System Time Distribution of Dynamic Traveling Repairman Problem under the PART-n-TSP Policy

Jiangchuan Huang and Raja Sengupta

***Abstract*— We propose the PART-n-TSP policy for the Dynamic Traveling Repairman Problem [1]. We compute a good approximation for the distribution of the system time, defined as the elapsed time between the arrival and the completion of each task. PART-n-TSP stabilizes the system for every load in $[0, 1)$. PART-n-TSP has lower system time variance than PART-TSP [14] and Nearest Neighbor [1] when the load is neither too small or too large. We show that PART-n-TSP is also optimal for system time expectation under light and heavy loads.**

I. INTRODUCTION

The Dynamic Traveling Repairman Problem (DTRP) [1] is an important variant of the vehicle routing problem: A convex region $\mathbf{A}$ of area A contains a moving server that travels at constant speed v. Tasks arrive according to a Poisson process with rate λ and have a location that is independent and identically distributed (i.i.d.) according to the probability density function (pdf) $f_X(x)$ within $\mathbf{A}$. Each task i has size B_i. B_i is i.i.d. according to $f_B(s)$. $E[B_i] = b$, which is assumed to be finite. Define load $\rho = \lambda b$. The system time of task i, denoted T_i, is defined as the elapsed time between the arrival and the completion of task i. When the system is stable, T_i converges to some T in distribution. T is called the (steady state) system time.

The expectation of system time $E[T]$ is the primary metric of DTRP because of its stochastic setting. However, the expectation is not enough to measure performance. On entering a McDonalds, one may ask not just "What is my expected service time?" but also "How certain is this value?" Users demand not only system times that are fast on average, but also system times that are predictable. In practice, highly variable system time can be even more frustrating than large expected system times [11]. Extending the expectation of system time to the distribution of system time is essential for applicability. Also, knowing the distribution of the system time T, together with its expectation $E[T]$ and variance $Var[T]$, enables the expectation-variance analysis of the system under uncertainties [5], since two policies at the same load level can be incomparable in the sense that one has low expectation of system time but high variance while the other has high expectation of system time but low variance. See for example Table I in Section III. The literature discusses the distribution of system time T, together with its expectation and variance, for the First Come First Serve (FCFS) policy and its variations such as the Stochastic Queue Median (SQM) policy and Partitioning-FCFS (PART-FCFS) [1]. This is in sharp contrast to queueing theory where the distribution of the system time or its moments are known for a wide variety of policies. See for example [13]. To illustrate our point, the expected system time of the FCFS, SQM and PART-FCFS policy is not as good as Nearest Neighbor (NN) [1] and Partitioning-Traveling Salesman Policy (PART-TSP) [14] or Divide & Conquer (DC) [12] at most load levels.

In this paper, we propose a policy called the Partitioning-n-Traveling Salesman Policy (PART-n-TSP) for the DTRP, and give a good approximation for the distribution of the system time that is easy to compute under this policy. We do this by utilizing approximation results for the distribution of system time T, together with $E[T]$ and $Var[T]$ known for polling systems [7]. Figure 5 shows that the cumulative distribution function (cdf) of the system time as computed by our method is very close to the cdf of the system time as obtained by Monte-Carlo simulation. We show that FCFS, PART-FCFS and TSP [1] are special cases of PART-n-TSP, meaning PART-n-TSP can be optimized to have better performance than the three. We also compare PART-n-TSP with PART-TSP [14] and NN [1] on $E[T]$ and $\sigma[T]$ in Table I, since the latter two are considered near optimal in the literature. The $E[T]$ and $\sigma[T]$ under PART-n-TSP are obtained by our approximation. The $E[T]$ and $\sigma[T]$ under PART-TSP and NN are obtained by simulation. The results show that NN achieves lower $E[T]$ than both PART-n-TSP and PART-TSP for all loads $\rho \in \{0.1 \dots 0.9\}$ simulated by us. PART-n-TSP achieves lower $E[T]$ than PART-TSP when ρ is not too small or too large, e.g. when $\rho \in \{0.3, \dots, 0.7\}$. Also, PART-$n$-TSP achieves lower $\sigma[T]$ than PART-TSP and NN when ρ is not too small or too large, e.g. when $\rho \in \{0.3, \dots, 0.7\}$. In real systems it may be desirable for ρ to be neither too small nor too large, since small ρ results in low server utilization, and large ρ in large system times. If so, PART-n-TSP would be good in practice as it achieves lower $\sigma[T]$ than PART-TSP and NN, and lower $E[T]$ than PART-TSP. Our approximation method also holds when the Poisson task arrival assumption is relaxed to renewal arrival. See Section II. We also show that PART-n-TSP is $E[T]$ optimal under light load ($\rho \to 0^+$) and asymptotically optimal under heavy load ($\rho \to 1^-$). See Theorem 1. The multi-customer variant of the vehicle routing problem is discussed in [9].

