Parallel Asynchronous Matrix Completion

Inderjit S. Dhillon
Computer Science & Mathematics
UT Austin

Joint work with

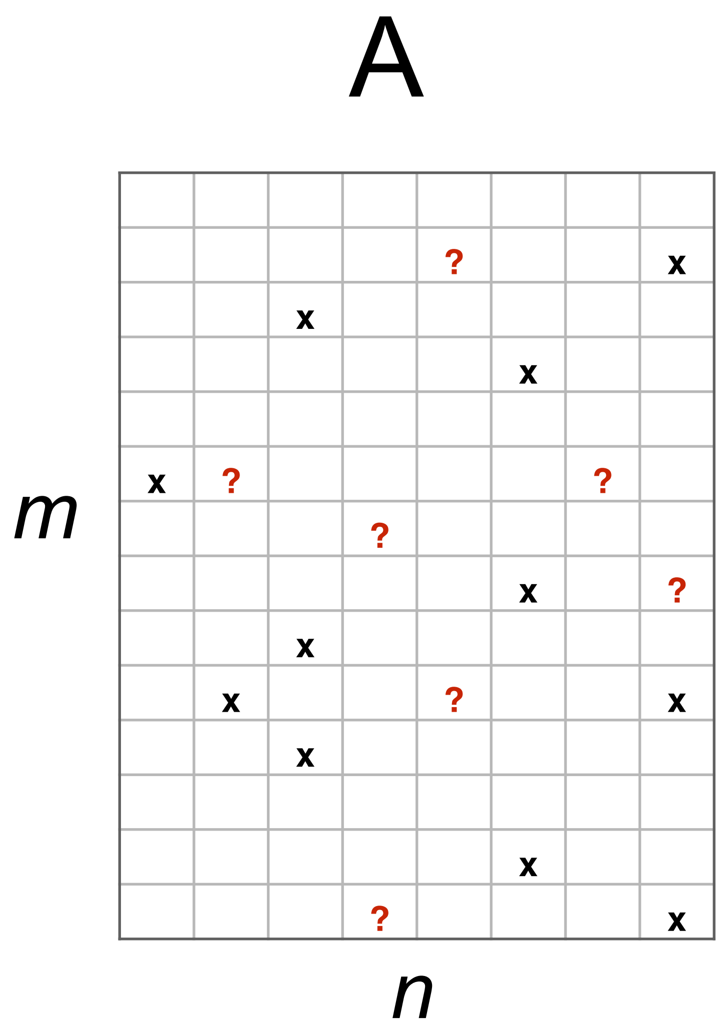
Cho-Jui Hsieh, Hsiang-Fu Yu, Hyokun Yun, S.V.N. Vishwanathan

Householder Symposium
Spa, Belgium
June 10, 2014

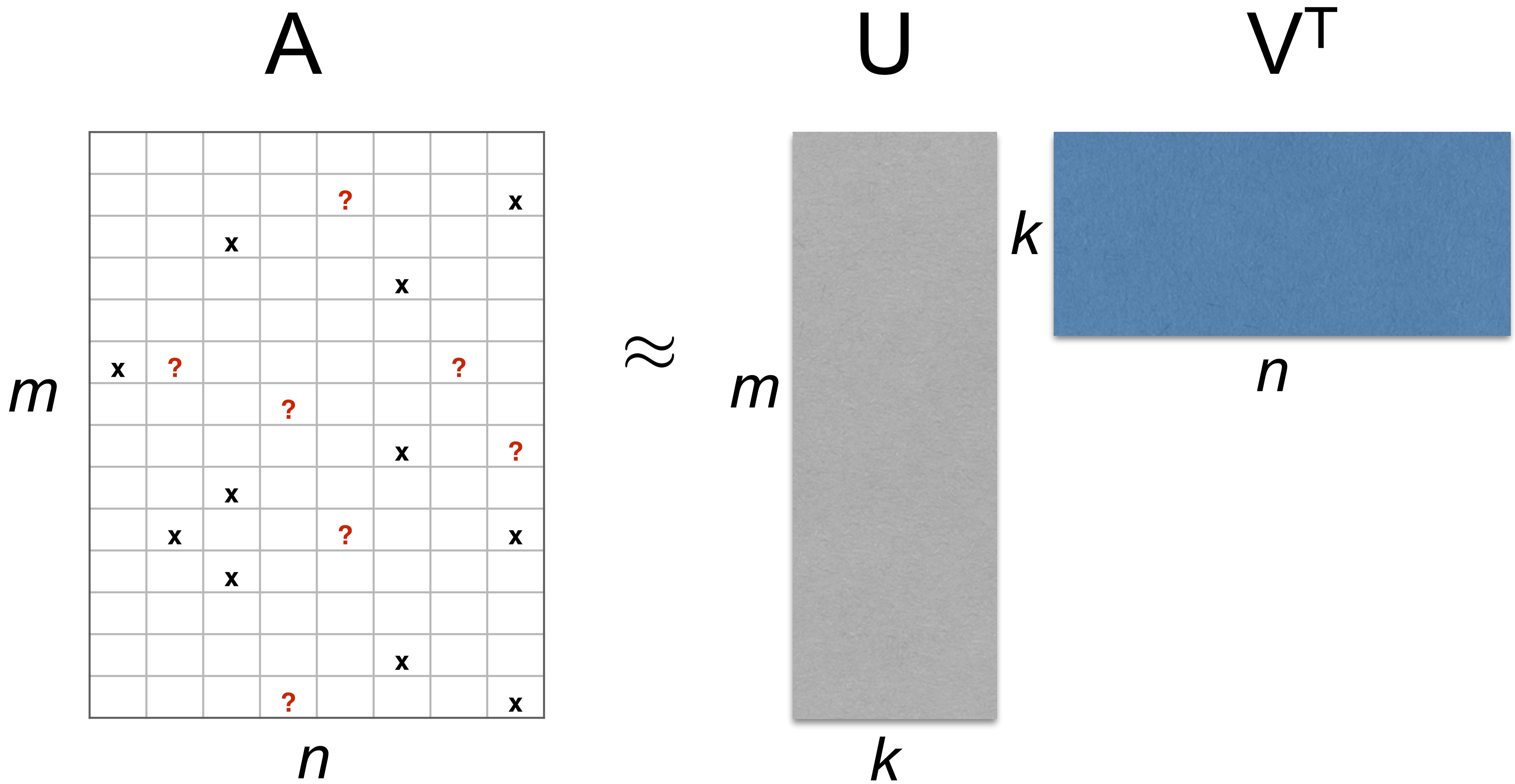
The $1M Netflix Prize [2006]

$1M was won in 2009

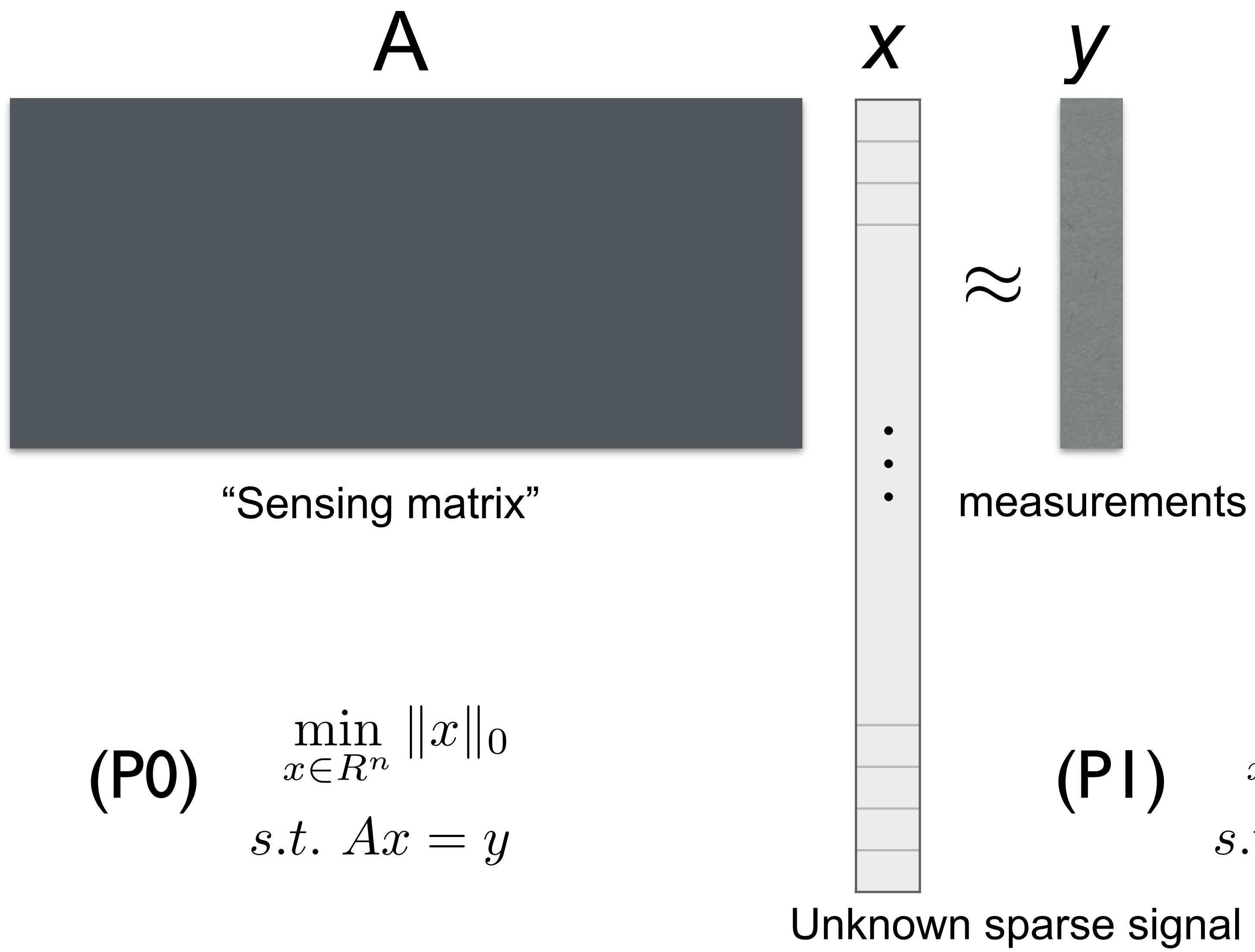
Missing Value Estimation



Low-rank Matrix Completion



Compressed Sensing



(P0) $\min_{x \in R^n} \|x\|_0$ $s.t.\ Ax = y$

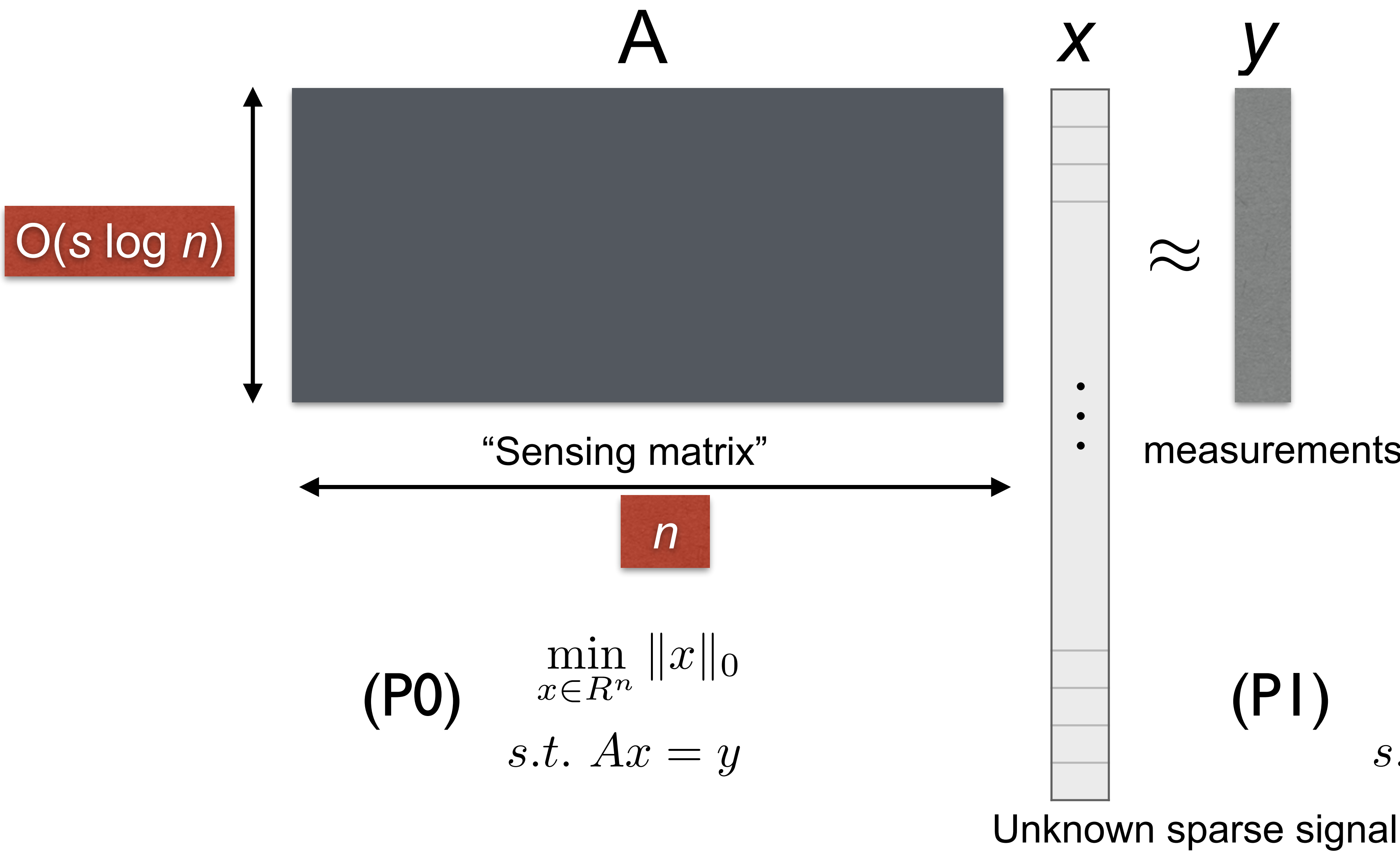
(P1) $\min_{x \in R^n} \|x\|_1$ $s.t.\ Ax = y$

[Candes,Tao 2005] If *A* obeys the *restricted isometry property*:

$$(1-\delta_s)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1+\delta_s)\|x\|_2^2, \quad \forall x : \|x\|_0 = s$$

for some constant δ_s and the solution x_0 to problem (P0) is *s*-sparse, then x_0 is the unique solution to the convex relaxation (P1) and can be *exactly* recovered by solving a linear program.

Compressed Sensing



(P0) $\min_{x \in R^n} \|x\|_0 \quad s.t.\ Ax = y$

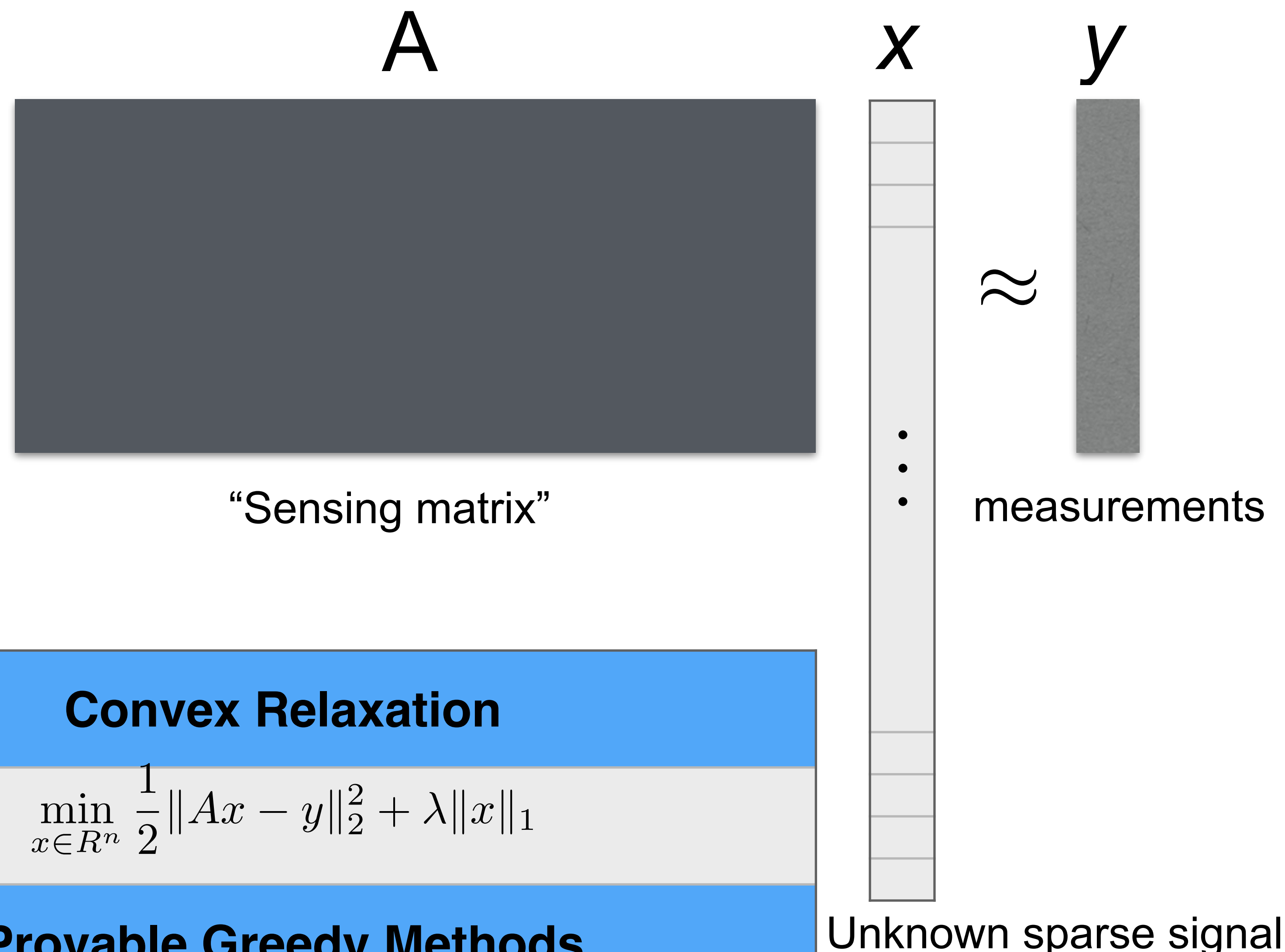
(P1) $\min_{x \in R^n} \|x\|_1 \quad s.t.\ Ax = y$

[Candes,Tao 2005] If *A* obeys the *restricted isometry property*:

$$(1-\delta_s)\|x\|_2^2 \le \|Ax\|_2^2 \le (1+\delta_s)\|x\|_2^2, \quad \forall x : \|x\|_0 = s$$

for some constant δ_s and the solution x_0 to problem (P0) is *s*-sparse, then x_0 is the unique solution to the convex relaxation (P1) and can be *exactly* recovered by solving a linear program.

