

# Optimal Decomposition in Linear Constraint Systems

Cees Witteveen and Michel Wilson and Tomas Klos

Algorithmics group, Department of Computer and Software Technology,  
Delft University of Technology, NL-2628 CD, Delft, The Netherlands

## Abstract

Decomposition is a technique to obtain complete solutions by assembling independently obtained partial solutions. In particular, constraint decomposition plays an important role in distributed databases, distributed scheduling and violation detection: It enables conflict-free local decision making, while avoiding communication overloading. One of the main issues in decomposition is the loss of flexibility due to decomposition. Here, flexibility roughly refers to the freedom in choosing suitable values for the variables in order to satisfy the constraints. In this paper, we concentrate on linear constraint systems and efficient decomposition techniques for them. Using a generalization of a flexibility metric developed for Simple Temporal Networks, we show how an efficient decomposition technique for linear constraint systems can be derived that minimizes the loss of flexibility. As a by-product of this decomposition technique, we propose an intuitively attractive flexibility metric for linear constraint systems where decomposition does not incur any loss of flexibility.

## Introduction

Decomposition is a technique to obtain total solutions by assembling partial solutions. Typically, these partial solutions are obtained by distributed local problem solving and does not require communication between the individual problem solvers. A well-known example is the maintenance of global integrity constraints in distributed databases (Gupta and Widom 1993; Alwan, Ibrahim, and Udzir 2009). If such global integrity constraints would have to be maintained centrally, communication costs would be too high. Decomposition of such global integrity constraints results in the addition of a set of local integrity constraints to each local database (site) in such a way that satisfaction of all the constraints at each site guarantees satisfaction of the global integrity constraint. Another example is peer-to-peer systems and sensor networks, where one is focusing on the detection of constraint *violations* (Agrawal et al. 2007). Here, constraints define a set of normal states of the total system, and violations of such constraints indicate potential anomalies (e.g., huge file exchanges or extreme sensor readings). In order to detect such anomalies in real time, decomposition is

used to establish constraint violations in a distributed way, without the need for communication during detection.

Note that these decomposition methods are performed *offline* and are not meant to improve the efficiency of finding solutions<sup>1</sup> online. Instead, they provide a means to distribute a constraint problem over a number of local problem solving processes (sites) such that each site is able to propose its own solution. These partial solutions constitute a set of conflict-free solutions and can be easily merged to create a total solution of the original constraint problem.

A major problem in decomposing constraint systems is the loss of *flexibility* due to decomposition (Hunsberger 2002a; Wilson et al. 2013). Here, flexibility roughly refers to the amount of freedom one has in choosing values for variables to satisfy the constraints. Decomposition, while aiming at providing enough flexibility to the local problem solvers, might seriously affect this flexibility. The effect of decomposition on the flexibility of constraint systems has been studied mainly in the context of Simple Temporal Networks (STNs)<sup>2</sup>, where these decomposition methods are known as *temporal decoupling* methods (Hunsberger 2002b; Boerkoel and Durfee 2012; Brambilla 2010).

Currently, the study of flexibility metrics and decoupling methods is restricted to STNs. In this paper, we would like to generalize both the flexibility metrics as well as the decomposition methods to general linear constraint systems such that decomposition now can be applied to a much broader range of applications. In that sense, our work can be seen as an example of a concrete method satisfying the requirements for a general decomposition framework as sketched by Brodsky et al. (Brodsky, Kerschberg, and Varas 2004). As an even more ambitious goal for this paper we aim at a tight integration of decomposition and flexibility computation in one framework. This means not only that we want to use a common (LP-)framework to specify the flexibility of original and decomposed systems, but also that we want to come up with a flexibility metric that allows us to compute an arbitrary decomposition in a very efficient way *without any loss of flexibility*. Using this (stronger) flexibility met-

<sup>1</sup>It has been shown that finding a solution of a constraint system is polynomially related to finding a decomposition for it, see (Planken, de Weerd, and Witteveen 2010).

<sup>2</sup>STNs (Dechter 2003) constitute a rather restricted subset of linear constraint systems.

ric, we are able to generalize a quite recent result obtained by (Wilson et al. 2013) to linear constraint systems.

Before presenting these results, we start with some preliminaries.

## Preliminaries

A linear constraint system is a tuple  $S = (X, C)$ , where  $X = \{x_1, \dots, x_n\}$  is a set of variables and  $C$  is a set of  $m$  constraints  $c_i$  each of the form  $a_{i,1}x_1 + \dots + a_{i,n}x_n \leq b_i$ . A linear constraint system is compactly represented by the matrix inequality  $\mathbf{Ax} \leq \mathbf{b}$  where  $\mathbf{A} = [a_{i,j}]$  is an  $m \times n$  matrix of coefficients,  $\mathbf{x} = [x_i]_{n \times 1}$  is an  $n \times 1$  vector of variables, and  $\mathbf{b} = [b_j]_{m \times 1}$  is an  $m \times 1$  vector of constants. We use the following conventions: Bold-face capital letters refer to matrices, and boldface lower case letters to vectors. Elements of vectors and matrices are referred to by italic lower case letters. If  $\mathbf{A}$  is an  $m \times n$  matrix,  $\mathbf{A}^t$  denotes the  $n \times m$  transpose of  $\mathbf{A}$ . The rows of  $\mathbf{A}$  are indicated by  $\mathbf{a}_i$ ,  $i = 1, 2, \dots, m$ , that is  $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_m]^t$ .

A solution  $\sigma$  of a constraint system  $S$  is an assignment  $\sigma(\mathbf{x}) = \mathbf{v}$  of the  $n \times 1$  vector  $\mathbf{x}$  of variables to an  $n \times 1$  vector  $\mathbf{v}$  of real numbers such that  $\mathbf{Av} \leq \mathbf{b}$ . Equivalently, we will refer to such an assignment  $\sigma$  as a function from  $X$  to  $\mathbb{R}$ . In this paper, we always assume a linear constraint system  $S$  to be *consistent* (there always exist at least one solution  $\sigma$ ), and *bounded* (every solution  $\sigma$  assigns a bounded real value  $v$  to a variable  $x$ ). We will use  $S = (X, C)$  or  $S : \mathbf{Ax} \leq \mathbf{b}$  to refer to a linear constraint system.

