

# Hybrid Model for *A Priori* Performance Prediction of Multi-Job Type Swarm Search and Service Missions\*

Meghan Chandarana<sup>1</sup>, Dana Hughes<sup>2</sup>, Michael Lewis<sup>3</sup>, Katia Sycara<sup>2</sup>, and Sebastian Scherer<sup>2</sup>

**Abstract**—In Swarm Search and Service (SSS) applications, swarm vehicles are responsible for concurrently searching an area while immediately servicing jobs discovered while searching. Multiple job types may be present in the environment. As vehicles move in and out of the swarm to service jobs, the coverage rate (i.e., area searched by the swarm per time step) changes dynamically to reflect the number of vehicles currently engaged in search. As a result, the arrival rates of jobs also changes dynamically. When planning SSS missions, the resource requirements, such as the swarm size needed to achieve a desired system performance, must be determined. The dynamically changing arrival rates make traditional queuing methods ill-suited to predict the performance of the swarm. This paper presents a hybrid method – Hybrid Model – for predicting the performance of the swarm *a priori*. It utilizes a Markov model, whose state representation captures the proportion of agents searching or servicing jobs. State-dependent queuing models are used to calculate the state transition function of the Markov states. The model has been developed as a prediction tool to assist mission planners in balancing complex trade-offs, but also provides a basis for optimizing swarm size where cost functions are known. The Hybrid Model is tested in previously considered constant coverage rate scenarios and the results are compared to a previously developed Queuing Model. Additional SSS missions are then simulated and their resulting performance is used to further evaluate the effectiveness of using the Hybrid Model as a prediction tool for swarm performance in more general scenarios with dynamically changing coverage rates.

## I. INTRODUCTION

Robot swarm missions, known as Swarm Search and Service (SSS) missions, require swarm vehicles to use decentralized (i.e., local) control laws to simultaneously search a given area and to identify and service any jobs that arise. For example, forest fire applications may require any found brush fire to be put out. Prior research, such as [1], has considered missions to be complete as soon as targets are found or the region has been searched. However, to successfully complete a real mission, jobs found must also be serviced. Multiple job types can be present, each requiring a different sized group of vehicles to break off from the swarm for a different specified length of time to successfully service it. Figure 1 depicts a sample wildfire mission where 2 job types are present: putting out small brush fires and attending to injured people. In this example, the swarm traverses the environment with

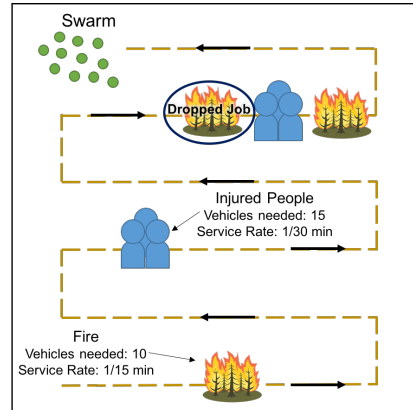


Fig. 1: Example SSS mission with two different job types.

a lawn mower pattern (yellow dashed line). When too many jobs arrive in quick succession, there may not be enough vehicles present in the swarm to service all jobs. This results in one or more of the jobs being dropped (i.e., not serviced). Once a job has been serviced, the vehicles used to service it return to the swarm and are available for reallocation. Each mission scenario consists of an environment with a different distribution of jobs types.

In many SSS applications, the environment is open and the coverage rate of the swarm dynamically varies with the number of vehicles actively searching. More specifically, as the number of vehicles in the swarm decreases (allocated to service a job), the rate of coverage also decreases. This results in an increase in time between the arrival of all job types and vice versa for an increase in swarm size.

For many applications, negative consequences can result from not immediately servicing jobs. In the case of forest fire missions, failure to put out identified brush fires that have been started with embers carried by the wind could result in the fire spreading more rapidly. This may lead to a loss of wildfire containment. Surveillance jobs may require the tracking of suspicious targets within the search area. In military applications, suspicious targets left without surveillance could lead to unanticipated attacks. Due to the costs associated with dropping a job and with deploying each vehicle, an inherent trade-off exists between swarm size and mission performance.

