Approximation Schemes for Multi-Budgeted Independence Systems

Dimitris Bakas
$\mu\pi\lambda\forall$

Network Algorithms and Complexity

General problem setting

- Ground set: S
- Solution set: $I \subseteq 2^{S}$.
- Weight: w : S $\rightarrow \mathbb{Z}_+$.
- Lengths: l_i : S $\rightarrow \mathbb{Z}_+$with budget $L_i \in \mathbb{Z}_+$ $\forall i \in [k] = \{1,2, \dots, k\}$

min/max $w(I)$ subject to $\mathrm{I} \in S, l_i(\mathrm{I}) \leq L_i$

Typical problems

- k-budgeted perfect matching
- k-budgeted spanning tree
- k-budgeted shortest path
- k-budgeted matroid independent set
- …

$k = 0$: polynomial-time

Some relevant previous results

Results for 1-budgeted problems

- FPTAS for 1-budgeted shortest path (Warburton [1987], Hassin[1992], Lorenz and Raz [2001])
- PTAS for 1-budgeted spanning tree (Ravi and Goemans [1996])
- PTAS for 1-budgeted matching and 1-budgeted matroid intersection independent set (Berger [2009])

Theorem 1

For $k \geq 2$, it is NP-complete to decide whether there is a feasible solution for

- k-budgeted shortest path
- k-budgeted perfect matching and
- k-budgeted spanning tree

We consider problems whose solutions form an independence system (S, F):

$$S \in F, S' \subseteq S \Rightarrow S' \in F$$

Theorem 2

P : a k-budgeted problem where the set of solutions is an independence system.

Algorithm A : computes in polynomial time an $(1-\delta)$ approximate solution to P violating each budget by a factor at most $(1 + \delta)$.

Then there is a PTAS for P.

Feasibilization

Filtering :

Guess the $h = \frac{k}{\varepsilon}$ heaviest elements E_H in opt

Scaling:

Scale down all the budgets by a factor (1 - δ) = (1 - $\frac{\varepsilon}{\kappa+1}$) to obtain E_L

Return : $E_H \cup E_L$

A PTAS for 2-budgeted matching

To obtain a PTAS it suffices to provide an efficient algorithm returning a solution of $value \geq opt \ - c \, w_{max}$

We present a polynomial algorithm for 2-budgeted matching returning a solution of

$$weight \geq opt \ - 6w_{max}$$

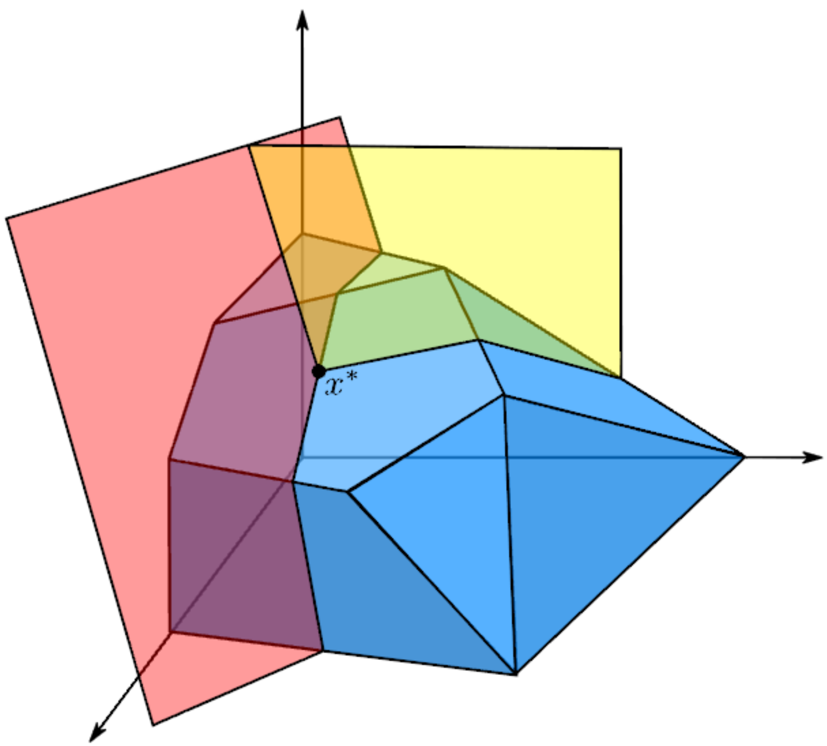
Patching procedure

Given two matchings x', x'' and a parameter $\mu \in [0,1]$, computes a matching z satisfying $l_z \leq l_{x_\mu}$ where $x_\mu = \mu x' + (1 - \mu) x''$.