II. SYSTEM TIME OF PART-n-TSP

A. *Partitioning-n-Traveling Salesman Policy*

Bertsimas et al. [1] introduced the traveling salesman policy (TSP). It is based on collecting tasks into sets of

This work was partly supported by NSF-CNS-1136141.

Dr. J. Huang and Dr. R. Sengupta are with Systems Engineering Group, Dept. Civil and Environmental Engineering, University of California at Berkeley. jiangchuan.huang@gmail.com

size n that are then served in a TSP path. To be precise, we call this the n-TSP. We generalize it to the PART-n-TSP as follows.

The PART-n-TSP includes the polling phase and the sequencing phase. The polling phase involves generating an r-partition $\{\mathbf{A}^k\}_{k=1}^{r}$ of the convex region $\mathbf{A}$ that is simultaneously equitable with respect to $f_X(x)$. In particular, when the region $\mathbf{A}$ is a square region $\mathbf{A}$ with size $a \times a$, and the tasks are uniformly distributed in $\mathbf{A}$ with pdf $f_X(x) = \frac{1}{A}$, $\mathbf{A}$ is divided into $r = m^2$ square partitions, each has size $\frac{a}{m} \times \frac{a}{m}$, where $m \geq 1$ is a given integer that parameterizes the policy. The server visits the partitions in a cyclic order. Partitions are numbered so that for any $k = 1, \ldots, r-1$, partition $k+1$ is adjacent to partition k, and partition r is adjacent to partition 1 when m is even, or to the diagonal of partition 1 when m is odd, as illustrated in Figure 1 for the case $m = 4$ and $m = 5$. The server cycles through the partitions in the order $1, \ldots, r, 1, \ldots, r, \ldots$. After the server finishes the tasks polled in the current partition under a sequencing policy, the server moves to an adjacent partition and serves it under the same sequencing policy.

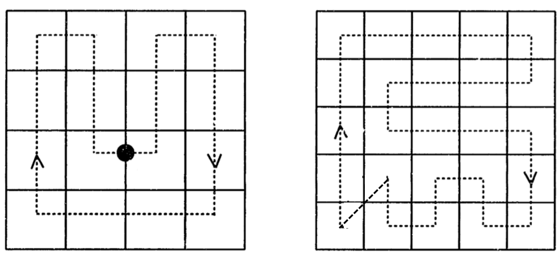

Fig. 1. Partition serving order in polling phase, cited & revised from [14].

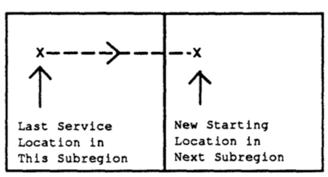

Fig. 2. Server moving to adjacent partition, cited from [1].

To move from one partition (polling station) to the next, the server uses the projection rule shown in Figure 2 as introduced in [1]. Its last location in a given partition is simply "projected" onto the next partition to determine the server's new starting location. The server then travels in a straight line between these two locations. This makes the distance traveled between partitions a constant, each starting location uniformly distributed, and independent of the locations of tasks in the new partition. In practice, one might use a more intelligent rule such as moving directly to the first or nearest task in the next partition. When m is even, the server always travels to an adjacent partition. Thus the switch time between two consecutive partitions is always $\Delta^k = \frac{a}{m}$, $k = 1, \ldots, r$. Thus the switch time of a cycle $\Delta = \sum_{k=1}^{r} \Delta^k = ma$. When m is odd, the server always travels to an adjacent partition except the last one. Thus $\Delta^k = \frac{a}{m}$ when $k = 1, \ldots, r-1$, and $\Delta^r = \frac{\sqrt{2}a}{m}$ as shown in Figure 1. Thus $\Delta = \sum_{k=1}^{r} \Delta^k = \frac{(m^2-1+\sqrt{2})a}{m}$. Then we have

$$\Delta = \begin{cases} ma, & \text{if } m \text{ is even.} \\ \frac{(m^2-1+\sqrt{2})a}{m}, & \text{if } m \text{ is odd.} \end{cases} \qquad (1)$$

In the sequencing phase, we use the exhaustive n-TSP policy to sequence the tasks in each partition. This policy is adopted from the TSP policy in [1]. We repeat it for the convenience of the reader. Let $\mathcal{N}_l^k$ denote the l-th set of n tasks to arrive in partition k. Each set $\mathcal{N}_l^k$ has cardinality n. For example, $\mathcal{N}_1^k$ is the set of tasks $1, \ldots, n$ in partition k, and $\mathcal{N}_2^k$ is the set of tasks $n+1, \ldots 2n$ in partition k, and so on. To serve a set, we form a TSP path of the n tasks in the set starting at the initial position of the server and ending at the location of the last task in the TSP path. A TSP path is the Hamiltonian path with the minimum length among all the Hamiltonian paths. A Hamiltonian path is a path that visits each task location exactly once starting at the initial position of the server.