In this paper we often refer to Simple Temporal Networks, abbreviated by STNs. An STN  $S = (X, C)$  (Dechter, Meiri, and Pearl 1991; Dechter 2003) is a special case of a linear constraint system where every constraint in  $C$  is a binary difference constraint  $x_i - x_j \leq c_{ij}$  for some constant  $c_{ij} \in \mathbb{R}$ .

## Generalizing the STN flexibility metric

Instead of a single solution to a constraint system  $S$ , we are interested in the *flexibility* of  $S$ . Intuitively, the flexibility of  $S$  is related to the freedom we have in choosing values  $v_i$  to assign to variables  $x_i$  in solutions  $\sigma$  for  $S$ . To introduce such a flexibility metric for linear systems, we first concentrate on STNs, for which a variety of flexibility metrics have been proposed.

Considering consistent STNs, it is well-known (Dechter 2003) that for every  $x \in X$  we can find its (unique) minimum<sup>3</sup> value  $est(x)$  and maximum value  $lst(x)$  that can occur in any solution of  $S = (X, C)$ . Moreover, the set of minimum values and the set of maximum values both constitute a solution to  $S$ . Finally, it holds that, for every variable  $x_i \in X$  and for every value  $v_i \in [est(x_i), lst(x_i)]$ , the assignment  $x_i \mapsto v_i$  can be extended to a solution of the constraint system. So for every (single!)  $x_i \in X$ , we are free to choose any value between  $est(x_i)$  and  $lst(x_i)$  without jeopardizing the satisfaction of the constraints.

<sup>3</sup>STNs are used to specify temporal scheduling problems. Using standard STN-terminology, we use  $est(x)$  and  $lst(x)$  as familiar notations to indicate the earliest (minimum) and latest (maximum) time value respectively, for the variable  $x$ .

These properties justified the idea that the flexibility  $flex(x_i)$  of a variable  $x_i$  can be associated with the difference  $lst(x_i) - est(x_i)$ . Therefore, quite a lot of flexibility metrics for STNs (Hunsberger 2002b; 2002a; Pollicella et al. 2004; 2005) essentially are based on defining the flexibility  $flex(S)$  of the constraint system  $S$  as the sum  $flex(S) = \sum_{x_i \in X} (lst(x_i) - est(x_i))$ .

We want to obtain a flexibility metric  $flex(S)$  for linear constraint systems as a conservative extension of this flexibility metric. The problem, however, when generalizing from STNs to linear systems, is that, in general, the set of maximum (minimum) values for variables does *not* constitute a solution to  $S$ .

**Example 1.** Consider the system  $S = (X, C)$  where  $X = \{x_1, x_2, x_3\}$  and  $C$  contains the following linear constraints:  $-x_i \leq 0$ , for  $i = 1, 2, 3$ ,  $x_1 + x_3 \leq 50$ ,  $x_2 + x_3 \leq 50$ , and  $x_1 \leq x_3$ . The matrix equation corresponding to  $S$  is  $\mathbf{Ax} \leq \mathbf{b}$ , where

$$\mathbf{A} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & -1 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 50 \\ 50 \\ 0 \end{bmatrix}$$

Observe that the set of individual maxima of the variables (i.e.,  $x_1 = 25, x_2 = 50, x_3 = 50$ ) does not constitute a solution to  $S$ . On the other hand, if we would maximize the *sum*  $x_1 + x_2 + x_3$  of all variables given  $\mathbf{Ax} \leq \mathbf{b}$ , there is a unique maximum value for this sum (75). Likewise, minimizing this sum given  $\mathbf{Ax} \leq \mathbf{b}$  results in a minimum value (0). Intuitively then, we could define the flexibility of  $S$  as the difference between the maximum and the minimum value of this sum. Hence, the resulting flexibility of this system would be  $75 - 0 = 75$ . There is however, one caveat: obviously, we should ensure that the value of any variable  $x_i$  in maximising the sum should be at least as large as its value in minimising the sum of variables. ■

Based on this idea, we propose to define the flexibility  $flex(S)$  of a linear system  $S : \mathbf{Ax} \leq \mathbf{b}$  as follows: Consider two vectors of variables  $\mathbf{x}^- = [x_i^-]_{n \times 1}$  and  $\mathbf{x}^+ = [x_i^+]_{n \times 1}$  and the following LP:

$$\begin{aligned} \max \quad & \sum_{x_i \in X} (x_i^+ - x_i^-) \\ \text{s.t. :} \quad & \mathbf{Ax}^- \leq \mathbf{b}, \\ & \mathbf{Ax}^+ \leq \mathbf{b}, \\ & \mathbf{x}^- \leq \mathbf{x}^+ \end{aligned} \tag{1}$$

Note that a maximizer  $\sigma$  of the LP (1), where  $\sigma(\mathbf{x}^-) = [v_i^-]_{n \times 1}$  and  $\sigma(\mathbf{x}^+) = [v_i^+]_{n \times 1}$ , can also be viewed upon as a set of non-empty<sup>4</sup> intervals  $I_X = \{[v_i^-, v_i^+] \mid i = 1, \dots, n\}$  for the variables in  $X$ . In fact,  $I_X$  is a set of intervals with maximum total length, such that both the lower bounds  $v_i^-$  of the intervals as well as the upper bounds  $v_i^+$  constitute a solution to  $S$ .

<sup>4</sup>Observe that consistency and boundedness of  $S$  always guarantee the existence of such a set of intervals.

This flexibility metric  $\max \sum_{x_i \in X} (x_i^+ - x_i^-)$  for linear constraint systems constitutes a conservative extension of the flexibility metric  $flex()$  defined for STNs:

**Proposition 1.** *Let  $S : \mathbf{Ax} \leq \mathbf{b}$  be an STN. Then for the LP (1) associated with this constraint system, it holds that*

$$\max \sum_{x_i \in X} (x_i^+ - x_i^-) = \sum_{x_i \in X} (lst(x_i) - est(x_i)).$$

*Proof.* Observe that

$$\begin{aligned} \max \sum_{x_i \in X} (x_i^+ - x_i^-) &\leq \max \left\{ \sum_{x_i \in X} x_i^+ \right\} - \min \left\{ \sum_{x_i \in X} x_i^- \right\} \\ &\leq \sum_{x_i \in X} lst(x_i) - \sum_{x_i \in X} est(x_i) \\ &= flex(S). \end{aligned}$$