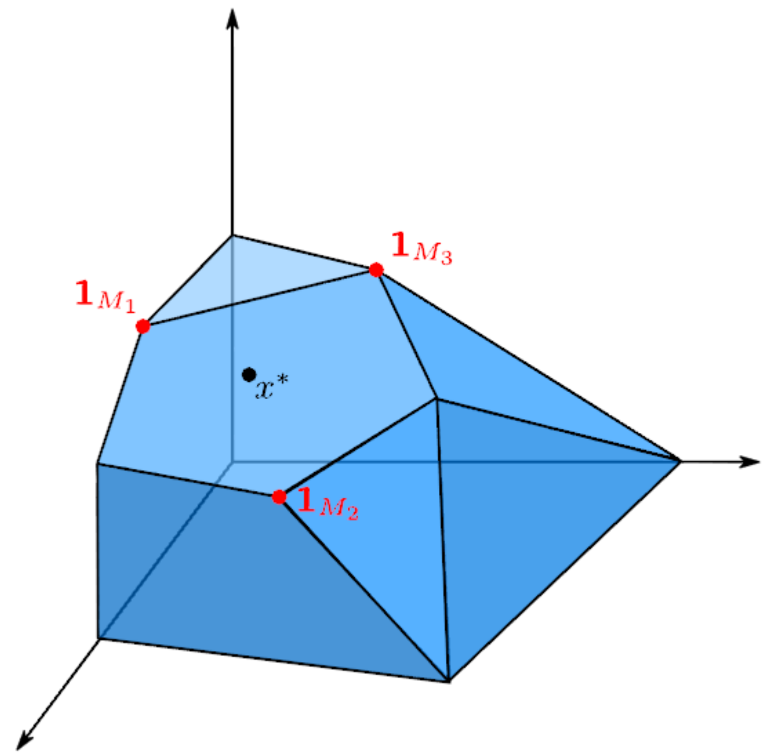
The weight $w(z)$ is close to $w(x_\mu)$.

Framework

Get optimal basic solution x^* to
max $w(x)$
$l_1(x) \leq L_i \; \forall i \in \{1,2\}$



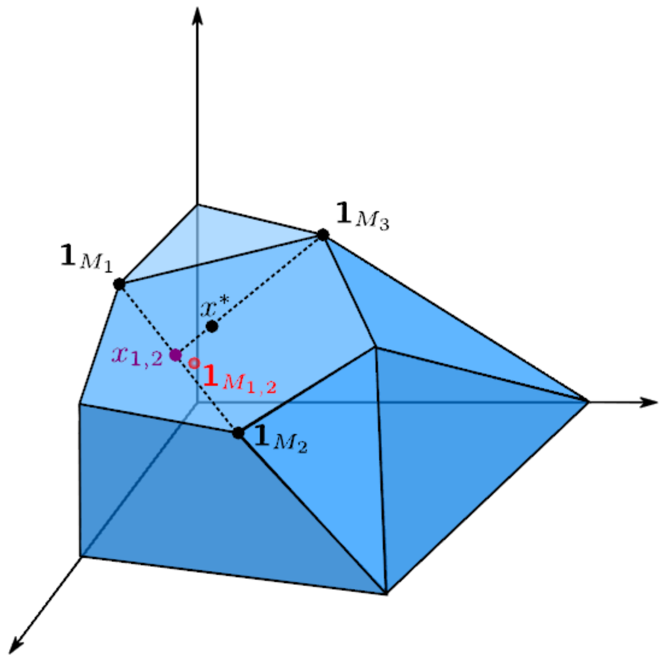
Compute convex combination
$x^* = a_1 1M_1 + a_2 1M_2 + a_3 1M_3$
of three matchings



Framework

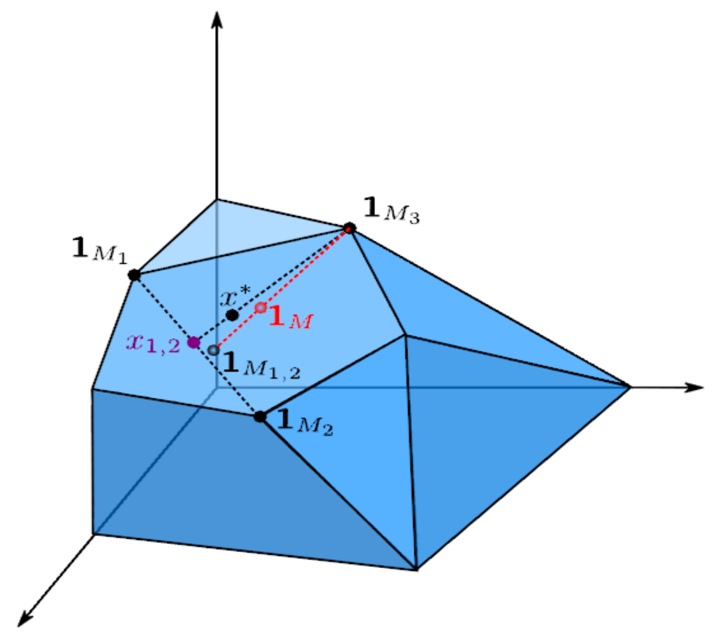
Merge M_1 and M_2 to get $M_{1,2}$ "close" to

$$x_{1,2} = \frac{1}{a_1 + a_2}(a_1 1M_1 + a_2 1M_2)$$



Then merge $M_{1,2}$ and M_3 to get M "close" to

$$x^* = a_1 + a_2(1M_{1,2} + 1M_3)$$



A property of polygonal curves in $\mathbb{R}^2$

For $a \in [0, \tau]$, let $f^a : [0, \tau] \to \mathbb{R}^2$ be the following polygonal curve:

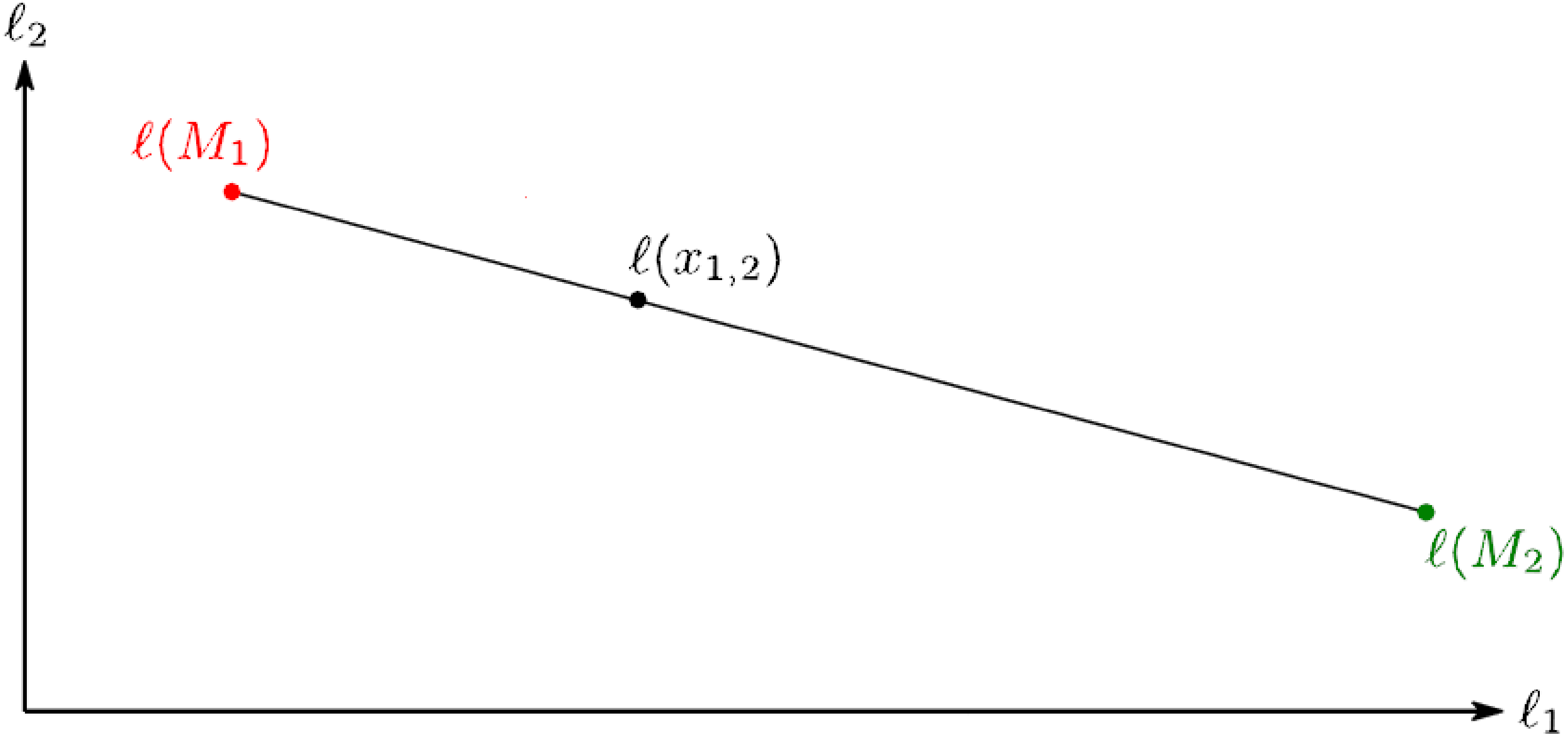
$$f^a(t) = \begin{cases} f(t+a) - f(a) + f(0) & if\ t + a < \tau \\ f(\tau) - f(a) + f(a + t - \tau) & if\ t + a \geq \tau \end{cases}$$

Lemma:

There are $a, t \in [o, \tau]$ such that

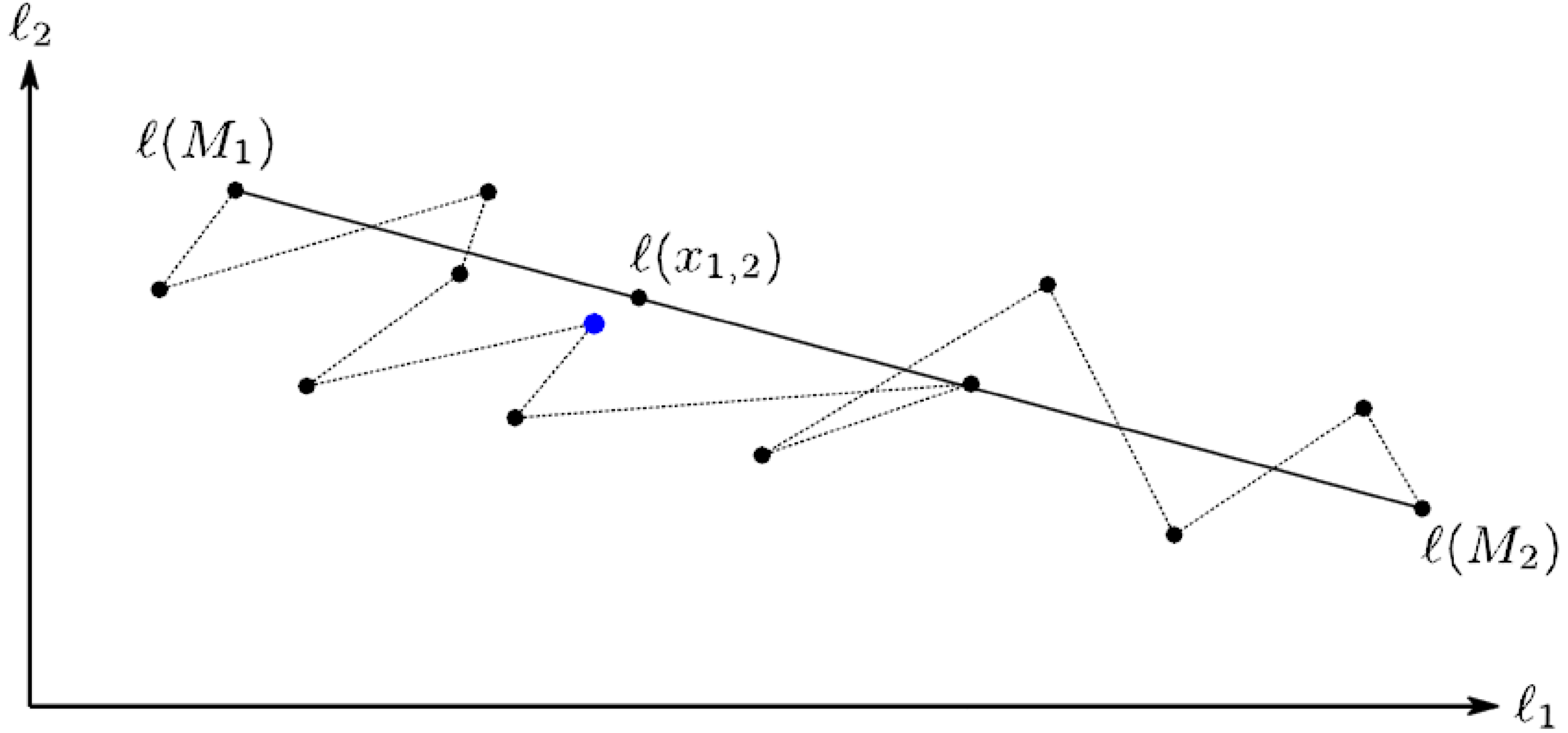
$$f^a(t) = \mu f(0) + (1 - \mu)\, f(\tau)$$