The server starts at some location in partition 1. If all tasks in set $\mathcal{N}_1^1$ have arrived, we form a TSP path over these tasks. Tasks are then served by following the TSP path. If all $\mathcal{N}_2^1$ tasks have arrived when the TSP path of $\mathcal{N}_1^1$ is completed, they are also served using a TSP path. Otherwise, the server moves to partition 2, and so on. The sets N_l^k are served in an FCFS order in each partition k.

The PART-n-TSP is in the class of Polling-Sequencing policies introduced in [10]. We know that $\rho + \lambda b_d < 1$ is the stability condition by Theorem 6 in [10], and the sequencing phase TSP has $b_d = 0$ by Remark of Theorem 5 in [10]. Thus PART-n-TSP is stable if and only if $\rho < 1$.

B. Computation of System Time Distribution

The system time T of a task has three components:

- W_O, the time a task waits for its set to form (wait for the last task in the set to arrive).
- W_P, waiting time of the set in the polling system.
- W_I, the time it takes to complete service of the task once the task's set enters service.

Thus,

$$T = W_O + W_P + W_I \qquad (2)$$

Since W_O, W_P and W_I are independent, the distribution of T can be obtained from the distributions of W_O, W_P and W_I through convolution. First, the distribution of W_O is obtained through the renewal assumption of task arrival. Second, if we treat the service time of each set as a job, and each partition as a polling station, then the system is a classic polling system on r polling stations with renewal arrival and known switch times. Thus W_P is the waiting time of each set (job) in this classic polling system. We utilize the closed form approximation for the classic polling system waiting time in [7] to compute W_P. Third, W_I is the sum of the TSP path traveled to reach the current task and the service times of the tasks before the current task in its set. The TSP path distribution is obtained by simulation and the task service times are known.

1) Distribution of W_O: We first obtain the distribution of W_O, together with its expectation and variance. Pick a random task. Let W_{Ol} be the waiting time of a task outside a set if it is the $(n-l)$-th task arrived in the set, $l = 0, \ldots, n-1$. Since we have equitable partitions, then the task arrival process inside each partition is Poisson with arrival rate $\frac{\lambda}{r}$. Thus $W_{O0} = 0$ and W_{Ol} is Erlang distributed with parameters $\left(l, \frac{\lambda}{r}\right)$, $l = 1, \ldots, n-1$. Thus the cdf of W_{Ol}, $F_{W_{Ol}}\left(t; l, \frac{\lambda}{r}\right) = 1 - \sum_{j=0}^{l-1} \frac{1}{j!} e^{-\frac{\lambda}{r}t} \left(\frac{\lambda}{r}t\right)^j$.

Since it is equally probable that a task is the $(n-l)$-th arrived task in the set, then
$P\left(W_O \le t\right) = \sum_{l=0}^{n-1} P\left(W_{Ol} \le t\right) \frac{1}{n}$
$= \frac{1}{n}\left(1 + \sum_{l=1}^{n-1}\left(1 - \sum_{j=0}^{l-1} \frac{1}{j!} e^{-\frac{\lambda}{r}t}\left(\frac{\lambda}{r}t\right)^j\right)\right)$. Thus,

$$P\left(W_O \le t\right) = \frac{1}{n}\left(1 + \sum_{l=1}^{n-1}\left(1 - \sum_{j=0}^{l-1} \frac{1}{j!} e^{-\frac{\lambda}{r}t}\left(\frac{\lambda}{r}t\right)^j\right)\right) \tag{3}$$

From the distribution of W_O, we get its mean and variance.

$$E\left[W_O\right] = \frac{(n-1)r}{2\lambda} \tag{4}$$

$$Var\left[W_O\right] = \frac{\left(n^2 + 6n - 7\right) r^2}{12\lambda^2} \tag{5}$$

2) Distribution of W_I: Before discussing W_P, we compute the distribution of W_I as follows. Let W_{Inj} be the waiting time of a task inside a set if it is the j-th task to be served in the set of n tasks, $j = 1, \ldots, n$. Then

$$W_{Inj} = \frac{D_{nj}}{v} + \sum_{i=1}^{j} B_i \tag{6}$$

where D_{nj} is the travel distance from the initial server position to the location of the j-th task through a TSP path in a partition. B_i is i.i.d.. D_{nj} is independent of B_i. Thus $\sum_{j=1}^{k} B_i$ is the convolution of j B_i's, and W_{Inj} is the convolution of $\frac{D_{nj}}{v}$ and $\sum_{i=1}^{k} B_i$.

Let L_{nj} be the D_{nj} value when there is only one partition, i.e., $r = 1$. When $r > 1$, we assume that

$$D_{nj} =_d \frac{cL_{nj}}{\sqrt{r}} \tag{7}$$

where $=_d$ means identically distributed with, and c is a positive constant. In particular, when region $\mathbf{A}$ and all the partitions $\mathbf{A}^k$ are squares, and $r = m^2$ with m an integer, $c = 1$.