Since  $S$  is an STN, both  $\sigma^+(x_i) = lst(x_i)$  and  $\sigma^-(x_i) = est(x_i)$ ,  $i = 1, \dots, n$ , are solutions of  $\mathbf{Ax} \leq \mathbf{b}$ . Hence,  $\max \sum_{x_i \in X} (x_i^+ - x_i^-) = \sum_{x_i \in X} (lst(x_i) - est(x_i)) = flex(S)$ .  $\square$

Every variable  $x_i \in X$  in an STN  $S$  enjoys the property<sup>5</sup> that, for every value  $v_i \in [est(x_i), lst(x_i)]$ , the assignment  $\sigma(x_i) = v_i$  is extendable to a complete solution  $\sigma$  of  $S$ . For maximizers of linear systems using the LP (1) the same property holds:

**Proposition 2.** *Let  $\mathbf{v}^t = [(\mathbf{v}^-)^t, (\mathbf{v}^+)^t]$  be a maximizer of the LP (1). Then for every  $x_i \in X$  and every  $v_i \in [v_i^-, v_i^+]$ , there exists a solution  $\sigma$  for  $\mathbf{Ax} \leq \mathbf{b}$  such that  $\sigma(x_i) = v_i$ .*

*Proof.* Note that both  $\mathbf{v}^-$  and  $\mathbf{v}^+$  are solutions of  $\mathbf{Ax} \leq \mathbf{b}$ . Clearly,  $v_i \in [v_i^-, v_i^+]$  implies  $v_i = \lambda v_i^+ + (1 - \lambda)v_i^-$  for some  $0 \leq \lambda \leq 1$ . Since every convex combination of solutions of an LP is a solution as well, it follows that  $\sigma(\mathbf{x}) = \lambda \mathbf{v}^+ + (1 - \lambda)\mathbf{v}^-$  is a solution of  $\mathbf{Ax} \leq \mathbf{b}$ , extending  $\sigma(x_i) = v_i$ .  $\square$

## Decomposition in linear constraint systems

Often, constraint problems have to be solved in a distributive context (see e.g. (Hunsberger 2002b; Boerkoel and Durfee 2013)) where communication between the problem solvers is not possible. We consider the case that the set  $X$  of variables in  $S = (X, C)$  is partitioned into  $k$  disjoint subsets  $X_i$ , each of them controlled by an independent agent. Besides  $X_i$ , such an agent also is given a set of local constraints  $C_i$ . While the task of each agent is to find a solution  $\sigma_i$  for its local constraint system  $S_i = (X_i, C_i)$ , as designers of the distributed system we have to make sure that the *merge*<sup>6</sup>  $\sigma = \bigsqcup_{i=1}^k \sigma_i$  of these partial solutions  $\sigma_i$  is a solution of the original system  $S$ , whatever choices are made for these  $\sigma_i$ . Then the *flexibility-maximizing decomposition* problem for constraint systems is the following problem:

<sup>5</sup>Intuitively, this property guarantees that  $[est(x_i), lst(x_i)]$  does not contain useless values.

<sup>6</sup>Think of the merge  $\bigsqcup_{i=1}^k \sigma_i$  as an assembling function  $\sigma$  such that for  $j = 1, \dots, n$ ,  $\sigma(x_j)$  is the value assigned by the agent controlling  $x_j$ .

Given a constraint system  $S = (X, C)$  and a partitioning  $\{X_i\}_{i=1}^k$  of  $X$ , how to come up with suitable constraint sets  $C_i$  for the subsets  $X_i$  such that (i) whatever solutions  $\sigma_i$  for the subsystems  $S_i = (X_i, C_i)$  are chosen, their merge  $\sigma = \bigsqcup_{i=1}^k \sigma_i$  is always a solution for the original system  $S$ , while (ii) the total flexibility  $\sum_{i=1}^k flex(S_i)$  of the distributed system is maximized.

For STNs Hunsberger (Hunsberger 2002b) approached this problem by making a distinction between *intra-agent constraints*, involving variables belonging to a single agent and *inter-agent constraints*, involving variables controlled by different agents. Then he proposed so-called *temporal decoupling* algorithms that, by tightening intra-agent constraints, make the set of inter-agent constraints obsolete. As a result, a set of decomposed subsystems  $S_i$  is returned such that each combination of solutions  $\sigma_i$  for these subsystems  $S_i$  can be merged into a complete solution of the original problem. Hunsberger observed that in some cases the added (tightened) constraints may severely limit the flexibilities of the individual subproblems, i.e., the sum  $\sum_{j=1}^k flex(S_j)$  of the flexibilities of the subsystems  $S_j$  could be considerably less than the flexibility  $flex(S)$  of the original system. Therefore, he proposed an algorithm that ensures a *locally optimal* decoupling, i.e., a decoupling such that no individual constraint can be loosened without violating the property of inducing a decoupling.

However, instead of locally optimal decompositions, we want to establish *globally* optimal decomposition methods not restricted to STNs, but applicable to linear constraint systems.

To model this idea of decomposition, without loss of generality, we may assume that the variables  $\mathbf{x}$  occurring in a (consistent and bounded) linear constraint system  $\mathbf{Ax} \leq \mathbf{b}$  are partitioned into a set of  $k$  vectors  $\mathbf{y}_i$  of variables such that  $[\mathbf{y}_1^t, \dots, \mathbf{y}_k^t] = \mathbf{x}^t$ . This corresponds to creating a partitioning of the set of variables  $X$ . The flexibility-maximizing decomposition problem for a linear constraint system  $S$  then is to find  $k$  matrices  $\mathbf{A}_i$  and  $k$  vectors  $\mathbf{b}'_i$  such that

1. for  $j = 1, \dots, k$ ,  $S_j : \mathbf{A}_j \mathbf{y}_j \leq \mathbf{b}'_j$  is a linear constraint system, and
2. whenever  $\sigma_i(\mathbf{y}_j) = \mathbf{v}_j$  is a solution to  $S_j$ , the solution  $\sigma$  defined by  $\sigma(\mathbf{x})^t = [\sigma(\mathbf{v}_1)^t, \dots, \sigma(\mathbf{v}_k)^t]$  is a solution to  $\mathbf{Ax} \leq \mathbf{b}$ , and
3. the sum  $\sum_{j=1}^k flex(S_j)$  is maximal.