The work in this paper aims to develop a model to predict, *a priori*, the performance of a swarm (of a given size) deployed for an SSS mission with a given job distribution and dynamically changing coverage and arrival rates. We present a Hybrid Model that estimates the predicted performance of the swarm by combining aspects of queuing theory with a

<sup>1</sup>The author is with the Dept. of Mechanical Engineering at Carnegie Mellon University, Pittsburgh, PA, USA {mchandar@cmu.edu}

<sup>2</sup>The authors are with the Robotics Institute at Carnegie Mellon University, Pittsburgh, PA, USA {danahugh@andrew.cmu.edu, katia@cs.cmu.edu, basti@andrew.cmu.edu}

<sup>3</sup>The author is with the School of Computing and Information at the University of Pittsburgh, Pittsburgh, PA, USA {ml@sis.pitt.edu}

\*Sponsored by NASA LaRC (NIA Activity 201020)

Markov chain state space representation. The Hybrid Model incorporates the following attributes: (1) a state space representation that captures the dynamically changing coverage rate resulting from vehicles moving in and out of the searching swarm to service jobs, (2) the use of queuing theory to determine the transition probabilities between swarm states and (3) the utilization of the stationary distribution to evaluate the average performance of the swarm.

The Hybrid Model is an extension to the previously developed Queuing Model [2], which only considered a special case of SSS missions where the swarm coverage rate – and therefore the job arrival rate – of the swarm remained constant. In applications where the costs of deploying robots and dropping jobs can be explicitly defined, the Hybrid Model can be used as a prediction tool to determine the optimal swarm size to deploy. In scenarios where humans are required for mission planning (e.g. to make swarm deployment decisions or for making legal, moral, or ethical decisions), the Hybrid Model can be used by human operators as a planning tool aid. The Hybrid Model prediction is compared against the Queuing Model prediction in constant coverage rate scenarios (Section IV-A). In addition, the Hybrid Model predictions are compared to the results of simulated SSS missions in dynamically changing coverage rate scenarios (Section IV-B).

## II. RELATED WORK

In static vehicle routing target locations are known in advance and the problem to be solved is planning explicit routes to them [3][4]. In dynamic vehicle routing (DVR), however, targets appear dynamically and the problem becomes determining policies for routing the vehicles [5]. Because demands are not known *a priori*, steady-state system performance must be optimized [5]. In the past, DVR policies have focused on assigning a predetermined vehicle or group of vehicles to a partition of an area. Each vehicle or group of vehicles then individually monitored their partition for new jobs and adjusted the optimal service order for the current jobs in their partition as required [5][6].

In [2] we developed a simple policy of optimally allocating the nearest vehicles as new jobs are detected while the swarm traverses the environment (as is the case in SSS missions). A Queuing Model was then developed to predict the performance of a swarm using these policies to service jobs by modeling the system as a variant of the DVR problem with time constraints. More specifically, the Queuing Model was able to predict the relationship between swarm size and the expected number of dropped jobs for SSS missions with a given job distribution.

The initial work concentrated on applications where the coverage rate of the swarm remained constant, such as in urban environments where narrow streets could eliminate the advantages of multiple robots searching abreast to cover greater area. As a result, the arrival rate for all job types remained constant throughout the environment, regardless of the number of vehicles currently in the swarm (i.e., the unallocated vehicles). Analysis showed that the Queuing

Model accurately modeled such SSS missions [2]. The use of the predicted relationship between swarm size and dropped jobs as a planning tool resulted in operators planning SSS missions that resulted in better overall performance, fewer used vehicles, and a lower unsearched area than those who planned the mission without the tool [7]. However, traditional queuing theory cannot be used to model systems where arrival rates change dynamically according to an unknown function. Therefore, it is ill-suited for more realistic applications where coverage rates vary dynamically.

Current research has explored how Markov chains can be used to model the macroscopic behaviors of the swarm as a whole [8]. In such models, a single Markov chain state space and its corresponding transitions can represent the entire swarm with each state representing the average number of robots in a particular state [9][10]. Markov chain representations have also been used for the probabilistic guidance of robots in a swarm in formation generation [11][12][13] and swarm splitting for task allocation [14].

