Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems
September 28 - October 2, 2004, Sendai, Japan

Determining the Fleet Size of Mobile Robots with Energy Constraints

Yongguo Mei, Yung-Hsiang Lu, Y. Charlie Hu, and C.S. George Lee
School of Electrical and Computer Engineering, Purdue University
{ymei, yunglu, ychu, csglee}@purdue.edu

***Abstract*— As robotics technologies improve, mobile robots can be used in many applications. A fundamental question is to decide the number of robots needed (i.e., the "fleet-size problem") to accomplish tasks. Previous studies did not consider the energy constraints of the fleet size problem. In this paper, we present a probabilistic method to decide the fleet size for serving random requests. A simplification method is provided for fast computation. Our method is computation efficient, with an average of 4% errors validated by an event-driven simulator.**

I. Introduction

Mobile robots can be used for surveillance, patrolling, hazard detection, and other applications. A basic problem is to determine the number of robots needed (i.e., the "fleet size") for providing satisfactory services. The fleet size depends on many factors. Important factors include: (a) Available energy. Most mobile robots carry energy sources with limited capability, such as batteries. These batteries need to be replaced or recharged. This makes the robots unavailable occasionally. (b) Power consumption. If a robot consumes more power, its batteries need recharging or replacement sooner. When a robot moves faster, it consumes more power; hence a robot's power consumption depends on its speed. (c) Service field. A larger service field needs more robots since they travel farther. A robot's speed is also important. If a robot can move faster, it can serve more requests. (d) Request rate. More robots are needed if requests arrival rates are higher. (e) Timing constraints. If each request has to be served shortly after the arrival, more robots are needed.

None of the existing solutions considers all the five factors listed above. Huang et al.[3] did not consider the effects of different speeds. Simulation can be adopted to determine the fleet sizes [7]; however, simulation can be slow because many scenarios need to be simulated. Mole [11] used request traces but the conclusion may not be applicable to other request patterns. Some used "trial-and-error" by first guessing the number needed and adding more robots if many tasks cannot be accomplished.

[1]This work was supported in part by the National Science Foundation under Grant IIS-0329061.

This paper presents a probabilistic method for determining the fleet size of mobile robots, considering all the five factors listed above. The requests are served according to the order they arrive. We define the *satisfying probability* as the probability that a new request can be served within the timing constraint. The request can be served if a robot can reach the requests location in time. Our method computes the probability if a given number of robots are available. The number of robots varies because robots are unavailable during battery recharging or replacement. Our estimation calculates the probability when a particular number of robots are available and then finds the overall satisfying probability. We develop a technique to reduce the computation. The technique is based on a simple observation: if a robot serves an earlier request, the robot is more likely to become idle when a new request arrives. This technique dramatically speeds up the estimation of fleet sizes. Our method has three major characteristics: (a) It is a probabilistic model that considers all major factors. (b) It is fast with our simplification technique. (c) It is validated by an event-driven simulation.

II. Related Work

Kirby [6] presented a mathematical model to estimate the fleet sizes and determined the number of vehicles to purchase. The method did not consider the dynamic properties of the requests, and simply assumed the probabilities of needing N robots were known. Mole [11] considered the periodic variation of the requests and used dynamic programming to minimize a cost function in determining the fleet size . Huang et al. [3] showed several heuristics to estimate the fleet size of automated guided vehicle systems. Their methods considered only one constant travel speed and assumed all the vehicles were always available. Some previous studies decided the fleet size based on simulation. For example, Lesyna [7] described a procedure to decide the fleet size by discrete-event simulations; however, no analytic model was provided. Thangiah et al. [15] used a genetic algorithm to decide the fleet sizes.

Mobile robots have been studied for many applications, such as pickup, delivery, carpet cleaning, and lawn mowing. Evans [2] described a systems of multiple mobile robots used in a hospital. Rossetti et al. [14] simulated a mobile robot delivery system used in a clinical laboratory. Some researchers investigate how to coordinate multiple mobile robots. Marapane et al. [9] presented a method for motion coordination using computer vision without direct communication among robots. Parker [12] designed an architecture to coordinate multiple robots by adaptively assigning tasks to the robots. For mobile robots, energy consumption is a critical issue. Aylett pointed out energy constraint would be the most important challenge for mobile robots [1].

