An Approximate Dynamic Programming Approach to the Dynamic Traveling Repairperson Problem

Hyung Sik Shin[1,3] Sanjay Lall[2]

In Proceedings of the IEEE Conference on Decision and Control, pp. 2286–2291, 2010

Abstract

This paper presents a novel suboptimal policy for the Dynamic Traveling Repairperson Problem (m-DTRP), a problem requiring dynamic planning for a team of vehicles. The suggested policy is adaptive, locally distributed, computationally efficient, and independent of traffic load intensities. It is shown that the policy is asymptotically optimal in the light traffic load case. Experimental results are provided to show that the performance in the moderate and heavy load cases is comparable to the best known policies.

1 Introduction

The problem of controlling a team of many entities such as agents, robots, or vehicles has received much attention from many fields. In this paper, we address the vehicle routing problem (VRP). Traditionally, VRP has been a problem in a static environment. In a static VRP, there is no dynamic element in the problem structure in the sense that all the relevant information is given and fixed a priori. For example, the famous TSP (Traveling Salesperson Problem) is a static environment VRP. However, many real world problems are inherently dynamic: environments change over time while a team of vehicles is routed to serve demands. For instance, many new service demands may arrive randomly while vehicles move around or serve prior demands. Furthermore, the service time of each demand may vary or be random.

According to problem structures, *e.g.*, static or dynamic, there are several classes of VRP. One of them is the so-called m-DTRP (m-Dynamic Traveling Repairperson Problem), which was formulated by Bertsimas and van Ryzin [3]. The m-DTRP considers the problem of minimizing the average system time of demands in a dynamic environment. Due to the dynamic structure of the problem, however, finding the optimal policy becomes a challenging task. Even though many suboptimal policies for the m-DTRP have been suggested so far, most of them were shown to behave well only in the two extreme cases: the light and heavy traffic load levels. The performances of the suggested policies in the moderate traffic load case, which are more important when considering practical applicability, are not well known. Furthermore, policies showing the best performance in the heavy load case are rather computationally demanding or complicated.

The main contribution of this paper is that it presents a very simple and computationally efficient policy, which is not only independent of traffic load levels but also comparable to the previously suggested policies in its performance. Furthermore, the policy is adaptive: it is independent of the number of vehicles, the network size, the service region area, and environment changes. Also, the policy is locally decentralized in the sense that it only requires communication between neighboring vehicles. Although this paper does not prove that the suggested policy is optimal over all traffic load levels, it is proved that the suggested policy is asymptotically optimal in the light traffic load level. Then it is shown by experiments that the policy also works well in the moderate and heavy traffic load levels.

2 Problem Formulation

The Dynamic Traveling Repairperson Problem (DTRP) was first formulated and studied comprehensibly by Bertsimas and van Ryzin [2, 3]. The m-DTRP problem can be formulated as follows.

Let the environment $\mathcal{A} \subset \mathbb{R}^d$ be a convex, compact set with volume A. The case of $d = 2$, *i.e.*, the planar environment, is only considered in this paper. The result of this paper, however, is easily extensible to higher dimensional environments.

Suppose that there are m vehicles in the environment $\mathcal{A}$ and they can move to serve demands arriving to $\mathcal{A}$. Suppose also that the vehicles are identical and each vehicle can move at a constant speed v and has unlimited fuel and target-servicing capacity. Let

$$p(t) = (p_1(t), \ldots, p_m(t)) \in \mathcal{A}^m$$

refers to the locations of the m vehicles at time t. A demand is served by a vehicle that travels to it. Upon arrival to a demand, the vehicle spends a random amount

[1]H. S. Shin is with the Department of Electrical Engineering at Stanford University, Stanford, CA 94305, USA. hyungsik.shin@stanford.edu

[2]S. Lall is with the Departments of Electrical Engineering and Aeronautics and Astronautics at Stanford University, Stanford, CA 94305, USA. lall@stanford.edu

[3]H. S. Shin was supported by Samsung Scholarship.

of on-site service time, which follows a given distribution function ψ. The first and second moments of the distribution function ψ are finite and are denoted by $\bar{s}$ and $\overline{s^2}$, respectively.