## III. METHOD OVERVIEW

The Hybrid Model presented below is an extension to the previously developed Queuing Model [2], which was used to estimate the predicted performance of an SSS mission *a priori*. In the Queuing model, a Poisson process is used to describe the arrival times of individual jobs, and the time required to complete each job is drawn from an exponential distribution. In this model, the coverage rate (i.e., area searched per time step) of the swarm remains constant, regardless of the size of the swarm.

The constant coverage rate assumption may be true in urban environments with constrained streets. However, in more general scenarios, the coverage rate varies as vehicles leave the swarm to service arriving jobs (i.e., is dependent on the number of vehicles left in the swarm). More specifically, as the number of vehicles in the swarm decreases, the rate of coverage will also decrease. Thus, jobs arrive less frequently to the system (i.e., the arrival rate decreases). The dynamically changing arrival rates make the Queuing Model an infeasible solution. For small environments, empirical studies can be used to simulate SSS systems with dynamically changing arrival rates. However, as the size of the environment grows, these simulations become computationally expensive. Therefore, we propose the use of a Hybrid Model that captures the dynamically changing coverage rate using a Markov chain state space representation. For each state, queuing theory is utilized to calculate the transition probabilities between itself and all other states using the information provided from the state representation about the distribution of robots over the jobs being serviced and the those free to search. The stationary distribution is then used to determine the expected *a priori* mission performance of the swarm.

### A. Queuing Theory

Consider an environment of size  $A$ . There are  $M$  job types present in the environment. Jobs are randomly distributed

in the environment such that each location has an equal probability of having a job (i.e., distributed randomly with a uniform spatial density function). Their locations are i.i.d. and are not known *a priori*. Given the expected distribution of jobs over the environment, the job density for each job type is given by  $\Phi = [\phi_1, \dots, \phi_M]$ , where  $\phi_m$  is the job density for job type  $m$ .  $\phi_m = \frac{n_{job}^m}{A}$ , where  $n_{job}^m$  is the number of expected jobs of type  $m$  in the environment.

In our formulation, the area covered by a swarm of  $n$  vehicles in a single time step,  $t_s$ , is given by  $\mathcal{F}(n)$  and is referred to as the coverage rate. This function is assumed to be known *a priori* and is used as an abstraction of the real-world attributes of the swarm, such as swarm formation, sensing range, and vehicle velocities. As vehicles cover area, jobs arrive to the system (i.e., jobs are dynamically sensed). As jobs are distributed randomly and uniformly, their arrivals follow a Poisson distribution. As in the queuing theory literature [15], for jobs that arrive according to a Poisson distribution: for a swarm of size  $n$ , the probability of  $k$  jobs of type  $m$  arriving is given by

$$p_a^m(\lambda_m(n), k) = e^{-\lambda_m(n)} \frac{\lambda_m(n)^k}{k!}, \quad (1)$$

where  $\lambda_m(n)$  is the expected number of jobs of type  $m$  to arrive in a single time step (the arrival rate)

$$\lambda_m(n) = \phi_m \cdot \mathcal{F}(n). \quad (2)$$

Each arriving job requires a specified amount of resources. The resource requirements for each job type are defined by the tuple  $j_m = \langle n_{service}^m, \mu_m \rangle$ , where  $n_{service}^m$  and  $\mu_m$  are the required number of service vehicles and the mean service time for a job of type  $m$  respectively. The set of all vehicle and service time requirements can be defined as  $\bar{n}_{service} = [n_{service}^1, \dots, n_{service}^M]$  and  $\bar{\mu} = [\mu_1, \dots, \mu_M]$  respectively. When jobs arrive and not enough vehicles are present in the swarm, the job is dropped (i.e., not serviced). Similar to other queuing formulations, the service times are assumed to follow an exponential distribution [15]. Therefore, the probability of a job of type  $m$  being completed in a single time step is

$$p_c^m = 1 - e^{-\mu_m}. \quad (3)$$

All service times include the travel time between the job site and the swarm.