III. Fleet Size Problem

A. Motivating Example

Suppose there are two customers A and B at location L_A and L_B. A post office is located in the middle of the two locations. Mobile robots are employed to pick up mails from the two customers to the post office. Once a customer makes a pickup request, one robot will travel from the post office to the customer's location, pick up the mail, and return to the post office.

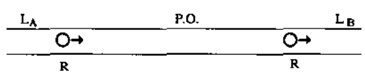

Fig. 1. Mobile Robots for Pickup

Suppose it takes a robot 20 minutes to travel from L_A or L_B to the post office in one way. We assume the pickup time is negligible and the robot returns immediately after the pickup. Thus, it takes 40 minutes for a robot to serve one request. Assume each customer issues a request every hour. Customer A's requests come at time 0, 1:00, 2:00, ..., while customer B's requests come at time 0:30, 1:30, 2:30, ... The timing constraint is that one robot has to reach the customer within 30 minutes after receiving the request; otherwise, the service fails. For this setting, we can easily see that one robot is inadequate to serve the two customers, while two robots will be enough. We can change the settings to determine the fleet size under different situations. Doubling the traveling speed reduces the time for serving one request to 20 minutes, so one robot will be enough. Meanwhile, more power is consumed at a higher speed. If each customer submits a request every 30 minutes, two robots will be inadequate.

This is an simple example of the fleet size problem for pickup and delivery. We need to consider other factors in solving real-world problems. Energy limitation is one of them. The robots carry batteries, and they will be unavailable when their batteries are being recharged or replaced. The robots are available with a probability that we called the availability. In section IV-B, we will discuss in details how to handle the energy constraints. Also, a robot may choose different speeds. The inter-arrival time between requests may not be known in advance. The locations of the customers may change. In the rest of the paper, we first describe the general settings of the problem and then explain our proposed probabilistic method.

B. Problem Description

We make the following assumptions in this paper. (a) Each robot operates (either serving a request or waiting for a request) continuously until its battery energy exhausts. The operation and recharging times are constants and known in advance. (b) Requests are independent. After a robot finishes serving a request and there is no pending request, the robot stays at the location of the request in order to conserve energy. (c) A fixed timing constraint, τ, is imposed. A request has to be served within the timing constraint; otherwise, the service is considered as a failure. We define the service time, δ, as the duration between a request is issued and the time when a robot arrives at the customer. (d) Customers are scattered in a 2-dimensional open field without any obstacles. The locations of customers from the center of the field are modeled by normal distributions. This is an approximation of the actual situation, where the customer density is usually higher around the center. (e) The request arrival is modeled by a Poisson process. (f) All robots move at the same speed, v, and the power consumed by a robot depends on the speed only. Several studies have been conducted to model the relationship between power consumption and speed [4], [5], [8], [10].

We need to model the time and the distance between two requests. Suppose the time between two consecutive requests follow an exponential distribution with mean $\frac{1}{\lambda}$. The probability density function (PDF) can be written as $\lambda e^{-\lambda t}$. Let $e_1, e_2, ...$ $(e_1 \leq e_2 \leq ...)$ be the arrival time of the requests and e_{N+1} be the newly arrived request. Let t_k represent the time between the last request and the k^{th} previous request. By definition, t_k equals to $(e_{N+1} - e_{N+1-k})$ as shown in Figure 2. The PDF for t_k is an Erlang distribution $\frac{\lambda(\lambda t_k)^{k-1}}{(k-1)!}e^{-\lambda t_k}$. The joint PDF of $t_1, t_2, ..., t_k$ is $f(t_1, t_2, ..., t_k) = \lambda^k e^{-\lambda t_k}, t_k > ... > t_2 > t_1 > 0$. The joint PDF of $t_j, t_{j+1}, ..., t_k$ is $f(t_j, t_{j+1}, ..., t_k) = \frac{t_j^{j-1}}{(j-1)!}\lambda^k e^{-\lambda t_k}, t_k > ... > t_{j+1} > t_j > 0$ for $1 \leq j \leq k$. Notice that $t_1, t_2, ..., t_N$ are not independent.

