On Optimal Constraint Decomposition, Monitoring, and Management in Distributed Environments

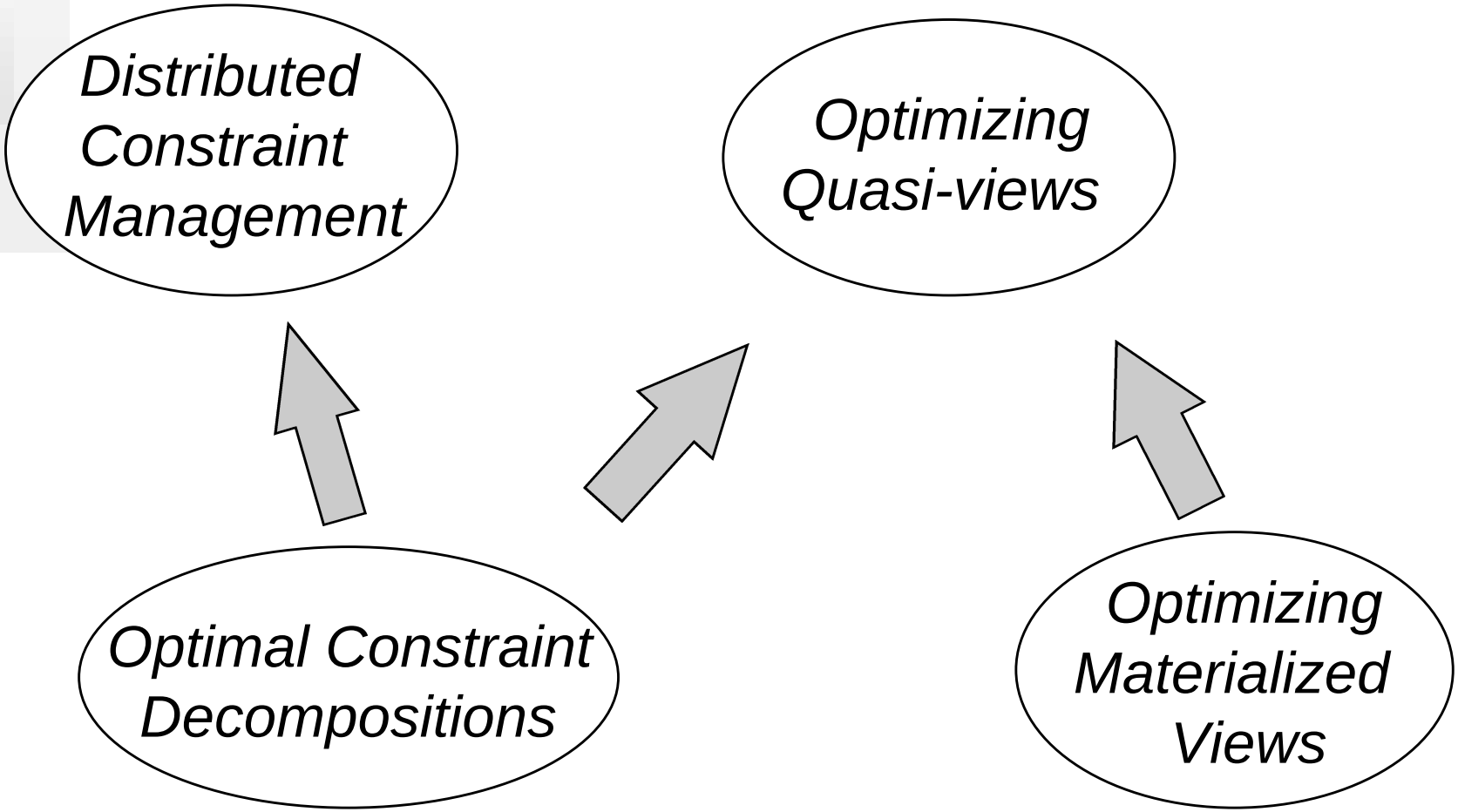
by
Samuel Varas

Advisors Dr. Larry Kerschberg,
Dr. Alexander Brodsky

School of Information Technology and Engineering
George Mason University

Summer 1998

Problem Definition



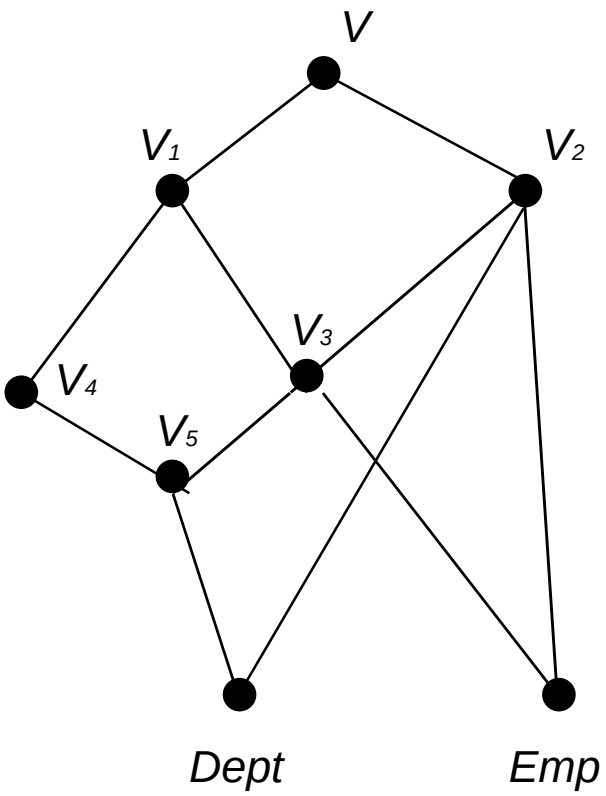
Optimizing Materialized Views

- *Two relations **Dept**(DName, Budget) and **Emp**(EName, DName, Salary).*
- *View (V) defined as follows**

```
SELECT      Dept.DName
FROM        Emp, Dept
WHERE       Dept.DName = Emp.DName
GROUP BY    Dept.DName, Budget
HAVING      SUM(Salary) > Budget
```

- *The salary of an employee is raised, we need to re-compute SUM(salary)*
- *To improve view maintenance, materialize intermediate views*

```
SELECT      Dname, SUM(Salary)
FROM        Emp
GROUP BY    DName
```