Compressed Sensing



Convex Relaxation
$$\min_{x \in R^n} \frac{1}{2}\|Ax - y\|_2^2 + \lambda \|x\|_1$$
Provable Greedy Methods
[Pati,Rezaiifar,Krishnaprasad1993] *Orthogonal Matching Pursuit* (OMP)
[Needell, Tropp 2009] *Compressive Sampling Matching Pursuit* (CoSaMP)
[Jain,Tewari,Dhillon 2011] *Orthogonal Matching Pursuit with Replacement* (OMPR)

Linear Sensing of Low-Rank Matrices

(P0) $$\min_{X \in \mathbb{R}^{m \times n}} \ \text{rank}\,(X) \quad s.t.\ \mathcal{A}(X) = b$$

(P1) $$\min_{X \in \mathbb{R}^{m \times n}} \ \|X\|_* \quad s.t.\ \mathcal{A}(X) = b$$

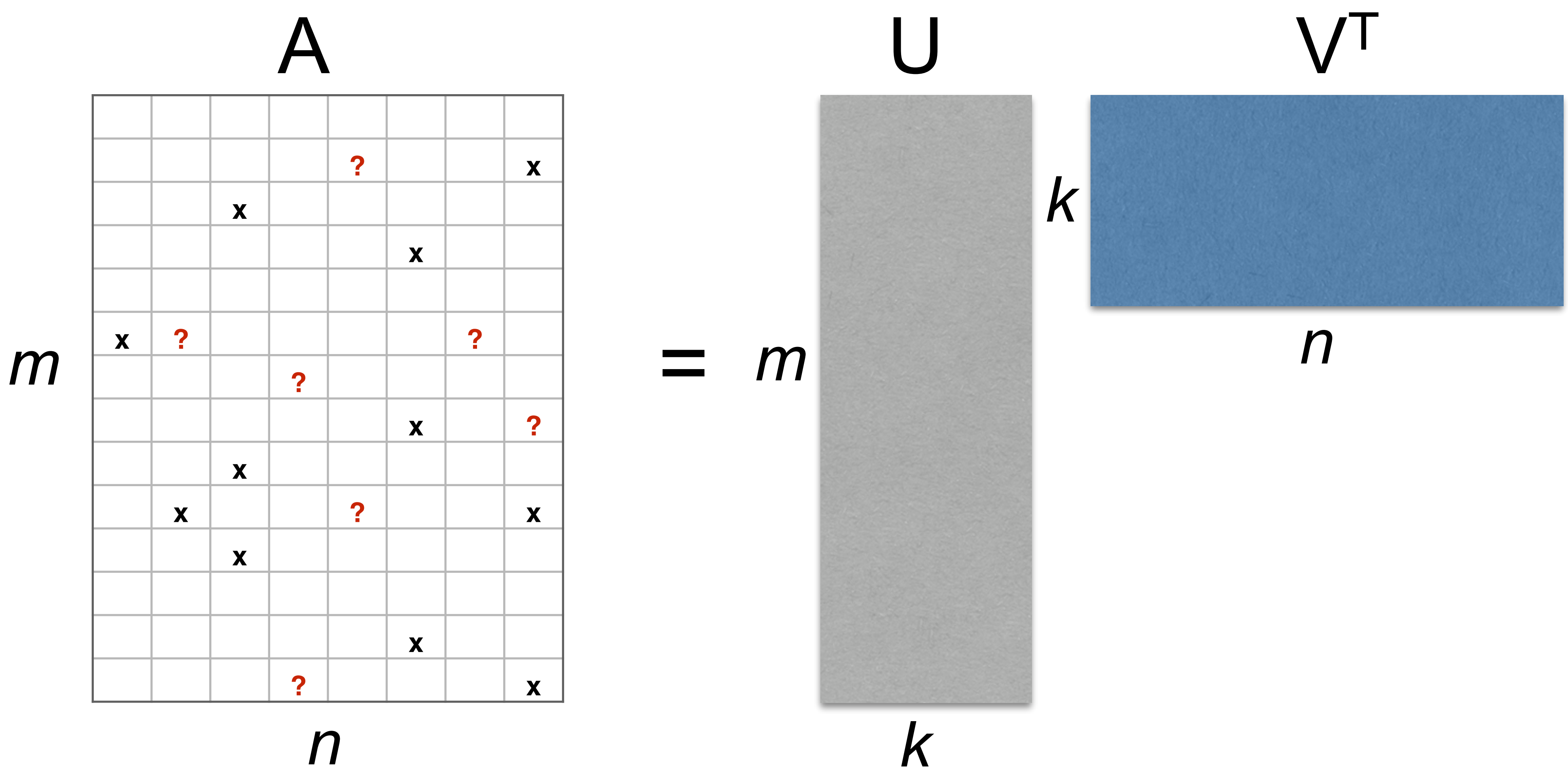
- Compressed sensing guarantees can be extended to low-rank matrix recovery problem
- Recovery depends on a *generalized Restricted Isometry Property* for linear maps:

[Recht,Fazel,Parrilo 2010] If $\mathcal{A}$ obeys the *restricted isometry property*:

$$(1 - \delta_r(\mathcal{A}))\|X\|_F \leq \|\mathcal{A}(X)\| \leq (1 + \delta_r(\mathcal{A}))\|X\|_F, \quad \forall X : \text{rank}(X) \leq r$$

for some constant δ_r and the solution X_0 to (P0) has rank r, then X_0 is the unique solution to the convex relaxation (P1).

Low-rank Matrix Completion

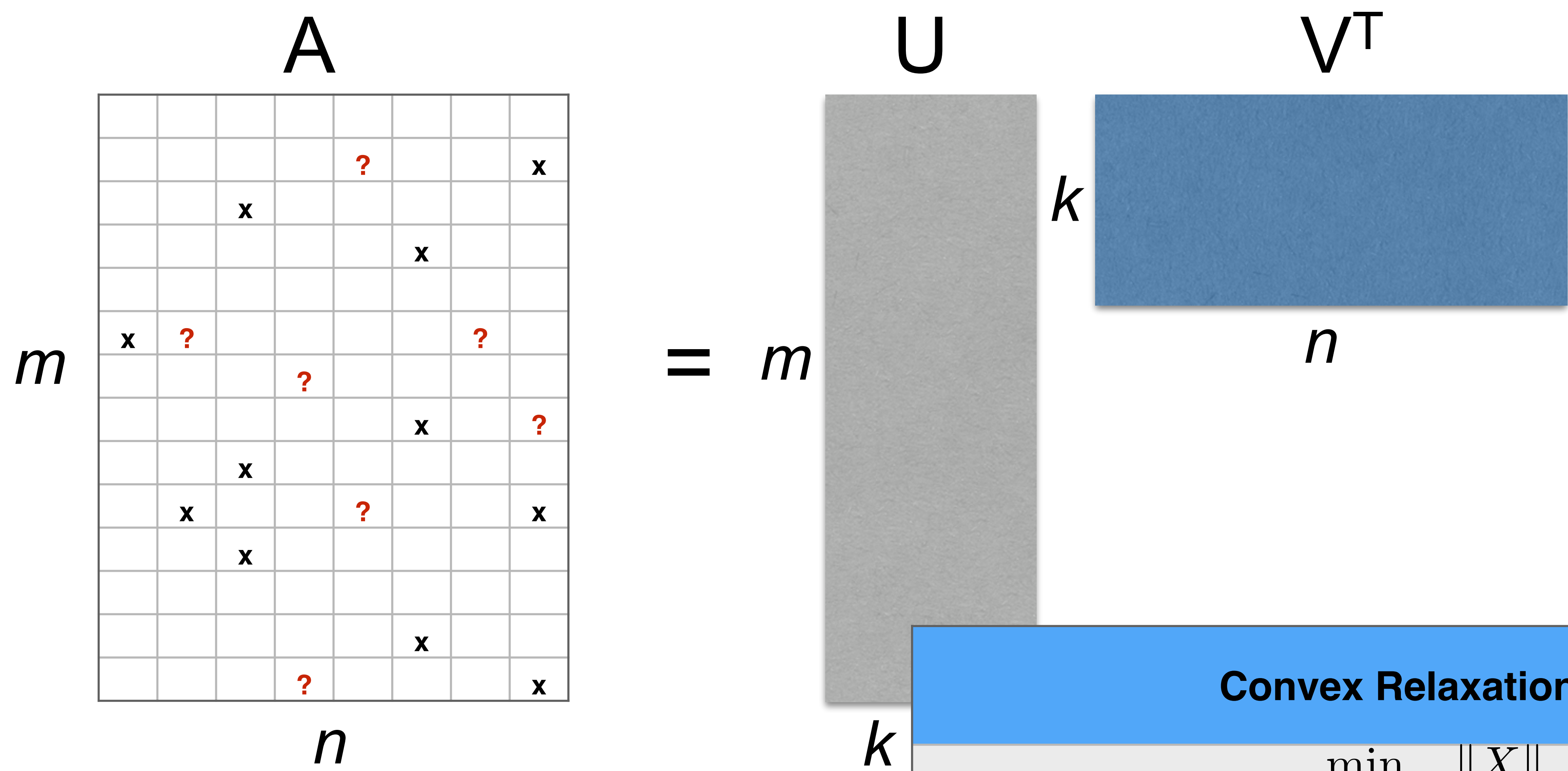


[Candes,Recht 2008] Nuclear norm minimization perfectly recovers most low-rank *n x n* matrices of rank *r*, if $O(n^{1.2} r \log n)$ entries (sampled uniformly at random) are observed.

[Recht 2009] Near-optimal sample complexity: $O(\max\{\mu_1, \mu_0\} nr \log^2 2n)$

[Gross 2009] A novel analysis yielding sample complexity $O(nr\nu \log^2 n)$

Low-rank Matrix Completion



Convex Relaxation

$$\min_{X \in \mathbb{R}^{m \times n}} \|X\|_*$$

$$\text{s.t.} \;\; X_{ij} = M_{ij}, (i,j) \in \Omega$$

Provable Greedy Methods

[Jain,Meka,Dhillon 2009]
Singular Value Projection (SVP)
[Jain,Netrapalli,Sanghavi 2013]
Alternating Least Squares (ALS)

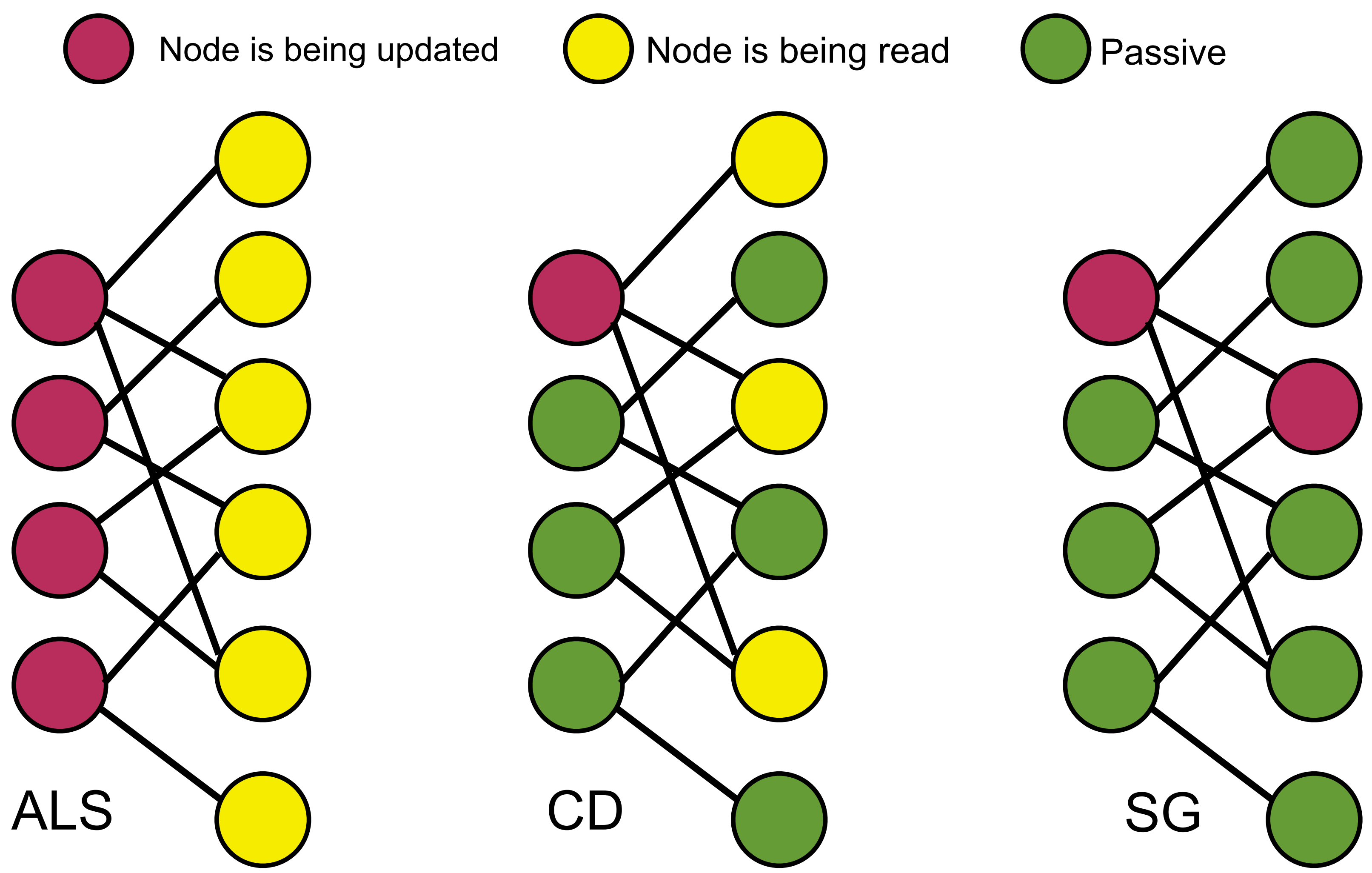
Low-rank Matrix Completion

$$\min_{U \in \mathbb{R}^{m \times k}, V \in \mathbb{R}^{n \times k}} \|W \odot (A - UV^T)\|_F^2 + \lambda(\|U\|_F^2 + \|V\|_F^2)$$

- How do we solve in practice?
- Typical time complexity for iterative procedure: $O(|\Omega|k^2 + (m+n)k^3)$
- Domains where recommender systems are used — Millions of rows and columns, and billions of observed samples
- Big Data — How do we scale to very large matrices?