Suppose a customer's location is modeled by two independent random variables (x, y) that follow normal distributions. The mean and standard deviation

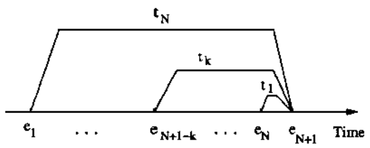

Fig. 2. Requests and Time Intervals, e_{N+1} is the Last Request.

are μ and σ, respectively; we choose the center's location so that μ is zero. We need to consider the distances between customers because the distances decide the traveling time and whether a robot can reach a customer in time. Let l_k represent the distance between the newly arrived and the k^{th} previous request (similar to the definition t_k). The distance follows Rayleigh distribution $g(l_k) = \frac{l_k}{2\sigma^2} e^{-\frac{l_k^2}{4\sigma^2}}$. The derivation of the probability functions is omitted here due to the paper limitation. The values of l_1, l_2, ... l_k have identical distributions because the customers' locations are independent.

IV. Determining Fleet Size

Our method determines the fleet size using two metrics: (a) *satisfying probability*, the probability that a new request can be served. (b) *satisfying interval*, the duration between two consecutive failures. Instead of finding the fleet size directly, we compute the two metrics if at most N robots are available. If the satisfying probability is too low or the satisfying interval is too short, we add more robots to the fleet and recompute the probabilities. If both are too large, we reduce the number of robots. We use binary search to obtain the appropriate number of robots.

A. Satisfying Probability with Unlimited Energy

Our first step calculates the satisfying probability without considering the energy limitation. Then, we add the limitation into the calculation. We show the satisfying probability of $1, 2, ...,$ and N robots. We compute the steady-state satisfying probabilities by assuming that the initial N requests are always satisfied if N robots are used.

1) One Robot: We first consider the scenario in which at most one robot is available. At e_2, the second request arrives. The robot may be in one of the two possible cases: (a) The robot is on the way to serve the first request e_1. (b) The robot has served the first request and is idle. Since it takes δ to serve the first request, the service completes at time $e_1 + \delta$. The first case occurs when $e_2 < e_1 + \delta$ and the second case occurs when $e_2 \geq e_1 + \delta$. The second request must be served before $e_2 + \tau$.

To determine whether the second request can be served, we have to consider the distance between the first and the second requests. In the first case, the robot can move to the second customer at $e_1 + \delta$. Hence, it can travel at most a distance of $v(e_2 + \tau - e_1 - \delta)$. If the distance between the two requests is shorter than $v(e_2 + \tau - e_1 - \delta)$, the second request can be served. The probability is a double integration of the distribution of the inter-arrival time and the distance between the two requests. The inter-arrival time must be smaller than δ in order to belong to the first case. The distance is shorter than $v(e_2 + \tau - e_1 - \delta)$. By definition, $t_1 = e_2 - e_1$. Consequently, the probability that the second request can be served is $\int_0^\delta [\int_0^{v(t_1+\tau-\delta)} f(t_1) g(l_1) dl_1] dt_1$.

In the second case, $e_2 > e_1 + \delta$ so $t_1 > \delta$. The second request can be served if the distance l_1 is smaller than $v\tau$. The probability is $\int_\delta^\infty [\int_0^{v\tau} f(t_1) g(l_1) dl_1] dt_1$. Because the two cases are exclusive, the overall satisfying probability is $\int_0^\delta [\int_0^{v(t_1+\tau-\delta)} f(t_1) g(l_1) dl_1] dt_1 + \int_\delta^\infty [\int_0^{v\tau} f(t_1) g(l_1) dl_1] dt_1$. The value of δ varies between 0 and τ to satisfy the first request. Let $h(\delta)$ be the PDF of δ. The overall satisfying probability of the second request is

$$\int_0^\tau \{ \int_0^\delta [\int_0^{v(t_1+\tau-\delta)} f(t_1) g(l_1) dl_1] dt_1 + \int_\delta^\infty [\int_0^{v\tau} f(t_1) g(l_1) dl_1] dt_1 \} h(\delta) d\delta \quad (1)$$

2) Two or More Robots: When there are two robots, we consider whether the third request can be served. When the third request arrives, there are three possible cases: both robots are idle, one is idle and the other is on the way to serve a request, and both are on their way to serve requests. To simplify the analysis, we assume δ is a constant $\frac{4\sigma}{v}$. This assumption is based on the following observation: The average distance to travel is $\sqrt{\pi}\sigma \approx 1.8\sigma$,so the average travel time is $\frac{1.8\sigma}{v}$. We use $\frac{4\sigma}{v}$ because it covers more than 98% of all possible travel time. For multiple robots, a new request may be served by any of these robots; the computation is complex. In contrast, the service fails if none of the robots can serve within the time constraint. It will be easier if we compute the probability of failure and its complement is the satisfying probability.