For a swarm of size  $N$ , vehicles are either allocated and servicing jobs or they are in the swarm searching area

$$N = N_{busy} + N_{search}, \quad (4)$$

where  $N_{busy}$  is the number of vehicles currently allocated to service jobs and  $N_{search}$  is the number of free vehicles available to continue searching the environment. If  $N_{search}$  were constant, then  $\mathcal{F}$  would be constant and the previously developed Queuing Model could be used to predict the performance of the swarm. However, as jobs arrive and are completed, vehicles dynamically move in and out of the swarm resulting in  $\mathcal{F}$  being state dependent.

## B. State Space Representation

To represent the different system states that exist in SSS mission scenarios, a discrete time Markov chain state space representation is used. Discrete time Markov chains are a stochastic model that are used to describe the evolution of a finite number of states over discrete time. Each state is assumed to be independent and memoryless. The transition matrix,  $T$ , is an  $|S| \times |S|$  matrix where  $|S|$  is the size of the state space  $S$ . Each element,  $T_{ij}$  defines the probability of transitioning from state  $i$  to another state  $j$  in a time step [16].

To capture the dynamically changing swarm size that results from jobs arriving and being completed in SSS missions, each state,  $s_i$ , is defined as  $s_i = [n_1, \dots, n_M]$ , where  $n_m$  is the number of jobs of type  $m$  that are currently being serviced at state  $s_i$ . The full state space is given by  $S = \{s_1, \dots, s_K | N - \bar{n}_{service} \cdot s_i \geq 0\}$ , where  $i \in \{1, \dots, K\}$ . Therefore, the valid states are those in which there are enough vehicles in the swarm to service the current jobs in the system. By using this state space representation each state can encode both the number of vehicles currently allocated (busy vehicles) and the number of vehicles left in the swarm to search the remaining area (search vehicles). For each state, the number of search vehicles left in the swarm is used to determine the coverage rate for that swarm in that state (i.e.,  $\mathcal{F}(N_{search})$ ).

## C. Transition Dynamics

At each time step, the following can happen: jobs arrive, jobs are completed, or nothing (i.e., the swarm stays in the same state). One or more things can happen within the same time step. For each state, the transition probabilities are found by calculating the state dependent arrival and completion probabilities. For each job type,  $m$ , the set of all arrival probabilities,  $P_a^m$ , is given by

$$\mathbf{P}_a^m = \{p_a^m(\lambda_m(s_i), k_m = 0), \dots, p_a^m(\lambda_m(s_i), k_m = \gamma_m), p_a^m(\lambda_m(s_i), k_m > \gamma_m)\}, \quad (5)$$

where  $k_m$  is the number of jobs that arrive of type  $m$  and  $\gamma_m$  is the maximum number of jobs of type  $m$  that can appear at the same time without any of the jobs being dropped if  $N_{search}(s_i) = N$  (i.e., no vehicles are already busy with other jobs).  $p_a^m(\lambda_m(s_i), k_m > \gamma_m)$  is found as follows

$$p_a^m(\lambda_m(s_i), k_m > \gamma_m) = 1 - \sum_{k_m=0}^{\gamma_m} p_a^m(\lambda_m(s_i), k_m). \quad (6)$$

For a state,  $s_i$ , a maximum of  $n_m(s_i)$  jobs of type  $m$  can be completed in a time step, where  $n_m(s_i)$  is the number of jobs of type  $m$  currently being serviced at state  $s_i$ . Therefore, the probability of  $\chi$  jobs of type  $m$  being completed is

$$p_c^m(\chi) = \binom{n_m(s_i)}{\chi} (p_c^m)^\chi (1 - p_c^m)^{(n_m(s_i) - \chi)}. \quad (7)$$

The set of all completion probability for each job type  $m$  in state  $s_i$  is

$$\mathbf{P}_c^m = \{p_c^m(\chi = 0), \dots, p_c^m(\chi = n_m(s_i))\}. \quad (8)$$