Matrix Completion Algorithms

- Alternating Least Squares (ALS): Fix W and solve for H (and vice versa)
- Coordinate Descent (CD): Choose a direction (1 through k) and update the corresponding feature in W and H
- Stochastic Gradient (SG): Sample an observation (i,j) at random and update the corresponding latent factors in W and H



Stochastic Gradient

- Well-suited for data arriving in an incremental fashion
- Continually update model parameters as data arrives
- Consider:

$$\min_{w \in \mathcal{W}} F(w) = E_Z[f(w;Z)]$$

- Idea: At each step *t* , obtain a gradient estimate at current parameter w_t such that it is "unbiased":

$$E_Z[\hat{g}_t] = \nabla_w F(w_t)$$

- Update $w_{t+1} = \Pi_{\mathcal{W}}(w_t - \eta_t \hat{g}_t)$

[Nemirovski 2009] If F is strongly convex and smooth w.r.t. *w,* and $E[\|\hat{g}_t\|^2]$ is bounded, then:

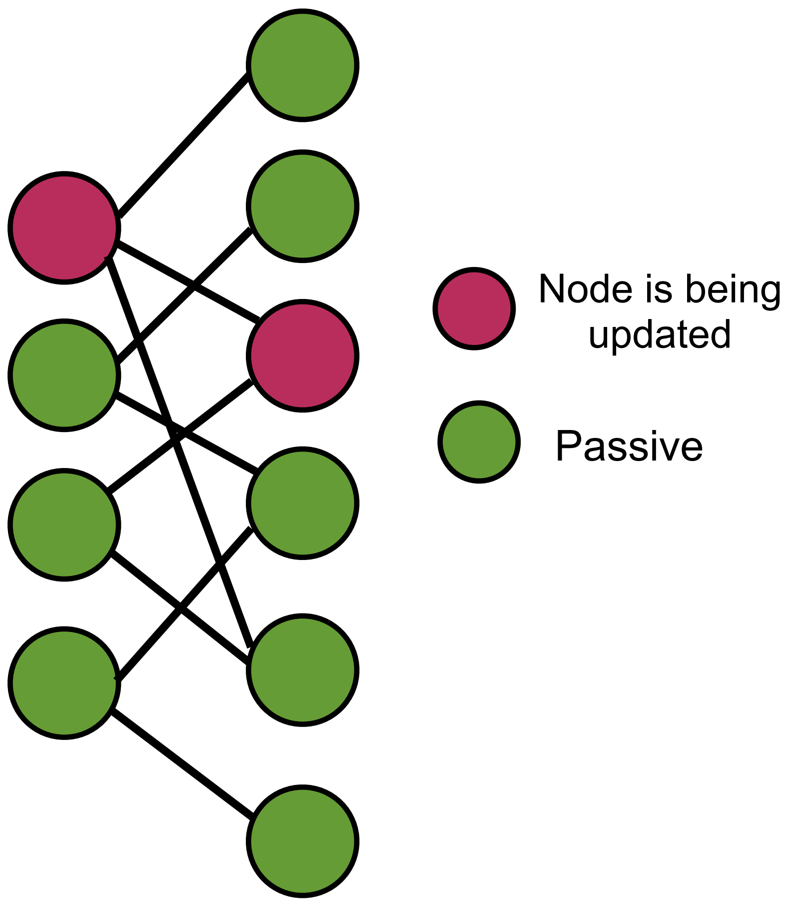
$$E[F(w_t) - F(w^*)] = O\left(\frac{1}{t}\right)$$

Stochastic Gradient: Matrix Completion

- Sample random rating (*i,j*) and update corresponding latent factors:

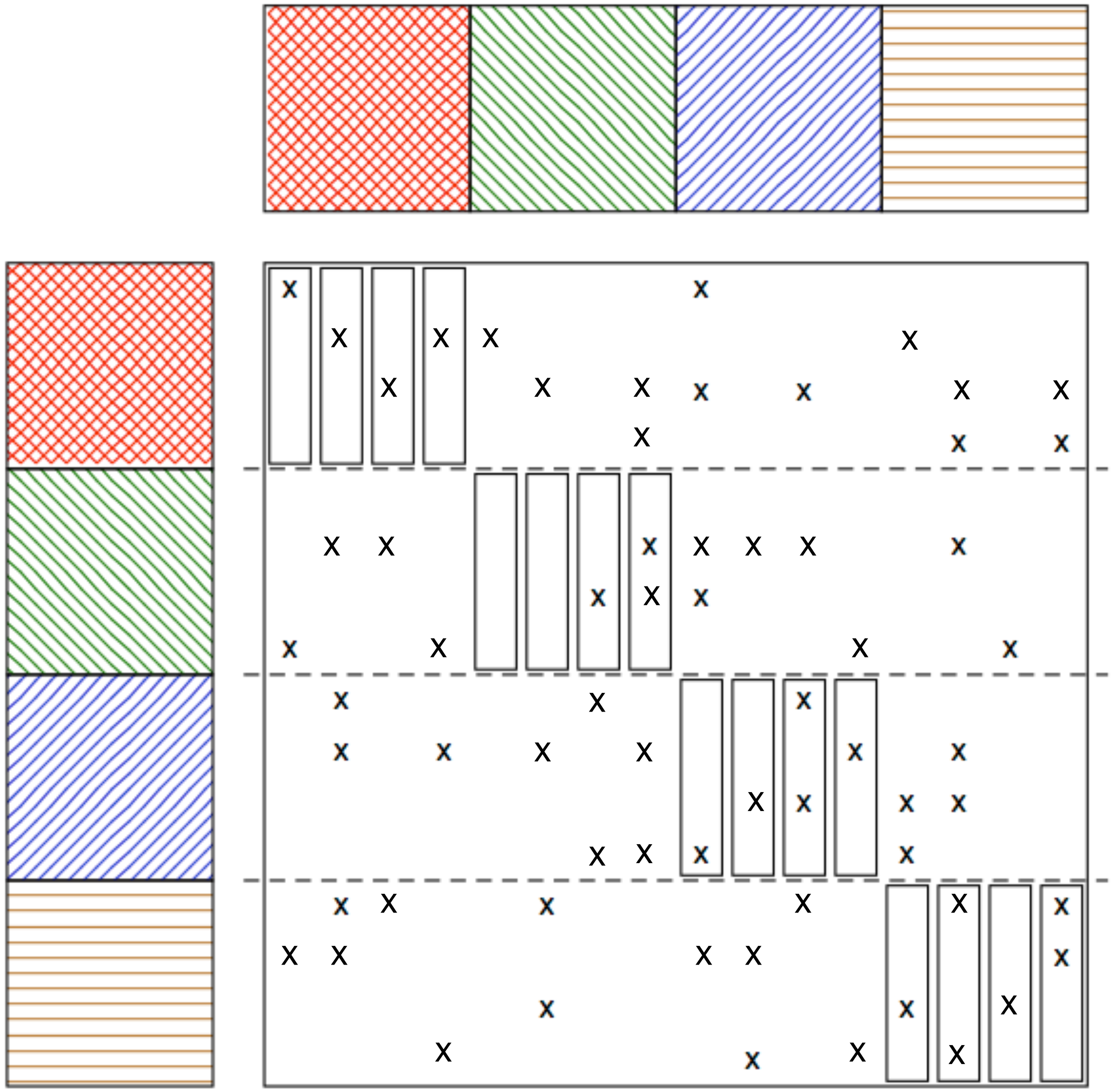
$$\mathbf{u}_i \leftarrow \mathbf{u}_i - \eta(A_{ij} - \langle \mathbf{u}_i, \mathbf{v}_j \rangle \mathbf{v}_j + \lambda \mathbf{u}_i)$$
$$\mathbf{v}_j \leftarrow \mathbf{v}_j - \eta(A_{ij} - \langle \mathbf{u}_i, \mathbf{v}_j \rangle \mathbf{u}_i + \lambda \mathbf{v}_j)$$

- Time per update $O(k)$
- Effective for very large-scale problems



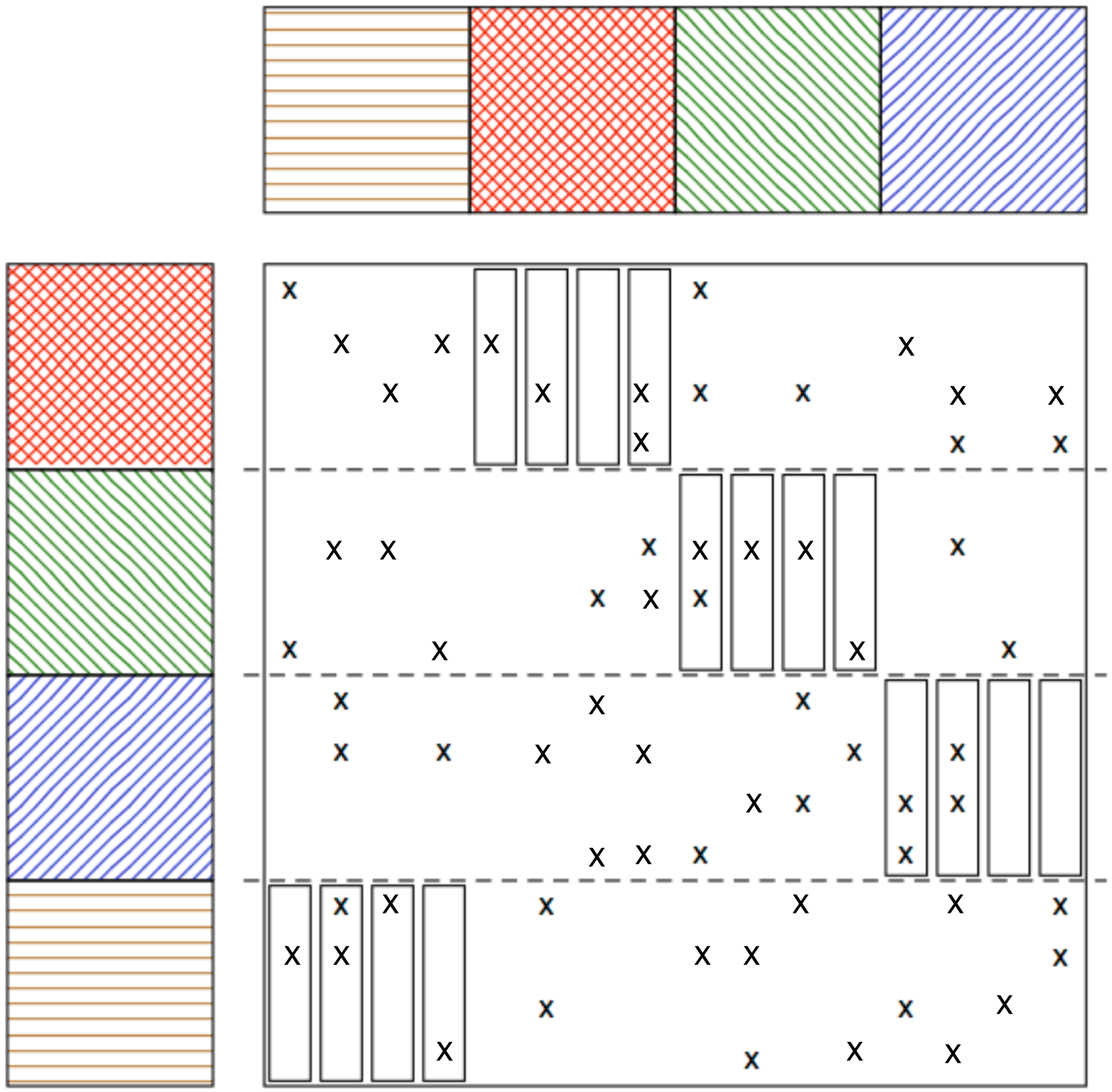
Distributed Stochastic Gradient Descent

- Decoupled updates — easy to parallelize [Gemulla et al 2011]
- But communication & computation are done in sequence



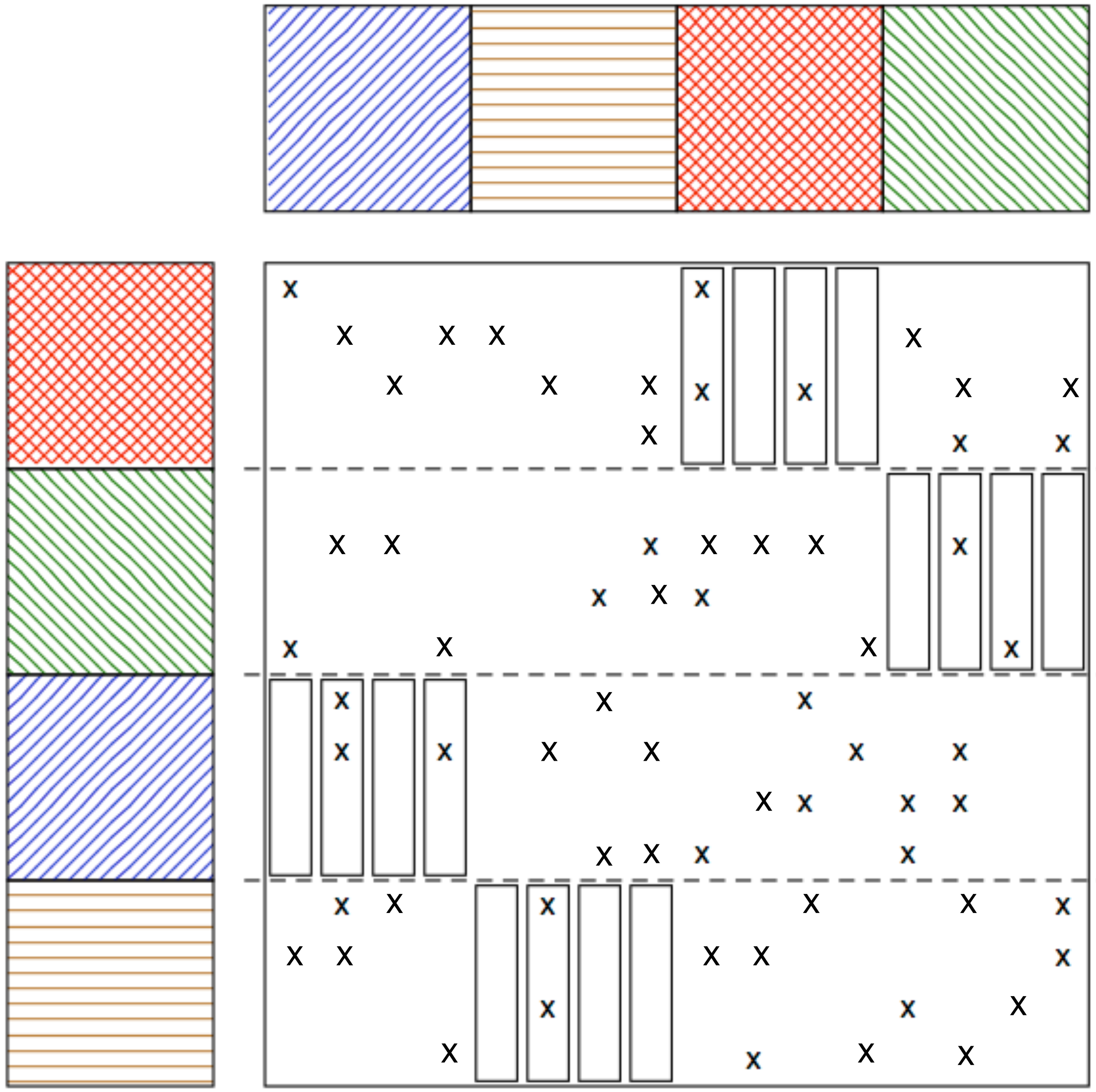
Distributed Stochastic Gradient Descent

- Decoupled updates — easy to parallelize [Gemulla et al 2011]
- But communication & computation are done in sequence



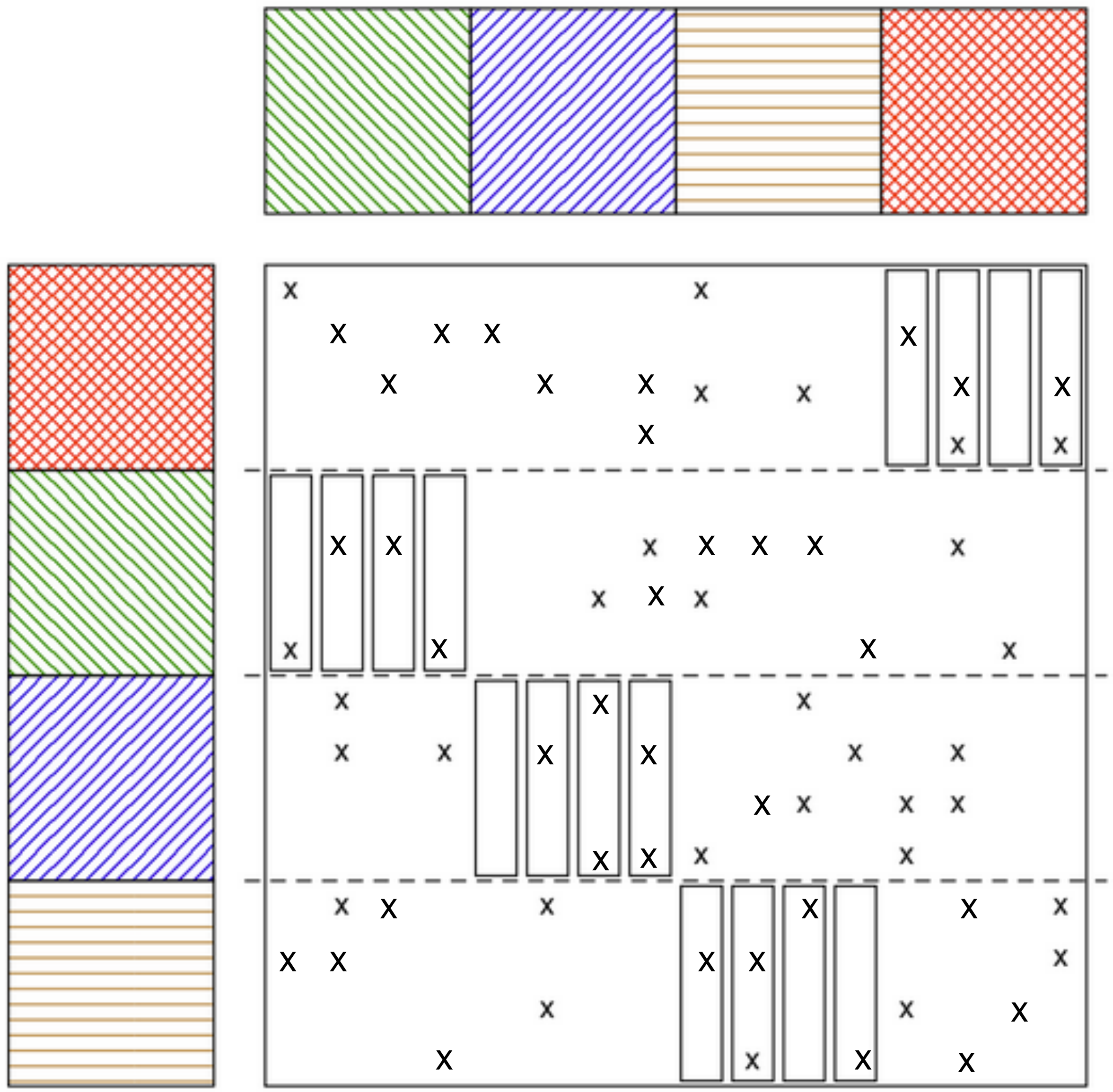
Distributed Stochastic Gradient Descent

- Decoupled updates — easy to parallelize [Gemulla et al 2011]
- But communication & computation are done in sequence



Distributed Stochastic Gradient Descent

- Decoupled updates — easy to parallelize [Gemulla et al 2011]
- But communication & computation are done in sequence



Our solution: NOMAD

- **Goal**: Keep CPU & network simultaneously busy.

