

Stochastic Collection and Replenishment (SCAR): Objective Functions

Andrew W Palmer, Andrew J Hill and Steven J Scheduling¹

Abstract—This paper introduces two objective functions for computing the expected cost in the Stochastic Collection and Replenishment (SCAR) scenario. In the SCAR scenario, multiple user agents have a limited supply of a resource that they either use or collect, depending on the scenario. To enable persistent autonomy, dedicated replenishment agents travel to the user agents and replenish or collect their supply of the resource, thus allowing them to operate indefinitely in the field. Of the two objective functions, one uses a Monte Carlo method, while the other uses a significantly faster analytical method. Approximations to multiplication, division and inversion of Gaussian distributed variables are used to facilitate propagation of probability distributions in the analytical method when Gaussian distributed parameters are used. The analytical objective function is shown to have greater than 99% comparison accuracy when compared with the Monte Carlo objective function while achieving speed gains of several orders of magnitude.

I. INTRODUCTION

Resources such as fuel, battery charge and data storage space are a necessary component of autonomous systems. However, autonomous agents have a finite capacity for these resources and thus at some point will require the resource to be replenished, particularly when persistent autonomy is the goal. The aims of this paper are to introduce the Stochastic Collection and Replenishment (SCAR) scenario, and to present and compare two SCAR objective functions—a Monte Carlo method and a faster analytical method.

The SCAR scenario consists of two inverse problems—collection and replenishment. Collection refers to the case where user agents in the field are collecting a resource such as data or physical samples, whereas the replenishment case refers to agents using a resource such as fuel or battery charge. In the collection case, one or more collection agents travel to the field agents and retrieve the resource that the user agents have gathered. In the replenishment case, the replenishment agents travel to the user agents and replenish the resources of the user agents. In both cases, the aim of SCAR is to allow the user agents to continue operating without returning to base.

Several replenishment scenarios exist in the literature—air refuelling of a fleet of planes or Unmanned Aerial Vehicles (UAVs) [1], [2], [3], [4], refuelling of a constellation of satellites [5], [6], and recharging of UAVs using Unmanned Ground Vehicles (UGVs) [7]. The main collection scenario in the literature is the use of a data mule to collect

data from a sensor field [8], [9], [10]. The above scenarios are formulated as an NP-hard optimisation problem, and the methods proposed for solving them include heuristics, simulated annealing, dynamic programming, and branch and bound. The objective functions generally revolve around minimising factors such as the total time or tardiness of the schedule, and the total fuel used or total distance travelled by the replenishment agent.

The above literature have several limitations. Firstly, the replenishment agents are assumed to have sufficient capacity to fully fill all of the agents. However, when persistent autonomy is considered, it is clear that the replenishment agents themselves will have to be replenished at some point. Secondly, it is assumed that each user agent is replenished only once. Again, it is necessary to consider replenishing the user agents multiple times when performing persistent tasks. Finally, the agent parameters are treated as deterministic. In a practical system, many parameters such as speed and usage rates are stochastic in nature. Disregarding this uncertainty can lead to selection of suboptimal schedules from a risk point of view [11].

The SCAR scenario aims to address the above limitations by incorporating replenishment of the replenishment agent, multiple replenishments of the user agents, and uncertainty in the agent parameters. An open issue in optimisation of stochastic systems is estimation of the expected cost. A popular approach in the literature for calculating an expected cost has been scenario-based methods such as Monte Carlo simulation [12], [13]. This is simple to implement, but can be computationally expensive. Gaussian quadrature integration was used in [14] to generate an expected cost, while other authors have attempted to transform the stochastic problem into a deterministic problem using chance constraints [15] or conservative estimates of the uncertain parameters [16].

The main contribution of this paper is the formulation of two methods for calculating the expected cost in a SCAR scenario. The first is a Monte Carlo method that uses parameter values sampled from probability distributions to estimate an expected cost, while the second is an analytical solution that incorporates the entire probability distribution. The SCAR problem is firstly defined in Section II, and mathematical models of the SCAR scenario are developed in Section III. A deterministic model is developed for use in the Monte Carlo method, and a stochastic model for use in the analytical method. These methods are valid for any probability distribution, and are compared in this paper using Gaussian distributed variables. Section IV introduces a number of approximation methods for working with Gaussian distributed variables, including a new approximation to the

This work has been supported by the Rio Tinto Centre for Mine Automation and the Australian Centre for Field Robotics, University of Sydney, Australia.

