חלקי הפרויקט:

1. סיכום המאמר – בפגישה עם תמיר הוא אמר לנו שנסכם את פרקים 1-3 של המאמר. הוספנו גם סיכום קצר של פרק 4 על TS במובן כללי יותר.
2. החלק הפרקטי – תמיר אמר לנו להשוואת את TS לucb1 שראינו בכיתה. מימשנו את שני האלגוריתמים על dataset שמצאנו באינטרנט של הקלקות על מודעות. והוספנו הסברים על הסקה בייסיאנית באופן כללי מול בגישה הפריקונסיסטית ועל הביצועים השונים שלהם על הdataset בבעית MAB

A Tutorial on Thompson Sampling - Daniel J. Russo

סוכם ע"י נעם שמיר וגיא חדד

רקע: המאמר סוקר את אלגוריתם Thompson Sampling (TS) ומציג שימושים ואספקטים שונים של האלגוריתם.
האלגוריתם פורסם לראשונה ב1933 ע"י ווילאם תומפסון כפתרון לבעיית הבנדיט עם שתי ידיות (תוצג בהמשך). האלגוריתם לא עורר עניין רב עד שבשנות ה- 2000 עם עידן המידע הוא חזר לכותרות. בשנים האחרונות יש התעניינות עצומה באלגוריתם והתאמות שלו נכנסו לשימוש בתחומים שונים כמו מערכות המלצה ו*a/b testing* למשל בבדיקות אפקטיביות של מודעות. שימושים שונים של האלגוריתם מהווים מרכיב חשוב בתחום המחקר ופיתוח בחברות רבות.

1. הקדמה

בעיית multi-armed bandit הייתה מוקד מחקר אינטנסיבי בתחומי הסטטיסטיקה, חקר ביצועים, הנדסת חשמל, מדעי המחשב וכלכלה. הבעיה מתארת מהמר שנכנס לקזינו עם מכונת מזל ולה כמה ידיות. לכל ידית יש הסתברות לזכייה בפרס שאינה ידועה למהמר ובלתי תלויה בתוצאות המשחקים הקודמים. המהמר מעוניין להשיג את ערך מקסימלי בסה״כ בכל המשחקים שלו. לשם כך הוא נוקט בגישה של explore -exploit, שלה 2 מטרות סותרות. האחת – לבחור בזרוע שתמקסם את הרווח בתור הנוכחי על סמך הידע שברשותו. השניה - לבחור בזרוע שתאפשר גילוי וחקירה של כיוונים אחרים (שאולי יתגלו כרווחיים יותר). שתי הגישות באות לידי ביטוי במדיניות המהמר ע״י משקל הסתברותי שונה שניתן לכל אחת מהן לאור המידע הקיים. האיזון והשילוב המתאים בין שתיהן הוא explore -exploit trade off.

העניין בסוג הבעיה הזו נובע מכך שאין התייחסות רק למידע ההיסטורי כדי להפיק תובנה כמו שעושים ב*supervised machine learning* אלא יש גם מימד של גילוי מידע חדש שעלול להגביר את הביצועים העתיידים.

דוגמא 1.1 – Bernoulli Bandit – בהנחה שיש לנו K פעולות שונות, כאשר לכל פעולה יש אופציה של כישלון/הצלחה. נסמן הצלחה ברווח של 1 וכשלון כרווח של 0. לכל ידית יש הסתברות $\theta_k \in [0,1]$ שלא ידועה לסוכן אך היא קבועה. הסוכן צריך למקסם את ההצלחות שלו למשך *T* סבבים כאשר *T>K*.