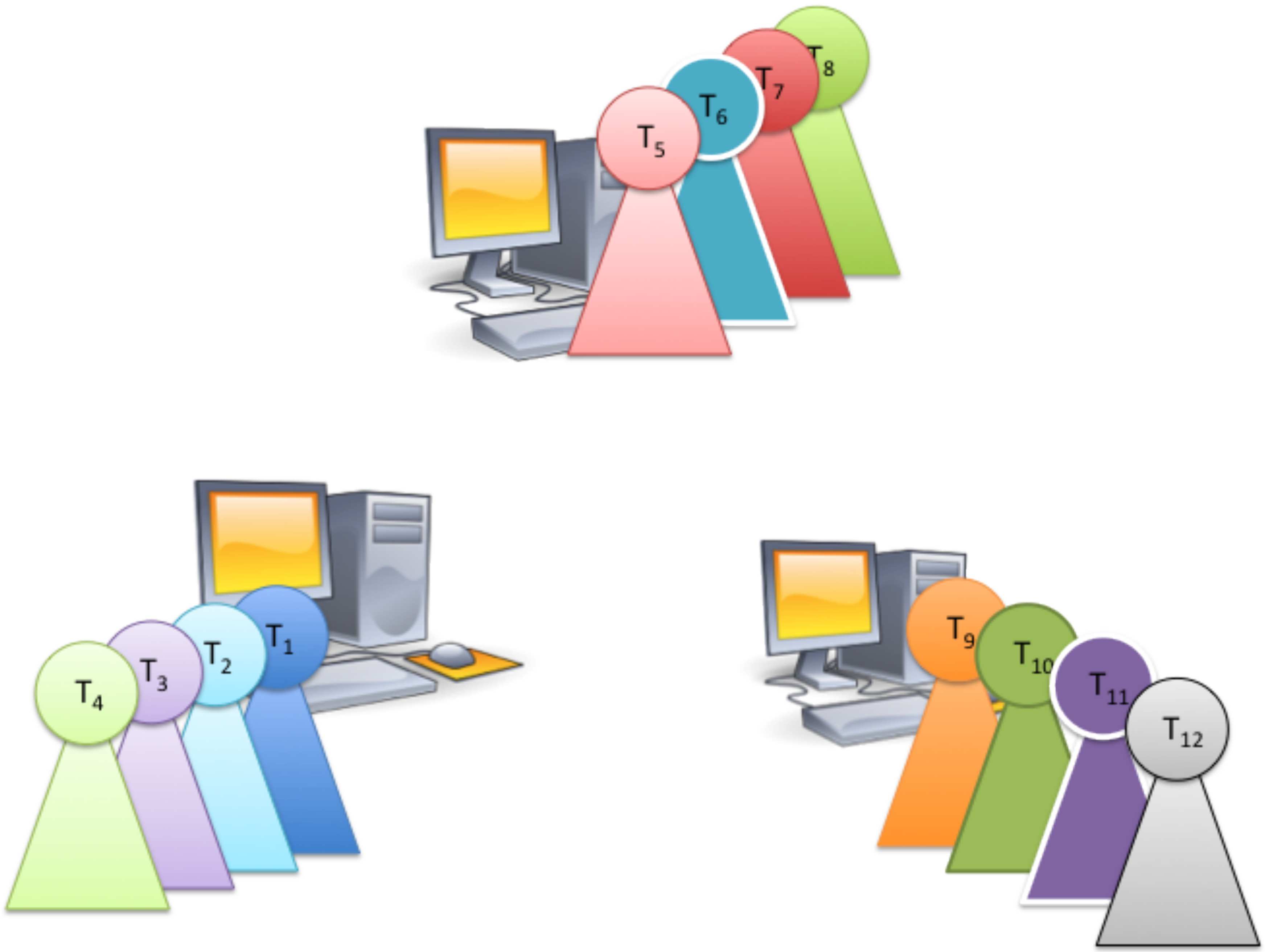
> Non-locking
> stOchastic
> Multi-machine algorithm for
> Asynchronous &
> Decentralized matrix factorization

- Stochastic gradient update rule: only a small set of variables involved
- *Nomadic token passing*: avoid conflict without explicit remote locking
- Fully asynchronous computation!

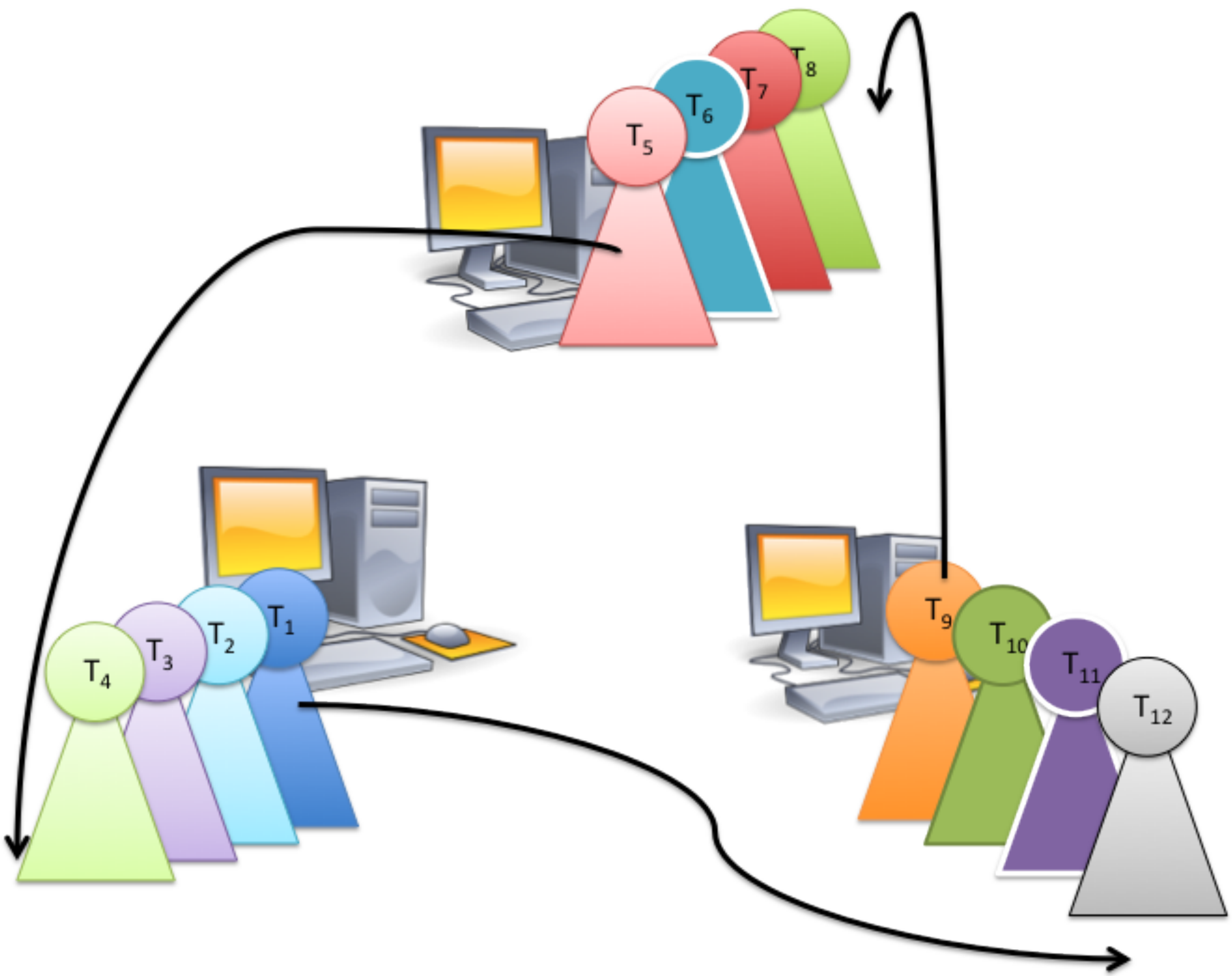
H. Yun, H.-F. Yu, C.-J. Hsieh, S. V. N. Vishwanathan, I. S. Dhillon. NOMAD: Non-locking, stOchastic Multi-machine algorithm for Asynchronous and Decentralized matrix completion. To appear in *Proceedings of the VLDB Endowment*, 2014.