**Example 2.** Assume that in the constraint system discussed in Example 1, we have two agents, one controlling the variables  $x_1$  and  $x_2$ , and the other the variable  $x_3$ . Then  $\mathbf{y}_1 = [x_1, x_2]$  and  $\mathbf{y}_2 = [x_3]$ . Consider the decomposition  $\{S_1 : \mathbf{A}_1 \mathbf{y}_1 \leq \mathbf{b}'_1, S_2 : \mathbf{A}_2 \mathbf{y}_2 \leq \mathbf{b}'_2\}$  of  $S$  where

$$\mathbf{A}_1 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{y}_1 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \mathbf{b}'_1 = \begin{bmatrix} 0 \\ 0 \\ 25 \end{bmatrix}$$

$$\mathbf{A}_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad \mathbf{y}_2 = [x_3], \quad \mathbf{b}'_2 = \begin{bmatrix} 25 \\ -25 \end{bmatrix}$$

Clearly, the merge of any pair of solutions  $\sigma_1$  and  $\sigma_2$  of these partial systems constitutes a solution to the total system. But

this decomposition comes with a high price: the total flexibility of these two systems equals  $25 + 0 = 25$ , that is one-third of the original flexibility. The question then arises whether this decomposition is a maximally-flexible one. ■

Like Hunsberger, in linear constraint systems with a partitioning  $[\mathbf{y}_1^t, \dots, \mathbf{y}_k^t] = \mathbf{x}^t$  of  $X$  we also distinguish *intra-* and *inter-agent* constraints:

1. An intra-agent constraint  $c_i \in \text{Intra}(C)$  is any constraint  $c_i : \mathbf{a}_i \mathbf{x} \leq b_i$  such that for all  $a_{i,h} \neq 0$  the corresponding variables  $x_h$  belong to a single block  $\mathbf{y}_j$ . For example, the first three constraints in  $\mathbf{A}$  (see Example 1) are intra-agent constraints.
2. If  $c_i : \mathbf{a}_i \mathbf{x} \leq b_i$  contains variables  $x_h$  and  $x_{h'}$  that (i) belong to different blocks of variables  $\mathbf{y}_j$  and  $\mathbf{y}_{j'}$ , respectively, while (ii)  $a_{i,h} \neq 0$  and  $a_{i,h'} \neq 0$ , then  $c_i \in \text{Inter}(C)$  is said to be an *inter-agent* constraint. For example, the last three constraints in  $\mathbf{A}$  are inter-agent constraints.

The idea of decomposition is to provide an additional set of intra-agent constraints such that all inter-agent constraints are implied. To see how this can be realised, consider the LP (1) for flexibility maximization. A maximizer of this LP consists of vectors  $\mathbf{v}^-$  and  $\mathbf{v}^+$ . Let  $I_X = \{[v_i^-, v_i^+] \mid x_i \in X\}$  be the corresponding set of intervals. From Proposition 2 we know that, for every single  $x_i$ , every value  $v_i \in [v_i^-, v_i^+]$  can be chosen without jeopardizing the satisfaction of any of the constraints. This property, however is no longer sufficient in a decomposed system: In a decomposed system agents might choose an arbitrary value for a variable  $x_i$  in its variable block  $\mathbf{y}_j$  *concurrently* and independently from the others. So, instead of extending an assignment for a single choice, we now have to be prepared for extending an assignment for *several concurrent choices* to a total solution. Hence, the set  $I_X$  of intervals has to satisfy an additional requirement: every *inter-agent constraint*  $c_i$  has to be satisfied whatever choices  $v_j$  in the intervals  $[v_j^-, v_j^+]$  have been made by the individual agents. So suppose the  $i$ -th constraint is an inter-agent constraint  $c_i \in \text{Inter}(C)$  and equals

$$\mathbf{a}_i \mathbf{x} = a_{i,1}x_1 + \dots + a_{i,m}x_m \leq b_i$$

Then it should hold that

$$\mathbf{a}_i \mathbf{x} \leq b_i \quad \forall \mathbf{x} \in [\mathbf{v}^- \leq \mathbf{x} \leq \mathbf{v}^+]$$

But that means that such a set of flexibility-maximizing intervals  $\{[v_i^-, v_i^+]\}_{i=1}^n$  can be derived from a maximizer of the following LP:

$$\begin{aligned} \max \quad & \sum_{x_i \in X} (x_i^+ - x_i^-) \\ \text{s.t. :} \quad & \mathbf{A} \mathbf{x}^- \leq \mathbf{b}, \\ & \mathbf{A} \mathbf{x}^+ \leq \mathbf{b}, \\ & \mathbf{a}_i \mathbf{x} \leq b_i \quad \forall \mathbf{x} \in [\mathbf{x}^- \leq \mathbf{x} \leq \mathbf{x}^+] \quad \forall c_i \in \text{Inter}(C), \\ & \mathbf{x}^- \leq \mathbf{x}^+ \end{aligned} \quad (2)$$

Here, as indicated in the preliminaries,  $\mathbf{a}_i$  denotes the  $i$ -th row of  $\mathbf{A}$  and  $b_i$  refers to the  $i$ -th entry of  $\mathbf{b}$ .

Clearly, the occurrence of an infinite set of constraints in the LP (2) prevents an efficient computation of the maximum

flexibility. Fortunately, there is an easy way to remove these infinite sets of constraints by showing that one extremal solution to the constraint  $\mathbf{A}_i \mathbf{x} \leq b_i$  implies the existence of solutions for all points  $\mathbf{x}^- \leq \mathbf{x} \leq \mathbf{x}^+$ :

**Proposition 3.** Let  $\sigma(\mathbf{x})^t = [(\mathbf{v}^-)^t, (\mathbf{v}^+)^t]$  be a maximizer of the LP (2) and let  $c_i : \mathbf{a}_i \mathbf{x} \leq b_i$  be a linear constraint. Define the vector  $\mathbf{v}_i^* = [v_{i,j}^*]_{n \times 1}$  such that for every  $j = 1, \dots, n$ ,

$$v_{i,j}^* = \begin{cases} v_j^+, & \text{if } a_{i,j} > 0 \\ v_j^-, & \text{if } a_{i,j} \leq 0. \end{cases}$$

Then  $\mathbf{a}_i \mathbf{v}_i^* \leq b_i$  iff  $\mathbf{a}_i \mathbf{v} \leq b_i, \forall \mathbf{v} \in [\mathbf{v}^- \leq \mathbf{v} \leq \mathbf{v}^+]$ .

