithm 1. CRITICAL MULTIPLIER ALGORITHM

(i) Define $U_t = \sum_{k=1}^{t} c_k$ for $t = 1, \ldots, T$.

(ii) Compute all of the critical multipliers λ^i of $UP(\lambda)$ and the corresponding solutions $x(\lambda^i)$. This can be done using sensitivity analysis or binary search.

(iii) Using the critical multipliers computed in the previous step, obtain the optimal solution $\bar{x}^t$ of problem $CP(U_t)$ for each $t = 1, \ldots, T$, as indicated by Theorem 3.1.

(iv) Construct $\bar{x} = (\bar{x}^1, \bar{x}^2, \ldots, \bar{x}^T)$. By Theorem 3.2, $\bar{x}$ is the optimal solution of the LP relaxation of MPPCKP, and its objective value is $\sum_{t=1}^{T} \gamma_t p \bar{x}^t$, where $\gamma_T = (\frac{1}{1+r})^T$ and $\gamma_t = (1 - \frac{1}{1+r})(\frac{1}{1+r})^t$.

4 Computational Results

Our computational tests have two goals in mind. Our first goal is to compare the performance of the Critical Multiplier algorithm with CPLEX LP algorithms. Our second goal is to assess the quality of the solutions obtained by using our proposed heuristics, and the time required to obtain them.

We denote the Greedy, Gershon and Expected TopoSort heuristics as **GrTS**, **GeTS**, and **ExTS**, respectively. Additionally, we denote by **CPXbest** the best time obtained by CPLEX 11 to solve the LP relaxation, among the different LP solvers included in this software. We refer to the Critical Multiplier algorithm as **CMA**. When comparing objective function values, we always present numbers divided by the upper bound obtained with **CMA**. This allows us to assess the proximity of solutions to the optimal value.

The dataset to evaluate these algorithms is composed of four mines. One is a ficticious mine called Marvin included in Whittle's software, and the remaining are real mines located in America, Asia and Chile. These four mines contain 53'668, 19'320, 772'800 and 4'320'480 blocks respectively. Precedence constraints are constructed using precedence cones of 45 degrees. All four mines consider a time horizon of 15 years and a discount rate of 10%. Before solving these problems we first apply a common pre-processing scheme [5] reducing the number of variables into problems with 119'262, 96'675, 1'333'245 and 52'400'325 variables, respectively.

The following table contains the running times of **CPXbest**, **CMA** and the running times and objective values of the Topological Sorting heuristics on each of our data set instances. As expected, CPLEX is unable to solve the LP relaxation of large instances in a reasonable time, but **CMA** can solve them to optimality in minutes. Additionally, it can be seen that values obtained by **GrTS** and **GeTS** are very poor for some of the instances, whereas the values obtained by **ExTS** are very good (all within 6% of optimality). Moreover, using these solutions as a starting point for a local search improving algorithm (see [1]) it is possible to obtain solutions at less than 1% of optimality with a few hours of extra computation.

Instance	Time **CMA**	Time **CPXbest**	Time **GrTS**	ObjVal **GrTS**	Time **GeTS**	ObjVal **GeTS**	Time **ExTS**	Objval **ExTS**
Marvin	12 s	1h 3m	< 1s	0.856	25s	0.867	< 1s	0.957
AmericaMine	4 s	19m 26s	< 1s	0.819	11s	0.905	< 1s	0.940
AsiaMine	2m 36s	10d+	< 1s	0.750	3h 33m	0.861	< 1s	0.986
Andina	1h 44m	N/A	< 1s	0.486	4d 17h	0.524	< 1s	0.977

References

[1] Amaya, J., D. Espinoza, M. Goycoolea, E. Moreno, T. Prevost and E. Rubio, *A scalable approach to optimal block scheduling*, in: *Proceedings of APCOM 2009* (2009), pp. 567–575.

[2] Bienstock, D. and M. Zuckerberg, *A new algorithm for precedence constrained production scheduling*, Optimization Online (2009).

[3] Boland, N., C. Fricke and G.Froyland, *A strengthened formulation for the open pit mine production scheduling problem*, Optimization Online (2007).

[4] Boyd, E. A., *Polyhedral results for the precedence-constrained knapsack problem*, Discrete Applied Mathematics **41** (1993), pp. 185 – 201.

[5] Caccetta, L. and S. Hill, *An application of branch and cut to open pit mine scheduling*, Journal of global optimization **27** (2003), pp. 349–365.

[6] Dagdelen, K. and T. Johnson, *Optimum open pit mine production scheduling by lagrangian parameterization*, Proceedings of 19th APCOM Symposium (1986).

[7] Gershon, M., *Heuristic approaches for mine planning and production scheduling*, Int. Journal of Mining and Geological Engineering **5** (1987), pp. 1–13.

[8] Ibarra, O. H. and C. E. Kim, *Approximation algorithms for certain scheduling problems*, Mathematics of Operations Research **4** (1978), pp. 197–204.

[9] Johnson, T., *Optimum open-pit mine production scheduling*, in: A. Weiss, editor, *A decade of digital computing in the mining industry*, AIME, New York, 1969 .

[10] Park, K. and S. Park, *Lifting cover inequalities for the precedence-constrained knapsack problem*, Discrete Applied Mathematics **72** (1997), pp. 219 – 241.

[11] Ramazan, S. and R. Dimitrakopolous, *Recent applications of operations research and efficient mip formulations in open pit mining*, Society for mining, metallurgy and exploration meeting transactions **316** (2004), pp. 73–78.

[12] Samphaiboon, N. and T. Yamada, *Heuristic and exact algorithms for the precedence-constrained knapsack problem*, Journal of optimization theory and applications **105** (2000), pp. 659–676.

[13] van de Leensel, R., C. van Hoesel and J. van de Klundert, *Lifting valid inequalities for the precedence constrained knapsack problem*, Mathematical Programming **86** (1999), pp. 161 – 185.

[14] Wilbaut, C., S. Hanafi and S. Salhi, *A survey of effective heuristics and their application to a variety of knapsack problems*, IMA J Management Math **19** (2008), pp. 227–244.