Nomadic Token Passing

Nomadic Token Passing



Nomadic Token Passing

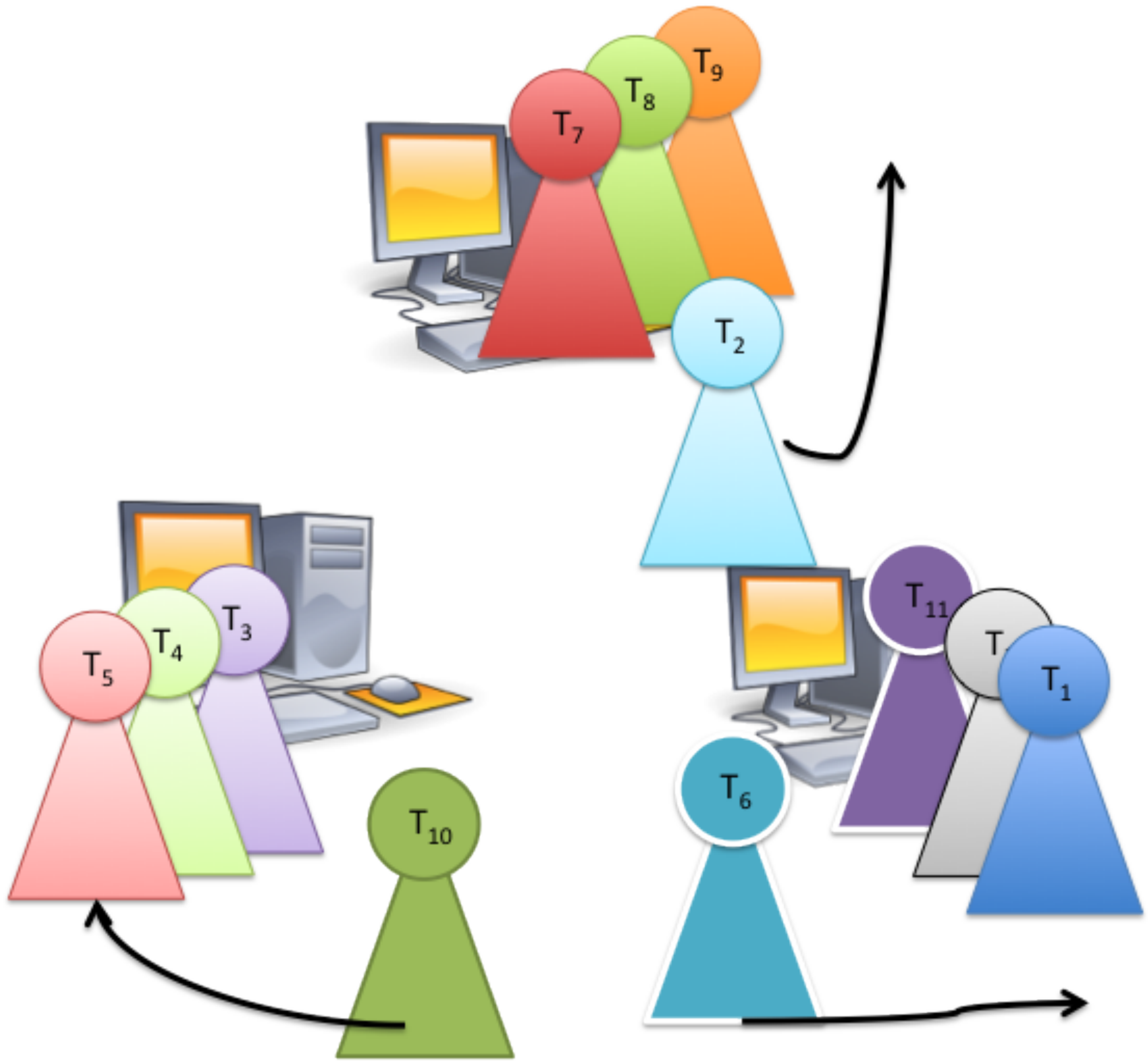


Nomadic Token Passing

Nomadic Token Passing

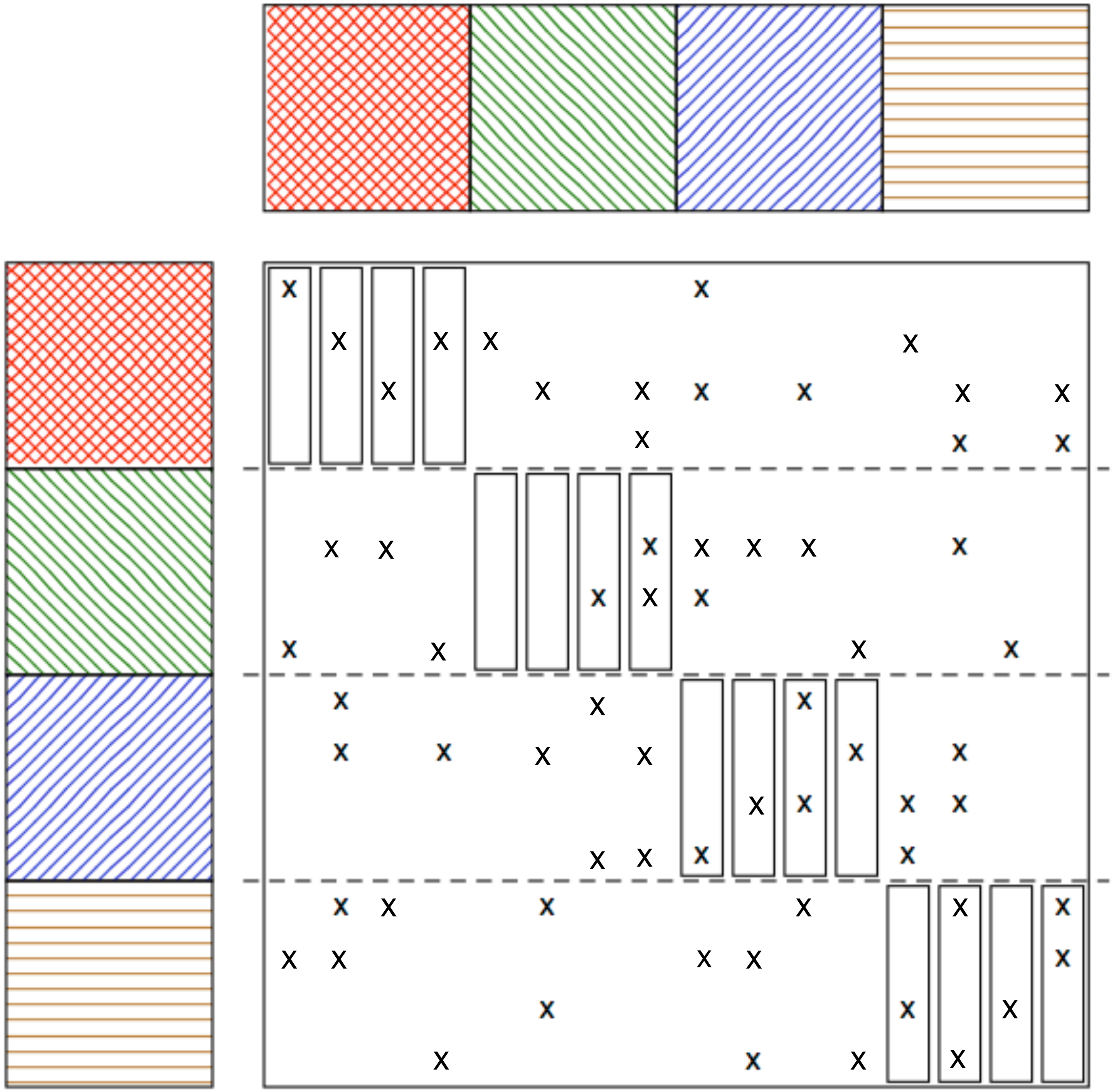
Nomadic Token Passing

Nomadic Token Passing

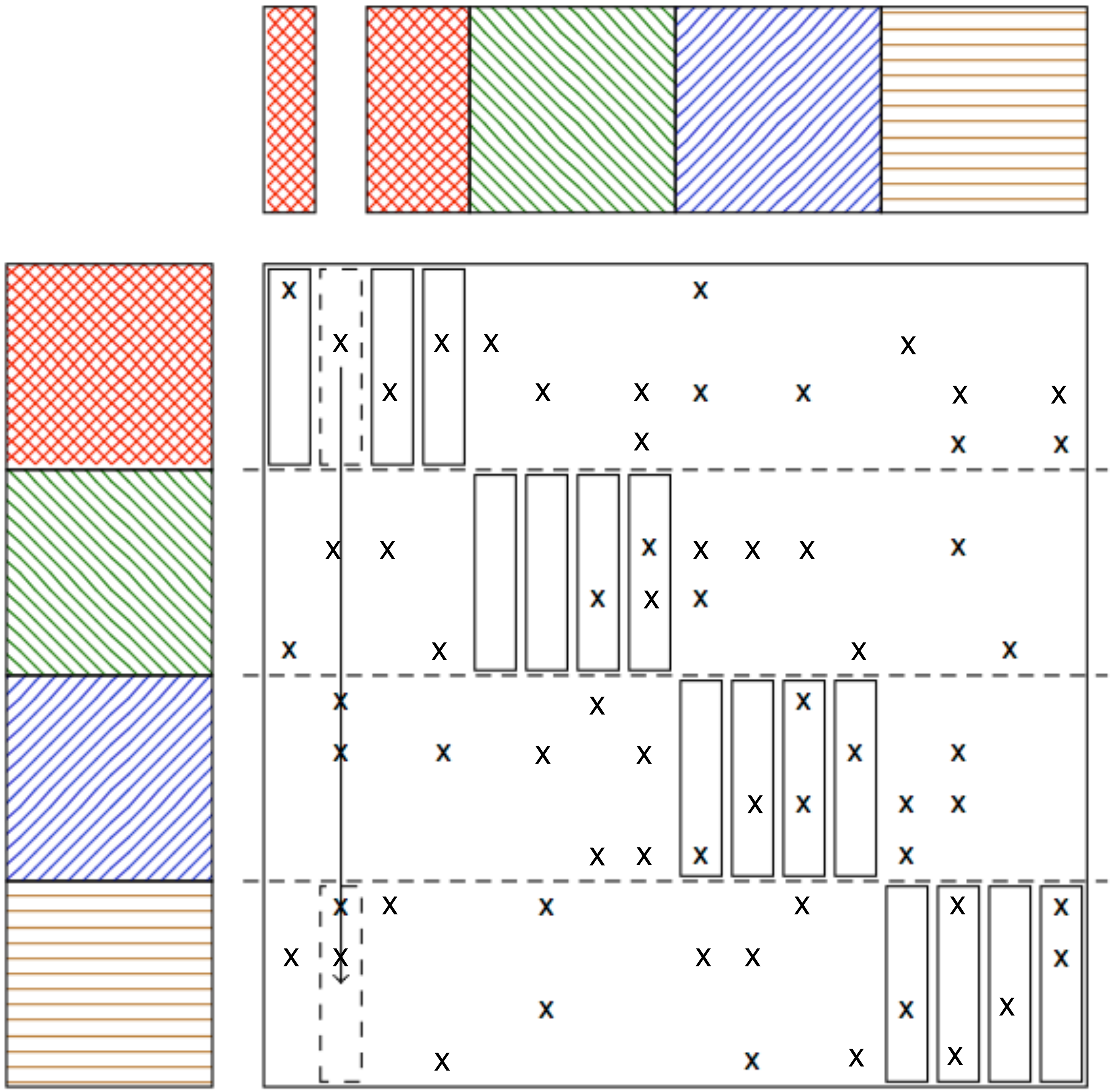


Nomadic Token Passing

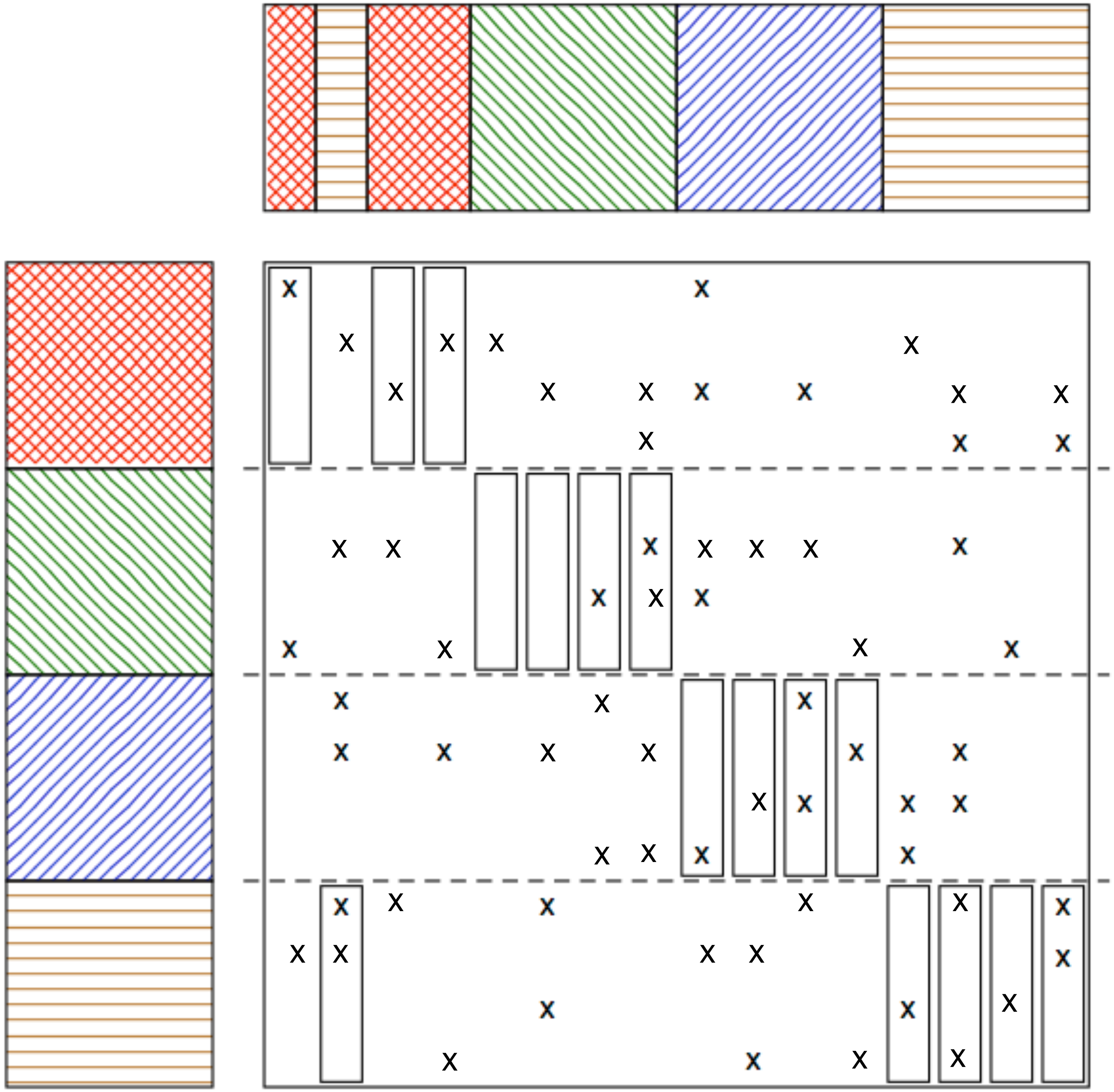
NOMAD Illustration



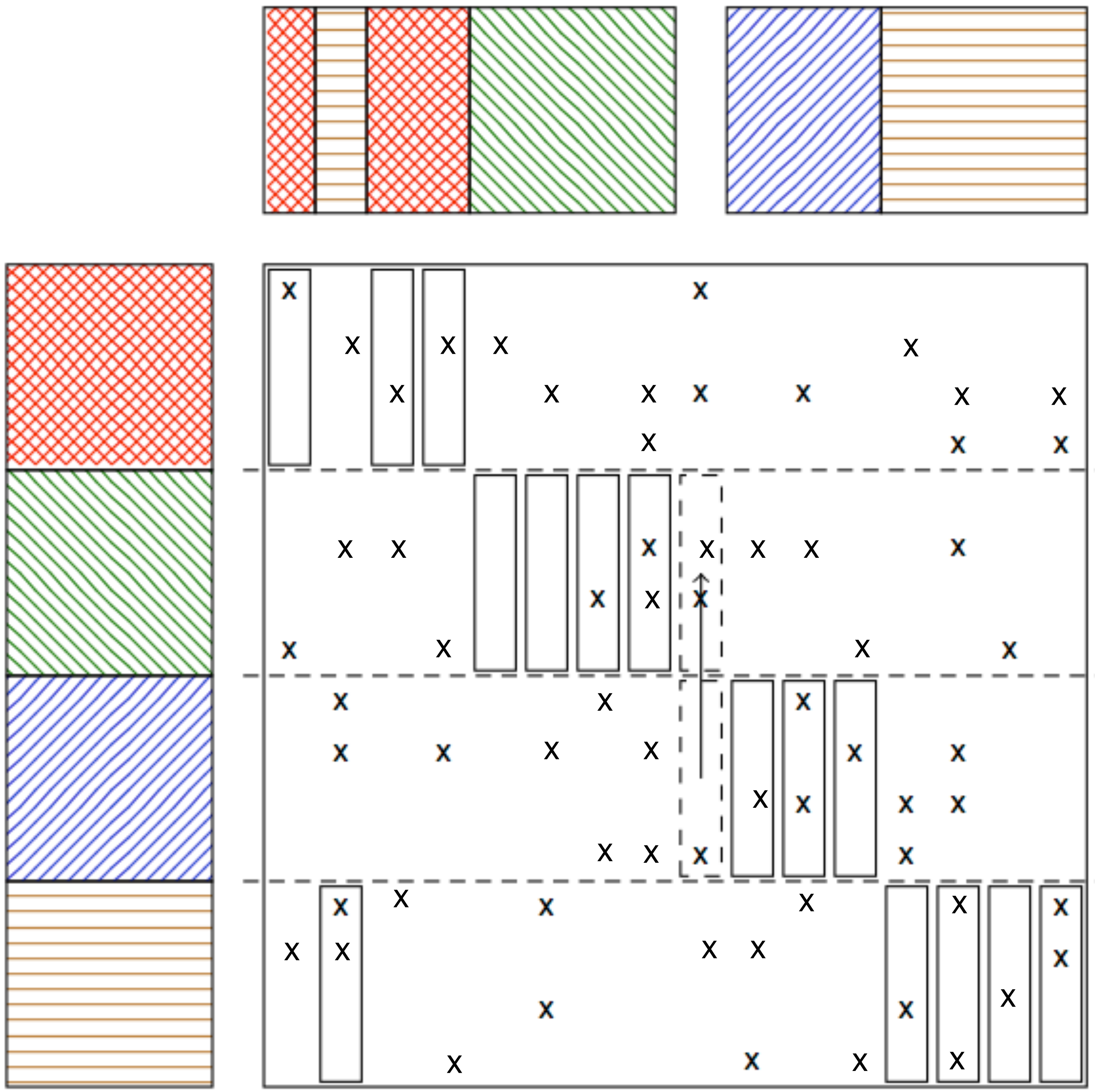
NOMAD Illustration



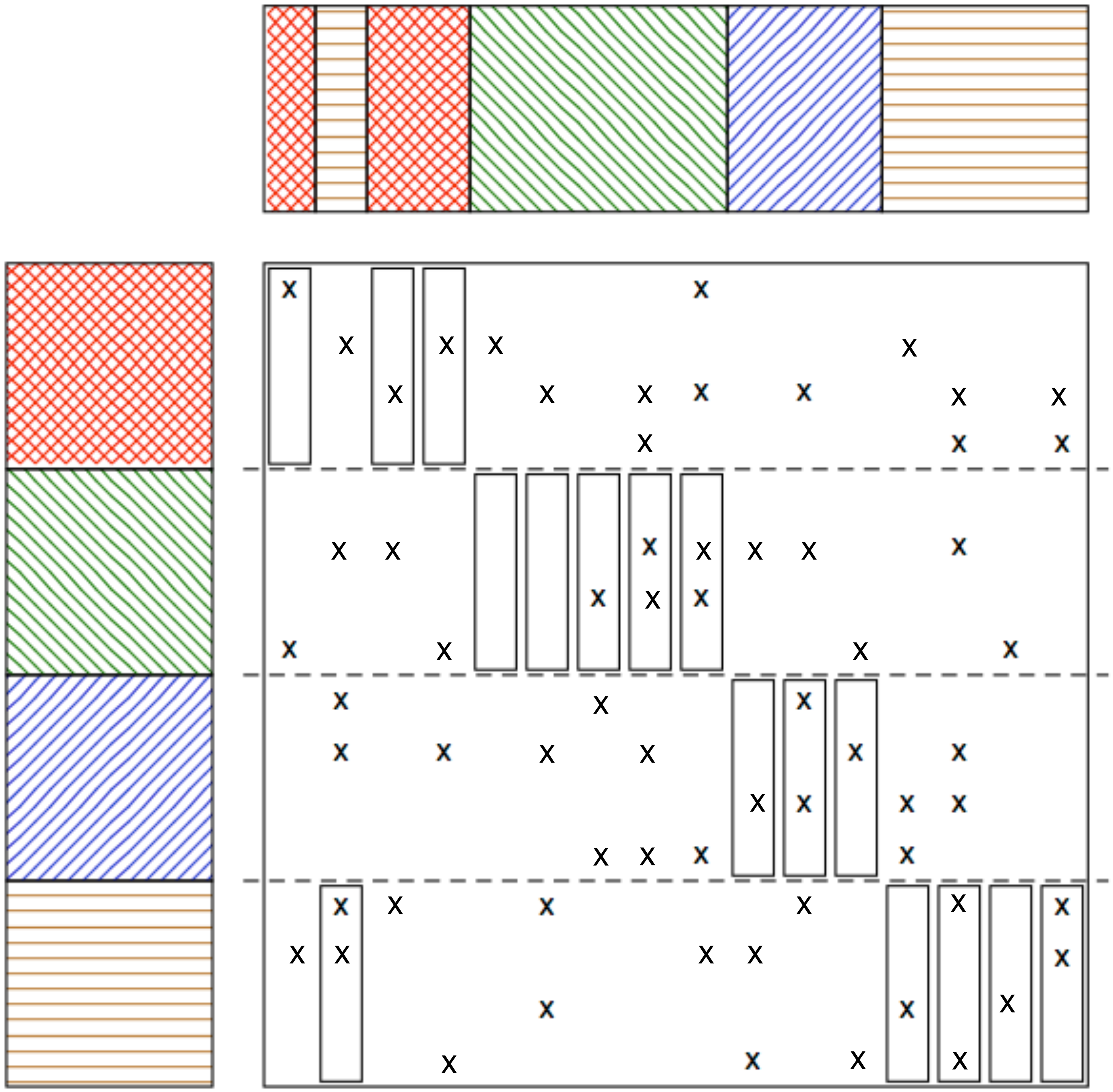
NOMAD Illustration



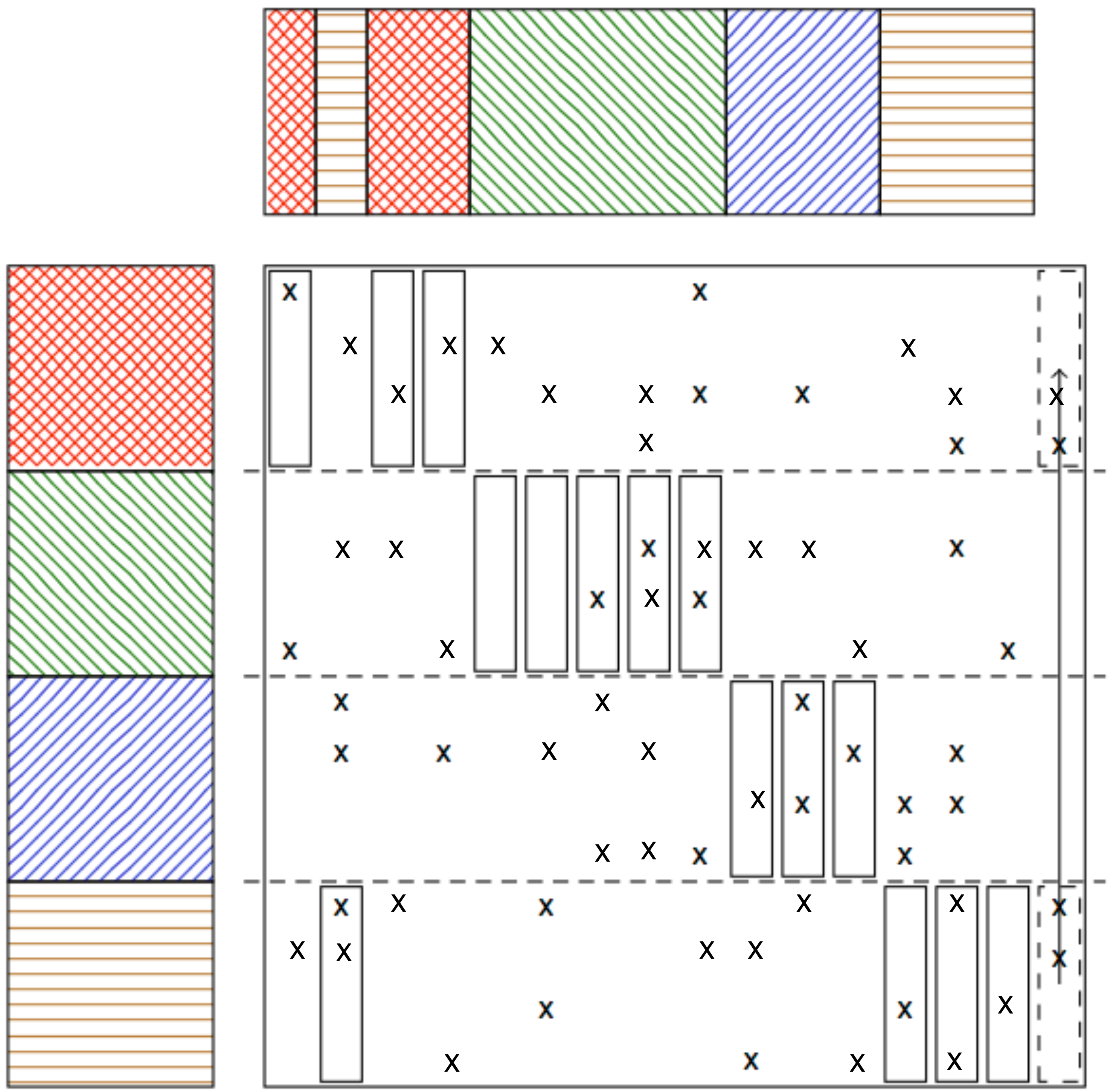
NOMAD Illustration



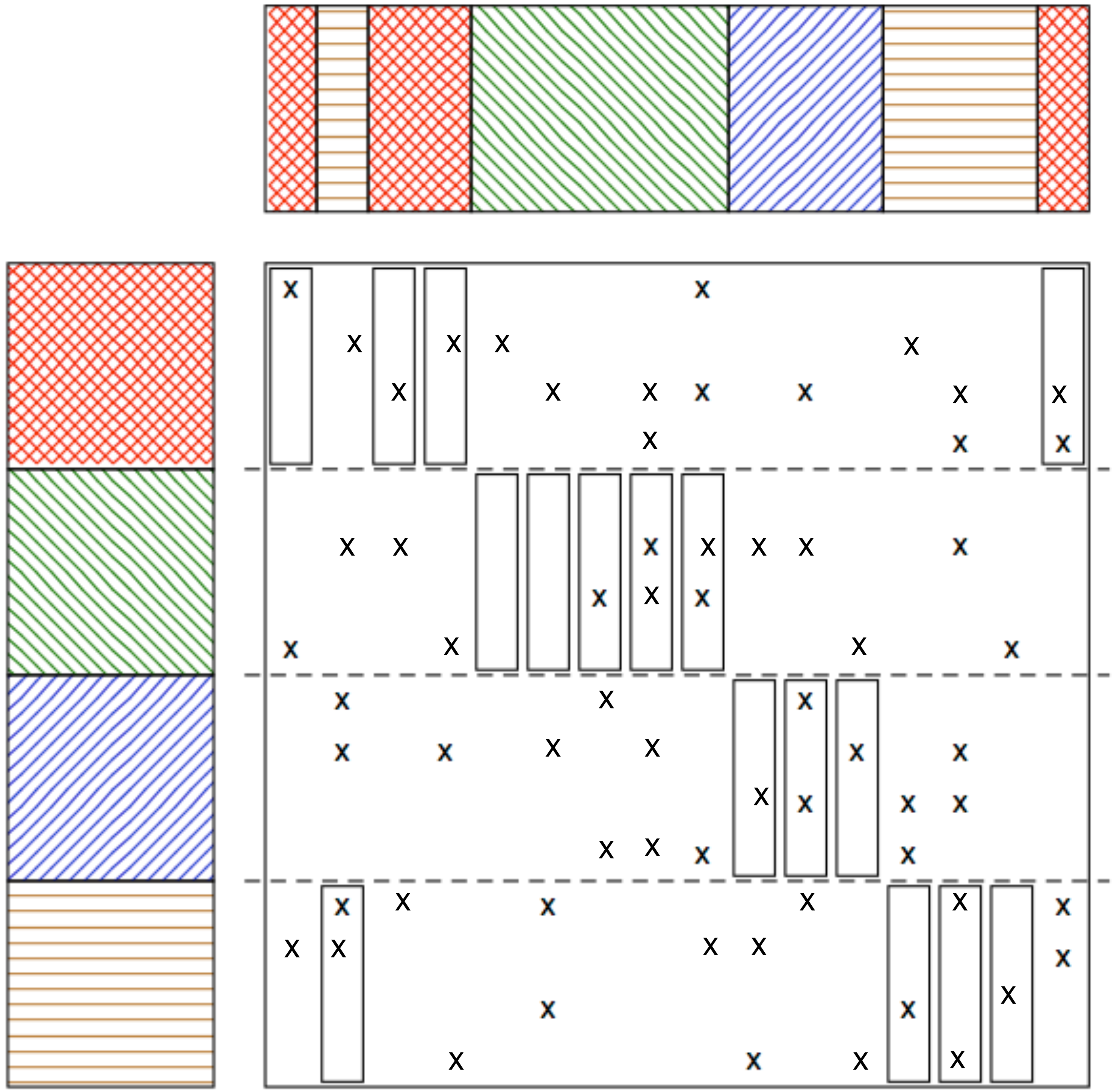
NOMAD Illustration



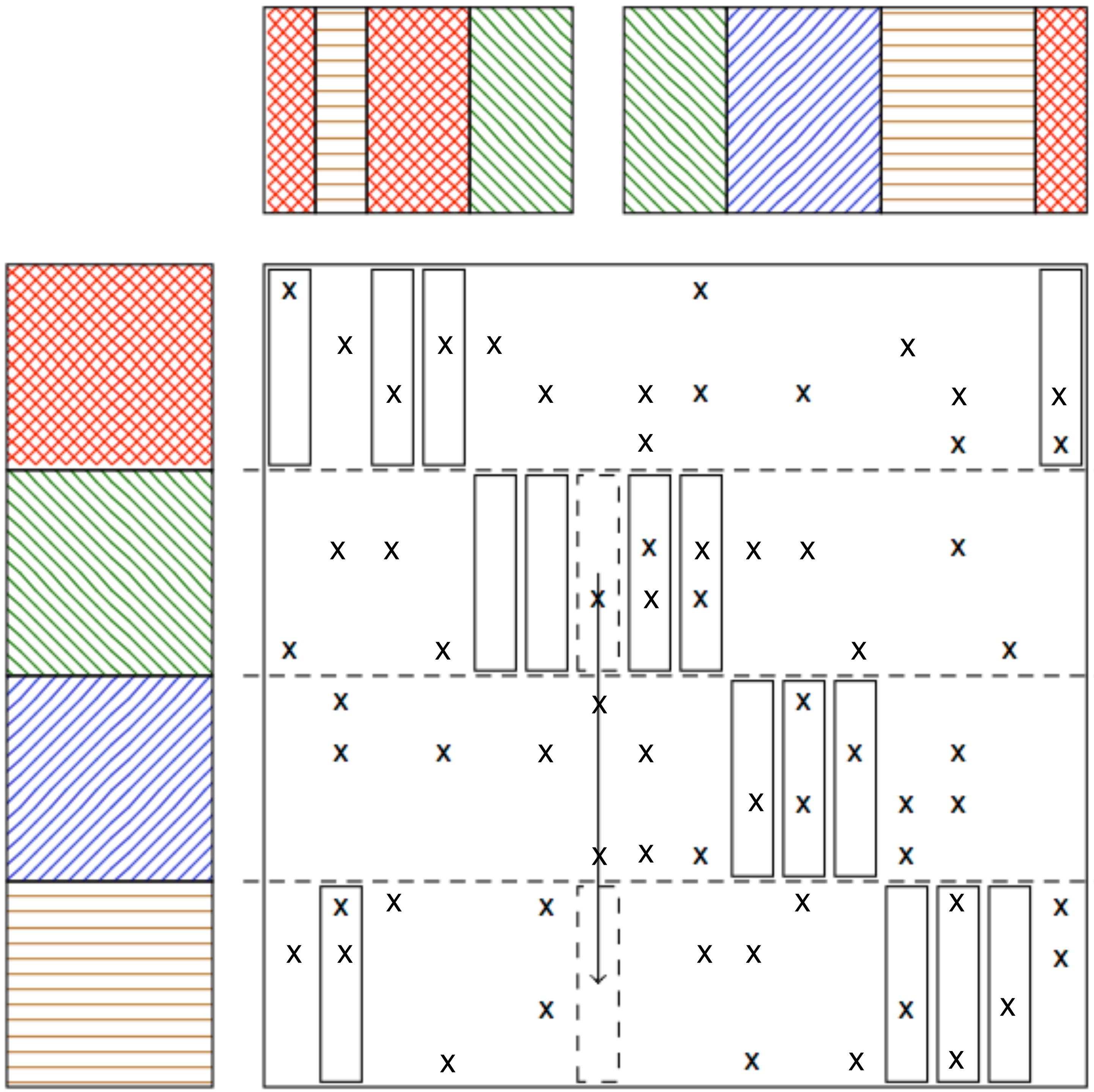
NOMAD Illustration



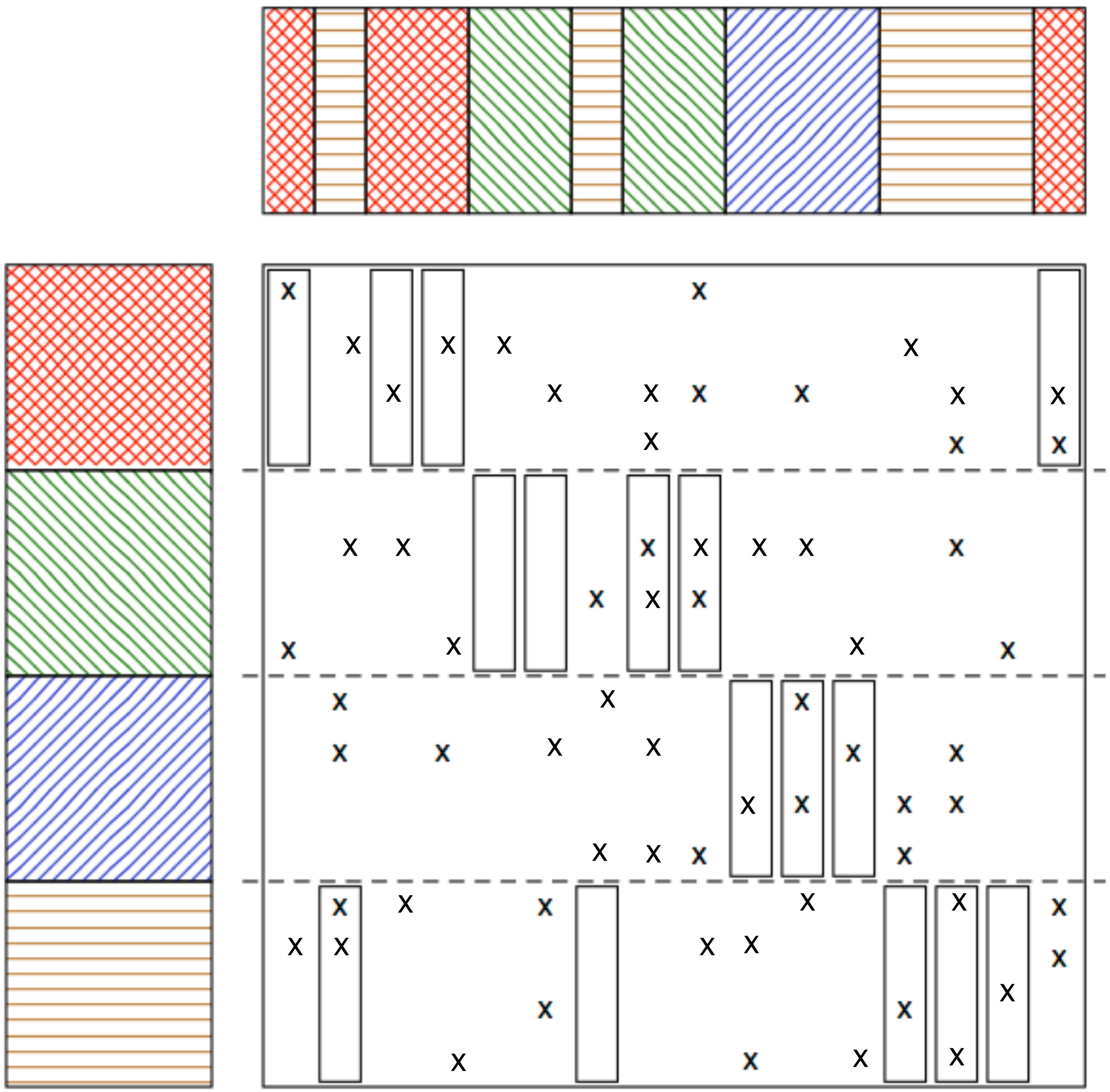