The joint probability of  $k$  jobs arriving and  $\chi$  jobs completing is calculated for each combination of jobs arriving and completing. Each joint probability,  $P_j$  contributes to the transition from state  $s_i$  to a new state,  $s_j$ . If  $s_j \in S$ , then the joint probability is added to the appropriate column in the transition matrix. In cases where  $s_j \notin S$ , too many jobs have arrived to the system at once and not all of them can be serviced given  $N_{search}(s_i)$ . Therefore, a policy for determining the service priority of arriving jobs must be defined. As a result of this policy, one or more jobs will be dropped. Once the policy has determined which possible state the joint probability can transition the current state to, the joint probability can be added to the appropriate column in the transition matrix. The total probability of dropping a job in state  $s_i$  is defined as:

$$P_{drop}(s_i) = \sum_{j \in D} P_j, \quad (9)$$

where  $D$  denotes the set of joint probabilities that, if allowed, would cause the system to transition to an impossible state (i.e.,  $N_{search}(s_i) \leq 0$ ).

#### D. Steady State Distribution

Using the transition matrix, the steady state (or limiting) distribution,  $\pi$ , of the system can then be found. The steady state distribution describes the probability of being in each of the states at any given time. The limiting distribution,  $\pi$ , is defined as

$$\pi = \lim_{p \rightarrow \infty} T^p \cdot \pi_0, \quad (10)$$

where  $\pi_0$  is the initial distribution. In the case of SSS missions, the system always starts off with zero jobs being serviced (i.e.,  $\pi_0 = [1, 0, \dots, 0]^T$ ).  $\pi$  is found by repeated matrix multiplications. However, since it is possible to move from any state to any other state in the finite state system and there is a non-zero probability of staying in the same state, the transition matrix is irreducible and aperiodic. Therefore, the limiting probability is unique [15] and it also satisfies the following

$$\pi = \pi \cdot T \text{ and } \sum_i \pi_i = 1. \quad (11)$$

Thus, the steady state distribution can be found by solving the linear system above. The steady state distribution can then be used to determine the expected number of dropped jobs,  $d$  as follows

$$d = (P_{drop} \cdot \pi) T_s, \quad (12)$$

where  $P_{drop} \cdot \pi$  is the dot product between the probability of dropping a job and the stationary distribution (i.e., a

weighted dropped job probability for one time step across all states).  $T_s = \bar{t}/t_s$  is the total number of time steps required to complete the mission, where  $\bar{t}$  is the weighted average mission time across all the states. The mission time for each state,  $s_i$ , is the average time  $N_{search}(s_i)$  vehicles would take to cover the search area,  $A$  (i.e.,  $A/N_{search}(s_i)$ ).

#### E. Constant Coverage Rate Scenarios

Although the Hybrid Model was designed to model SSS missions with varying coverage rates, the state space representation can also model missions with constant coverage rates. In such scenarios the coverage rate function which defines the relationship between the real mission dynamics (e.g., vehicle formation, sensing speed, velocity, etc.) is given by  $\mathcal{F} = r$ , where  $r$  is a constant and is not dependent upon the number of available search vehicles. An example is shown in Section IV-A. The results are also compared against those given by the previously developed Queuing Model [2].

### IV. EXPERIMENTAL RESULTS

The validation results shown in this section will consider a wildfire scenario where the swarm is tasked with monitoring injured people who may be trapped due to the rapidly expanding fire (job type 1) and putting out brush fires that have sparked (job type 2). A grid-based environment, 50x50 in size, is used to simulate mission where the swarm searches the area using a lawn-mower (boustrophedon) pattern. Jobs are spread across the environment randomly from a uniform distribution. The environment is broken down in to 1x1 cells. Only one job can be present in each cell. The Hybrid Model's ability to predict system performance is compared against the previously developed Queuing Model in Section IV-A. The Hybrid Model is further evaluated in scenarios with varying coverage rates in Section IV-B. For both scenarios, two test configurations were simulated where the number of each job was varied, as summarized in Table II. Swarm sizes of 20, 25, 30 and 35 were tested.