We obtain the empirical distribution of L_{nj}, together with $E\left[L_{nj}\right]$ and $Var\left[L_{nj}\right]$ for different n and $j = 1, \ldots, n$ through simulation on the TSP path. The ant colony optimization algorithm [6] is used to heuristically search for the TSP path. Both the number of ants and the number of iterations are set to 1000. The distribution of D_{nj} is calculated from L_{nj} by scaling in (7). We write L_n for L_{nn} and D_n for D_{nn}. Figure 3 shows the values of $E\left[L_n\right]$ and $Var\left[L_n\right]$ for different n. Figure 4 shows the pdf of L_{nj} for a set of $n = 5$ tasks. The tasks are uniformly distributed on a square of size 1×1.

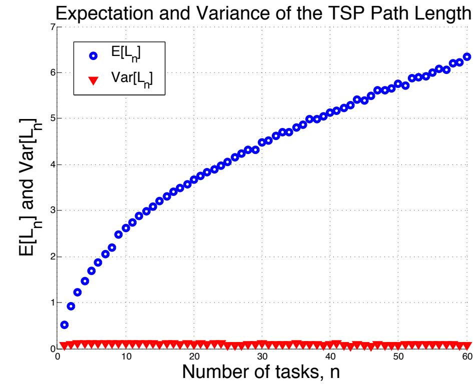

Fig. 3. Expectation & variance of TSP path length for n tasks in a square.

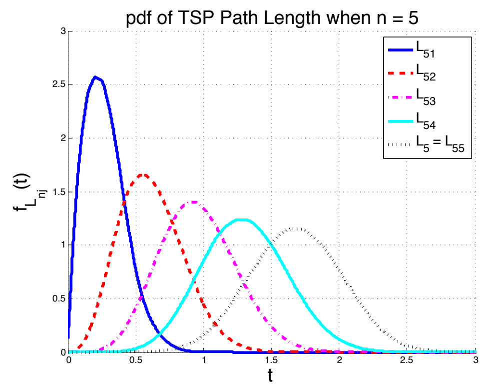

Fig. 4. The pdf of L_{nj} for 5 tasks uniformly distributed in a square.

Since it is equally probable that the task is the k-th served task in the TSP path, $k = 1, \ldots, n$, then

$$P\left(W_I \le t\right) = \frac{1}{n}\sum_{j=1}^{n} P\left(W_{Inj} \le t\right) \tag{8}$$

Then the pdf of W_I, $f_{W_I}(t) = \frac{1}{n}\sum_{j=1}^{n} f_{W_{Inj}}(t)$.
$E\left[W_I\right] = \frac{1}{n}\sum_{j=1}^{n} E\left[W_{Inj}\right]$.
$E\left[W_I^2\right] = \frac{1}{n}\sum_{j=1}^{n} E\left[W_{Inj}^2\right]$.
$Var\left[W_I\right] = E\left[W_I^2\right] - E\left[W_I\right]^2$.

The distribution of W_I can be calculated from (6) and (8). The distribution of W_I does not have a closed form, but can be arbitrarily accurate through simulation. Observe that in order to obtain the distribution of D_{nj} and W_I for partitions with different sizes parameterized by r, we do not need to rerun the simulation for each partition with different size. We only need to run it once for L_{nj} and store the data. D_{nj} and W_I are obtained by scaling and convolution.

3) Distribution of W_P: The analysis of W_P uses the results from [7] by establishing the PART-n-TSP to be equivalent to a classic polling system over jobs that are the sets $\mathcal{N}_l^k$.

Since the task arrival process is Poisson with arrival rate λ, the distribution of the interarrival time of sets, A, is Erlang of order n and arrival rate λ, i.e., $A \sim Erlang(n, \lambda)$. Let A_k be the interarrival time of sets that fall in partition $\mathbf{A}^k$. Then $A_k \sim Erlang(n, \frac{\lambda}{r})$. Thus

$$E[A_k] = \frac{nr}{\lambda}, Var[A_k] = \frac{nr^2}{\lambda^2} \tag{9}$$

The arrival rate of a set is

$$\lambda^s = \frac{\lambda}{n} \tag{10}$$

The arrival rate of a set in partition $\mathbf{A}^k$, $k = 1, \ldots, r$, is

$$\lambda_k^s = \frac{\lambda^s}{r} = \frac{\lambda}{nr} \tag{11}$$

The size of a set, or the time needed to travel to and execute all the tasks in the set, is W_{Inn} as given in (6). We write W_n for W_{Inn}. The size of each set W_n is i.i.d.. Thus, if we treat each set as a job with size W_n, and each partition as a polling station, then the system is a classic polling system on r polling stations with renewal (Erlang) arrival of rate λ^s, job size W_n, and switch time Δ^k. The load is

$$\rho^s = \lambda^s E[W_n] \tag{12}$$

The load in partition $\mathbf{A}^k$, $k = 1, \ldots, r$, is

$$\rho_k^s = \frac{\rho^s}{r} \tag{13}$$

W_P is the waiting time of each set (job) in this classic polling system. Exhaustive or gated PART-n-TSP correspond to exhaustive or gated FCFS on sets, respectively.