NOMAD Illustration



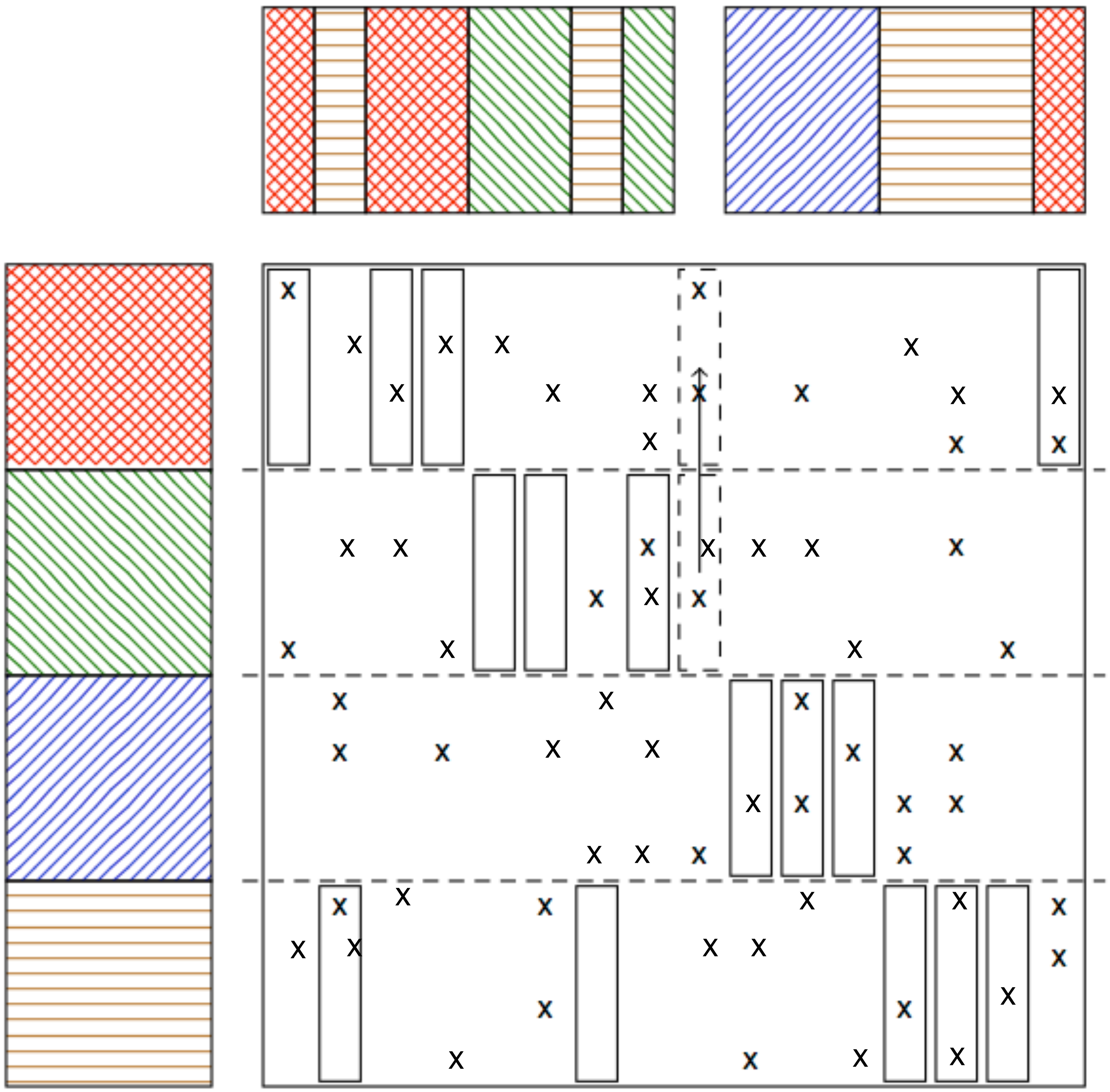
NOMAD Illustration



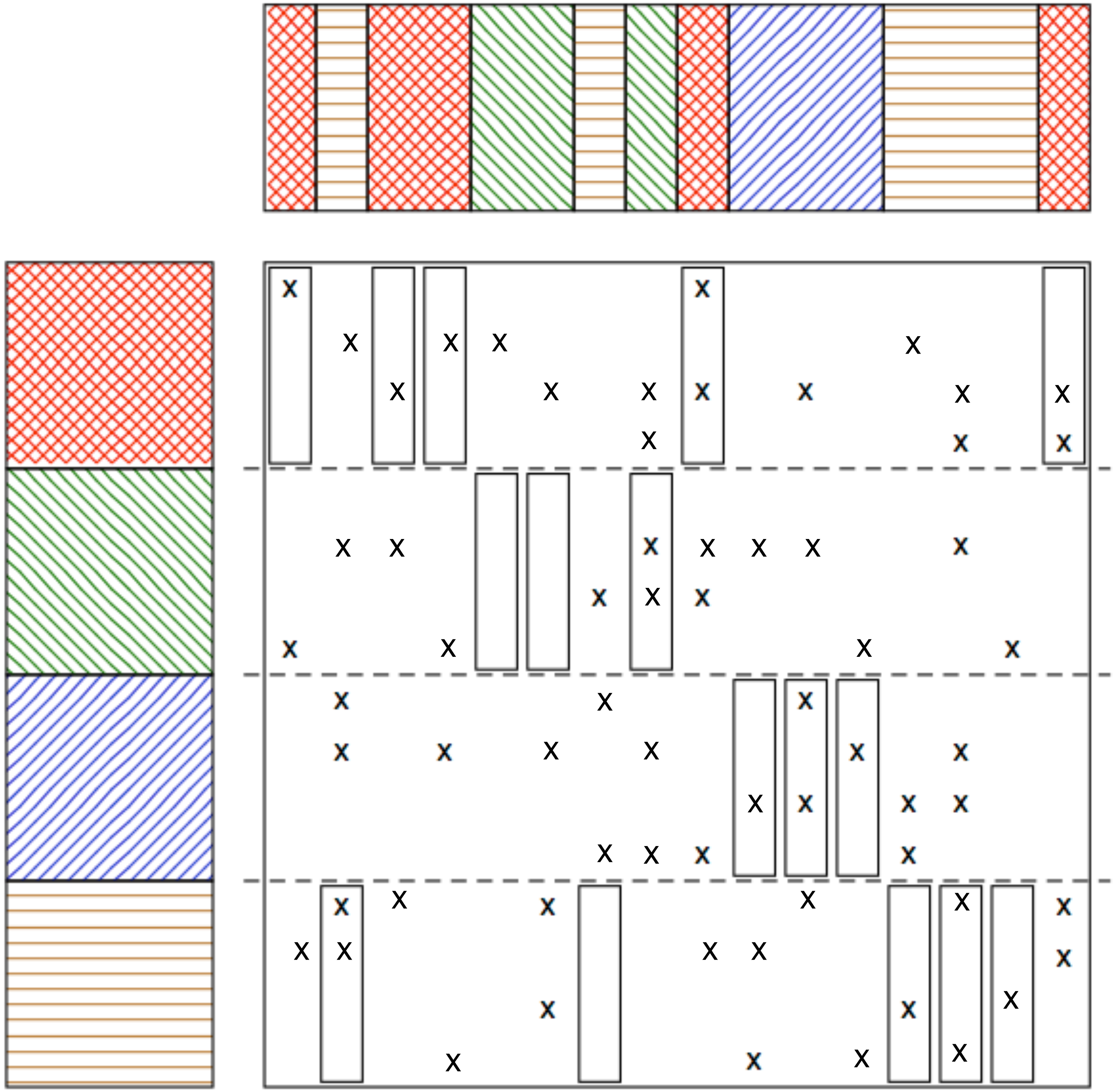
NOMAD Illustration



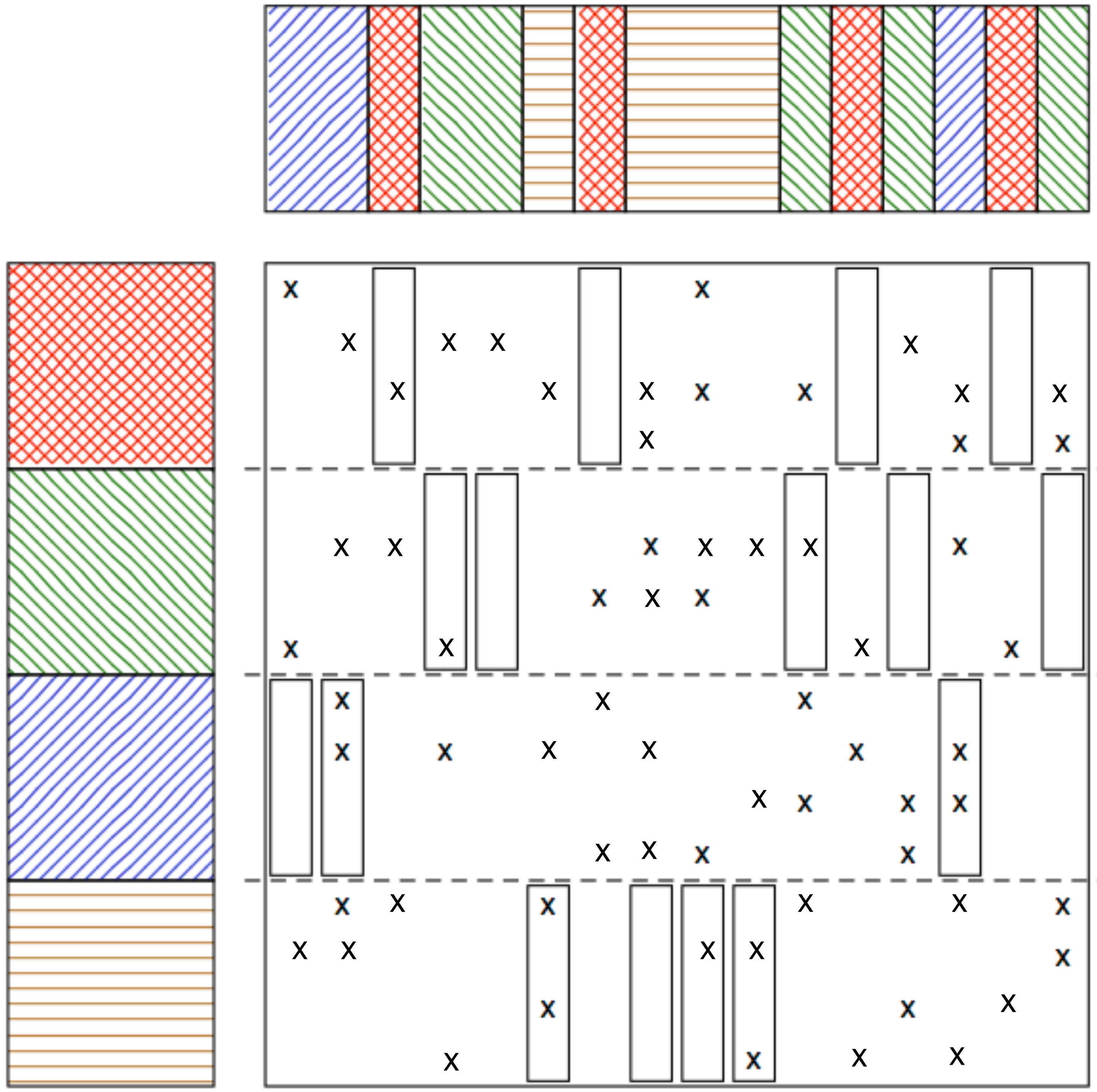
NOMAD Illustration



NOMAD Illustration

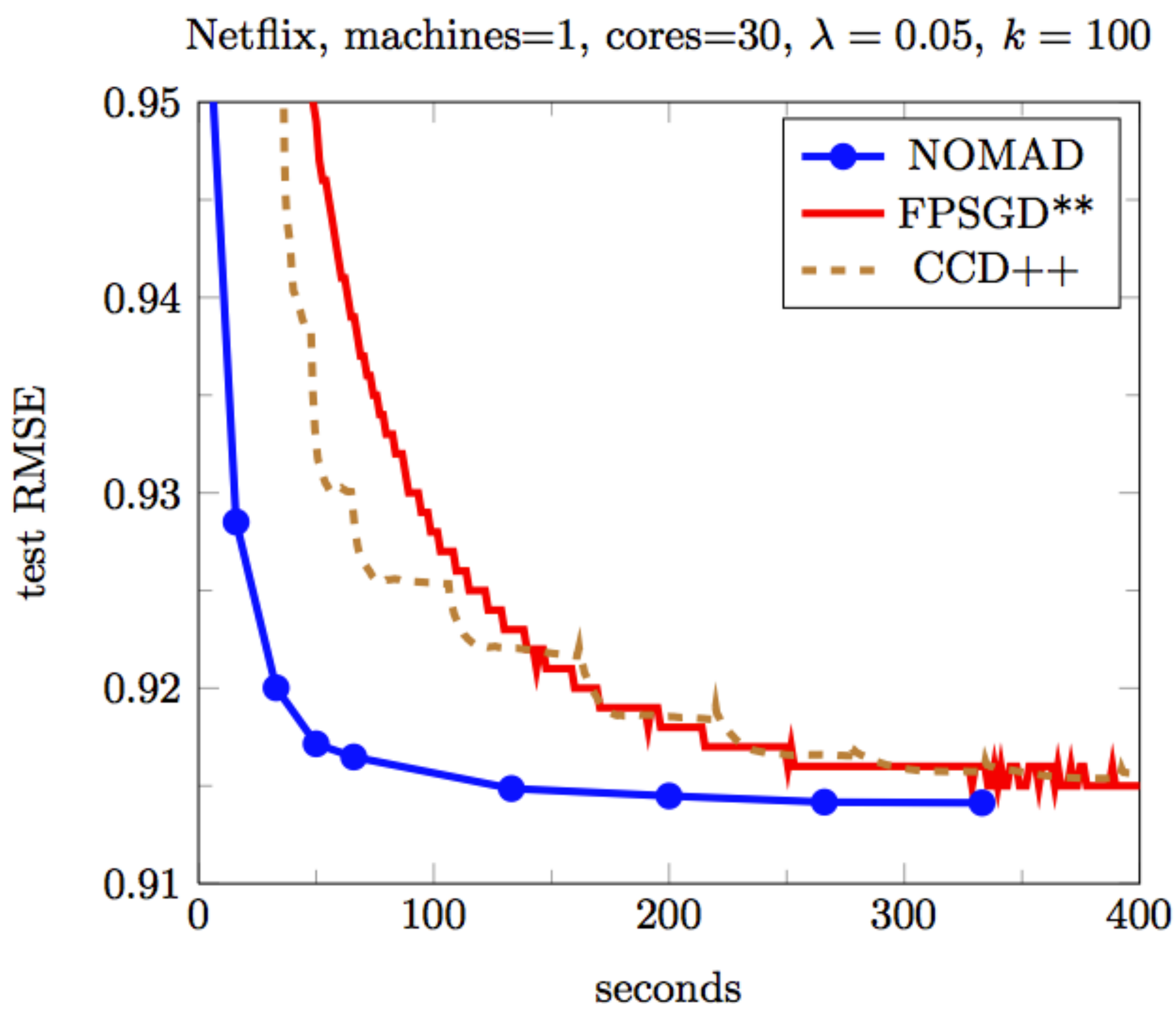


NOMAD Illustration

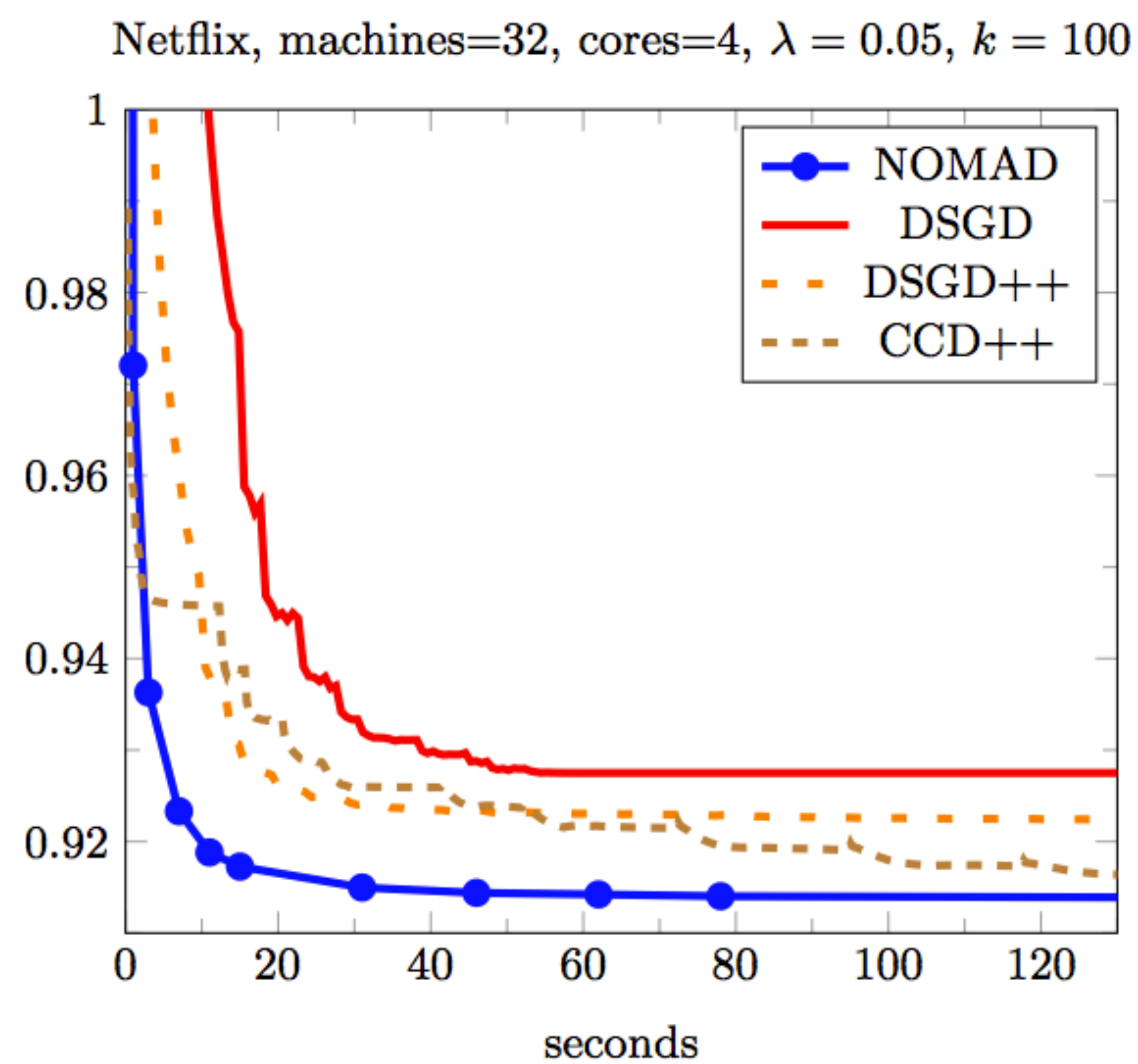


Results: Recommender Systems

Netflix dataset: 480,189 users, 17,770 movies, ~100M ratings



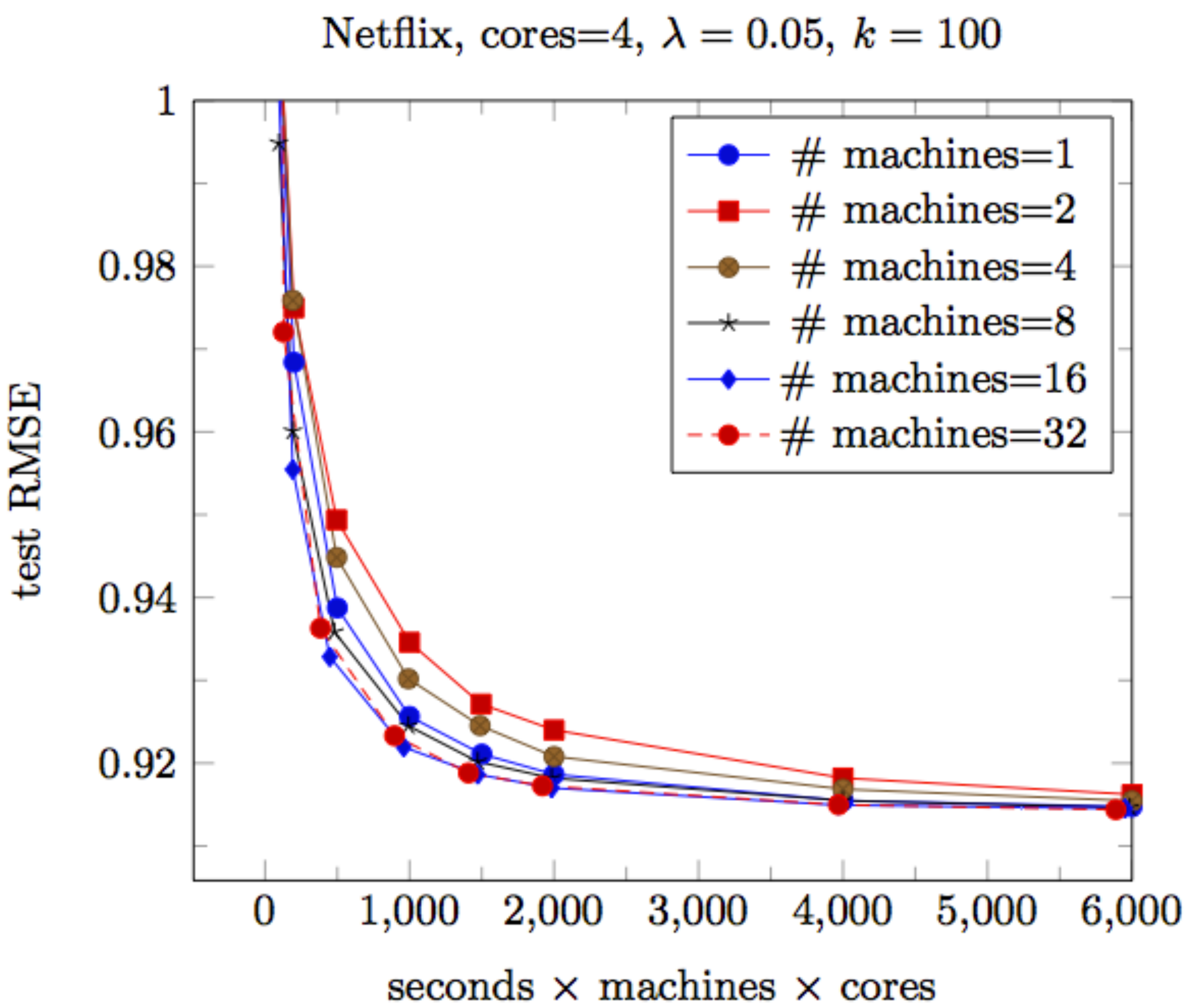
Multicore



Distributed

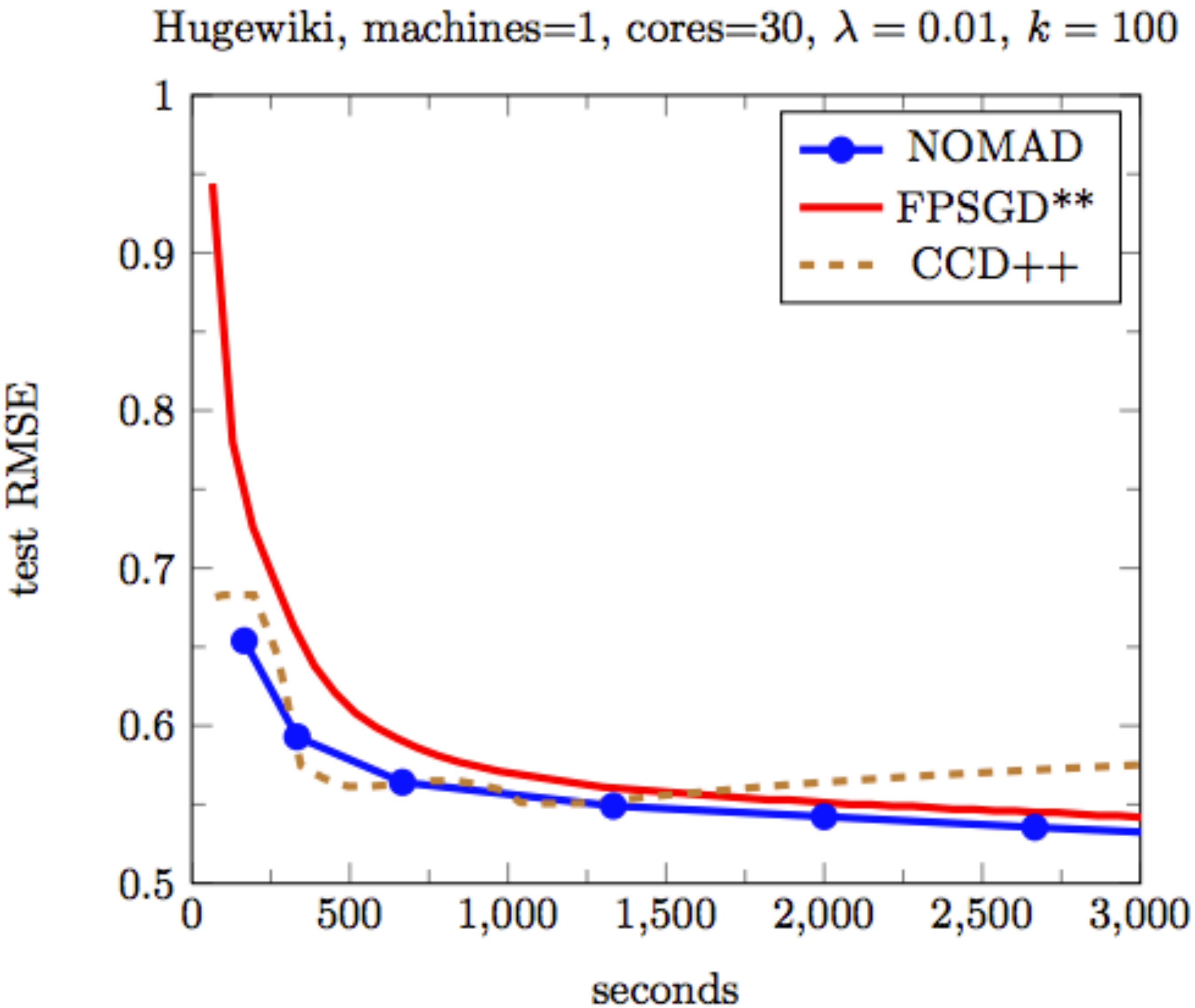
Results: Recommender Systems

Netflix dataset: 480,189 users, 17,770 movies, ~100M ratings

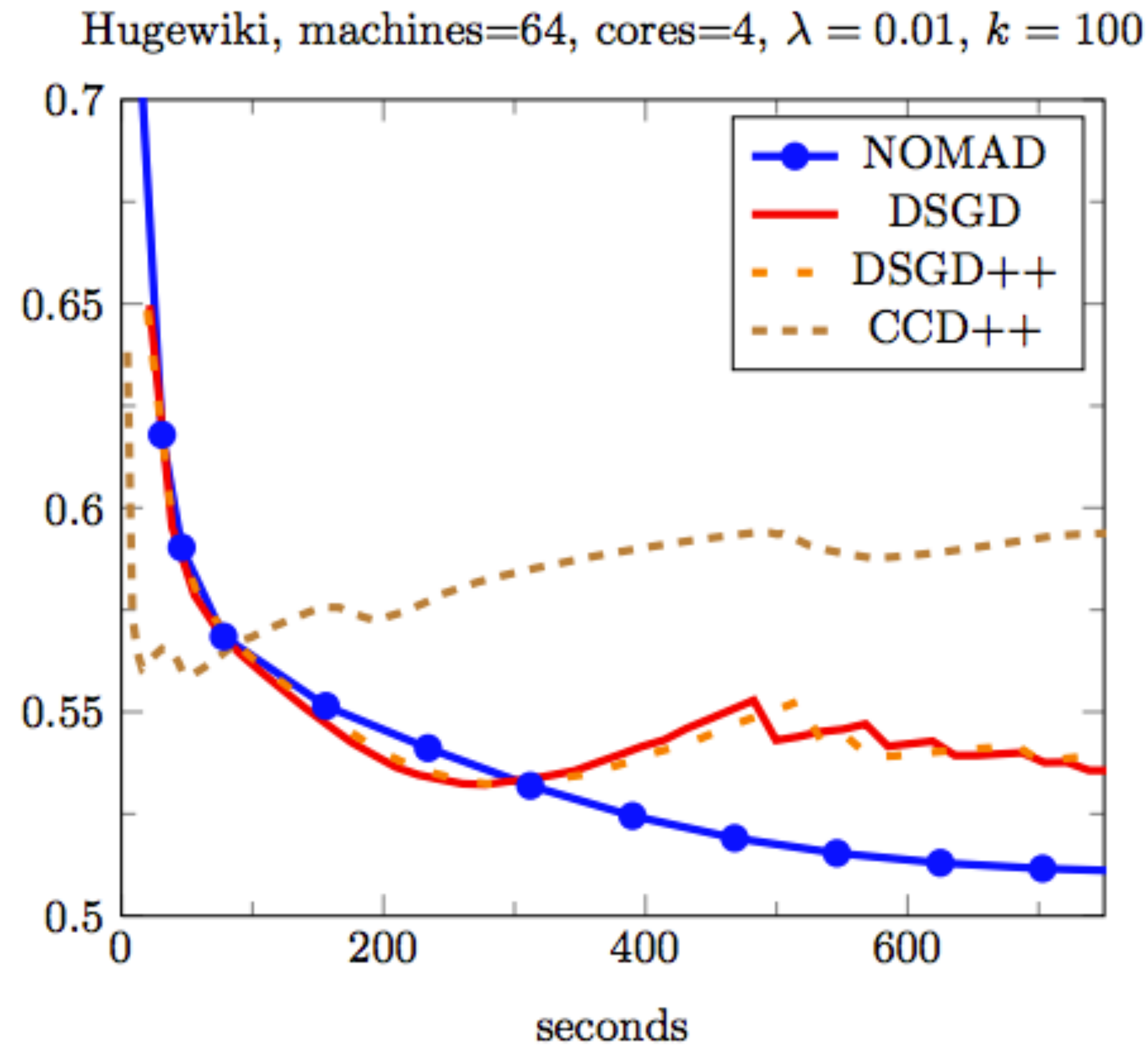


Results: Recommender Systems