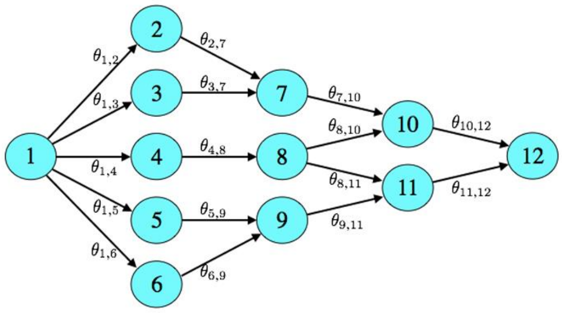
גירסה נאיבית לפתרון, תקציב משך זמן קבוע לחקירה של כל ידית בצורה רנדומלית אחידה ולאחר מכן בחירה בידית בעלת התוצאות הטובות ביותר. נשים לב שזאת גישה בזבזנית מאוד והיא עלולה להיכשל לגמרי במקרים מורכבים יותר. אלגוריתם *TS* מציע פתרון יעיל ומהיר יותר כפי שיוצג בהמשך.

אפשר לדמות את הבעיה הזאת לבעיות שימושיות רבות כמו מודעות פרסום באינטרנט כאשר הפרס יהיה מספר הקליקים והמודעות השונות יהיו הידיות השונות.

נציג דוגמה נוספת שממדלת עולם מורכב יותר:

דוגמא 1.2 – בעיית המסלול הקצר – סוכן הולך מביתו לעבודה מדי יום אך אינו יודע מה המסלול הקצר ביותר מבין כל הדרכים האפשריות. נגדיר גרף *G = (V,E)* כאשר צומת מסמל

צומת במסלול וקשת מסמלת דרך. משקל הקשת מסמן את הזמן הממוצע θ_e. נרצה למזער את $\sum_{t=1}^{T}\sum_{e\in x_t} y_t, es$ כאשר $y_e t$ מסמל את המרחק הקונקרטי שלקח בזמן t.



בעיית המסלול הקצר דומה מבחינה קונספטואלית לבעית המודעות, אך מספר האופציות האפשריות הוא גדול בהרבה, אקספונציאלי במספר הקשתות. בפתרונות נאיביים – הפתרון הוא ארוך בהרבה, עד כדי בלתי אפשרי. כמובן שניתן להוסיף עוד אלמנטים לשאלה שיכולים להפוך אותה לריאליסטית יותר כמו קורלציה בין הקשתות השונות או הגדרה אחרת של הבעיה כזמן כולל בלבד. מסיבות אלו - נרצה לחפש מודל ׳גמיש׳ יותר שנוכל להתאים לו סגנונות שונים של הבעיה. הפתרון לכך הוא אלגוריתם – *Thompson Sampling*.

2. החלטות חמדניות

אלגוריתמים חמדניים הם אולי הגישה הנפוצה והפשוטה ביותר לבעיית החלטה *online*.
שיטת הפעולה תהיה לרוב כזו:

1. נבנה מודל על סמך ההיסטוריה שידועה לנו.
2. נבחר את הפעולה האופטימלית על סמך המודל הידוע לנו והאומדן לפרמטר של ההתפלגות $\hat{\theta}$.

נגדיר את תוחלת הפרס כ$r_t = r(y_t)$

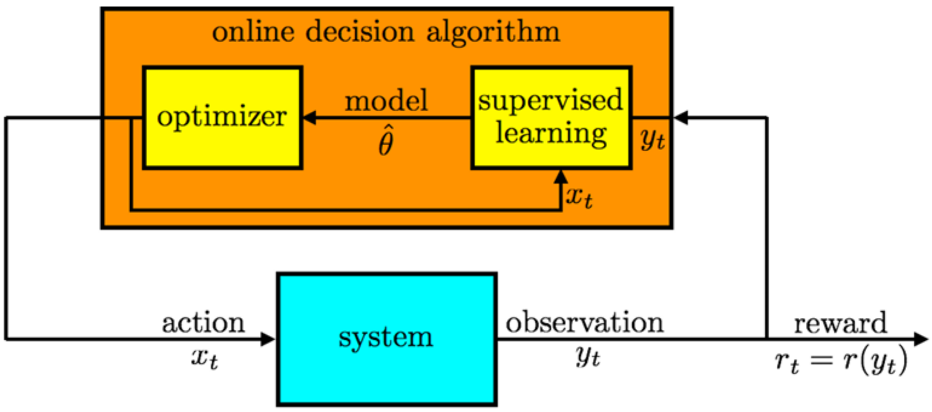


Figure 2.1: Online decision algorithm.