Dorsman et al. [7] provide closed form approximations for the distribution of the steady state waiting time of a job, W_P, for polling systems under a renewal arrival process with gated or exhaustive policies when the sequencing policy is FCFS. They claim that for exhaustive-FCFS policies,

$$P(W_P \le t) \approx P(UI \le (1 - \rho^s)t) \tag{14}$$

where U is uniformly distributed on $[0, 1]$, and I is Gamma distributed with parameters

$$\alpha = \frac{2E[\Delta]\delta}{\sigma^2} + 1, \beta = \frac{2E[\Delta]\delta + \sigma^2}{2\sigma^2(1 - \rho^s)E[W_{Boon}]}. \tag{15}$$

where $\Delta = \sum_{k=1}^{r} \Delta^k$ is the total switch time in a cycle. When the region $\mathbf{A}$ and partitions $\mathbf{A}^k$ are squares as shown in Figure 1, Δ is given in (1). ρ^s is given in (12).

To explain δ, σ^2 and $E[W_{Boon}]$, we denote by $\hat{y}$ the value of each variable y that is a function of ρ^s evaluated at $\rho^s = 1$. $\delta = \sum_{j=1}^{r} \sum_{k=j+1}^{r} \hat{\rho^s}_j \hat{\rho^s}_k$, where $\hat{\rho^s}_k$ is given in (13) evaluated at $\rho^s = 1$. Since we have equitable partitions, then

$$\hat{\rho^s}_k = \frac{1}{r} \tag{16}$$

for all $k = 1, \ldots, r$. Thus

$$\delta = \frac{r(r-1)}{2r^2} = \frac{r-1}{2r}. \tag{17}$$

Again by [7]
$\sigma^2 = \sum_{k=1}^{r} \hat{\lambda^s}_k \left(Var[W_n] + \hat{\rho^s}_k^2 Var\left[\hat{A}_k\right] \right)$.

Since $\hat{\lambda} = n\hat{\lambda^s}$ by (11), then $Var\left[\hat{A}_k\right] = \frac{nr^2}{\hat{\lambda}^2} = \frac{r^2}{n\hat{\lambda^s}^2}$ by (9). Also, $\hat{\lambda^s}_k = \frac{\hat{\lambda^s}}{r}$ by (11), then substituting (16) we have $\sigma^2 = \hat{\lambda^s} \left(Var[W_n] + \frac{1}{r^2} \frac{r^2}{n\hat{\lambda^s}^2} \right)$. Thus,

$$\sigma^2 = \hat{\lambda^s} \left(Var[W_n] + \frac{1}{n\hat{\lambda^s}^2} \right), \tag{18}$$

where by (12)

$$\hat{\lambda^s} = \frac{1}{E[W_n]}. \tag{19}$$

From (6) we know

$$E[W_n] = \frac{E[D_n]}{v} + nb \tag{20}$$

$$Var[W_n] = \frac{Var[D_n]}{v^2} + n\sigma_B^2 \tag{21}$$

where $E[D_n]$ and $Var[D_n]$ are obtained from $E[L_n]$ and $Var[L_n]$ by (7), and $E[L_n]$ and $Var[L_n]$ are obtained from simulation as shown in Figure 3. Thus σ^2 is known substituting (19), (20) and (21).

Finally by Boon et al. [3], for equitable partitions

$$E[W_{Boon}] = \frac{K_0 + K_1\rho^s + K_2(\rho^s)^2}{1 - \rho^s} \tag{22}$$

where $K_0 = E[\Delta^+]$. Δ^+ is called the residual of the random variable Δ with $E[\Delta^+] = \frac{E[\Delta^2]}{2E[\Delta]}$. In our case, Δ is deterministic. Thus,

$$K_0 = E\left[\Delta^+\right] = \frac{\Delta}{2} \tag{23}$$

with Δ given in (1). $K_1 = \hat{\rho^s}_k \left(\left(c_{\hat{A}_k}^2\right)^4 \mathbf{1}\{c_{\hat{A}_k}^2 \le 1\} + 2\frac{c_{\hat{A}_k}^2}{c_{\hat{A}_k}^2 + 1} \mathbf{1}\{c_{\hat{A}_k}^2 > 1\} - 1 \right) E[W_n^+]$ $+ E[W_n^+] + \hat{\rho^s}_k (E[\Delta^+] - E[\Delta])$, where

$$c_{\hat{A}_k}^2 = \frac{Var\left[\hat{A}_k\right]}{E\left[\hat{A}_k\right]^2}, \tag{24}$$

and $\mathbf{1}\{\Omega\}$ is the indicator function defined as $\mathbf{1}\{\Omega\} = 1$ if Ω is true, and $= 0$ otherwise. By (9) we have

$$c_{\hat{A}_k}^2 = \frac{1}{n}. \tag{25}$$

Substituting (1), (16), (23) and (25) we have

$$K_1 = E\left[W_n^+\right] \left(\frac{1}{rn^4} - \frac{1}{r} + 1 \right) - \frac{\Delta}{2r}, \tag{26}$$