In the first case, both robots are idle so $e_3 - e_2 = t_1 \geq \delta$, $e_3 - e_1 = t_2 \geq \delta$, and $t_2 > t_1$ (by definition). The third request cannot be served if neither robot can travel to the location in time. In other words, $v\tau < l_1$ and $v\tau < l_2$. The probability of failures is $\int_\delta^\infty \int_{t_1}^\infty \int_{v\tau}^\infty \int_{v\tau}^\infty f(t_1, t_2) g(l_1) g(l_2) dl_2 dl_1 dt_2 dt_1$.
In the second case, the first request has been served but the second request has not been served: $t_2 > \delta$ and $t_1 < \delta$. The probability that the third request cannot be served is $\int_0^\delta \int_\delta^\infty \int_{v(\tau+t_1-\delta)}^\infty \int_{v\tau}^\infty f(t_1, t_2) g(l_1) g(l_2) dl_2 dl_1 dt_2 dt_1$.

In the third case, the failure probability is $\int_0^\delta \int_{t_1}^\delta \int_{v(\tau+t_1-\delta)}^\infty \int_{v(\tau+t_2-\delta)}^\infty f(t_1,t_2)g(l_1)g(l_2)dl_2dl_1 dt_2dt_1$.

For N robots, we need to consider $N+1$ cases: all are idle, one is serving a request, two are serving requests, ..., and all are serving requests. If the new request arrives when k robots are serving requests, then the following condition must hold: $t_1 < t_2 < ... < t_k < \delta < t_{k+1} < ... < t_N$. The new request cannot be served if none of the N robots can serve it. The last k robots cannot serve this new request if $l_i > v(t_i + \tau - \delta)$ for $1 \le i \le k$. The other $N-k$ robots cannot serve this new request if $l_i > v\tau$ for $k < i \le N$. The failure probability is $\int_0^\delta \int_{t_1}^\delta ... \int_{t_{k-1}}^\delta \int_\delta^\infty ... \int_{t_{N-1}}^\infty f(t_1,t_2,...,t_N) \{\Pi_{i=1}^k \int_{v(t_i+\tau-\delta)}^\infty g(l_i)dl_i\}\{\Pi_{i=k+1}^N \int_{v\tau}^\infty g(l_i)dl_i\} dt_N...dt_1$. The overall failure probability is the sum for all possible values of k from 0 and N.

$$\sum_{k=0}^N \int_0^\delta \int_{t_1}^\delta ... \int_{t_{k-1}}^\delta \int_\delta^\infty ... \int_{t_{N-1}}^\infty f(t_1,t_2,...,t_N)\{\Pi_{i=1}^k \int_{v(t_i+\tau-\delta)}^\infty g(l_i)dl_i\} \{\Pi_{i=k+1}^N \int_{v\tau}^\infty g(l_i)dl_i\}dt_N...dt_1 \quad (2)$$

3) A Simplification Technique: When N is larger than 3, the computation of the multiple integral (equation (2)) is quite complex. When the number of robots increases from 2 to 4, the computation time increases from 1.6 to 1098 seconds, or 686 times, on a 2GHz-Pentium-4 machine with 0.5GB memory. We can simplify the computation by considering only those robots that serve the first j requests: $e_1,...,e_j$. An appropriate value of j could be 2, 3 or 4. This is based on the observation that these robots are most likely to finish their assigned requests earlier than the other robots. Hence, they are more likely to be idle for serving a new request.