Demands arrive to the environment $\mathcal{A}$ following a Poisson process with rate λ. Upon arrival, each demand assumes a location in $\mathcal{A}$ according to a given spatial distribution function $\varphi : \mathcal{A} \to \mathbb{R}_+$, independently. The spatial density φ is assumed normalized so that

$$\varphi(\mathcal{A}) \triangleq \int_{\mathcal{A}} \varphi(q)\, dq = 1.$$

We will consider static state-feedback control policies, which are maps from the current state $x(t)$ of the system to the velocities of all the vehicles $\dot{p}(t)$. The sets of policies and states of the system can be defined in many different ways. We will define them precisely in section 5.

Given a policy μ, let T_j be the total amount of time that the j-th arrived demand spends in the system, *i.e.*, the waiting time before service plus on-site service time. If the system is stable, *i.e.*, the number of demands waiting to be served is not increasing to infinity over time, then we can define the steady-state system time under the policy μ as

$$T_\mu \triangleq \lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} \mathbb{E}[T_j].$$

The objective of the m-DTRP problem is to find an optimal policy, which minimizes the steady-state system time.

This paper focuses on such policies that make decisions only when a vehicle is idle or completes the service of a demand. In other words, once a vehicle is assigned a demand by a policy, the vehicle moves to the demand and serves it. Also, a policy does not assign a demand to multiple vehicles. Once a policy assigns a demand to a vehicle, then the other vehicles never have the demand as their targets.

3 Preliminaries

In this section, we introduce two important problems, *the continuous multi-median problem* and *the Euclidean traveling salesperson problem*. They play a crucial role in the m-DTRP problem.

3.1 The continuous multi-median problem

The formulation of the continuous multi-median problem in this subsection is almost the same as the one in [5]. Suppose that a set $\mathcal{A} \subset \mathbb{R}^d$ is convex and compact. Suppose also that m distinct points $p = (p_1, \ldots, p_m) \in \mathcal{A}$ and a spatial probability density φ on $\mathcal{A}$ are given. Let a random point q be drawn according to φ. Then, define a function $H_m(p, \mathcal{A})$ as follows:

$$\begin{aligned} H_m(p, \mathcal{A}) &\triangleq \mathbb{E}\left[\min_{i\in\{1,\ldots,m\}} \|p_i - q\|\right] \\ &= \sum_{i=1}^{m} \int_{\mathcal{V}_i(p,\mathcal{A})} \|p_i - q\| \varphi(q)\, dq \end{aligned}$$

where $\mathcal{V}_i(p, \mathcal{A})$ is called the Voronoi cell of the generator p_i. The Voronoi cell of the generator p_i, $\mathcal{V}_i(p, \mathcal{A})$, is the set of points in $\mathcal{A}$ such that every point in $\mathcal{V}_i(p, \mathcal{A})$ is closer to p_i than to any other p_j for $j \neq i$.

The collection $\mathcal{V}(p, \mathcal{A}) \triangleq (\mathcal{V}_1(p, \mathcal{A}), \ldots, \mathcal{V}_m(p, \mathcal{A}))$ is called the Voronoi partition of the set $\mathcal{A}$ generated by the m points p. Therefore, each set consisting of finite points in $\mathcal{A}$ has its own Voronoi partition. From the above definition, $H_m(p, \mathcal{A})$ is the expected distance from a random point on $\mathcal{A}$ to the closest point in p. Therefore, $H_m(p, \mathcal{A})$ depends on p and the density function φ.

Given a density function φ on $\mathcal{A}$, the **m-median** of the set $\mathcal{A}$ is defined as follows:

$$p_m^*(\mathcal{A}) = \arg\min_{p\in\mathcal{A}^m} H_m(p, \mathcal{A}).$$

In other words, the m-median of the set $\mathcal{A}$, $p_m^*(\mathcal{A})$, is the global minimizer of $H_m(p, \mathcal{A})$ as a function of p. The continuous multi-median problem is to find the global minimizer $p_m^*(\mathcal{A})$ given $\mathcal{A}$ and φ. Let $H_m^*(\mathcal{A}) = H_m(p_m^*(\mathcal{A}), \mathcal{A})$ be the global minimum of H_m. It is known that the discrete version of the m-median problem for $d \geq 2$ is NP-hard [8].