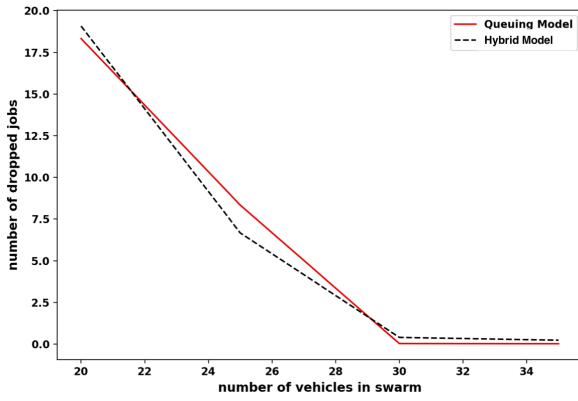
TABLE I: Resource Requirements for Each Job Type

Job Type	Required No. of Veh.	$\mu_m$ (jobs/time unit)
1	10	1/10
2	15	1/2

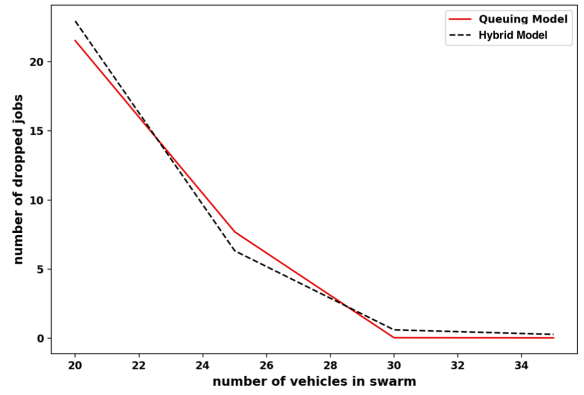
TABLE II: Test Configurations

Configuration	Type 1 Jobs	Type 2 Jobs
1	10	10
2	15	10

The required resources for each job type is displayed in Table I. Job Type 1 requires 10 vehicles while Job Type 2 requires 15. Table I also shows the mean service rate for each job type. The two different job configurations shown in Table II are explored for both the varying (Section IV-B) and constant (Section IV-A) coverage rate validations. In Configuration 1 there are 10 jobs of each type present in the environment. Configuration 2 has 15 jobs of type 1 present and 10 jobs of type 2 present.

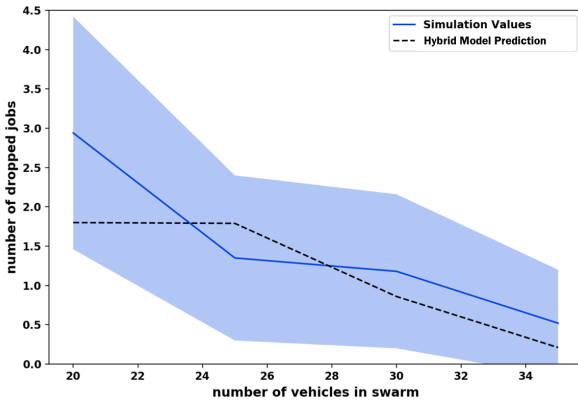


(a) Configuration 1

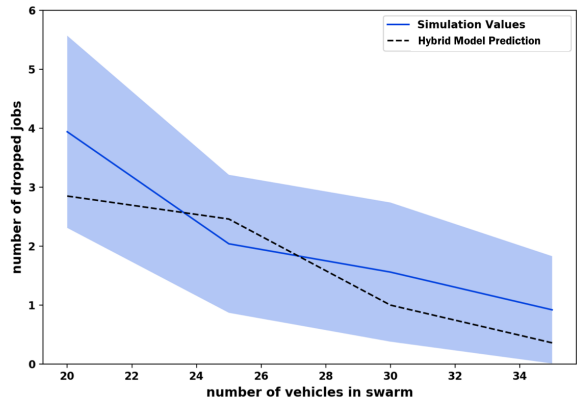


(b) Configuration 2

Fig. 2: Comparison between the Hybrid Model and the previously implemented Queuing Model [2].



(a) Configuration 1



(b) Configuration 2

Fig. 3: Comparison between the simulated performance of the swarm and the predicted performance from the Hybrid Model.