Suppose the new request arrives when k ($0 \le k \le j$) robots in the first group are serving requests; this leads to the condition $t_{N-j+1} < ... < t_{N-j+k} < \delta < t_{N-j+k+1} < ... < t_N$. None of the k robots can serve the new request if $l_{N-j+1} > v(\tau + t_{N-j+1} - \delta)$, ..., $l_{N-j+k} > v(\tau + t_{N-j+k} - \delta)$. None of the rest $j-k$ idle robots in the first group can serve the new request if $l_{N-j+k+1} > v\tau$, $l_{N-j+k+2} > v\tau$, ..., and $l_N > v\tau$. We replace N in equation (2) by considering only first j robots and obtain the following failure probability:

$$\sum_{k=0}^j \int_0^\delta \int_{t_{N-j+1}}^\delta ... \int_{t_{N-j+k-1}}^\delta \int_\delta^\infty ... \int_{t_{N-1}}^\infty f(t_{N-j+1},t_{N-j+2},...,t_N) \{\Pi_{i=N-j+1}^{N-j+k} \int_{v(t_i+\tau-\delta)}^\infty g(l_i)dl_i\} \{\Pi_{i=N-j+k+1}^N \int_{v\tau}^\infty g(l_i)dl_i\} dt_N...dt_{N-j+k+1}dt_{N-j+k}...dt_{N-j+2}dt_{N-j+1} \quad (3)$$

B. Satisfying Probability with Limited Energy

If we consider energy constraints, some robots may be unavailable during recharging. Let X represent the number of available robots at the time a new request comes; X is a random variable whose distribution depends on the robots' working and recharging time. If there are at most N robots, the probabilities $P(X=k), k=1,...,N$ can be decided based on the robot's energy limitation, as showed in the following subsection. If the satisfying probability of k robots with unlimited energy is $\gamma(k)$, then after considering the energy limitation, the satisfying probability of next request with at most N robots is $\Sigma_{k=1}^N P(X=k)\gamma(k)$.

1) Robot Availability: A robot is either working or being recharged. Let T_w and T_c represent the working time and the recharging time, respectively. The batteries are recharged immediately after its energy exhausts. The availability of the robot is defined as $\alpha = \frac{T_w}{T_w+T_c}$. A robot is available when it is either traveling or waiting for a new request. When the robot is waiting, it does not move and consumes no power. Thus, the working time is divided into two parts: traveling time T_t and waiting time T_a, $T_w = T_t + T_a$. If we define the traveling ratio r_t as the quotient of traveling time divided by the working time: $r_t = \frac{T_t}{T_w}$, then $\alpha = \frac{T_t}{T_t + T_c \times r_t}$. Let E_c be the battery's energy capacity and $\beta(v)$ be the power of the robot while traveling at speed v. The traveling time is the ratio between E_c and $\beta(v)$, $T_t = \frac{E_c}{\beta(v)}$. The availability is $\alpha = \frac{E_c}{E_c + T_c \times r_t \times \beta(v)}$. Consider a fleet of N robots. We can compute the probability $P(X=k)$ when k robots are available ($k \le N$). The probability follows Bernoulli distributions and is $C_k^N \alpha^k (1-\alpha)^{N-k}$. Here, C_k^N is the combination of k out of N.

C. Satisfying Interval

Another important metric is the satisfying interval between two consecutive failures. We use η to represent the satisfying probability. Let M represent the number of services that the robots can serve until a failure occurs. For example, $M=1$ shows that the robots can serve the first request but fail the second request. M is a geometric distributed random variable, and $P(M=k) = \eta^k(1-\eta)$. If M is k, k requests are satisfied so the satisfying interval is t_k. The satisfying interval is t_k with the probability of $\eta^k(1-\eta)$. The value of t_k is also a random variable and its expected value is $\frac{k}{\lambda}$. Consequently, the expected satisfying interval is:

$$\sum_{k=0}^{\infty} \frac{k}{\lambda}\eta^k(1-\eta) = \frac{\eta}{(1-\eta)\lambda}. \quad (4)$$

V. A Case Study

A. Experimental Setup

We used a commercial robot called PPRK developed at Carnegie Mellon University [13] for our experiments. We measured the average robot's power at different speeds [10]. We studied the effect of different speeds, energy capacity, and request arrival rates. The default energy capacity was $20736J$ for four AA batteries (1200 mAh and 1.2V). The default speed was $0.11m/s$ because it was the most power-efficient speed [10]; at this speed, the robot's power was $0.869W$. Suppose a robot was serving requests 80% of time ($r_t = 0.8$). The working time was 8.3 hours. The battery charging time was 3 hours so the robots' availability was 0.734. The default arrival rate λ was $0.005/s$, and the default standard deviation σ was $25m$. The timing constraint was set to $5\sigma/v$; the value could change according to the real service policy. The simplification method was used for computing the satisfying probability for three or more robots.