3.2 The Euclidean traveling salesperson problem

The Euclidean traveling salesperson problem (TSP) has received much attention from various fields. It is known that TSP is an NP-hard problem.

TSP can be formulated as follows: Suppose that a set D of finite points in $\mathbb{R}^d$ is given. The objective is to find the minimum length tour such that every point in D is visited at least once.

The Euclidean TSP problem has the following interesting asymptotic property. Let $X_1, \ldots, X_n$ be independently and uniformly distributed random points in a square of area A. Define L_n as the optimal tour length through the n points. Then there exists a constant β_{TSP} such that

$$\lim_{n\to\infty} \frac{L_n}{\sqrt{n}} = \beta_{TSP}\sqrt{A}$$

with probability one [7]. It is known that $\beta_{TSP} \approx 0.712$ [6].

4 Prior Work

Bertsimas and van Ryzin were the first to formulate the m-DTRP problem and suggest several policies [2, 3].

In [2], they introduced the single vehicle DTRP problem and presented several policies including an asymptotically optimal policy in the light load case. Because it is difficult to find an optimal policy in the heavy load case, they suggested several policies for the heavy load case and proved that some of them show *constant factor* suboptimal performances to a lower bound of the optimal system time. They also illustrated by simulation that the **Nearest Neighbor (NN) policy** shows the best performance among the suggested policies in the single vehicle case [2]. Roughly speaking, the NN policy directs each vehicle to the nearest unassigned demand upon service completions. In [3], they extended the single vehicle problem to the multiple vehicles problem (m-DTRP), and performed similar analysis as in [2].

4.1 Lower bounds

Bertsimas and van Ryzin provided two lower bounds of the optimal steady-state system time for the m-DTRP problem that are useful in the light and heavy load cases [2, 3]. Before presenting the lower bounds, it is convenient to define the following quantity, which captures the notion of traffic intensity or traffic load level.

The traffic load level, which is denoted as ρ, is defined as $\rho = \frac{\lambda \bar{s}}{m}$. If ρ is close to zero, then it means that the traffic load is light. On the contrary, if ρ is close to one, then it means that the traffic load is heavy. If ρ is greater than or equal to one, then the system is unstable, *i.e.*, the number of demands waiting in the system increases to infinity as time grows.

The lower bounds of the optimal steady-state system time in the light and heavy load cases are given as follows. In the light load case ($\rho \to 0^+$), the following is satisfied:

$$T^* \geq \frac{H_m^*(\mathcal{A})}{v} + \bar{s} \tag{1}$$

where $H_m^*(\mathcal{A})$ is the optimal value of the function $H_m(p, \mathcal{A})$ as defined previously. In the heavy load case ($\rho \to 1^-$), the following is satisfied:

$$T^* \geq \gamma^2 \frac{\lambda A}{m^2 v^2 (1-\rho)^2} - \frac{\bar{s}(1-2\rho)}{2\rho} \tag{2}$$

where $\gamma \geq \frac{2}{3\sqrt{2\pi}} \approx 0.266$. The lower bound (1) is known to be tight in the light load case: Bertsimas and van Ryzin suggested an asymptotically optimal policy for the case. The lower bound (2) is not known to be tight.

The lower bound (1) is given as the sum of the expected on-site service time and the expected minimum time for a demand to be reached by a server waiting at the nearest median location on $\mathcal{A}$. In other words, this lower bound says that the optimal system time is achieved if it is possible that each server can be located at the m-median location of $\mathcal{A}$ before any new demand arrives in the corresponding Voronoi cell.

The lower bound (2) indicates an interesting asymptotic behavior of the optimal system time as the traffic intensity increases. The optimal system time increases at least as fast as $\frac{1}{(1-\rho)^2}$. In fact, it is known that the system time of many stable policies in the heavy load case is approximately given as

$$T_\mu \approx \gamma_\mu^2 \frac{\lambda A}{m^2 v^2 (1-\rho)^2} \tag{3}$$

where γ_μ is a constant depending only on the policy μ [3].