נתאר הצלחה כפרס של 1 וכישלון כ0. ידיעת הסוכן תתבטא במושגים של התפלגות פוסטריורית.

נראה זאת בדוגמא כאשר *K=3* ו $\theta \in R^3$. ננסה את פעולות 1 ו2 1000 פעמים כל אחת ואת פעולה 3 ננסה רק 3 פעמים.

ונניח שמספר ההצלחות שקיבלנו הוא 600, 400 ו1 בהתאמה. עם שיקלול של פריור יוניפורמי - נקבל הסתברויות פוסטריורית של 0.6, 0.4 ו0.4 בהתאמה. נדגיש שלגבי פעולה 3 תהיה לנו אי ודאות מאוד גבוהה ביחס לשתי הפעולות הראשונות.

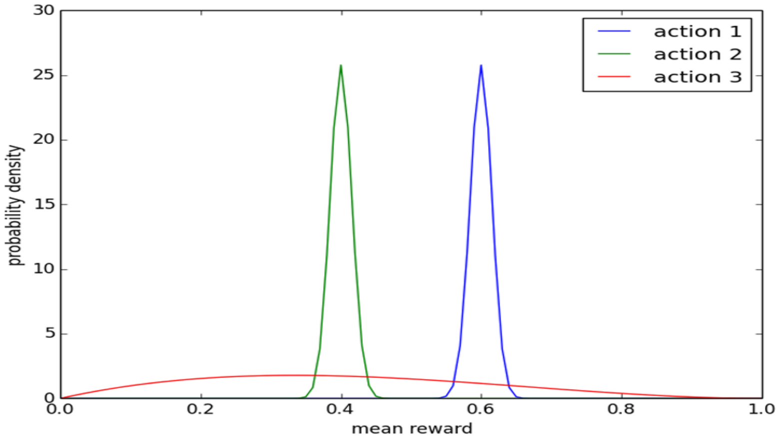


Figure 2.2: Probability density functions over mean rewards.

אלגוריתם חמדני יבחר כנראה את פעולה 1 אינסוף פעמים. נראה הגיוני להעדיף את פעולה 1 על 2 בגלל שלגמרי לא סביר סטטיסטית ש$\theta_2 > \theta_1$. מצד שני – אלגוריתם חמדני לא יבדוק אולי $\theta_3 > \theta_1$ למרות חוסר הודאות לגבי ערך האומדן.

Dithering היא גישה לחקור פעולות בצורה רנדומלית כאשר בהדרגה עוברים לאלגוריתם חמדני. גרסה נפוצה של הגישה נקראת $greedy - \epsilon$ שבה אנחנו בוחרים פעולה חמדנית בהסתברות של $1 - \epsilon$, ובהסתברות ϵ בוחרים פעולה בצורה רנדומלית. הבעיה בגישה הזו שהיא בוחרת גם פעולות שאין סיבה לחקור אותם כמו פעולה 2 בדוגמא לעיל. *TS* באה להתמודד גם עם בעיה זו.

3. *TS for the Bernoulli Bandit*

נסתכל שוב על דוגמה 1.1 - Bernoulli Bandit. נשנה את האלגוריתם כך שהפעם הסוכן יגדיר את הפריור שלו מהתפלגות בטא:

$$p(\theta_k) = \frac{\Gamma(\alpha_k + \beta_k)}{\Gamma(\alpha_k)\Gamma(\beta_k)} \theta_k^{\alpha_k - 1}(1 - \theta_k)_k^{\beta - 1}$$

ונעדכן את ההתפלגות שלנו לפי כלל בייס. נעדכן את הפרמטרים לפי הכלל הבא:

$$(\alpha_k, \beta_k) \leftarrow \begin{cases} (\alpha_k, \beta_k), & if\ x_t \neq k \\ (\alpha_k, \beta_k) + (r_t, 1 - r_t,), & if\ x_t = k \end{cases}$$

הממוצע של פונקציית בטא $\frac{\alpha_k}{\alpha_k + \beta_k}$ והדחיסות של ההסתברות גדלה ככל ש$\alpha_k + \beta_k$.