where, by definition of a residual,

$$E\left[W_n^+\right] = \frac{E\left[W_n^2\right]}{2E[W_n]} = \frac{Var[W_n] + E[W_n]^2}{2E[W_n]} \tag{27}$$

with $E[W_n]$ and $Var[W_n]$ given in (20) and (21). Finally, by [3]

$$K_2 = \frac{1 - \hat{\rho^s}_k}{2} \left(\frac{\sigma^2}{2\delta} + E[\Delta] \right) - K_0 - K_1 \tag{28}$$

is known by substituting (1), (16), (17), (18), (23) and (26). Thus, we can calculate the closed form approximation for the cdf of W_P by (14).

After obtaining the distributions of W_O, W_P and W_I, we are able to compute the distribution of $T = W_O + W_P + W_I$ through convolution. In the three components of T, W_O is accurate and known in closed form in (3). W_P has a closed form approximation in (14). W_I does not have a closed form, but can be arbitrarily accurate through simulation of TSP paths. It is also easy to obtain W_I because we only need to run the simulation once for the empirical distribution of L_{nj} and store the data, then calculate the distribution of D_{nj} and W_I by scaling in (7) and convolution in (8) for partitions with different sizes parameterized by r.

We can check that the computation of the distributions of W_O, W_P, W_I, and thus T, also holds when the Poisson task arrival assumption is relaxed to renewal arrival. When the task location is not uniformly distributed, or the region $\mathbf{A}$ has an irregular shape, the r equitable partitions will not be similar to the region $\mathbf{A}$. The scaling formula (7) will only be an approximation. In this case, we need to run r simulations for the empirical distribution of D_{nj} in each partition.

Figure 5 shows the pdf of T obtained from the convolution of its three components, and the empirical pdf of T obtained through simulation under the exhaustive PART-n-TSP when the region $\mathbf{A}$ is a square of size 1×1 with $m = 2$ and $n = 5$, and $m = 2$ and $n = 10$, separately. The tasks are uniformly distributed in the square with size $B_i \sim Unif[0, 1]$. The tasks arrive according to a Poisson process with rate $\lambda = 1$. Thus load $\rho = \lambda b = 0.5$. The simulated empirical pdf of T is regarded as the "true" value. We can see that the approximated values are very close to the true (simulated) values.

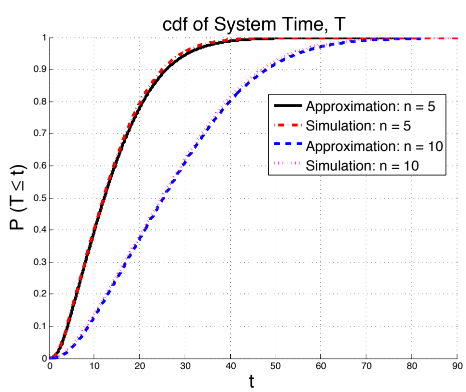

Fig. 5. Approximated & simulated cdf of system time under PART-n-TSP.

III. Comparison of PART-n-TSP, PART-TSP and NN

Bertsimas et al. [1] compared the $E[T]$ of SQM, FCFS, PART-FCFS, Space Filling Curve (SFC), NN and n-TSP through simulation, and concluded that NN achieves lower $E[T]$ than other policies simulated. PART-TSP [14] or DC [12] were proven to be $E[T]$ optimal under the light and heavy loads. Here the focus is on $Var[T]$ or $\sigma[T]$.

Since the approximation for the cdf of T for PART-n-TSP is both good and easy to compute, we can optimize the two parameters n and r to minimize $E[T]$ or other performance metrics when the region $\mathbf{A}$ and partitions $\mathbf{A}^k$ are squares. Table I gives the r^* and n^* in the range $r \in \{1^2, 2^2, \ldots, 10^2\}$ and $n = \{1, \ldots, 60\}$ that minimize $E[T]$ under exhaustive PART-n-TSP and the corresponding $E[T]$ and $\sigma[T]$ values for different ρ values. The region is a square of size 1×1. Task size $B_i \sim Unif[0, 0.5]$ and $B_i \sim Unif[0, 1]$, separately. Noting that FCFS is PART-n-TSP when $r = 1$ and $n = 1$, PART-FCFS is PART-n-TSP when $n = 1$, and n-TSP is PART-n-TSP when $r = 1$. Thus by optimizing on r and n, PART-n-TSP has better performance than FCFS, PART-FCFS and n-TSP.

We compare PART-n-TSP with PART-TSP [14] and NN [1] since they are considered near optimal in the literature. We simulate PART-TSP and NN under the same setting. The number of partitions for PART-TSP is set to be the optimal number of partitions for PART-n-TSP. The average number of tasks served inside each gate, denoted by $E[n]$, is also shown in the table. We generate $N = 100,000$ tasks for each load ρ value. Only the 25,000th to the 75,000th tasks are used to calculate $E[T]$ and $\sigma[T]$ to make sure that the steady state data are used. We have checked this by randomly sampling time segments in this range.