4.2 An optimal policy in the light load case

In [2, 3], Bertsimas and van Ryzin presented an asymptotically optimal policy in the light load case, which is called the Stochastic Queue Median (SQM) policy.

> **The mSQM Policy**
> Locate one vehicle at each of the m-median locations of the environment $\mathcal{A}$. When demands arrive, assign each of them to the nearest median location and its corresponding vehicle. Let each vehicle serve its assigned demands in First-Come First-Served (FCFS) manner, returning to its median after each service completion.

It is intuitively plausible that the mSQM policy achieves the lower bound (1) asymptotically in the light load case. In other words, $T_{mSQM} \to T^*$ as $\rho \to 0^+$. This convergence result was shown in [3]. Although this policy achieves the lower bound asymptotically in the light load case, it quickly destabilizes the system as the traffic load increases, *i.e.*, $\rho \to 1^-$.

4.3 A good policy in the heavy load case

It is difficult to analyze the m-DTRP problem in the heavy load case and the lower bound in (2) is not known to be tight. A policy showing the asymptotically best performance in the heavy load case, the Modified $G/G/m$ policy, was presented in [3].

> **The Modified $G/G/m$ Policy**
> For some fixed integer $k \geq 1$, divide $\mathcal{A}$ into k subregions of equal measure using radial cuts centered at a common depot (suppose that a common depot is at the median of $\mathcal{A}$), *i.e.*, form k wedges of area A/k. Within each region, form sets of demands of size n/k and, as sets are formed, deposit them in a queue. Service the queue in FCFS manner with the first available vehicle by following optimal TSP tours connecting all the demands in the set starting and ending at the depot. Optimize over n.

In [3], it is shown that

$$\frac{T_{ModG/G/m}}{T_*} \leq \frac{\beta_{TSP}^2}{2\gamma^2} \quad \text{as } \rho \to 1^-$$

where γ was given in (2). This result is the best known constant-factor approximation for the system time in the heavy load case. In [4], it is conjectured that the Modified $G/G/m$ policy is, in fact, asymptotically optimal in the heavy load case.

4.4 The sRH and mRH policies

Frazzoli and Bullo suggested another policy, the Receding Horizon Median/TSP policy [5]. The policy is decentralized, spatially distributed, and is provably locally optimal in the light load case. Also, it achieves the same performance as the best known policies in the single vehicle, heavy load case.

> **The sRH Policy**
> While the set of demands is empty, move toward $p_1^*(\mathcal{A})$ if the vehicle is not located in $p_1^*(\mathcal{A})$, otherwise stop. While the set of demands is not empty, do the following: (i) for a given $\eta \in (0, 1]$, find a path that maximizes the number of demands reached within $\tau = \max\{\text{diam}(\mathcal{A}), \eta L_{TSP}\}$ time units; (ii) service this optimal fragment from the current location. Repeat.

For multiple vehicles case, Frazzoli and Bullo extended the sRH policy to the mRH policy.

> **The mRH Policy**
> For all $i \in \{1, \ldots, m\}$, the i-th vehicle computes its Voronoi cell $\mathcal{V}_i(p, \mathcal{A})$, where $p = (p_1, \ldots, p_m)$ is the m locations of vehicles. Then, executes the sRH($\mathcal{V}_i(p, \mathcal{A})$) policy with the single following modification; while the vehicle is servicing demands in an optimal fragment on $\mathcal{V}_i(p, \mathcal{A})$, it will shortcut all demands already serviced by other vehicles.

Frazzoli and Bullo showed that the mRH policy is locally asymptotically optimal in the light load case. However, no analytic result was available on the mRH policy in the heavy load case, hence experimental results were given [5]. Note that the mRH policy assumes that each vehicle can determine its Voronoi cell and the locations of all outstanding events in it in real time.

5 The ADP Policy

Even though many good policies for the m-DTRP problem have been suggested so far, they are not guaranteed to show good performances in the moderate traffic load case, which is more important than the cases of light and heavy load when considering practical applicability. Furthermore, many policies are computationally demanding.