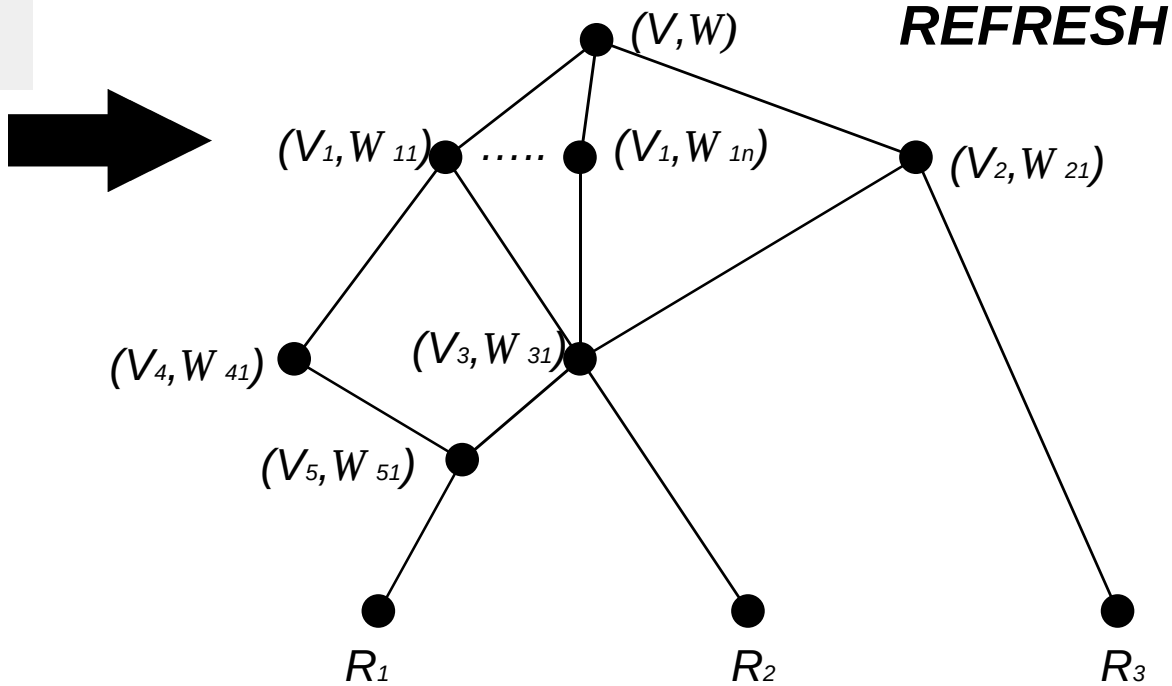


(*) Ross 1996

Optimizing Quasi-views

- $T_1,..,T_m$ be quasi -views or relations names
- $A_1,\ldots,A_n$ be attributes names

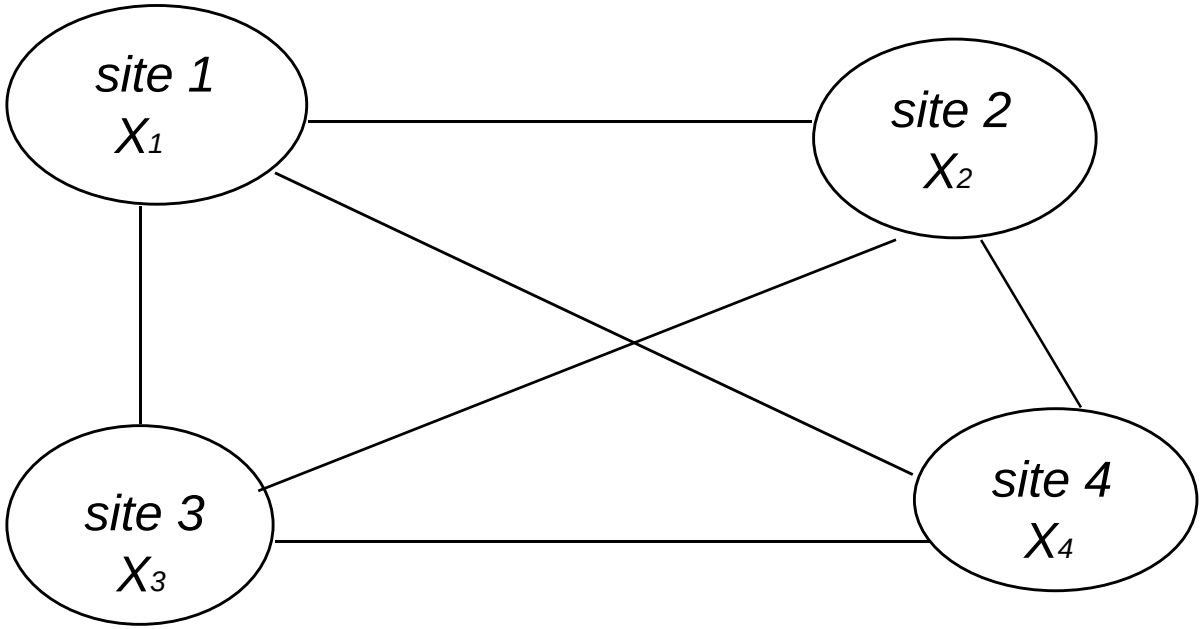
DEFINE QUASI-VIEW *QV* ***AS***
SELECT $A_1,\ldots,A_n$
FROM $T_1,\ldots,T_m$
WHERE *C*
REFRESH WHEN *W*