The $E[T]$ and $\sigma[T]$ of PART-TSP and NN are compared to those of PART-n-TSP. The minimum $E[T]$ and $\sigma[T]$ at each load level of the three policies are bolded. From Table I, we can see that NN achieves lower $E[T]$ than PART-n-TSP and PART-TSP for all $\rho \in \{0.1 \ldots 0.9\}$ in both $B_i \sim [0, 0.5]$ and $B_i \sim [0, 1]$. PART-n-TSP achieves lower $E[T]$ than PART-TSP when ρ is not too small or too large, e.g. when $\rho \in \{0.3, \ldots, 0.7\}$ for $B_i \sim [0, 0.5]$, and when $\rho \in \{0.4, \ldots, 0.8\}$ for $B_i \sim [0, 1]$. PART-n-TSP has higher $E[T]$ than PART-TSP when ρ is low because it is better to have the average number of tasks in a set to be between 1 and 2 as done by PART-TSP, but PART-n-TSP can only set it to be either 1 ro 2, resulting in higher $E[T]$. PART-n-TSP has higher $E[T]$ than PART-TSP when ρ is high because $r^* > 1$ when ρ is high. Then there is a switching time between partitions. By setting n to be a fixed number under PART-n-TSP, the server might arrive at a partition, and find the number of tasks to be less than n. Then the server would switch to the next partition without serving any task, resulting in a switching cost but no tasks served.

PART-n-TSP behaves like a "standardized" version of PART-TSP. While the fixed n reduces flexibility, it increases certainty. Thus $Var[T]$ or $\sigma[T]$ should be lower. Indeed, as shown in Table I, PART-n-TSP achieves lower $\sigma[T]$ than PART-TSP and NN when ρ is not too small or too large, e.g. when $\rho \in \{0.3, \ldots, 0.7\}$ for $B_i \sim [0, 0.5]$, and when $\rho \in \{0.4, \ldots, 0.8\}$ for $B_i \sim [0, 1]$. The performance of PART-n-TSP on $\sigma[T]$ when ρ is too small or too large is not as good for the same reasons affecting $E[T]$ as explained in the previous paragraph.

TABLE I

COMPARISON OF (α=PART-n-TSP), (β=PART-TSP) AND (γ=NEAREST NEIGHBOR) ON EXPECTATION AND STANDARD DEVIATION OF SYSTEM TIME.

		Task size $B_i \sim Unif[0, 0.5]$									Task size $B_i \sim Unif[0, 1]$								
Load ρ		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
r^* of α		1	1	1	1	1	1	1	4	25	1	1	1	1	1	1	1	1	9
n^* of α		1	2	2	4	12	24	57	59	57	1	1	1	2	3	6	15	42	41
$E[n]$ of β		1.05	1.25	1.75	3.46	9.08	23.3	61.2	63.9	49.7	1.02	1.1	1.25	1.59	2.37	4.57	13.4	40	31.1
$E[T]$	α	1.04	1.71	1.70	2.80	5.90	9.99	20.3	81.4	321	1.20	1.51	2.15	2.23	2.94	4.96	10.8	26.3	192
	β	0.95	1.26	1.87	3.30	5.97	10.9	24.4	51.7	181	**1.16**	1.37	1.71	2.35	3.63	6.24	12.9	27.9	93.8
	γ	**0.94**	**1.21**	**1.66**	**2.46**	**3.81**	**6.37**	**12.7**	**32.6**	**154**	**1.16**	**1.36**	**1.66**	**2.16**	**2.93**	**4.50**	**8.10**	**18.0**	**78.7**
$\sigma[T]$	α	0.69	1.19	**0.94**	**1.39**	**2.74**	**4.36**	**8.56**	31.7	140	0.69	0.91	1.59	**1.30**	**1.59**	**2.47**	**4.85**	**11.2**	80.7
	β	0.48	0.77	1.22	2.11	3.27	5.35	13.3	**27.1**	**101**	**0.54**	**0.75**	**1.07**	1.64	2.58	4.22	7.63	14.6	**56.2**
	γ	**0.47**	**0.76**	1.26	2.14	3.57	6.18	12.5	31.1	147	**0.54**	0.76	1.10	1.71	2.64	4.42	8.24	18.3	75.9

IV. OPTIMALITY OF PART-n-TSP UNDER LIGHT AND HEAVY LOADS

The PART-n-TSP can be modified to yield optimal $E[T]$ under light load ($\rho \to 0^+$). First the PART-n-TSP becomes FCFS policy when setting $r = 1$ and $n = 1$. Then under FCFS policy, let the server return to the median of region $\mathbf{A}$ when it becomes idle. Under light load this is the Stochastic Queue Median (SQM) policy [1], where the server travels directly to the task location from the median, executes the task, and then returns to the median after completion. SQM is proven to be $E[T]$ optimal under light load [1], proving the optimality of PART-n-TSP under light load.