*Proof.* The only-if direction is obvious since, by definition,  $\mathbf{v}^- \leq \mathbf{v}_i^* \leq \mathbf{v}^+$ . Conversely, assume that  $\mathbf{a}_i \mathbf{v}_i^* \leq b_i$  and let  $\mathbf{v} = [v_j]$  be an arbitrary vector such that  $\mathbf{v}^- \leq \mathbf{v} \leq \mathbf{v}^+$ . Without loss of generality, we may assume that there exists a  $0 \leq k \leq m$  such that  $a_{i,j} > 0$  for  $1 \leq j \leq k$  and  $a_{i,j} \leq 0$  for  $m \geq j > k$ . Then it holds that

$$\sum_{j=1}^m a_{i,j} v_j \leq \sum_{j=1}^k a_{i,j} v_j^+ + \sum_{j=k+1}^m a_{i,j} v_j^- = \sum_{j=1}^m a_{i,j} v_{i,j}^* \leq b_i$$

Hence,  $\sigma(\mathbf{x}) = \mathbf{v}$  satisfies the constraint  $c_i$ , too. □

As a consequence, we can replace the infinite set of equations belonging to an inter-agent constraint in LP (2) by one single constraint, resulting in the following LP equivalent to LP (2):

$$\begin{aligned} \max \quad & \sum_{x_i \in X} (x_i^+ - x_i^-) \\ \text{s.t. :} \quad & \mathbf{A} \mathbf{x}^- \leq \mathbf{b}, \\ & \mathbf{A} \mathbf{x}^+ \leq \mathbf{b}, \\ & \mathbf{a}_i \mathbf{x}_i^* \leq b_i \quad \forall c_i \in \text{Inter}(C), \\ & \mathbf{x}^- \leq \mathbf{x}^+. \end{aligned} \quad (3)$$

Here,<sup>7</sup>  $\mathbf{x}_i^*$  is defined analogously to  $\mathbf{v}_i^*$ .

**Example 3.** Consider the constraint system discussed in Example 1, with the partitioning  $\mathbf{y}_1 = [x_1, x_2]$  and  $\mathbf{y}_2 = [x_3]$ . Using the matrix  $\mathbf{A}$  derived in Example 1, we can compute the flexibility of the decomposition of  $S$  by computing:

$$\begin{aligned} \max \quad & \sum_{x_i \in X} (x_i^+ - x_i^-) \\ \text{s.t. :} \quad & \mathbf{A} \mathbf{x}^+ \leq \mathbf{b}, \\ & \mathbf{A} \mathbf{x}^- \leq \mathbf{b}, \\ & x_1^+ + x_3^+ \leq 50, \\ & x_2^+ + x_3^+ \leq 50, \\ & x_1^+ - x_3^- \leq 0, \\ & \mathbf{x}^- \leq \mathbf{x}^+. \end{aligned}$$

The flexibility of this system is 50. A maximizer is  $\mathbf{v}^t = [(\mathbf{v}^-)^t, (\mathbf{v}^+)^t] = [[0, 0, 25]^t, [25, 25, 25]^t]$ . Note that this (maximal) flexibility of this decomposition is larger than the flexibility of the decomposition discussed in Example 2. ■

<sup>7</sup>Note that both  $\mathbf{a}_i \mathbf{x}^+ \leq b_i$  and  $\mathbf{a}_i \mathbf{x}^- \leq b_i$  occurring in  $\mathbf{A} \mathbf{x}^+ \leq \mathbf{b}$  and  $\mathbf{A} \mathbf{x}^- \leq \mathbf{b}$ , respectively, are implied by  $\mathbf{a}_i \mathbf{x}^* \leq b_i$ .

One of our goals in this paper is not only to compute the maximum flexibility of a decomposed system, but also to compute an optimal decomposition realizing this maximum flexibility. One of the nice features of our framework is that, once we have determined the flexibility of a decomposed system, the decomposition itself can be easily derived from the solution of the LP (3), as we will show in the next section.

### How to decompose a linear constraint system

To compute an actual decomposition of a linear constraint system  $S : \mathbf{Ax} \leq \mathbf{b}$  using a partitioning  $[\mathbf{y}_1^t, \dots, \mathbf{y}_k^t] = \mathbf{x}^t$  of  $\mathbf{x}$ , we need to specify a decomposition  $\{S_j : \mathbf{A}_j \mathbf{y}_j \leq \mathbf{b}'_j\}_{j=1}^k$  with the required properties as mentioned before. To obtain this decomposition, we first solve the LP (3) and then use a maximizer  $\mathbf{v}^t = [(\mathbf{v}^-)^t, (\mathbf{v}^+)^t]$  for this LP to obtain the decomposition as follows:

Using the partitioning  $[\mathbf{y}_1^t, \dots, \mathbf{y}_k^t] = \mathbf{x}^t$  of  $\mathbf{x}$ , every constraint  $c_i : \mathbf{a}_i \mathbf{x} \leq b_i$  in  $S$  can be written as:

$$\mathbf{a}_i \mathbf{x} = \mathbf{a}_{(i,1)} \mathbf{y}_1 + \dots + \mathbf{a}_{(i,k)} \mathbf{y}_k \leq b_i$$

where  $\mathbf{a}_i = [\mathbf{a}_{(i,1)}, \dots, \mathbf{a}_{(i,k)}]$  is the  $i$ -th row of  $A$ . Since  $\mathbf{v}^t = [(\mathbf{v}^-)^t, (\mathbf{v}^+)^t]$  is a maximizer for LP (3), it holds that

$$\mathbf{a}_i \mathbf{v}_i^* \leq b_i,$$

where  $\mathbf{v}_i^* = [v_{i,j}^*]$  is defined as:

$$v_{i,j}^* = \begin{cases} v_j^+, & \text{if } a_{i,j} > 0 \\ v_j^-, & \text{if } a_{i,j} \leq 0. \end{cases}$$

Let us partition the vector  $\mathbf{v}_i^*$  into  $[v_{i,1}^*, v_{i,2}^*, \dots, v_{i,k}^*]^t$  according to the partitioning of  $\mathbf{x}$ . Then  $\mathbf{a}_i \mathbf{v}_i^*$  can be written as

$$\mathbf{a}_i \mathbf{v}_i^* = \mathbf{a}_{(i,1)} v_{i,1}^* + \dots + \mathbf{a}_{(i,k)} v_{i,k}^* \leq b_i \quad (4)$$

Using this equation, we replace the inter-agent constraint  $c_i : \mathbf{a}_i \mathbf{x} \leq b_i$  by a set of *intra-agent* constraints as follows: For every vector  $\mathbf{y}_j$ , create the constraint

$$\mathbf{a}_{(i,j)} \mathbf{y}_j \leq b'_{i,j} \quad (5)$$

where  $b'_{i,j} = \mathbf{a}_{(i,j)} \mathbf{v}_{i,j}^*$  and add this constraint to  $S_j$ .