Distributed Constraint Management

Global Constraint $X_1 + X_2 + X_3 + X_4 \leq 10,$

$X_1, X_2, X_3, X_4 \geq 0$



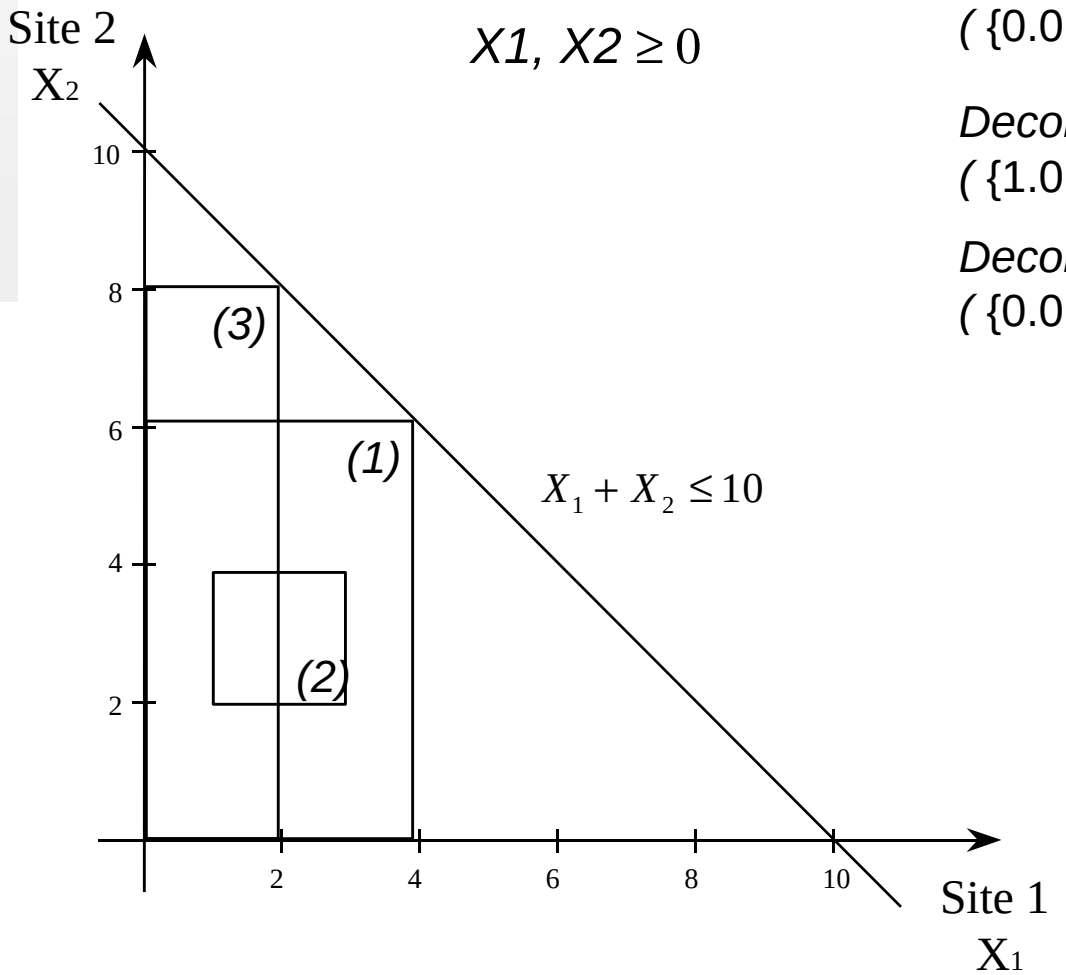
Overview of Contributions

- ***Distributed Constraint Management and Optimal Constraint Decompositions.***
  - *Protocol with guaranteed Consistency, Availability, and Optimality.*
  - *Decomposition Optimization Framework.*
  - *Parametric Mathematical Programming Characterization.*
  - *Algorithms and implementation.*
- ***Optimizing Materialized Views***
  - *Generic optimization framework.*
  - *Complete characterization to represent view evaluation plans.*
  - *Recursive Optimization Problem (Shortest Path Problem).*
  - *Algorithms.*
- ***Quasi-views and their Optimization.***
  - *Quasi-views optimization framework.*
  - *Solution based on concurrent view and constraint optimization problems.*
  - *Special refresh condition case corresponds to constraint decomposition.*

Safe Decompositions for two Sites

W is $X1 + X2 \leq 10$

$X1, X2 \geq 0$



Decomposition (1):
({$0.0 \leq X_1 \leq 4.0$}, {$0.0 \leq X_2 \leq 6.0$})

Decomposition (2):
({$1.0 \leq X_1 \leq 3.0$}, {$2.0 \leq X_2 \leq 4.0$})

Decomposition (3):
({$0.0 \leq X_1 \leq 2.0$}, {$0.0 \leq X_2 \leq 8.0$})

Feasible Decompositions

Each scenario is defined by one or more of the following properties.

