CP-Based Local Search for Minimum Cost Linear Extension

T.O. Bonates, L.F.S. Oliveira, L.V. Oliveira, and P.C.L. Silva

UFERSA

Abstract. Breve descrição do modelo CP para busca local. Pode ser usado para CPAIOR 2013, ou apenas na nova versão do artigo completo.

1 Introduction

Let E be a n-set and let us write $E = \{x_1, \dots, x_n\}$. A partially ordered set (or simply, a *poset*) defined on E is a pair $\mathbf{P} = (E, \leq)$, where $\leq$ is a binary relation on E that is transitive, reflexive and antisymmetric, *i.e.*, $\leq$ is a binary relation satisfying, respectively:

(i) $x_i \leq x_j, \;\; x_j \leq x_k \;\Rightarrow\; x_i \leq x_k, \;\; \forall\, x_i, x_j, x_k \in E$;
(ii) $x_j \leq x_i, \;\; \forall\, x_i \in E$;
(iii) $x_i \leq x_j, \;\; x_j \leq x_i \;\Rightarrow\; x_i = x_j, \;\; \forall\, x_i, x_j \in E$.

In a partial order $\mathbf{P} = (E, \leq)$ there might exist pairs of elements $x_i, x_j \in E$, with $x_i \neq x_j$, such that we have neither $x_i \leq x_j$ nor $x_j \leq x_i$ in $\mathbf{P}$. Two elements x_i, x_j of a poset $\mathbf{P}$ are said to be *comparable* if $x_i \leq x_j$ or $x_j \leq x_i$ in $\mathbf{P}$; otherwise, they are said to be *incomparable*, which is denoted by $x_i \sim x_j$. We say that x_j *covers* x_i in $\mathbf{P}$ if $x_i < x_j$ and if there is no element x_k in $\mathbf{P}$ satisfying $x_i < x_k < x_j$.

A linear extension of a poset $\mathbf{P} = (E, \leq)$ is a permutation $\mathbf{L}$ of the elements of E such that x_i comes before x_j in $\mathbf{L}$ whenever $x_i < x_j$, for all comparable pairs $x_i, x_j \in E$. It is a well-known fact that every poset admits at least one linear extension (see, *e.g.*, [**?**]). Let us write a linear extension $\mathbf{L}$ of poset $\mathbf{P}$ as

$$\mathbf{L} = (x_{i_1}, x_{i_2}, \dots, x_{i_n}) \in E^n,$$

with $x_{i_1} <_{\mathbf{L}} x_{i_2} <_{\mathbf{L}} \dots <_{\mathbf{L}} x_{i_n}$. It is typical for a poset to admit more than one linear extension. Thus, it is natural to consider an optimization problem that asks for the best linear extension of a poset $\mathbf{P}$, according to some criterion.

Let $c : E \times E \to \mathbb{R}_+$ be a cost function that associates a non-negative real value with every pair $x_i, x_j \in \mathbf{P}$, while satisfying the following conditions:

$$\begin{cases} c(x_i, x_j) = 0, & \text{if } x_i \leq x_j; \\ 0 < c(x_i, x_j) < +\infty, & \text{if } x_i \sim x_j; \\ c(x_i, x_j) = +\infty, & \text{if } x_i > x_j. \end{cases} \tag{1}$$

We will write the total cost of a linear extension $\mathbf{L} = (x_{i_1} x_{i_2} \ldots x_{i_n})$ as

$$c(\mathbf{L}) = \sum_{j=1}^{n-1} c(x_{i_j}, x_{i_{j+1}}) \tag{2}$$

Given a poset $\mathbf{P}$ and a cost function c as described above, the Minimum Cost Linear Extension problem (thereby referred to as MCLE) asks for a linear extension $\mathbf{L}$ of $\mathbf{P}$ which minimizes (2). In other words, we are interested in finding $\mathbf{L}^*$ satisfying

$$\mathbf{L}^* \in \arg\min\{c(\mathbf{L}) : \mathbf{L} \text{ is a linear extension of } \mathbf{P}\}.$$

The MCLE was originally defined in [**?**] and further studied in [**?**]. In [**?**] the authors showed that the MCLE generalizes the so-called bump number and jump number problems on posets. Specifically, if we define $c(x_i, x_j) = 1$, for $x_i \sim x_j$, the problem becomes equivalent to the jump number problem. Since the jump number problem is known to be NP-hard (see [**?**]), the MCLE is also NP-hard. In this work, however, we concentrate on the general version of the problem, in which $\mathbf{P}$ is an arbitrary poset and c is an arbitrary non-negative real-valued cost function satisfying conditions (1).

2 Related Work

Literature Review: essentially Liu1 and Liu2. Use of CP for local search.

3 CP-Based Search of the Neighborhood of a Linear Extension

Given a linear extension $\mathbf{L}_0$ of a poset $\mathbf{P} = (E, \leq)$, with $E = \{x_1, \ldots, x_n\}$, we are interested in finding a linear extension $\mathbf{L}_1$ within a certain neighborhood of $\mathbf{L}_0$ such that $c(\mathbf{L}_1) < c(\mathbf{L}_0)$.