Algorithm 1 BernGreedy(K, α, β)

for $t = 1, 2, \ldots$ **do**
  #estimate model:
  for $k = 1, \ldots, K$ **do**
    $\hat{\theta}_k \leftarrow \alpha_k / (\alpha_k + \beta_k)$
  end for

  #select and apply action:
  $x_t \leftarrow \text{argmax}_k \, \hat{\theta}_k$
  Apply x_t and observe r_t

  #update distribution:
  $(\alpha_{x_t}, \beta_{x_t}) \leftarrow (\alpha_{x_t} + r_t, \beta_{x_t} + 1 - r_t)$
end for

Algorithm 2 BernTS(K, α, β)

for $t = 1, 2, \ldots$ **do**
  #sample model:
  for $k = 1, \ldots, K$ **do**
    Sample $\hat{\theta}_k \sim \text{beta}(\alpha_k, \beta_k)$
  end for

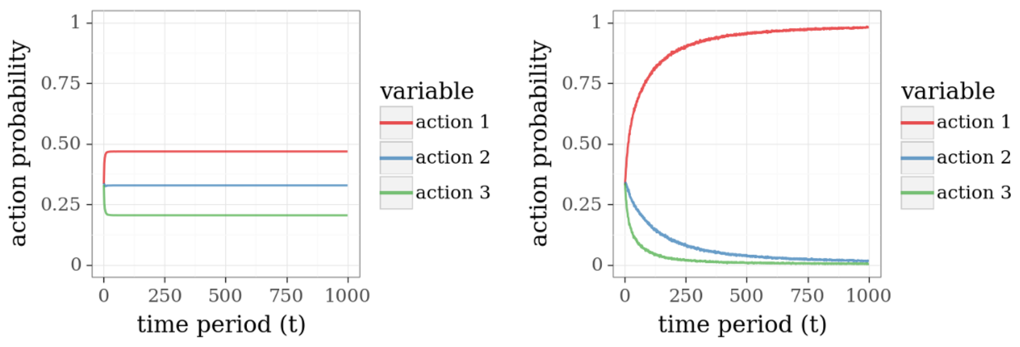
  #select and apply action:
  $x_t \leftarrow \text{argmax}_k \, \hat{\theta}_k$
  Apply x_t and observe r_t

  #update distribution:
  $(\alpha_{x_t}, \beta_{x_t}) \leftarrow (\alpha_{x_t} + r_t, \beta_{x_t} + 1 - r_t)$
end for

באלגוריתם 1 נאמוד את הפרמטר $\hat{\theta} = \frac{\alpha_k}{\alpha_k + \beta_k}$

באלגוריתם 2 – *TS* – ההבדל הוא שאנחנו דוגמים מפונקציית הבטא שקיבלנו. היתרון של *TS* על עקרון ה *dithering* שהוצג הוא שהאלגוריתם לא נותן בחקירה משקל שווה לכל הפעולות אלא מתחשב בהתפלגות הבטא על סמך הידע הקודם. *TS* עוזר לחקור יותר במקרה שיש אי ודאות עם סיכוי לאופטימליות עתידית.