In this section, we present a novel policy, which is very simple and applicable regardless of traffic load levels. The policy is also spatially distributed and adaptive to environment changes. The computation requirement is not demanding and the amount of information that need to be communicated between neighboring vehicles is not much. The main idea of the policy is that each vehicle estimates the waiting times of its neighboring demands approximately and makes decision based on the estimates. This idea can be well phrased in the context of the Approximate Dynamic Programming.

The Dynamic Programming has received much attention not only as its own academic interest but also as a problem-solving principle for many applications. In many practical cases, however, it is hard to apply dynamic programming recursion directly. In order to circumvent these difficulties, the approximate dynamic programming and *Q-factor* is often employed [1].

The new policy which we present in this section is based on approximating the Q-factor. Roughly speaking, Q-factor $Q(x, u)$ is the sum of the current cost incurred by an action u under a state x and the optimal cost-to-go thereafter. If we can compute the optimal cost-to-go of each state for the m-DTRP problem, then we may be able to compute the exact Q-factor. However, we are only able to approximate Q-factor because it is almost impossible to compute the exact optimal cost-to-go. The new policy in this paper will use the finite-horizon nearest-neighbor policy for approximating the Q-factor of each state and action pair.

Before the policy is suggested, the following definitions are introduced to define system states and possible policies for the m-DTRP. In the following definitions, time notation t is omitted, *e.g.*, $D = D(t)$, and so on.

Definition 1. *Let D be the set of demands that are being served currently or waiting to be served,* i.e.*, all demands still in the system. Let m be the number of vehicles. We define o_i to be the* target demand *of vehicle i, for all $i = 1, \ldots, m$ as follows:*

$$o_i = \begin{cases} d & \textit{if vehicle } i \textit{ is heading toward or} \\ & \quad \textit{serving a demand } d, \\ \text{idle} & \textit{if vehicle } i \textit{ is idle.} \end{cases}$$

The target demand of each vehicle will be a part of the system state.

Definition 2. *Let m be the number of vehicles. We define r_i to be the* elapsed service time *of vehicle i for all $i = 1, \ldots, m$ as follows:*

$$r_i = \begin{cases} s & \textit{if vehicle } i \textit{ has been serving a} \\ & \quad \textit{demand for } s \textit{ time units,} \\ \text{noserving} & \textit{if vehicle } i \textit{ is not serving a demand.} \end{cases}$$

The elapsed service time of each vehicle will also be a part of the system state.

Definition 3. *The **state** x of the system is defined as* $x = (p, D, (o_1, \ldots, o_m), (r_1, \ldots, r_m)) \in \mathcal{A}^m \times 2^{\mathcal{A}} \times (\mathcal{A} \cup \{\text{idle}\})^m \times (\mathbb{R}_+ \cup \{\text{noserving}\})^m$*, where* $p = (p_1, \ldots, p_m)$ *refers to the locations of the* m *vehicles.*

Note that the state of the system retains all the information about the system except the arrival times of all the demands in D. Because demand arrival follows a Poisson process, state transitions are independent of the prior demand arrival times due to *memoryless* property of Poisson processes, assuming that policies do not consider the order of demand arrivals. Note that, in the m-DTRP problem, we do not care the order of demand arrivals because the objective function considers only the mean of system time, not the variance.

With the above definitions, we consider policies defined as follows.

Definition 4. *A stationary state-feedback **policy** is a map* $\mu : \mathcal{A}^m \times 2^{\mathcal{A}} \times (\mathcal{A} \cup \{\text{idle}\})^m \times (\mathbb{R}_+ \cup \{\text{noserving}\})^m \to \mathcal{A}^m$*, which determines the destination points of all the idle vehicles at every time. In other words,* $\mu(x) \in \mathcal{A}^m$ *is the vector of the destination points of all the* m *vehicles.*

Now, we define a set consisting of the nearest unassigned demands for each vehicle.