B. Results

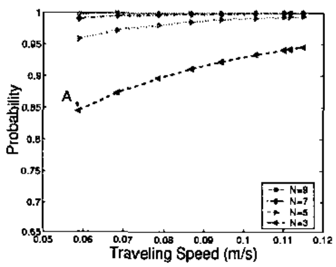

Fig. 3. Satisfying Probability with Unlimited Energy

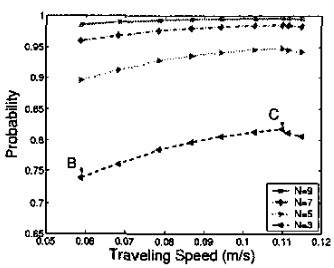

Fig. 4. Satisfying Probability with Limited Energy

Figures 3 and 4 show the satisfying probabilities at different speeds without and with energy limitations. They demonstrate that the energy limitations affect the probabilities significantly. For example, the probability is 0.845 and 0.739 at speed $0.0589m/s$ and at most three robots without and with energy limitation

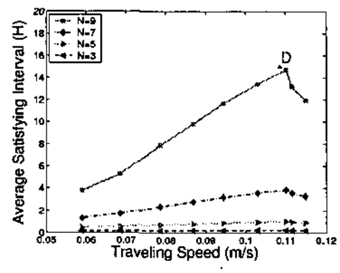

Fig. 5. Satisfying Interval

(point A and B). Figure 3 shows that the probabilities monotonically increase as the robots' speeds increase. However, as explained in [10], the energy efficiency (meter per Joule) decreases when the speed exceeds $0.11m/s$. As a result, Figure 4 shows that the satisfying probabilities decline when the speeds are too high (beyond point C in the figure). This effect is clearer in Figure 5 when we consider the satisfying intervals (point D). Figure 4 also shows "diminishing returns" as the number of robots increases. When the number increases from 3 to 5 at speed $0.0786m/s$, the probability increases from 0.785 to 0.928. If two more robots are added (totally 7 robots), the probability increases by only 0.049.

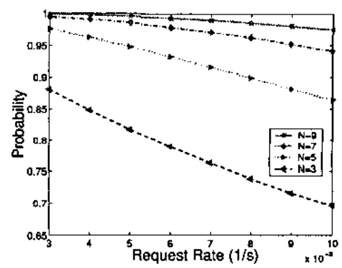

Fig. 6. Satisfying Probability

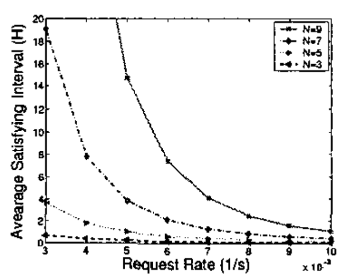

Fig. 7. Satisfying Interval

Figures 6 and 7 show how the satisfying probability and the satisfying interval change at different request arrival rates. The probability and interval decrease when the arrival rate increases. With five robots, when the arrival rate changes from $0.004/s$ to $0.008/s$, the satisfying probability decreases from 0.963 to 0.899 and the satisfying interval decreases from 1.8H to 0.31H. For the two curves in the two figures for at most nine robots, we can see that the satisfying probability changes little while the satisfying interval declines dramatically. From equation (4), we can see that their relationship is not linear. The interval changes quickly when the probability is close to 1. This clearly shows the importance of using different metrics in determining the fleet size.

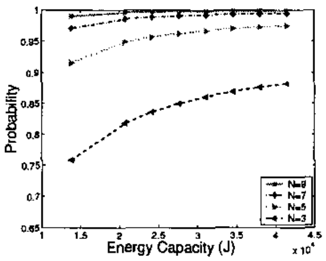

Fig. 8. Satisfying Probability

Figure 8 shows how energy capacity affects the satisfying probability. We can see that the probability increases when the energy capacity increases. This is because the robots can work for longer time, thus increasing the availability. With seven robots, the satisfying probability increases from 0.948 to 0.975, when the energy capacity doubles from $20736J$ to $41472J$.