Applying this procedure for all constraints results in a set of subsystems  $\{\mathbf{A}_j \mathbf{y}_j \leq \mathbf{b}'_j\}_{j=1}^k$  derived from  $\mathbf{Ax} \leq \mathbf{b}$ . Note that some of these constraints might be trivial if all the coefficients in a row of  $\mathbf{A}_j$  are 0. Therefore, these constraints (rows) can be omitted.

It is easy to see that this set  $\{\mathbf{A}_j \mathbf{y}_j \leq \mathbf{b}'_j\}_{j=1}^k$  is a decomposition of  $S : \mathbf{Ax} \leq \mathbf{b}$ : For  $j = 1, \dots, k$ , let  $\sigma_j(\mathbf{y}_j)$  be an arbitrary solution to the subsystem  $S_j : \mathbf{A}_j \mathbf{y}_j \leq \mathbf{b}'_j$ . Finally, let  $\sigma(\mathbf{x})$  be the merge of all the solutions  $\sigma_j(\mathbf{y}_j)$ . We have to show that  $\sigma(\mathbf{x})$  is a solution of the original system  $S$ .

Consider an arbitrary constraint  $c_i : \mathbf{a}_i \mathbf{x} \leq b_i$  of  $S$ . By construction, for  $j = 1, \dots, k$  there exist constraints  $c_{i,j} : \mathbf{a}_{(i,j)} \mathbf{y}_j \leq b'_{i,j}$  in  $S_j$  such that  $\mathbf{a}_{(i,j)} \sigma_j(\mathbf{y}_j) \leq b'_{i,j}$  and  $b'_{i,j} = \mathbf{a}_{(i,j)} \mathbf{v}_{i,j}^*$ . Hence, since  $\sigma$  is the merge of the local solutions  $\sigma_j$ :

$$\mathbf{a}_i \sigma(\mathbf{x}) = \sum_{j=1}^k \mathbf{a}_{(i,j)} \sigma_j(\mathbf{y}_j) \leq \sum_{j=1}^k b'_{i,j} = \sum_{j=1}^k \mathbf{a}_{(i,j)} \mathbf{v}_{i,j}^* \leq b_i$$

where the last inequality holds by the inequalities (4) and (5). Hence, the merge  $\sigma(\mathbf{x})$  of all solutions  $\sigma_j(\mathbf{y}_j)$  satisfies  $c_i$ .

**Example 4.** Continuing Example 3, consider a maximizer  $\mathbf{v}^- = [0, 0, 25]^t$  and  $\mathbf{v}^+ = [25, 25, 25]^t$  for a decomposition of  $S$ . Using this maximizer, the constraints  $-x_1 \leq 0$ ,  $-x_2 \leq 0$  are added to  $S_1$ , and  $-x_3 \leq 0$  is added to  $S_2$ . The constraint  $x_1 + x_3 \leq 50$  is split into two constraints  $x_1 \leq 25$  and  $x_3 \leq 25$ . These are added to  $S_1$  and  $S_2$ , respectively. Likewise,  $x_2 + x_3 \leq 50$  is split into  $x_2 \leq 25$  and  $x_3 \leq 25$ . Finally,  $x_1 \leq x_3$  is split into  $x_1 \leq 25$  and  $-x_3 \leq -25$ . The outcome of this procedure is the following decomposition  $[\mathbf{A}_1 \mathbf{y}_1 \leq \mathbf{b}'_1, \mathbf{A}_2 \mathbf{y}_2 \leq \mathbf{b}'_2]$  of  $S$  where

$$\mathbf{A}_1 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{y}_1 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \mathbf{b}'_1 = \begin{bmatrix} 0 \\ 0 \\ 25 \\ 25 \end{bmatrix}$$

$$\mathbf{A}_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad \mathbf{y}_2 = [x_3], \quad \mathbf{b}'_2 = \begin{bmatrix} 25 \\ -25 \end{bmatrix}$$

■

Although the procedure specified above results in a decomposition of the original system  $\mathbf{Ax} \leq \mathbf{b}$ , we also have to show that it results in a flexibility-maximizing decomposition, i.e., the sum of the flexibilities  $flex(S_i)$  of the subsystems equals the flexibility as computed by the LP (3). So let  $S = (X, C)$  be partitioned by  $\{X_j\}_{j=1}^n$ . Let  $F$  be the maximum flexibility of this system as computed by the LP (3) and  $\mathbf{v}^t = [(\mathbf{v}^-)^t, (\mathbf{v}^+)^t]$  be a corresponding maximizer. Let the resulting decomposition using  $\mathbf{v}^t$  be  $\{S_j : (X_j, C_j)\}_{j=1}^k = \{S_j : \mathbf{A}_j \mathbf{y}_j \leq \mathbf{b}'_j\}_{j=1}^k$ . We show that  $\sum_{j=1}^k flex(S_j) = F$ , where  $flex(S_j)$  is the maximum flexibility computed using the original LP (1) for  $\mathbf{A}_j \mathbf{y}_j \leq \mathbf{b}'_j$ .

1.  $\sum_{j=1}^k flex(S_j) \leq F$ . Note that  $\sum_{j=1}^k flex(S_j) = flex(S')$  where  $S' = (X, \bigcup_{j=1}^k C_j)$  and the flexibility is computed using LP (1). Let  $F'$  be the flexibility of  $S'$  computed with the LP (3). Since there are no inter-agent constraints, in this case LP (3) is equivalent to the LP (1), hence  $flex(S') = F'$ . Now consider  $S'' = (X, \bigcup_{j=1}^k C_j \cup C)$  with the same partitioning  $\{X_j\}_{j=1}^n$  as  $S$  and let LP (3) compute its maximum flexibility  $F''$ . On one hand, since every constraint in  $C$  is either equal to a constraint in  $\bigcup_{j=1}^k C_j$  or implied by it, it should hold that  $F' = F''$ . On the other hand,  $\bigcup_{j=1}^k C_j \cup C$  contains  $C$ , and, since the partitioning is identical,  $F'' \leq F$ . Therefore, we have  $\sum_{j=1}^k flex(S_j) = flex(S') = F' = F'' \leq F$ .
2.  $F \leq \sum_{j=1}^k flex(S_j)$  Consider the maximizer  $\mathbf{v}^t = [(\mathbf{v}^-)^t, (\mathbf{v}^+)^t]$  for the LP (3) applied to  $S$ . For  $j = 1, \dots, k$ , let  $\mathbf{v}_j^t = [(\mathbf{v}_j^-)^t, (\mathbf{v}_j^+)^t]$  be the parts of the maximizer corresponding to the partitioning of  $\mathbf{x}$ . We show that, for  $j = 1, \dots, k$ , both  $\mathbf{v}_j^-$  and  $\mathbf{v}_j^+$  are solutions to  $\mathbf{A}_j \mathbf{y}_j \leq \mathbf{b}'_j$ . This is easy to see since