- ***Safety of (C_1,…,C_M)**, i.e.,*

$$C_1 \wedge .... \wedge C_M \models \Omega$$

- ***Local Consistency**, i.e., for a given instance y_i, it must satisfy its local constraint C_i, $1 \leq i \leq M$.*
- ***Partial-constraint preservation**, i.e., some constraints are fixed.*
- ***Resource bound partition**, i.e., the global resource bound is partitioned in subset of sites, and each subset cannot increase such bound.*

Optimizing Decompositions

$$\begin{aligned} \text{Max} \quad & f(s) \\ \text{s.t.} \quad & s \in S \end{aligned}$$

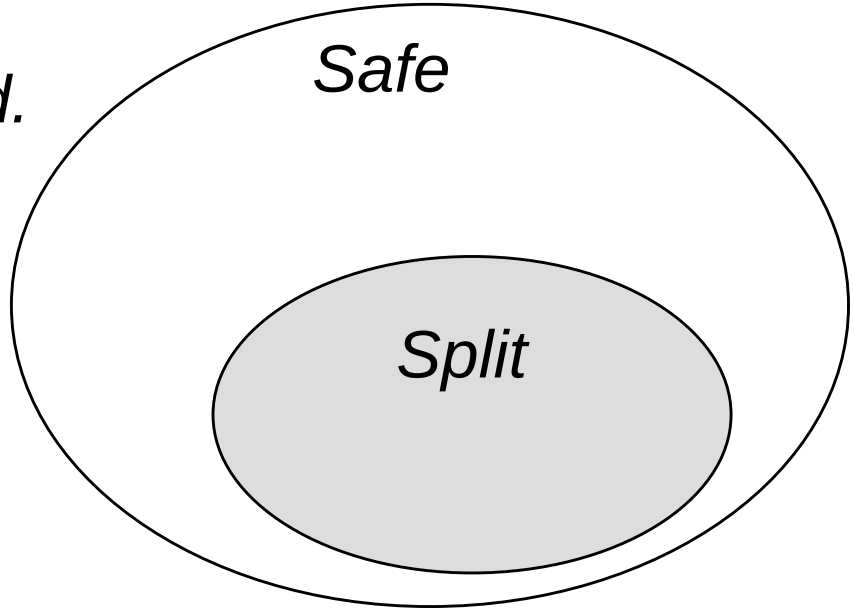
- *W is a global constraint*
- *S the set of FEASIBLE decompositions $C_1,\ldots,C_M$ of W*
- *f: $S \rightarrow R$ an optimization function*
  - *probability of not violating local constraints*
  - *the expected number of updates before an update violates local constraints*
  - *the average of the expected overall cost of manipulations during an update*

Problems in Reduction

- *No FINITE Parametric Characterization of S*
  - *One variable per site => interval characterization*
  - *Two or more variables per site => ?*
- *Concurrent Decompositions: No Full Knowledge of Constraints and Instances*
  - *Partial decomposition (small number of sites)*
  - *How do we guarantee safety, local consistency, and optimality?*
- *How to analytically express optimization criteria in terms of parameters?*

Solutions - Finite Characterization

- *Split:* $\Omega \equiv A\vec{x} \leq \vec{b}$ *is a tuple of constraints* $(A_1\vec{y}_1 \leq \vec{r}_1, \ldots, A_M\vec{y}_M \leq \vec{r}_M)$ *such that* $\sum_{i=1}^{M} \vec{r}_i \leq \vec{b}$
- *Split finitely characterized.*
- *Optimal safe decomposition can ALWAYS be found among SPLITS.*



Decomposition Sets

Safe Decomposition Parametric Characterization

- *W be A x £ b, Ai be the columns of A associated to yi, ri real vectors, and f a monotonic function*
- *Sites K+1,…,M define subset θ.*