Definition 5. *Let* U *be the set of demands not assigned as targets to any vehicle,* i.e.*,*

$$U = \big\{ d \in D \mid d \neq o_i \ \forall i = 1, \ldots, m \big\}.$$

For any positive integer $l \in \mathbb{Z}_+$*, let* U_l^i *be the set of the* l *nearest demands from the vehicle* i *in* U *for all* $i = 1, \ldots, m$*. If* $|U| < l$*, then we let* $U_l^i = U$*. In other words,* U_l^i *can be defined inductively as follows:*

$$\begin{cases} U_0^i = \emptyset, \\ U_j^i = U_{j-1}^i \cup \Big(\arg\min_{d \in U \setminus U_j^i} \|d - p_i\| \Big), \ j = 1, \ldots, l. \end{cases}$$

We assume that the *argmin* above can have at most one element.

Now, we introduce a new policy called the ADP policy. Roughly speaking, the ADP policy computes the total sum of waiting times of k demands, which are visited by a unit speed vehicle. In particular, the approximate Q-factor $\tilde{Q}(x, u)$ is defined to be the total sum of waiting times of a set of k demands provided that the k demands are visited in the nearest order except the first demand u. Finally, the policy picks up the demand u which has the least total sum of waiting times assuming that there is no new demand arrival until k demands are visited. Note that the ADP policy tries to consider the *waiting time* rather than the *traveling distance*. This is important because the objective of the m-DTRP is to minimize the expected waiting time, hence solving the Euclidean TSP, which minimizes the total distance, is not enough.

The ADP Policy
Let k and l be fixed positive integers. Initially, locate the m vehicles to the m-median locations $p_m^*(\mathcal{A})$ of the environment $\mathcal{A}$. When demands arrive, dispatch the nearest idle vehicle, if any, to the demand. After completing the service of any demand, each vehicle does the following: Suppose that the vehicle is the i-th vehicle. If U is empty, move to its own median location $(p_m^*(\mathcal{A}))_i$. Otherwise, compute the approximate Q-factor $\tilde{Q}_i(x, u)$ for all $u \in U_l^i$ and move to the minimizer $u^* \in \arg\min_{u \in U_l^i} \{\tilde{Q}_i(x, u)\}$ and serve it, where $\tilde{Q}_i(x, u)$ is given as

$$\tilde{Q}_i(x, u) \triangleq M \|p_i - u\| + \sum_{j=1}^{M-1} (M - j) \|d_{j+1} - d_j\|$$

where $M = \min\{k, |U|\}$, p_i is the vehicle location, and d_j is defined recursively as $d_1 = u$, $d_{j+1} \in \arg\min_{d \in U \setminus \{d_1, \ldots, d_j\}} \|d - d_j\|$. Optimize over k and l.

The ADP policy considers the l nearest demands for possible choices for the next target. The policy can be understood as a receding horizon policy. For each possible nearest demand, the policy estimates the total time that would affect the cost directly by choosing the demand. The policy uses the NN policy to estimate the total time by summing up all the waiting times of k demands visited by the nearest manner. In other words, k determines the amount of receding demands that are to be considered.

The following theorem says that the ADP policy is asymptotically optimal in the light load case, *i.e.*, the performance is identical to the mSQM policy. The proof is similar to the proof of Theorem 4.3 of [5], hence is omitted due to space constraint.

Theorem 6. *The ADP Policy is asymptotically optimal in the light load case,* i.e.*,*

$$T_{ADP} \to T^* \text{ as } \rho \to 0^+$$

Since it is hard to analyze the performance of the ADP policy except in the light load case, experimental results in the moderate and heavy load cases are presented in the next section.

6 Experimental Results

In [2, 4], the nearest neighbor (NN) policy showed a performance comparable to the best known policy over all traffic load ranges, even though it is not an asymptotically optimal policy in the heavy traffic load case.

The system time of a policy μ in the heavy load case is known to follow the approximation (3), hence the constant γ_μ in (3) determines the heavy traffic load performance of a policy μ. By simulation results, it was shown

that $\gamma_{NN} \approx 0.64$ and $\gamma_{TSP} \approx 0.51$, where TSP represents the best known policy in the heavy load case [4].

In this section, experimental results showing the performances of the NN policy and the ADP policy are presented. Simulations were performed for the single vehicle and three vehicles cases varying the traffic intensity ρ. All the vehicles were set to move at unit speed and the environment $\mathcal{A}$ was defined to be a square of area $A = 25$. The on-site service times were randomly generated by exponential random variables, whose mean $\bar{s}$ was varied to simulate various traffic intensities.