$$\mathbf{A}_j \mathbf{v}_j^- \leq \mathbf{A}_j \mathbf{v}_j^* \leq \mathbf{b}'_j \quad \text{and} \quad \mathbf{A}_j \mathbf{v}_j^+ \leq \mathbf{A}_j \mathbf{v}_j^* \leq \mathbf{b}'_j.$$

So, for  $j = 1, \dots, k$ ,  $flex(S_j) \geq \sum_{x_{i_j} \in X_j} (v_{i_j}^+ - v_{i_j}^-)$ .

Summing up, we derive

$$\sum_{j=1}^k flex(S_j) \geq \sum_{j=1}^k \sum_{x_{i_j} \in X_j} (v_{i_j}^+ - v_{i_j}^-) = F.$$

### A strong flexibility metric

In the LP (3) every inter-agent constraint  $\mathbf{a}_i \mathbf{x}^* \leq b_i$  entails two constraints  $\mathbf{a}_i \mathbf{x}^+ \leq b_i$  and  $\mathbf{a}_i \mathbf{x}^- \leq b_i$  occurring in  $\mathbf{A} \mathbf{x}^+ \leq \mathbf{b}$  and  $\mathbf{A} \mathbf{x}^- \leq \mathbf{b}$ , respectively. Hence, these constraints can be removed from the LP-specification without affecting the solution. As a limiting case then, when the (finest) partitioning of the variables results in  $n$  blocks containing only one variable  $x_i$ , the LP-specification obtained from LP (3) is:

$$\begin{aligned} \max \quad & \sum_{x_i \in X} (x_i^+ - x_i^-) \\ \text{s.t. :} \quad & \mathbf{a}_i \mathbf{x}_i^* \leq b_i \quad i = 1, 2, \dots, m, \\ & \mathbf{x}^- \leq \mathbf{x}^+. \end{aligned} \quad (6)$$

Defining the matrices  $\mathbf{A}^+ = [a_{i,j}^+]$  and  $\mathbf{A}^- = [a_{i,j}^-]$  by

$$a_{i,j}^+ = \begin{cases} a_{i,j} & \text{if } a_{i,j} > 0 \\ 0 & \text{else} \end{cases} \quad a_{i,j}^- = \begin{cases} a_{i,j} & \text{if } a_{i,j} < 0 \\ 0 & \text{else} \end{cases}$$

we can easily rewrite this LP to the following simple LP:

$$\begin{aligned} \max \quad & \sum_{x_i \in X} (x_i^+ - x_i^-) \\ \text{s.t. :} \quad & \mathbf{A}^+ \mathbf{x}^+ + \mathbf{A}^- \mathbf{x}^- \leq \mathbf{b}, \\ & \mathbf{x}^- \leq \mathbf{x}^+ \end{aligned} \quad (7)$$

Let us denote the flexibility of a constraint system  $S$  computed for the finest partition  $\{x_i\}_{i=1}^n$  by  $flex^*(S)$ , i.e.,  $flex^*(S)$  is the outcome of solving LP (7). This metric exhibits a stronger property than the weak flexibility metric  $flex(S)$  we have discussed before: Consider the set of intervals  $\{[v_i^-, v_i^+]\}_{i=1}^n$  generated by a maximizer  $\mathbf{v}^t = [(\mathbf{v}^-)^t, (\mathbf{v}^+)^t]$  for LP (7). This metric not only assures that for every single variable  $x_i$  every value  $v_i$  in the interval  $[v_i^-, v_i^+]$  can occur in a solution  $\sigma$ , but also that this property holds for *any concurrent choice* of values for a *set* of variables: For any subset  $X' \subseteq X$  of variables, values  $v_i$  in the intervals  $[v_i^-, v_i^+]$  might be chosen *simultaneously* without jeopardizing the satisfaction of all the constraints:

**Proposition 4.** *Let  $S : \mathbf{A} \mathbf{x} \leq \mathbf{b}$  be a constraint system with the finest partitioning  $X = \{x_i\}_{i=1}^n$ . Let  $\mathbf{v}^t = [(\mathbf{v}^-)^t, (\mathbf{v}^+)^t]$  be a maximizer for the LP (7). Then for every  $\mathbf{v}$  such that  $\mathbf{v}^- \leq \mathbf{v} \leq \mathbf{v}^+$ , it holds that  $\mathbf{A} \mathbf{v} \leq \mathbf{b}$ .*

*Proof.* Let  $\mathbf{v}$  be such that  $\mathbf{v}^- \leq \mathbf{v} \leq \mathbf{v}^+$ . Then we have

$$\mathbf{A} \mathbf{v} = \mathbf{A}^+ \mathbf{v} + \mathbf{A}^- \mathbf{v} \leq \mathbf{A}^+ \mathbf{v}^+ + \mathbf{A}^- \mathbf{v}^- \leq \mathbf{b}. \quad \square$$

Therefore, we call  $flex^*(\cdot)$  a *strong flexibility metric*. As the careful reader might have noticed, we can easily show that an optimal decomposition of linear constraint systems does not need to result in any loss of flexibility using this strong flexibility metric  $flex^*(\cdot)$ :

**Proposition 5.** *Let  $S$  be a linear constraint system  $\mathbf{A} \mathbf{x} \leq \mathbf{b}$  and let  $[\mathbf{y}_1, \dots, \mathbf{y}_k]$  be a partitioning of  $\mathbf{x}$ . Then there exists a decomposition  $\{S_j\}_{j=1}^k$  of  $S$  according to  $[\mathbf{y}_1, \dots, \mathbf{y}_k]$  such that  $flex^*(S) = \sum_{j=1}^k flex^*(S_j)$ .*