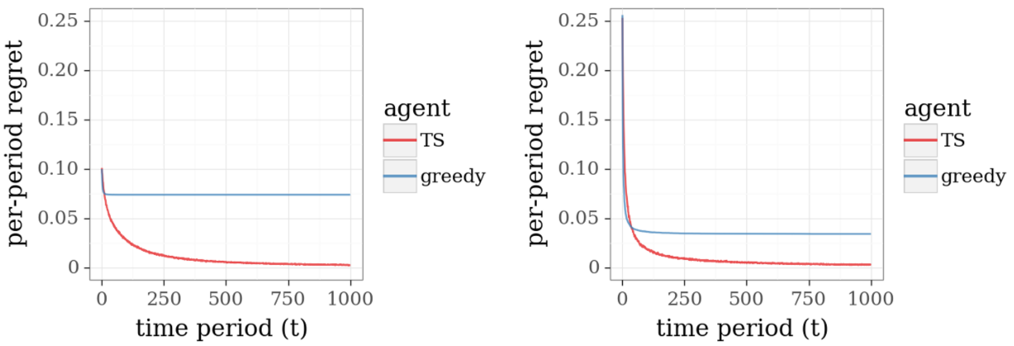
באלגוריתם חמדני - לא תמיד נקבל אפילו התכנסות לנקודה אופטימלית גם בגרסה הבסיסית של הבעיה.
למשל אם $\theta_1 = 0.9$ $\theta_2 = 0.8$ $\theta_3 = 0.7$ -
אלגוריתם חמדני לא תמיד יתכנס לפעולה 1 אף על פי שהיא הטובה ביותר.
האלגוריתם יכול 'להיתקע' על פעולה מסויימת גם אם היא לא אופטימלית, רק בגלל שבהתחלה היא נראית אופטימלית ולכן הוא ממשיך איתה.
למשל בדוגמה שלנו - אם האלגוריתם העריך את פעולה 1 ופעולה 2 ב *0.5* ובחר את פעולה 3 בכל האיטרציות עד ההתכנסות ל0.7.



(a) greedy algorithm (b) Thompson sampling

Figure 3.1: Probability that the greedy algorithm and Thompson sampling selects an action.

בדרך כלל, אלגוריתמים של *online decision* נמדדים לפי החרטה (*regret*). נגדיר את החרטה לתור כ $max_k \, \theta_k - \theta_{x_t}$. אגב, מקובל לבדוק θ שונים כדי למנוע הטיות. נראה תוצאות של חרטה ביחס לדוגמא לעיל עם תוצאות על בחירת ערכים בצורה רנדומית.



(a) $\theta = (0.9, 0.8, 0.7)$ **(b)** average over random θ

Figure 3.2: Regret from applying greedy and Thompson sampling algorithms to the three-armed Bernoulli bandit.

4. *TS* - הגדרה כללית

עכשיו נרצה להגדיר *TS* בצורה כללית. נגדיר סט ... *x1, x2, x3* של פעולות בגודל סופי או אינסופי. אחרי פעולה x_t נקבל תוצאה y_t שמוגרל מ $q_\theta(*\,|x_t)$. הסוכן נהנה מפרס של $r_t = r(y_t)$.

אנחנו נרצה למקסם את תוחלת הפרס שמבוטא ע"י:

$$E_{q_\theta}[r(y_t)|x_t = x] = \sum_o q_\theta(o|x)r(o)$$

כאשר ההסתברות p מתעדכנת בהתאם ל $\hat{y}_t$. ואם θ סופי אז ניתן לרשום את ההסתברות המותנת ע"י כלל בייס:

$$P_{p,q}(\theta = u\,|x_t, y_t) = \frac{p(u)q_u(y_t|x_t)}{\sum_v p(v)q_v(y_t|x_t)}$$

Algorithm 3 Greedy$(\mathcal{X}, p, q, r)$

for $t = 1, 2, \ldots$ **do**
  #estimate model:
  $\hat{\theta} \leftarrow \mathbb{E}_p[\theta]$

  #select and apply action:
  $x_t \leftarrow \text{argmax}_{x\in\mathcal{X}} \mathbb{E}_{q_{\hat{\theta}}}[r(y_t)|x_t = x]$
  Apply x_t and observe y_t

  #update distribution:
  $p \leftarrow \mathbb{P}_{p,q}(\theta \in \cdot|x_t, y_t)$
end for

Algorithm 4 Thompson$(\mathcal{X}, p, q, r)$

for $t = 1, 2, \ldots$ **do**
  #sample model:
  Sample $\hat{\theta} \sim p$

  #select and apply action:
  $x_t \leftarrow \text{argmax}_{x\in\mathcal{X}} \mathbb{E}_{q_{\hat{\theta}}}[r(y_t)|x_t = x]$
  Apply x_t and observe y_t