Merging two matchings



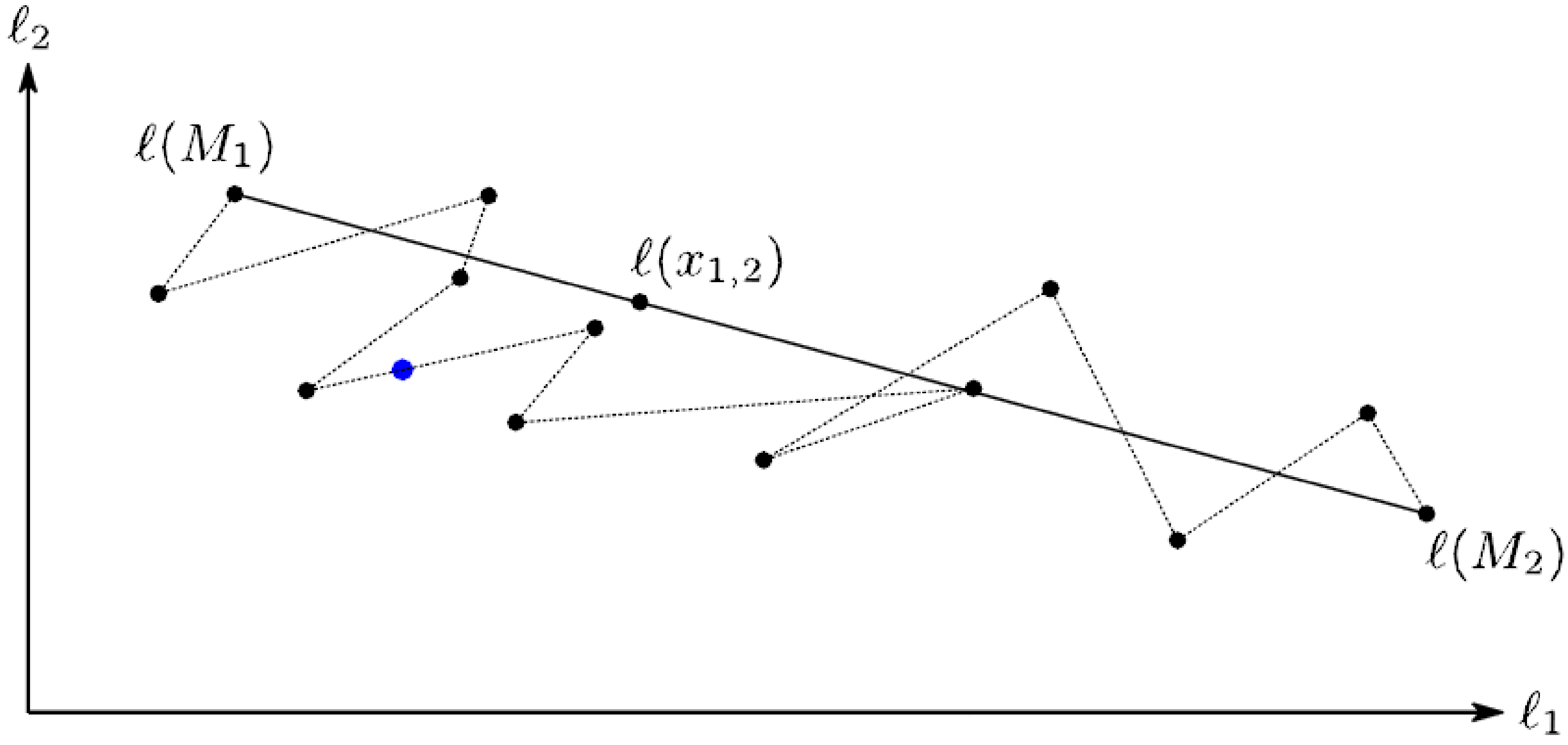
Goal : Get almost matching y with $l(y) \ = \ l(x_{1,2})$

Merging two matchings



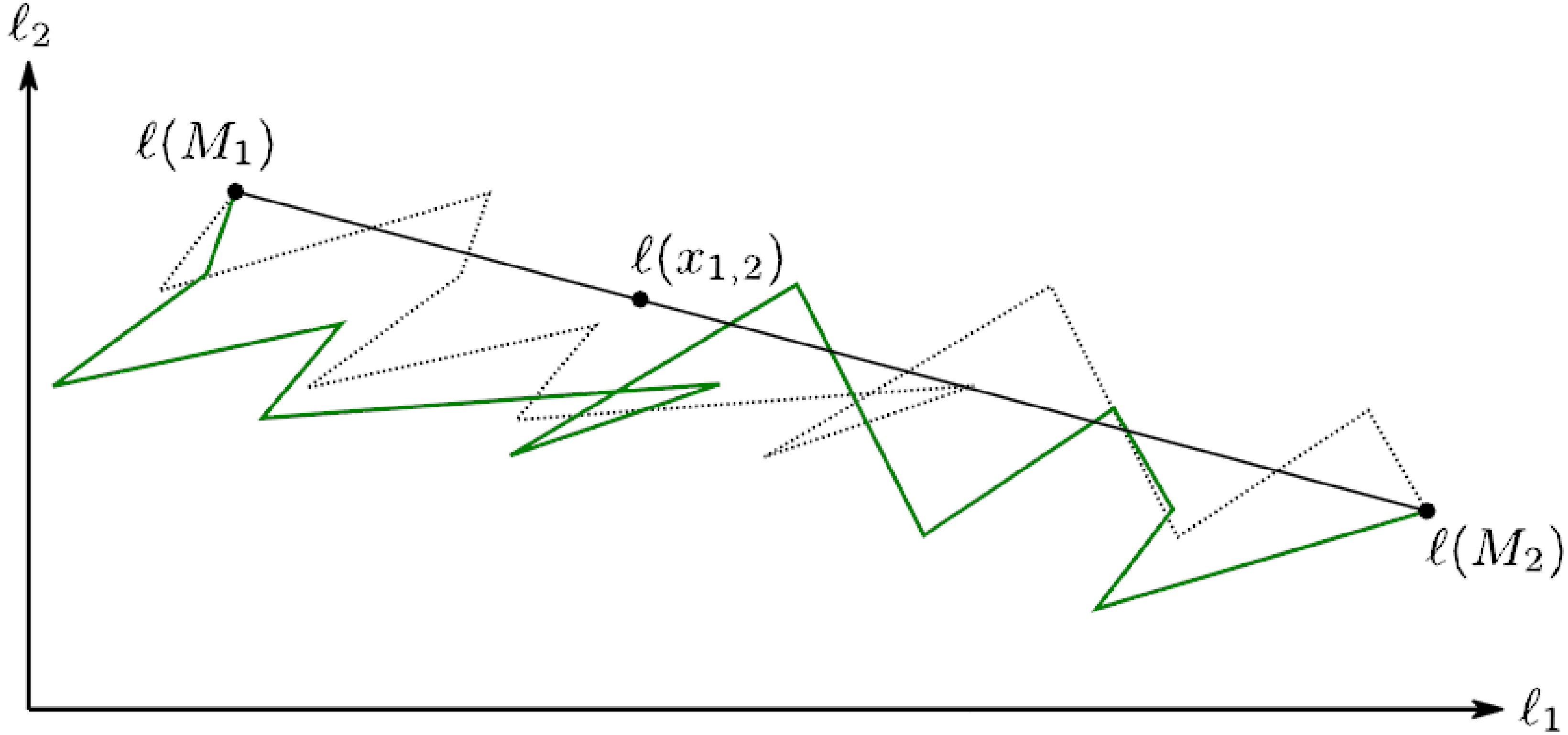
Flips can be performed with a fractional start and/or endpoint.