HugeWiki dataset: 50,082,603 users, 39,780 movies, ~2.7B ratings



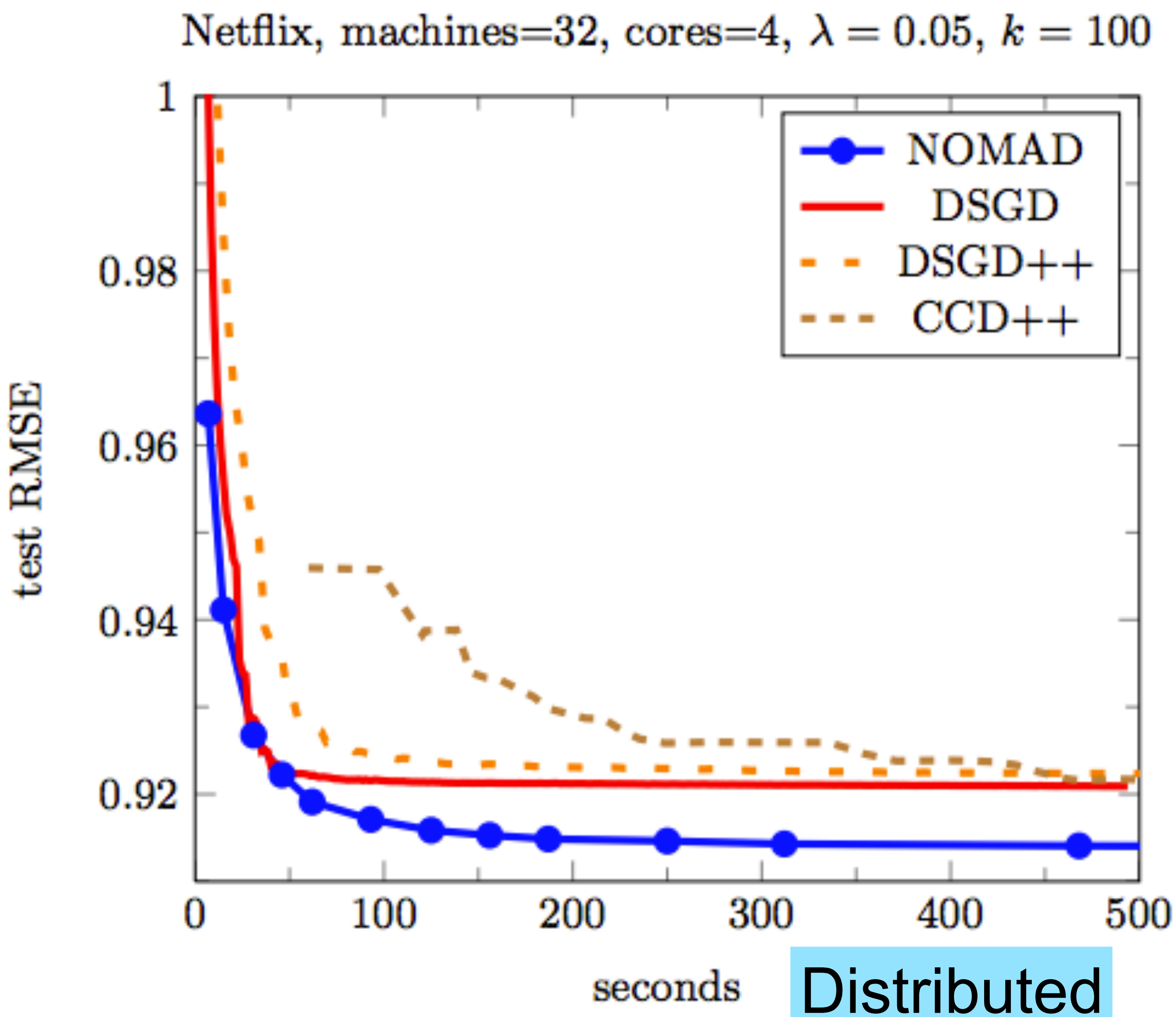
Multicore



Distributed

Results on Amazon Web Services

Netflix dataset: 480,189 users, 17,770 movies, ~100M ratings



Distributed

Conclusions

- Many “latent factor” models arise in data analysis (SVD,NMF,MC,…)
- NOMAD: Parallel Asynchronous Algorithm
  - Distributed, Lock-free & Serializable
  - Works well in practice
  - Convergence/recovery proof? Blocking? 2-D tiling? Communication Avoiding?
  - Can be applied to other latent-factor models, such as, “probabilistic topic modeling”
- Lots of demand for such scalable software in industry
  - No good solutions exist, so opportunities are great

References

- Yun, H., Yu, H., Hsieh, C., Vishwanathan S., & Dhillon, I (2014). *NOMAD: Non-locking, stOchastic Multi-machine algorithm for Asynchronous and Decentralized matrix completion*. To appear in Proceedings of the VLDB Endowment.
- Gemulla, R., Nijkamp, E., Haas, P. J., & Sismanis, Y (2011). *Large-scale matrix factorization with distributed stochastic gradient descent*. In Proceedings of the 17th ACM SIGKDD (pp. 69-77). ACM.
- Jain, P., Meka, R., & Dhillon, I.S (2010). *Guaranteed Rank Minimization via Singular Value Projection*. In NIPS (Vol. 23, pp. 937-945).
- Jain, P., Netrapalli, P., & Sanghavi, S (2013). *Low-rank matrix completion using alternating minimization.* Proceedings of the 45th annual ACM Symposium on theory of computing.