For the purpose of formulating the CP model for local search, we shall define a linear extension of poset $\mathbf{P}$ as a vector $e \in \{1, \ldots, n\}^n$ satisfying:

(a) $e_i \neq e_j$, $\forall i, j \in \{1, \ldots, n\}, i \neq j$;
(b) $e_i < e_j$, $\forall i, j \in \{1, \ldots, n\}$ with $x_i <_{\mathbf{P}} x_j$;

Let us define integer decision variables $V_i \in \{1, \ldots, n\}$, for $i = 1, \ldots, n$, with each V_i corresponding to the position of the i-th element in the linear extension. Conditions (a) and (b) above translate into the following constraints on the V variables:

$$\texttt{all-different}(V_1, \ldots, V_n) \tag{3}$$

$$V_i < V_j, \ \forall i, j \in \{1, \ldots, n\} \text{ such that } x_i <_{\mathbf{P}} x_j. \tag{4}$$

For any instantiation of V satisfying constraints (3) and (4), let us define s_k as the index of the k-th order statistic of $\{V_1, \ldots, V_n\}$, for $k = 1, \ldots, n$.

Clearly, the sequence $(x_{s_1}, x_{s_2}, \ldots, x_{s_n})$ is a linear extension of $\mathbf{P}$, to which we shall refer as the linear extension *induced* by V. Conversely, for an arbitrary linear extension $\mathbf{L} = (x_{k_1}, \ldots, x_{k_n})$ of $\mathbf{P}$, defining $V_{k_j} = j$ satisfies constraints (3) and (4).

We have thus established a correspondence between variables V satisfying (3) and (4) and linear extensions of $\mathbf{P}$. In what follows, we describe additional constraints and decision variables that can be included in the CP model in order to: (i) define an appropriate neighborhood within which $\mathbf{L}_1$ must lie, and (ii) keep track of incomparable pairs of elements of E that are adjacent in $\mathbf{L}_1$.

3.1 Neighborhood-Defining Constraints

Let $\mathbf{L}_0 = (x_{i_1}, \ldots, x_{i_n})$ and let us define $p \in \{1, \ldots, n\}^n$ with $p_{i_j} = j$, $j = 1, \ldots, n$. Let us also define $\mathbf{L}_1$ to be the linear extension induced by the vector V of decision variables. In order to detect which elements x_j $(j = 1, \ldots, n)$ of E change their absolute positions in $\mathbf{L}_1$ with respect to $\mathbf{L}_0$, we can write the reified constraint:

$$V_j \neq p_{i_j} \Leftrightarrow D_j, \tag{5}$$

where each D_j is a binary (Boolean) decision variable corresponding to whether or not the position of x_j in $\mathbf{L}_1$ differs from that in $\mathbf{L}_0$.
By constraining the number of variables D_j that take value 1 to be at most w we restrict the CP model to explore a neighborhood around $\mathbf{L}_0$, namely the set comprised of linear extensions in which at most w elements of E change their positions with respect to $\mathbf{L}_0$. The parameter w will be called the *width* of the neighborhood, which will be denoted by $N_w(\mathbf{L}_0)$. We shall thus write

$$\sum_{j=1}^{n} D_j \leq w. \tag{6}$$

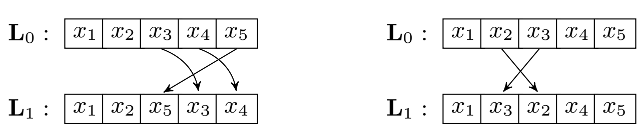

Fig. 1. Examples of $N_3(\mathbf{L}_0)$-movements allowed by the CP model. Arrows correspond to elements that changed their absolute positions with respect to the original linear extension $\mathbf{L}_0$.

3.2 Assessing the Cost of an Extension

In order to take into account the cost of having an incomparable pair $x_i, x_j \in \mathbf{P}$ that happens to be adjacent in $\mathbf{L}_1$ with $V_i < V_j$, we shall write

$$V_j - V_i = 1 \Leftrightarrow B_{i,j}, \tag{7}$$

where $B_{i,j}$ is a binary decision variable. We shall then use variables $B_{i,j}$ to define an objective function, which is to be minimized and is given by

$$\sum_{\substack{i,j=1 \\ x_i \sim x_j}}^{n} [\ c(x_i, x_j) B_{i,j} + c(x_j, x_i) B_{j,i}\ ]\,. \tag{8}$$

3.3 Redundant Constraints

The following set of additional constraints are redundant, but are included into the model with the goal of improving efficiency by means of more effective propagation:

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} B_{i,j} \leq 1,\ \forall i = 1, \ldots, n \tag{9}$$

$$B_{i,j} + B_{j,i} \leq 1,\ \forall i, j = 1, \ldots, n,\ x_i \sim x_j. \tag{10}$$

3.4 An Alternative Neighborhood

An alternative notion of neighborhood is the one in which, instead of having variables D_j to detect changes from $\mathbf{L}_0$ to $\mathbf{L}_1$, we introduce variables $F_{i,j}$, each corresponding to the fact of whether or not the incomparable pair (x_i, x_j) contributes to the total cost of either $\mathbf{L}_0$ or $\mathbf{L}_1$. By writting

$$\sum_{\substack{i,j=1 \\ x_i \sim x_j}}^{n} (F_{i,j} + F_{j,i}) \leq w \tag{11}$$

we constrain the cardinality of the symmetric difference between the sets of pairs that contribute to the total costs of extensions $\mathbf{L}_0$ and $\mathbf{L}_1$.

4 Computational Results

We embedded the CP model described in the previous section within a GRASP framework, in which a randomized version of Algorithm **??** is used as the constructive phase.

5 Conclusions and Future Work

Check the average improvement that we obtain by applying the CP local search to the solution produced by the randomized version of Liu2.

Compare the best solution obtained in this way with that obtained by running the deterministic version of Liu2.

Is the CP-based local search reasonable in terms of running time?