Since it was proved that the ADP policy is asymptotically optimal in the light load case, experiments were performed for the moderate and heavy load cases. For the ADP policy, experiments were performed varying the k values. The value of l was set to be equal to k for all the cases. However, these used values of k and l are not optimal, hence the performance of the ADP policy can be better than the experimental results presented here.

Figures 1 and 2 show the average system times for the single vehicle and multiple vehicles cases, respectively, where 'LB' stands for the lower bounds (1) and (2) presented in [2, 3]. From the experimental results, it can be observed that the ADP policy always shows better performances than the NN policy.

Using the experimental results, the constant γ_{ADP} can be estimated by (3) and is given as $\gamma_{ADP} \leq 0.57$. Therefore, the performance of the ADP policy in the heavy traffic case is close to the one of the best known policy. The experimental results indicate that the ADP policy shows a performance comparable to the best known one.

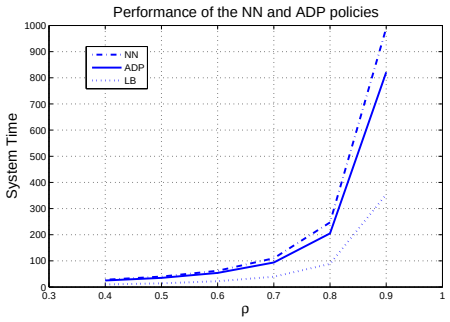

Figure 1: The average system times of the NN and ADP policies for the single vehicle case

7 Conclusion

We presented a novel policy, the ADP policy, for the m-DTRP. The ADP policy has many advantages over previously suggested ones. First of all, it can be applied universally regardless of traffic load intensities. Secondly, it is locally decentralized and highly adaptive to environment changes. Finally, it requires much less computation than the best known ones. In particular, many of the

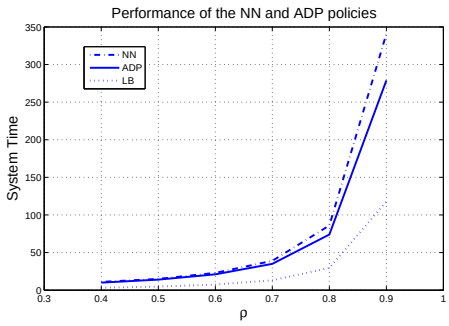

Figure 2: The average system times of the NN and ADP policies for the three vehicles case

previously suggested policies require the solution of the TSP (Traveling Salesperson Problem) at every decision moment in real time.

We showed that the ADP policy is asymptotically optimal in the light load case. Experimental results were provided to illustrate the performance in the moderate and heavy load cases. Over all traffic load intensities, the ADP policy was shown to be superior to the NN policy, which is known to show a performance close to the best known ones.

References

[1] D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 3rd edition, 2007.

[2] D. J. Bertsimas and G. van Ryzin. A stochastic and dynamic vehicle routing problem in the euclidean plane. *Operations Research*, 39:601–615, 1991.

[3] D. J. Bertsimas and G. van Ryzin. Stochastic and dynamic vehicle routing in the euclidean plane with multiple capacitated vehicles. *Operations Research*, 41(1):60–76, 1993.

[4] D. J. Bertsimas and G. van Ryzin. Stochastic and dynamic vehicle routing with general demand and interarrival time distributions. *Advances in Applied Probability*, 25:947–978, 1993.

[5] E. Frazzoli and F. Bullo. Decentralized algorithms for vehicle routing in a stochastic time-varying environment. *IEEE Conf. on Decision and Control*, pages 3357–3363, December 2004.

[6] D. S. Johnson, L. A. McGeoch, and E. E. Rothberg. Asymptotic experimental analysis for the held-karp traveling salesman bound. *Proc. 7th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 341–350, 1996.

[7] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys(eds). *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. John Wiley, Chichester, U.K., 1985.

[8] N. Megiddo and K. J. Supowit. On the complexity of some common geometric location problems. *SIAM Journal on Computing*, 13(1):182–196, 1984.