Merging two matchings

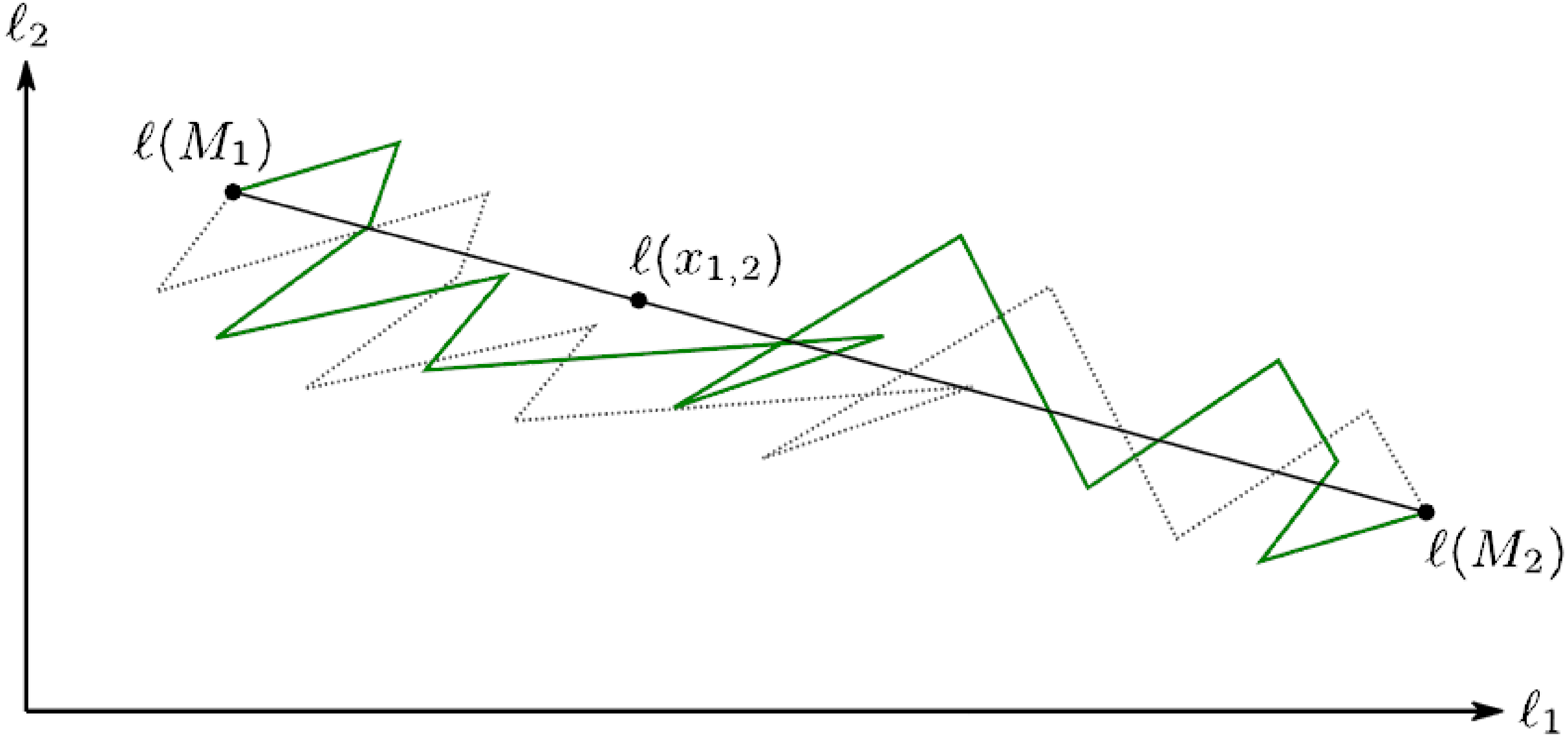


Flips can be performed with a fractional start and/or endpoint.