  #update distribution:
  $p \leftarrow \mathbb{P}_{p,q}(\theta \in \cdot|x_t, y_t)$
end for

בשני האלגוריתמים נרצה למקסם את התוחלת של הפרס. כאשר נעדכן לפי הכלל:

$$(\alpha_k, \beta_k) \leftarrow (\alpha + r_t 1_{xt}, \beta + (1 - r_t) 1_{xt})$$

כאשר 1_{x_t} הוא וקטור שכל הערכים בכניסות התואמות לפעולה שנבחרה שווה ל1 והאחרים שווים ל0.

דוגמא 4.1 - *independent travel time* - נתבונן ששוב בבעיה 1.2 – המסלול הקצר בגרף. נניח שכל θ_e נדגם בצורה בלתי תלויה מהתפלגות לוג-נורמלית ולכן $\ln(\theta_e) \sim N(\mu_e, \sigma_e^2)$. נקבל שכל $e \in E$ מתפלג לוג-נורמלי עם פרמטרים $\ln(e) - \frac{\sigma^2}{2}$ וσ^2 כך ש$E\left[y_{t_e} \middle| \theta_e\right] = \theta_e$. תכונת ההצמדה (*Conjugacy*) מאפשרת כלל עדכון פשוט להתפלגות:

$$(\alpha_k, \beta_k) \leftarrow \left(\frac{\frac{1}{\sigma_e^2}\mu_e + \frac{1}{\hat{\sigma}_e^2}\left(\ln\left(y_{t_e}\right) + \frac{\hat{\sigma}_e^2}{2}\right)}{\frac{1}{\sigma_e^2} + \frac{1}{\hat{\sigma}_e^2}}, \frac{1}{\frac{1}{\sigma_e^2} + \frac{1}{\hat{\sigma}_e^2}}\right)$$

המוטיבציה מאחורי צורת ייצוג כזאת היא שהסוכן יודע את המרחק d_e אך לא בטוח לגבי זמני ההגעה שלו. לכן זה יהיה טבעי לבחור את הפריור כך שהתוחלת תהיה שווה למרחק. לכן על ידי קיבוע של התוחלת $\mu_e = \ln(d_e) - \frac{\sigma_e^2}{2}$ ובנוסף נקבל גם רמה של חוסר ודאות מהפרמטרים. פריור השונות של ממוצע זמן הנסיעה לקשת יהיה $d_e^2(e_e^{\sigma_e{}^2} - 1)$.

נוכל להריץ את אלגוריתמים 3 ו4 בצורה יעילה חישובית. שניהם יאתחלו כל איטרציה t עם פרמטרים (μ_e, σ_e). האלגוריתם החמדני יציב את $\hat{\theta} = E_p[\theta_e] = e_e^{\mu + \frac{\sigma_e^2}{2}}$ ואילו *TS* ידגום מהתפלגות לוג-נורמלית את $\hat{\theta}$. כל אלגוריתם יבחר את הפעולה *x* אשר ממקסמת את תוחלת הפרס. אפשר למצוא בכל איטרציה את הפתרון ע"י אלגוריתם *Dijkstra* למשל. לאחר מכן נעדכן את הפרמטרים לפי הכלל שהצגנו לעיל.

במאמר מתואר ניסוי של האלגוריתם על גרף בעל 184,756 דרכים שונות להגיע ליעד. התוצאות מראות ש*TS* התכנס מאוד מהר לאופטימום בעוד שהאלגוריתם החמדני היה רחוק מהאמת. בנוסף בדקו גם $\epsilon - greedy$ עם ערכים שונים. קל לראות ששיטה זו איטית משמעותית מ*TS*.