### A. Comparison to Queuing Model

Although the Hybrid Model's state space was designed to capture the dynamically changing coverage rate of the swarm as it transitions from one state to another, it can also be used to predict the performance in missions where the coverage rate remains constant (e.g., urban settings) by setting the coverage rate function  $\mathcal{F}$  equal to a constant. To evaluate the Hybrid Model's ability to model constant coverage rate scenarios, it is compared with the previously developed Queuing Model [2]. The Queuing Model assumed a constant coverage rate of 1 cell per time step (i.e.,  $\mathcal{F}(n) = 1$ ). Both models were used to predict the swarm performance in the two environment configurations given by Table II. The swarm size versus the predicted number of dropped jobs is shown in Figure 2. Figure 2a displays the comparison between the Hybrid Model and the Queuing Model for Configuration 1, while Figure 2b shows the comparison for Configuration 2. The results of the Hybrid Model are shown with the black dashed line, while the Queuing Model results are shown with the solid red line.

### B. Simulation Validation

The grid world SSS mission was simulated in MATLAB. The two configurations in Table II were simulated using

the resource requirements for the two job types shown in Table I. The location of all jobs was randomly and uniformly distributed for each simulated mission. Each cell could only be occupied by 1 job. The swarm was assumed to travel in a line formation where every robot sensed 1 cell per unit of time ( $\mathcal{F}(n) = n$ ). The dropped job policy used prioritized servicing jobs that required fewer vehicles.

For each swarm size tested, the results for 100 simulated missions were averaged. Figure 3 displays the swarm size versus the number of dropped jobs. The predicted values from the Hybrid Model are shown with the black dashed line, the simulated results are shown in blue. The solid color line represents the mean value, while the standard deviation is shown with the surrounding colored region. Configuration 1 and 2 are shown in Figures 3a and 3b respectively.

## V. DISCUSSION

Figure 2 indicates that the Hybrid Model is able to effectively predict swarm performance for cases where the swarm maintains a constant coverage rate. The predicted values given by the Hybrid Model are very similar to those found using the Queuing Model in both configurations. This illustrates the appropriateness of the Hybrid Model for use in predicting the performance of swarms in previously considered constant coverage rate scenarios.

The results in Figure 3 show that the Hybrid Model is also able to accurately model SSS missions with dynamically changing coverage rates for configurations where the relative arrival rates between the job types is the same (Figure 3a) and when they differ (Figure 3b). For both configurations the predicted values from the Hybrid Model fall within one standard deviation around the mean of the simulated values. For the same swarm size, the number of dropped jobs increases as the total job density in the environment increases. In both the prediction and the empirical study the number of dropped jobs decreases as the swarm size increases. The results further illustrate the appropriateness of the Hybrid Model as a prediction tool for a broader variety of SSS missions, including those with varying coverage rates.

For illustration purposes the results presented in this paper were found using a dropped job policy that prioritized servicing jobs with a lower vehicle requirement. However, the Hybrid Model formulation is flexible enough to allow for any dropped job policy to be used. The choice of policy affects the determination of which feasible state an identified dropped job transition leads to. In other words, it modifies the determination of the leaf node transitions in the transition matrix. Other possible policies could include jobs being serviced in a random order or jobs being serviced in order of their cost where cost is defined as the product of the number of vehicles needed and their service time, etc.

Furthermore, the formulation of the Hybrid Model gives the model the ability to capture different real world features and constraints on SSS missions such as the swarm's formation, sensor coverage, vehicle velocity, etc. These real world constraints and their effect on the coverage rate of the swarm given the number of vehicles currently in the swarm are mapped in to the functional relationship represented by  $\mathcal{F}(n)$ . By including this direct bridge between the real world scenario and the formulation for the arrival rates of each job type, the model is able to accurately describe the dynamics of the real mission. The functional relationship represented by  $\mathcal{F}(n)$  can be determined for each unique SSS mission scenario, thus ensuring that the Hybrid Model is reflective of the real world state dynamics of the swarm.