*Proof.* Consider a maximum-flexibility decomposition  $\{S_{x_i}\}_{i=1}^n$  induced by the finest partitioning  $\{x_i\}_{i=1}^n$  of  $X$  using the LP (7). For  $i = 1, \dots, n$ , let  $S_{x_i} = (\{x_i\}, C_{x_i})$ . It follows that  $flex^*(S) = \sum_{i=1}^n flex^*(S_{x_i})$ . Since  $\{S_{x_i}\}_{i=1}^n$  is a decomposition, a decomposition induced by  $[\mathbf{y}_1, \dots, \mathbf{y}_k]$  can be obtained by taking  $S_j = (X_j, \bigcup_{x_i \in X_j} C_{x_i})$  for  $j = 1, \dots, k$ , where  $X_j$  is the set of variables occurring in  $\mathbf{y}_j$ . The resulting (strong) flexibility of the decomposed system induced by  $\{X_j\}_{j=1}^k$  now equals

$$\begin{aligned} \sum_{j=1}^k flex^*(S_j) &= \sum_{j=1}^k \sum_{x_i \in X_j} flex^*(S_{x_i}) \\ &= \sum_{i=1}^n flex^*(S_{x_i}) = flex^*(S). \end{aligned}$$

Hence, there exists at least one decomposition realizing a total flexibility equal to the original flexibility  $flex^*(S)$ .  $\square$

This result generalizes a recent result obtained by (Wilson et al. 2013) for STNs to linear constraint systems.

## Conclusion and Discussion

In this paper we concentrated on the decomposition of linear constraint systems. We introduced a generalisation  $flex(\cdot)$  of a flexibility metric proposed for STNs and used this metric to compute a maximum-flexible decomposition of a linear constraint system induced by a partitioning of the set of variables. Using an LP-approach, we finally proposed a strong flexibility metric  $flex^*(\cdot)$  as an alternative to the flexibility metric derived from STNs. We showed that using this strong flexibility metric, an arbitrary decomposition of a linear constraint system can be achieved without any loss of flexibility.

We remark that the ratio  $flex(S)/flex^*(S)$  of these flexibility metrics indicates how much dependencies there exist between the variables  $X$  in  $S$ : whenever this ratio is close to one, the values of variables within the intervals determined by the  $flex$  metric can be chosen almost independently, while a large ratio indicates that choosing a value within an interval can heavily influence the choice of values for other variables. Geometrically,  $flex(S)$  determines the smallest box containing the polyhedron associated with  $S$  while  $flex^*(S)$  determines the largest box contained in the polyhedron determined by  $S$ .

Finally, we would like to remark that LP (7) can also be used outside a decomposition setting: Given an arbitrary LP with objective function  $\max f(\mathbf{x})$  resulting in a maximum  $m$ , we can add an additional constraint in the form of  $f(\mathbf{x}) \geq m$  to the set of constraints. Then applying LP (7) to the resulting system and using a maximizer  $\mathbf{v}^t = [(\mathbf{v}^-)^t, (\mathbf{v}^+)^t]$  to obtain a set of intervals  $\{[v_i^-, v_i^+]\}_{i=1}^n$ , we know which values for the individual variables  $x_i$  do not affect the maximum value  $m$ . This provides a way to obtain flexible solutions to LP-problems.

## Acknowledgments

The authors thank the anonymous reviewers for their constructive remarks and well-considered suggestions for improvement of the paper.

## References

- Agrawal, S.; Deb, S.; Naidu, K. V. M.; and Rastogi, R. 2007. Efficient detection of distributed constraint violations. In *Proceedings of the 23rd International Conference on Data Engineering (ICDE 2007), April 15-20, 2007, Istanbul, Turkey*, 1320–1324. IEEE.
- Alwan, A.; Ibrahim, H.; and Udzir, N. I. 2009. Improved integrity constraints checking in distributed databases by exploiting local checking. *Journal of Computer Science and Technology* 24(4):665–674.
- Boerkoel, J., and Durfee, E. 2012. A distributed approach to summarizing spaces of multi-agent schedules. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI-12)*, 1742–1748. AAAI Press, MenloPark, CA.
- Boerkoel, J., and Durfee, E. 2013. Distributed reasoning for multi-agent simple temporal problems. *Journal of AI Research* 47:95–156.
- Brambilla, A. 2010. *Artificial Intelligence in Space Systems: Coordination Through Problem Decoupling in Multi-Agent Planning for Space Systems*. Lambert Academic Publishing.
- Brodsky, A.; Kerschberg, L.; and Varas, S. 2004. Optimal constraint decomposition for distributed databases. In Maher, M., ed., *Advances in Computer Science - ASIAN 2004. Higher-Level Decision Making*, volume 3321 of *Lecture Notes in Computer Science*, 301–319. Springer Berlin Heidelberg.
- Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal constraint networks. *Artificial Intelligence* 49(1-3):61–95.
- Dechter, R. 2003. *Constraint Processing*. Morgan Kaufmann.
- Gupta, A., and Widom, J. 1993. Local verification of global integrity constraints in distributed databases. *SIGMOD Rec.* 22(2):49–58.
- Hunsberger, L. 2002a. Algorithms for a temporal decoupling problem in multi-agent planning. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI-02)*, 468–475.
- Hunsberger, L. 2002b. *Group Decision Making and Temporal Reasoning*. Ph.D. Dissertation, Harvard University, Cambridge, Massachusetts.
- Planken, L.; de Weerd, M.; and Witteveen, C. 2010. Optimal temporal decoupling in multiagent systems. In VanderHoek; Kaminka; Lesperance; Luck; and Sen., eds., *Proceedings of the Ninth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-10)*, 789–796.
- Policella, N.; Smith, S. F.; Cesta, A.; and Oddi, A. 2004. Generating robust schedules through temporal flexibility. In Zilberstein, S.; Koehler, J.; and Koenig, S., eds., *Proceedings of the 14 International Conference on Automated Planning & Scheduling (ICAPS'04)*, 209–218.
- Policella, N.; Wang, X.; Smith, S.; and Oddi, A. 2005. Exploiting temporal flexibility to obtain high quality schedules. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-05)*, 1199–1204. AAAI Press, Menlo Park, CA.
- Wilson, M.; Klos, T.; Witteveen, C.; and Huisman, B. 2013. Flexibility and decoupling in the Simple Temporal Problem. In Rossi, F., ed., *Proceedings of the 23th International Joint Conference on Artificial Intelligence (IJCAI-2013)*, 2422 – 2428. AAAI Press, Menlo Park, CA.