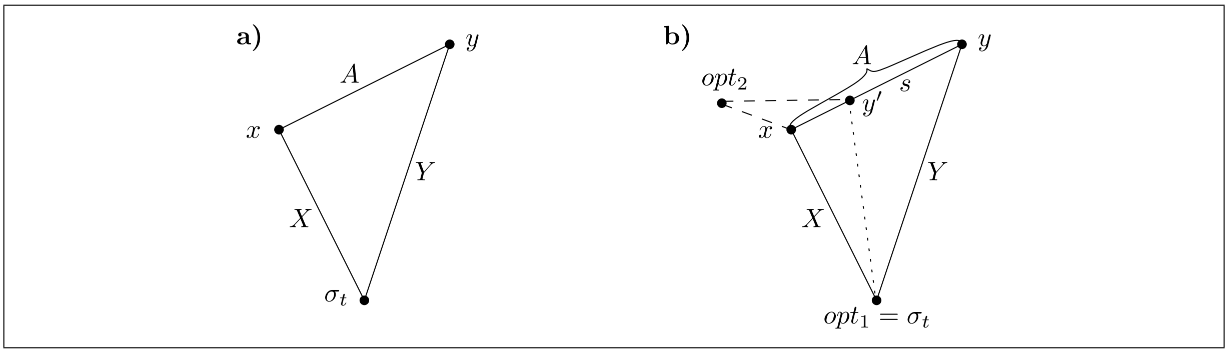

Rysunek 1: Pierwsza i druga częśc kroku algorytmu SlackCoverage

0.1 Algorytm Slack Coverage

W tej sekcji zdefiniujemy algorytm SlackCoverage (SC) działający dla 2 serwerów na płaszczyźnie i pokażemy, że jest on 3-konkurencyjny. Dla dowolnego $\gamma \in [0, 1]$ mamy:

Serwery algorytmu są w x, y; serwer x jest tym, który jest bliższy σ_t.
Niech A, X, Y będą zdefiniowane jak na rysunku ??a.
$s \leftarrow \gamma \cdot (X + A - Y)$.
Przesuń y o s w stronę x.
Przesuń x do σ_t.

Zanim udowodnimy konkurencyjność algorytmu SC zauważmy, że $\mathrm{SC}_{1/2}$ zastosowany na linii jest równoważny algorytmowi DC.

Twierdzenie 0.1 ([?])**.** *Algorytm* $\mathrm{SC}_{1/2}$ *jest* 3*-konkurencyjny.*

Dowód. W dowodzie wykorzystamy podobną funkcję jak w dowodzie konkurencyjności algorytmu DC. Niech a i b będą dowolnymi nieujemnymi liczbami rzeczywistymi, których wartości ustalimy później. Niech

$$\Phi = a \cdot M_{\min} + b \cdot d(x, y) \ ,$$

gdzie $M_{\min}$ jest ponownie kosztem minimalnego skojarzenia między serwerami SC a serwerami Opt. Jak w poprzednim dowodzie dzielimy krok t na dwie części i dla każdej z nich chcemy pokazać, że zamortyzowany koszt algorytmu jest nie większy niż koszt Opt razy współczynnik konkurencyjności.

W pierwszej części Opt przesuwa swoje serwery. Przy przesunięciu dowolnego z nich na odległość d, koszt minimalnego skojarzenia może wzrosnąć o d, a zatem potencjał może wzrosnąć o $a \cdot \mathrm{Opt}$. Zatem w pierwszej części kroku algorytm jest a-konkurencyjny.

Dla pokazania konkurencyjności w drugiej części kroku obliczmy najpierw koszt algorytmu SC. SC płaci X za przesunięcie serwera x oraz s za przesunięcie serwera y.

$$\mathrm{SC} = X + s \ .$$

Jaka jest zmiana potencjału związana ze zmianą odległości między x i y? Na początku kroku ta odległość wynosiła A. Serwer y przesuwa się o $\gamma \cdot (X + A - Y) \leq \gamma \cdot A$ po prostej łączącej x i y, przybliżając się do σ_t. Niech y' oznacza nową pozycję serwera y, tak jak na rysunku ??b. Zatem $d(y', \sigma_t) \leq d(y, \sigma_t) = Y$. Stąd otrzymujemy

$$\Delta d(x, y) \leq Y - A \ .$$

Jaka jest zmiana wartości $M_{\min}$? Na początku drugiej części kroku Opt musi mieć jeden z serwerów (nazwijmy go opt_1) w punkcie σ_t. Nazwijmy drugi z serwerów opt_2. Rozważmy dwa przypadki.

Przypadek 1. Na początku drugiej części kroku x jest skojarzony z σ_t. Wtedy ruch x powoduje zmniejszenie się kosztu skojarzenia o X, a ruch y powoduje zwiększenie się $M_{\min}$ o co najwyżej s. Stąd

$$\Delta M_{\min} \leq s - X \ .$$

Zatem dla dowodu konkurencyjności wystarczy, żeby

$$X + s + a \cdot (s - X) + b \cdot (Y - A) \leq 0 \tag{1}$$

Przypadek 2. Na początku drugiej części kroku x jest skojarzony z opt_2 (a więc y jest skojarzony z σ_t). Zatem początkowa wartość $M_{\min}$ to $d(x, opt_2)+Y$. Po przesunięciu serwerów w minimalnym skojarzeniu x zostaje skojarzony z σ_t, a y z opt_2. Zatem końcowa wartość $M_{\min}$ to $d(y', s_2) \leq (A-s)+d(x, opt_2)$. Stąd w tym przypadku

$$\Delta M_{\min} \leq A - s - Y \quad .$$

Zatem dla dowodu konkurencyjności wystarczy, żeby

$$X + s + a \cdot (A - s - Y) + b \cdot (Y - A) \leq 0 \tag{2}$$

Łatwo sprawdzić, że dla $\gamma = 1/2$, $a = 3$ i $b = 2$ nierówności **??** i **??** są spełnione, a zatem algorytm SC jest a-konkurencyjny. ■