***Theorem**. Let S be the set of all safe decompositions of W, and SS the set of all Splits of W. Then*

$$\max\ f(s)\ s.t.\ s \in S\ =\ \max\ f(s)\ s.t.\ s \in SS$$

***Theorem**. Solving the original optimization problem is equivalent to solving the following parametric problem*

$$\max \quad f(\vec{r}_1,....,\vec{r}_M)$$

$$s.t. \quad \sum_{i=1}^{M} \vec{r}_i \le \vec{b}$$

Safety

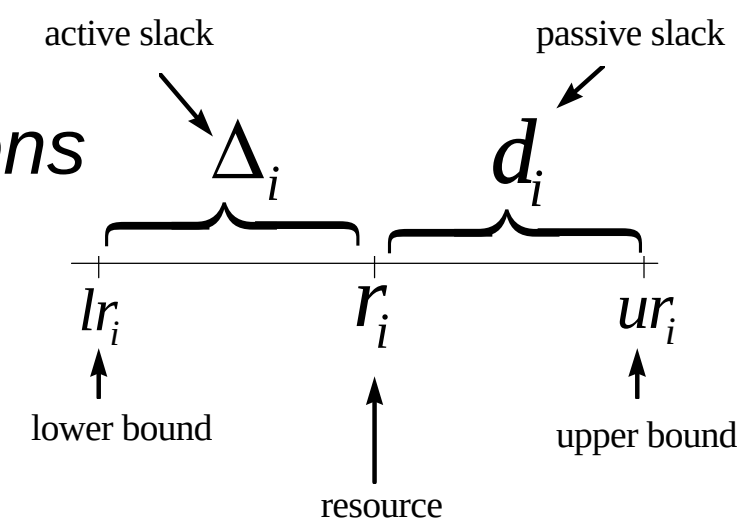
Partial-constraint preservation → $\vec{r}_i = \vec{r}_i^{\,0}, \quad 1 \le i \le k$

Local Consistency → $A_i \vec{y}_i^{\,0} \le \vec{r}_i, \quad k+1 \le i \le M$

Solutions - Resource Distribution

- *Compact Splits* $(A_1\vec{y}_1 \le \vec{r}_1, \ldots, A_M\vec{y}_M \le \vec{r}_M)$ *are represented by resource distributions*

  $((l\vec{r}_1, \vec{r}_1, u\vec{r}_1), \ldots, (l\vec{r}_M, \vec{r}_M, u\vec{r}_M))$
- *Compact representation*
- *Autonomous decompositions*
- *Optimality (by a subsets and all sites)*



Resource Distribution

- $A\vec{x} \le \vec{b}$ *be a global constraint,* f *a monotonic function and* f_q *be its q-localizer,* $D(\vec{r}_1^{\,0},\ldots,\vec{r}_k^{\,0})$ *be a partial compact split, and* $\vec{B}_q$ *be a resource partition,*
- *If* $(\vec{r}_{k+1}^{\,'},\ldots,\vec{r}_M^{\,'})$ *is a solution of*

$$\begin{aligned} &\max \quad f_{è}(\vec{r}_{k+1},\ldots,\vec{r}_M) \\ &s.t. \quad \vec{r}_q \le \vec{B}_q \\ &\qquad l\,\vec{r}_i \le \vec{r}_i, \; k+1 \le i \le M \end{aligned}$$

- *Then,* $(\vec{r}_1^{\,0},\ldots,\vec{r}_k^{\,0},\vec{r}_{k+1}^{\,'},\ldots,\vec{r}_M^{\,'})$ *is a solution of*

$$\begin{aligned} &\max \quad f(\vec{r}_1,\ldots,\vec{r}_M) \\ &s.t. \quad \vec{r}_q \le \vec{B}_q \\ &\qquad l\vec{r}_i \le \vec{r}_i, \; k+1 \le i \le M \\ &\qquad \vec{r}_i = \vec{r}_i^{\,0}, \; 1 \le i \le k \end{aligned}$$

Solutions - Objective Function Characterization

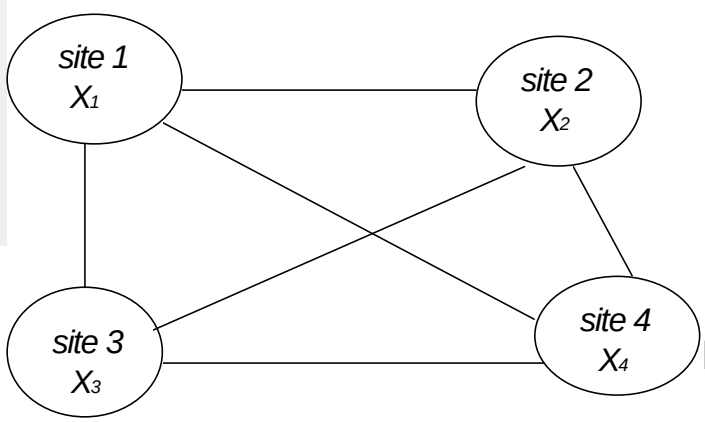
- *Assumption: Database instances are uniformly distributed over the space defined by* W*, i.e., each database instance on* W *has the same probability of occurring.*
- *Consider* ***f*** *as maximizing the probability of not violating local constraints* **.**
- *Then,* ***f*** *is proportional to the* ***volume*** *of a split (parametric characterization),* ***f*** *is a monotonic function, and* ***f*** *can be expressed in terms of a subset* θ *of sites (*θ*–localizer).*