We used an event-driven simulator to validate our probabilistic model. The simulator contained two modules: robots and an event generator. For a robot, we considered its traveling speed, energy capacity, power consumption and location. The event generator generated requests whose time and location followed the corresponding distributions. A data structure was used to record the requests, including their arrival times and locations; another data structure was used to record the robots including their locations and the times they would finish serving the corresponding requests. When a new request arrived, we first checked the robots that were waiting. If those waiting robots could not serve the new request, the rest of the robots were checked. We considered the new request as a failure when none of the robots could serve it. The failure was counted and the satisfying probability was computed as the number of served requests divided by the number of all generated requests. Simulation could accurately estimate the satisfying probabilities and intervals because it simulated what actually happened. A major drawback of simulation was the time needed to simulate. Our probabilistic method was at least an order of magnitude faster than the simulation. Compared with the results from the simulator, our probabilistic method with simplified computations had errors within 4% in the satisfying probabilities and intervals. Without simplification technique in section IV-A, the computation took too long.

VI. Conclusion

This paper presents a probabilistic analysis method for the fleet size problem in mobile-robot systems. A simplification method is developed to provide efficient computation. The results show that the energy limitations have significant effects on the fleet size. The results are validated by simulations within 4% errors.

References

[1] R. Aylett. *Robots: Bringing Intelligent Machines To Life.* Barrons, 2002.

[2] J. Evans. Helpmate: An Autonomous Mobile Robot Courier for Hospitals. In *IROS*, pages 1695–1700, 1994.

[3] P.-C. Huang and F.-H. Liu. Estimation of the Fleet Size for a Multi-load Automated Guided Vehicle System. In *Proceedings of National Science Council: Physical Science and Engineering*, pages 244–253, 2001.

[4] R. Katoh, O. Ichiyama, T. Yamamoto, and F. Ohkawa. A Real-time Path Planning of Space Manipulator Saving Consumed Energy. In *International Conference on Industrial Electronics, Control and Instrumentation*, pages 1064–1067, 1994.

[5] B. K. Kim and K. G. Shin. Suboptimal Control of Industrial Manipulators With A Weighted Minimum Time-Fuel Criterion. *IEEE Transactions on Automatic Control*, 30(1):1–10, January 1985.

[6] D. Kirby. Is Your Fleet the Right Size? *OR*, 10(4):252–252, 1959.

[7] W. R. Lesyna. Sizing Industrial Rail Car Fleets Using Discrete-Event Simulation. In *Winter simulation Conference*, pages 1258–1261, 1999.

[8] Y. Li and G. M. Bone. Are Parallel Manipulators More Energy Efficient. In *International Symposium on Computational Intelligence in Robotics and Automation*, pages 41–46, 2001.

[9] S. Marapane, M. Holder, and M. Trivedi. Coordinating Motion of Cooperative Mobile Robots Through Visual Observation . In *IEEE International Conference on Systems, Man, and Cybernetics*, pages 2260 – 2265, 1994.

[10] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. S. G. Lee. Energy-Efficient Motion Planning for Mobile Robots. In *International Conference on Robotics and Automation*, pages 4344–4349, 2004.

[11] R. H. Mole. Dynamic Optimization of Vehicle Fleet Size. *Operational Research Quarterly*, 26(1):25–34, 1975.

[12] L. E. Parker. ALLIANCE: An Architecture for Fault Tolerant Multirobot Cooperation. *IEEE Transactions on Robotics and Automaton*, 14(2):220–240, 1998.

[13] G. Reshko, M. T. Mason, and I. R. Nourbakhk. Rapid Prototyping of Small Robots. Technical report, CMU-RI-TR-02-11, Carnegie Mellon University, 2002.

[14] M. D. Rossetti, A. Kumar, and R. A. Felder. Mobile Robot Simulation of Clinical Laboratory Deliveries. In *Winter Simulation Conference*, pages 1415–1422, 1998.

[15] S. R. Thangiah, K. E. Nygard, and P. L. Juell. Gideon: A Genetic Algorithm System for Vehicle Routing with Time Windows. In *Conference on Artificial Intelligence Applications*, pages 322–325, 1991.