Merging two matchings



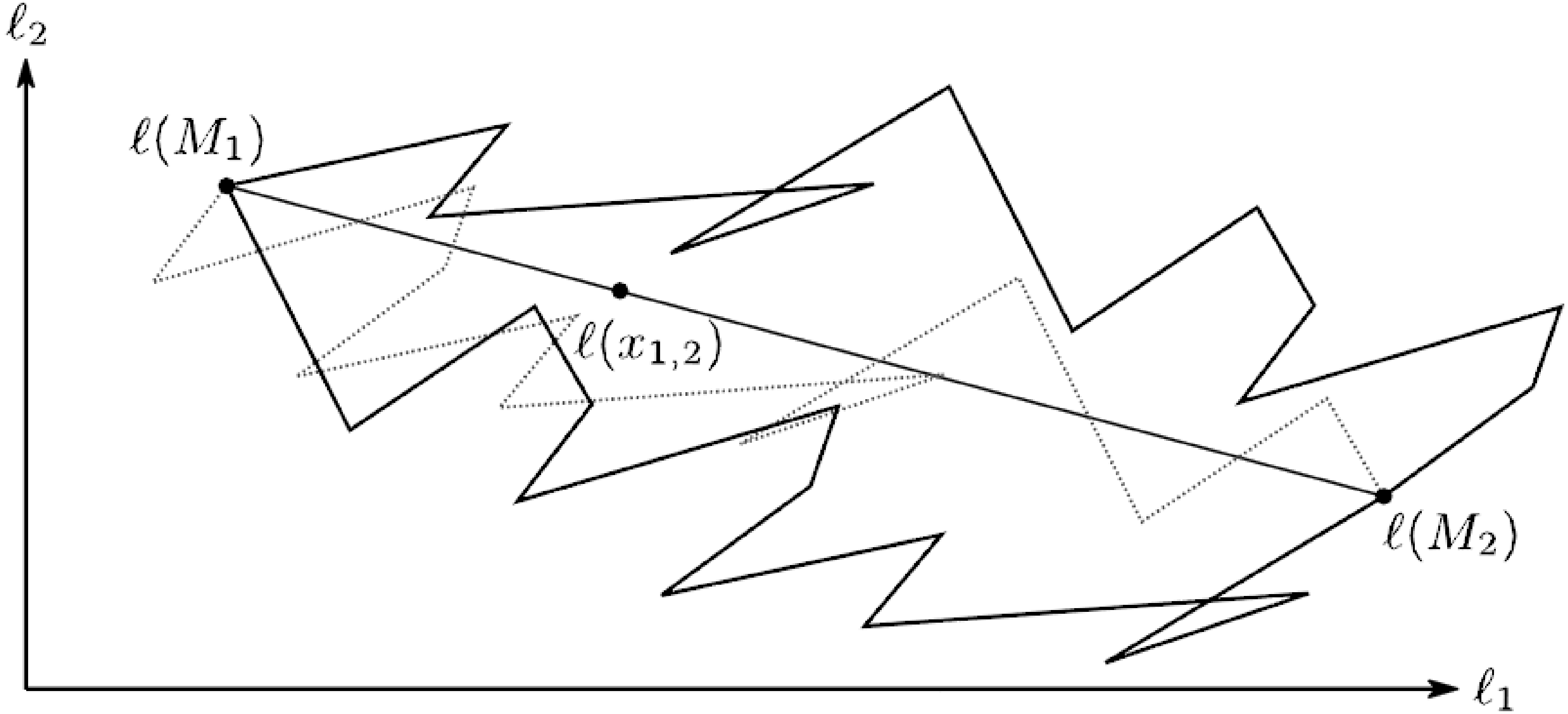
Merging two matchings



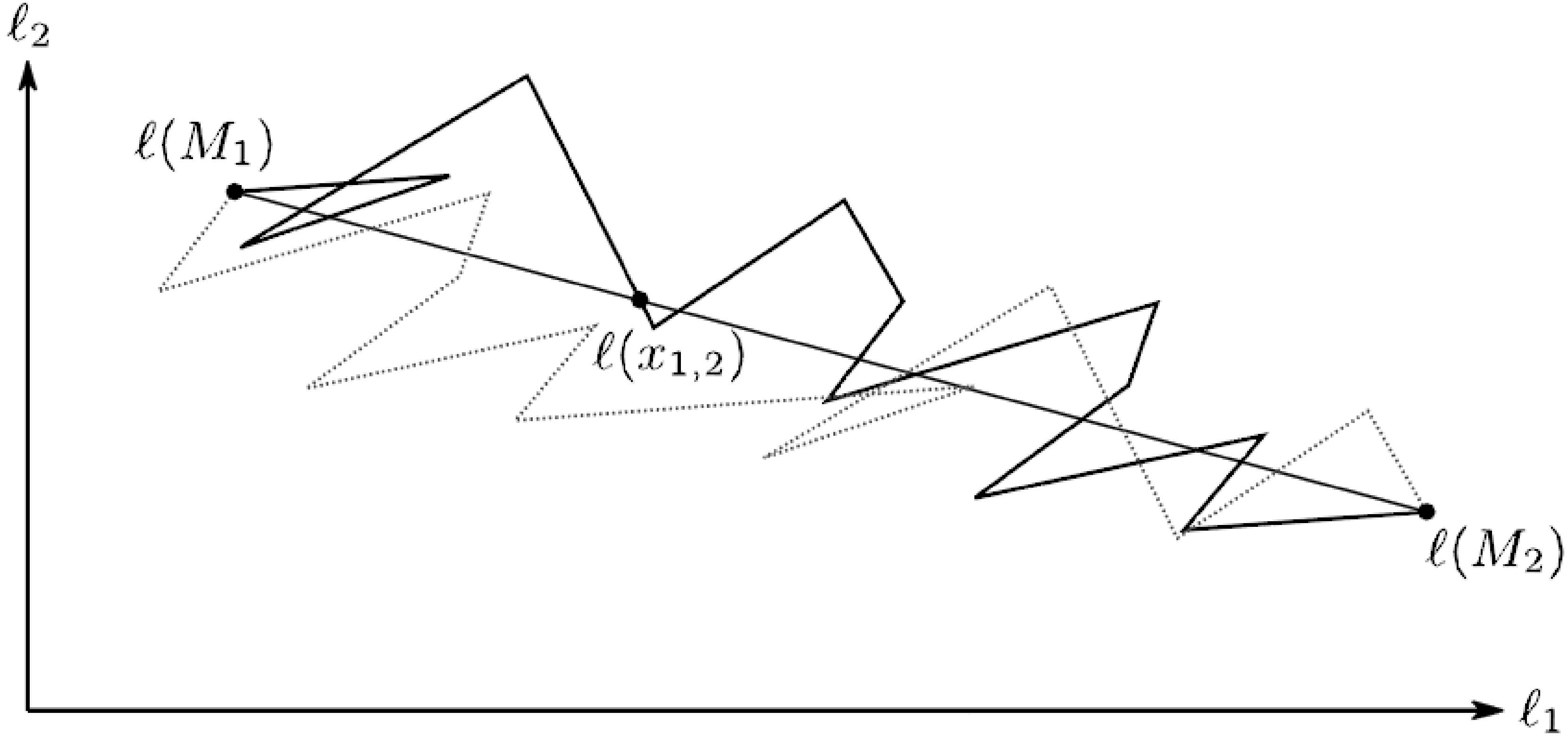
Is there a starting point such that the curve contains $l\,(x_{1,2})$?

Yes, such a starting point exists.

Merging two matchings



Merging two matchings



Theorem:

Starting with M_1, it is possible to flip a subinterval of edges to obtain an almost matching y with $l_i(y) = l_i(x_{1,2})$

Claim

The matching $M_{1,2}$ obtained from the almost matching y (merge of M_1 and M_2) satisfies:

$$\mathrm{w}\left(M_{1,2}\right) \geq w\left(x_{1,2}\right) - 2w_{max}$$

Summary

1. Merging M_1 and M_2 by edge flips we get an almost matching y : $l_i(y) = l_i(x_{1,2})$
2. Removing at most 2 units of y we get a matching $M_{1,2}$:

$$l_i(M_{1,2}) \leq l_i(y) = l_i(x_{1,2}) \text{ for } i = 1,2$$

$$\mathrm{w}(M_{1,2}) \geq \mathrm{w}(x_{1,2}) - 2w_{max}$$

3. Applying the same merging procedure to $M_{1,2}$ and M_3, we obtain a matching M:

$$l_i(M) \leq l_i(x^*) \leq L_i \text{ for } i = 1,2$$

$$\mathrm{w}(M) \geq \mathrm{w}(x^*) - 6w_{max}$$

How to extend to any constant number of budgets?

- Only steps that needs to be generalized (merging of matchings)
- Can we flip?
- Conjecture : Generalized Necklace Splitting Problem

Conclusions

- Is there a fully-polynomial PTAS (FPTAS) for 1-budgeted Spanning Tree?
- For 1-budgeted Matching? (a PTAS is known)

If we find an FPTAS for the second problem, we will have a deterministic algorithm for exact matching with polynomial weights, which is a long-standing open problem.

Thank you!