Algorithm and Experiments

- *The optimization problem has linear constraints, and a non-linear objective function (based on volume representation).*
- *For design-time and the individual variable partition case we use the Frank-Wolfe algorithm (concave objective function).*
- *For the general variable partition case, we use a local search algorithm, where for each local search we use the Frank-Wolfe algorithm.*

Distributed Protocol Assumptions

Global Constraint $X_1 + X_2 + X_3 + X_4 \leq 10$,
$X_1, X_2, X_3, X_4 \geq 0$



- *M distributed sites, each one has a local transaction manager.*
- *Global constraint* $A\vec{x} \leq \vec{b}$ *each site maintains a triple* $(l\vec{r}_i, \vec{r}_i, u\vec{r}_i)$
- *If an update violates a local constraint, then a coordinator is selected. It tries to create a new resource distribution in a (small) subset of sites, checking for*

$$1.\ l\vec{r}_q \leq u\vec{r}_q$$
$$2.\ l\vec{r}_{q*} \leq u\vec{r}_{q*}$$

- *To move resources, the primitive TRANSFER-RESOURCE runs under a distributed transaction manager*
- *Failure Model: both sites and communication links may fail.*

CSOL - Properties

- *Consistency*
- *Decomposition Safety*
- *Last-resort Update Refusal*
- *Optimality*

***Theorem**: Any protocol that uses exclusively the RESOURCE-TRANSFER primitive for exchanging resources, is guaranteed to satisfy CSOL-properties.*

Related Work on Distributed Constraint Management

- *Local Verification of Global Constraints*
  - *Garcia-Molina, Gupta, Qian, Simon, Grufman, Huyn.*
  - *Barbará, Soparkar, Mazumdar*
- *Local Verification on Numerical Constraints*
  - *D. Barbará and H. Garcia-Molina, 1992.*
  - *N. Soparkar and A. Silberschatz, 1990.*
  - *S. Mazumdar and Z. Yuan, 1998.*
- *Our work can significantly extend previous work because:*
  - *General linear constraints.*
  - *Optimization-based framework.*
  - *Different decomposition scenarios.*
  - *Compact local constraint representation.*

Related Work on Quasi-View Optimization

- *Quasi-copies, R. Alonso, 1990.*
- *Quasi-views, L. Seligman, 1994, 1997.*

Conclusions- Constraint Management

- *We have presented a generic and powerful domain-independent framework for finding optimal decompositions for a range of design and update-time scenarios.*
- *A comprehensive solution for the case of general linear constraints.*
  - *Reducing the problem to mathematical programming.*
  - *Developing algorithms for it.*
  - *Providing a distributed protocol to manage local updates and concurrent distributed constraint decompositions.*

Conclusions

- *Optimizing Materialized Views*
  - *Characterization of View Decomposition using Expression-DAGs.*
  - *The optimization problem can be represented as finding the Shortest Path in an expression-DAG.*
  - *Linear time Algorithm for a Expression-DAG, when a complete path is materialized.*
- *Optimizing Quasi-views*
  - *We extend the quasi-view concept to multiple DBs,*
  - *Optimization Framework in terms of view and constraint decomposition.*
  - *Framework is useful for both distributed and non distributed environments.*

Future Work

- ***Constraint Decomposition*** *(new types of constraints, replicated variables, different types of objective functions)*
- ***Distributed Protocol*** *(architecture instantiations, level of service)*
- ***Materialized views*** *(special classes of problems).*
- ***Quasi views*** *(different types of refresh conditions, and different models of level of service).*