## VI. CONCLUSION AND FUTURE WORK

This paper presented a Hybrid Model for predicting the performance of a swarm during Swarm Search and Service missions. The state space representation provided a method for capturing the dynamically changing arrival rates caused by the difference in coverage rates. The predicted performance given by the Hybrid Model was compared against a previously validated Queuing Model. Results show little difference between the performance values predicted by the Hybrid Model and the Queuing Model, illustrating the appropriateness of the Hybrid Model for previously considered scenarios. In addition, a validation study was performed to compare the predicted system performance from the Hybrid Model to simulation results. The results illustrate that the Hybrid Model is able to describe the performance of the

swarm for more general missions where the coverage rate dynamically changes with swarm size.

While the previously developed Queuing Model was restricted to dropping jobs when resources were exceeded, future work could extend the Hybrid Model to substitute pauses for the dropping of jobs by including additional pausing states. Instead of dropping a job, the job would be delayed until sufficient vehicles returned to service it. Expected pausing times would then be reflected through increased mission times making the model better suited to applications in which servicing of all jobs is crucial. In addition, the use of the model as a planning tool aid for human operators who are tasked with determining resource requirements while balancing complex interactions between mission parameters will be evaluated.

## REFERENCES

- [1] R. Calvo, J. R. de Oliveira, M. Figueiredo, and R. Romero, "Inverse aco applied for exploration and surveillance in unknown environments," in *COGNITIVE 2011, the Third International Conference on Advanced Cognitive Technologies and Applications*. Citeseer, 2011.
- [2] M. Chandarana, M. Lewis, K. Sycara, and S. Scherer, "Determining effective swarm sizes for multi-job type missions," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4848–4853.
- [3] R. W. Beard, T. W. McLain, M. A. Goodrich, and E. P. Anderson, "Coordinated target assignment and intercept for unmanned air vehicles," *IEEE transactions on robotics and automation*, vol. 18, no. 6, pp. 911–922, 2002.
- [4] G. Arslan, J. R. Marden, and J. S. Shamma, "Autonomous vehicle-target assignment: A game-theoretical formulation," *Journal of Dynamic Systems, Measurement, and Control*, vol. 129, no. 5, pp. 584–596, 2007.
- [5] F. Bullo, E. Frazzoli, M. Pavone, K. Savla, and S. L. Smith, "Dynamic vehicle routing for robotic systems," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1482–1504, 2011.
- [6] J. W. Durham, R. Carli, P. Frasca, and F. Bullo, "Discrete partitioning and coverage control for gossiping robots," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 364–378, 2012.
- [7] M. Chandarana, M. Lewis, B. D. Allen, K. Sycara, and S. Scherer, "Swarm size planning tool for multi-job type missions," in *2018 Aviation Technology, Integration, and Operations Conference*, 2018, p. 3846.
- [8] K. Lerman, A. Martinoli, and A. Galstyan, "A review of probabilistic macroscopic models for swarm robotic systems," in *International Workshop on Swarm Robotics*. Springer, 2004, pp. 143–152.
- [9] A. Martinoli and K. Easton, "Modeling swarm robotic systems," in *Experimental Robotics VIII*. Springer, 2003, pp. 297–306.
- [10] S. Konur, C. Dixon, and M. Fisher, "Analysing robot swarm behaviour via probabilistic model checking," *Robotics and Autonomous Systems*, vol. 60, no. 2, pp. 199–213, 2012.
- [11] B. Açıkmeşe and D. S. Bayard, "A markov chain approach to probabilistic swarm guidance," in *2012 American Control Conference (ACC)*. IEEE, 2012, pp. 6300–6307.
- [12] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, "Inhomogeneous markov chain approach to probabilistic swarm guidance algorithm," in *5th Int. Conf. Spacecraft Formation Flying Missions and Technologies*, 2013.
- [13] —, "Probabilistic swarm guidance using optimal transport," in *2014 IEEE Conference on Control Applications (CCA)*. IEEE, 2014, pp. 498–505.
- [14] S. Berman, Á. Halász, M. A. Hsieh, and V. Kumar, "Optimized stochastic policies for task allocation in swarms of robots," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 927–937, 2009.
- [15] M. Harchol-Balter, *Performance modeling and design of computer systems: queueing theory in action*. Cambridge University Press, 2013.
- [16] J. R. Norris, *Markov chains*. Cambridge university press, 1998, no. 2.