Under heavy load ($\rho \to 1^-$), the following lower bound holds [2].

$$E[T] \geq \frac{\beta_{TSP,2}^2 \lambda \left(\int_A f_X^{\frac{1}{2}}(x)\, dx \right)^2}{2v^2(1-\rho)^2} \tag{29}$$

The following theorem shows that PART-n-TSP achieves the heavy-load lower bound (29) when $r \to \infty$. Thus PART-n-TSP is asymptotically optimal in $E[T]$ under heavy load.

Theorem 1: Under PART-n-TSP, when $\rho \to 1^-$ and $n \to \infty$, the system time for the DTRP satisfies

$$E[T] \leq \left(1 + \frac{1}{r}\right) \frac{\beta_{TSP,2}^2 \lambda \left(\int_A f_X^{\frac{1}{2}}(x)\, dx \right)^2}{2v^2(1-\rho)^2} \tag{30}$$

where r is the number of partitions.

Proof: See the proof of Theorem 3.1 on page 37 in [8] for the proof. ∎

The PART-n-TSP is optimal under light load. Moreover, when $r \to \infty$, the PART-n-TSP policy achieves the heavy-load lower bound (29). Therefore the PART-n-TSP is both optimal under light load and arbitrarily close to optimality under heavy load. Notice that with $r = 10$ the PART-n-TSP is already guaranteed to be within 10% of the optimal performance under heavy load.

V. CONCLUSION

We give a good approximation for the distribution of the system time that is easy to compute under the PART-n-TSP policy by utilizing the approximation results on system time T in the polling systems [4], [7]. PART-n-TSP stabilizes the system for every load $\rho \in [0, 1)$ utilizing the stability result in [10]. We compare PART-n-TSP with PART-TSP [14] and Nearest Neighbor [1] on $E[T]$ and $\sigma[T]$ in Table I, since the latter two are considered near optimal in the literature. The results show that in practice PART-n-TSP achieves lower $\sigma[T]$ than PART-TSP and NN and lower $E[T]$ than PART-TSP when the load ρ is not too small or too large. We also show that PART-n-TSP is $E[T]$ optimal under light load ($\rho \to 0^+$) and asymptotically optimal under heavy load ($\rho \to 1^-$) in Theorem 1.

REFERENCES

[1] Dimitris J Bertsimas and Garrett Van Ryzin. A stochastic and dynamic vehicle routing problem in the euclidean plane. *Operations Research*, 39(4):601–615, 1991.

[2] Dimitris J Bertsimas and Garrett Van Ryzin. Stochastic and dynamic vehicle routing with general demand and interarrival time distributions. *Advances in Applied Probability*, pages 947–978, 1993.

[3] M. A. A. Boon, E. M. M. Winands, I. J. B. F. Adan, and A. C. C. van Wijk. Closed-form waiting time approximations for polling systems. *Perform. Eval.*, 68(3):290–306, March 2011.

[4] Onno Boxma, Josine Bruin, and Brian Fralix. Sojourn times in polling systems with various service disciplines. *Performance Evaluation*, 66(11):621 – 639, 2009.

[5] Prabuddha De, Jay B. Ghosh, and Charles E. Wells. Expectation-variance analyss of job sequences under processing time uncertainty. *International Journal of Production Economics*, 28(3):289 – 297, 1992.

[6] Marco Dorigo and Luca Maria Gambardella. Ant colonies for the travelling salesman problem. *Biosystems*, 43(2):73 – 81, 1997.

[7] J. L. Dorsman, R. D. van der Mei, and E. M. M. Winands. A new method for deriving Waiting-Time approximations in polling systems with renewal arrivals. *Stochastic Models*, 27:318 – 332, 2011.

[8] Jiangchuan Huang. *From the Real Vehicle to the Virtual Vehicle*. University of California, Berkeley, 2013.

[9] Jiangchuan Huang, Christoph M. Kirsch, and Raja Sengupta. Cloud computing in space. *INFORMS Journal on Computing*, 2015. accepted.

[10] Jiangchuan Huang and Raja Sengupta. Stability of dynamic traveling repairman problem under polling-sequencing policies. In *European Control Conference (ECC) 2013*, pages 614–619, July 2013.

[11] Michael K. Hui and Lianxi Zhou. How does waiting duration information influence customers' reactions to waiting for services?1. *Journal of Applied Social Psychology*, 26(19):1702–1717, 1996.

[12] M. Pavone, E. Frazzoli, and F. Bullo. Adaptive and distributed algorithms for vehicle routing in a stochastic and dynamic environment. *Automatic Control, IEEE Transactions on*, 56(6):1259–1274, June 2011.

[13] H. Takagi. *Queueing analysis: a foundation of performance evaluation, vol. 1 : vacation and priority systems*. Queueing Analysis. North-Holland, 1991.

[14] H. Xu. *Optimal policies for stochastic and dynamic vehicle routing problems*. Cambridge, 1994.