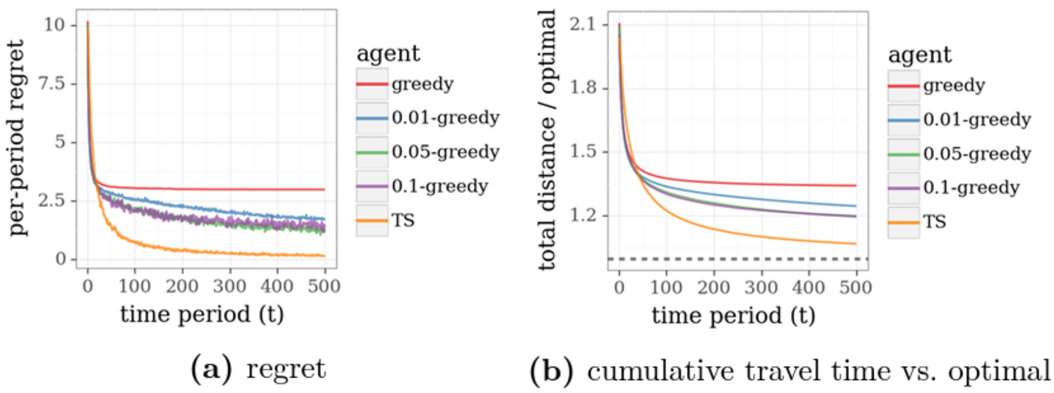


Figure 4.1: Performance of Thompson sampling and ϵ-greedy algorithms in the shortest path problem.

דוגמא 4.2 – ***correlated travel times*** **–** בדוגמא 4.1 הנחנו אי תלות, עכשיו נניח שהתפלגות הנצפית תתאפיין כך: $y_{t,e} = \zeta_{t,e}\eta_t\nu_{t,l(e)}\theta_e$ כאשר $\zeta_{t,e}$ מייצג את ה- *idiosyncratic* ביחס לקשת. η_t מייצג את הפקטור שמשוייך לכל הקשתות. *L(e)* הוא אינדיקטור אם הקשת מעל או מתחת של הגדר הבינומי. ו- ν מסמן את ההשפעה על הקשתות מעליו ומתחתיו.

שוב משתמרת תכונת ה *Conjugacy* ולכן מאפשרת לעדכן בצורה יעילה את ההתפלגות:

$$\phi_e = \ln(\theta_e) \qquad and \qquad z_{t,e} = \begin{cases} \ln(y_{t,e}), & if\ e \in x_t \\ 0, & otherwise \end{cases}$$

ונעדכן את מטריצת השונות בהתאם.

כעת, נוכל להריץ את *TS* בצורה יעילה חישובית. גם פה נרצה למקסם את תוחלת הפרס ונשתמש ב*Dijkstra* כדי למצוא את המסלול הקצר ביותר ואז נעדכן את ההתפלגות לפי הכלל לעיל.

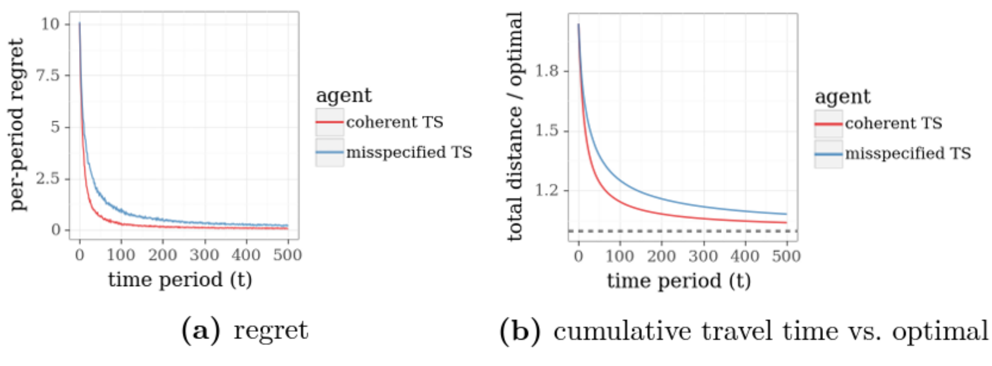


Figure 4.3: Performance of two versions of Thompson sampling in the shortest path problem with correlated travel times.