¹The authors are with the Australian Centre for Field Robotics, University of Sydney, Australia. Email addresses: {a.palmer;a.hill;s.scheduling}@acfr.usyd.edu.au

Inverse Gaussian Distribution. A computational study of the objective functions is undertaken in Section V, followed by some concluding remarks in Section VI.

II. PROBLEM DEFINITION

In the SCAR scenario there are multiple user agents, which have a limited capacity of a resource. In the replenishment problem, the user agents use the resource over time, while one or more replenishment agents travel around replenishing the supplies of the user agents. To replenish a user agent, the replenishment agent must travel to the user agent, setup before beginning replenishment, replenish the user agent, and then pickup before it can travel to the next user agent. The user agent may continue to use the resource while being replenished. The inverse problem of the user agents collecting the resource is identical except for the direction of the resource flow.

The replenishment agents also have a limited capacity of the resource and must periodically travel to a replenishment point to be replenished. The fleet of user agents is heterogeneous, as is the fleet of replenishment agents. The parameters of the agents such as usage rate, replenishment rate, speed, setup and pickup time are stochastic. The user agents may be positioned apart from one another, and the motion of all agents may be constrained by roads or obstacles.

Both user and replenishment agents can be replenished multiple times in a single schedule. Any user agent can be replenished by at most one replenishment agent at any time, and any replenishment agent can only service at most one user agent at any time. Preemption is not considered; i.e. once a replenishment agent has begun replenishing a user agent, it must continue until either the user agent is fully replenished or the replenishment agent has exhausted its supply of the resource.

User agents operate in their own small operational areas such that they are unable to reach the replenishment point. As the distances travelled by the replenishment agent are significantly larger than the size of these operational areas, any variations in the travel times of the replenishment agent due to the movements of the user agents can be accounted for in the uncertainty of the setup and pickup times of the replenishment agent.

The consequence of a user agent running out of the resource may be either catastrophic (for example, the loss of the agent) which corresponds to a hard deadline, or non-catastrophic which corresponds to a soft deadline. A cost is incurred if the deadline is not met. A hard deadline incurs a step cost, while a soft deadline yields a linearly increasing cost. A single replenishment agent scenario with soft deadlines for the user agents has been assumed for this paper.

III. MODEL FORMULATION

This section presents a mathematical model for both the deterministic and stochastic single replenishment agent problems. The mathematics for the collection problem is identical if the resource considered is the free space remaining instead of the actual resource collected.

A. Parameters

Each user agent, i , has the following parameters:

- $u_{i,max}$ is the resource capacity
- u_i is the current resource level
- r_i is the resource usage rate

The replenishment agent, a , has the following parameters:

- $u_{a,max}$ is the resource capacity
- u_a is the current resource level
- f_a is the resource replenishment rate into the user agent
- $t_{s,a}$ is the setup time
- $t_{p,a}$ is the pickup time
- v_a is the velocity

The replenishment point, r , has the following parameters:

- $t_{s,r}$ is the setup time
- $t_{p,r}$ is the pickup time
- f_r is the resource replenishment rate into the replenishment agent

The distance between the replenishment agent and a user agent is denoted $d_{a,i}$ and the distance between the replenishment agent and the replenishment point is denoted $d_{a,r}$. In the following equations, an upper-case variable denotes a probability distribution. Units do not matter as long as they are consistent.

B. Deterministic Equations

The cost, c , of a schedule when using a soft deadline is a ratio between 0 and 1 which indicates what proportion of time the agents are empty. A value of 0 means that none of the user agents were empty at any point during the schedule, while a value of 1 means that all of the user agents were empty for the entire time required to complete the schedule. A schedule denotes a series of tasks for the replenishment agents in the order to be executed. There are two possible task types—replenishing a user agent, or replenishing the replenishment agent. The first involves the replenishment agent travelling to the user agent, while the second requires the replenishment agent to travel to the replenishment point. An example schedule for a single-replenishment agent, four-user agent system might be [0, 2, 'r', 3, 1]. In this schedule, the replenishment agent would replenish user agents 0 and 2 before visiting the replenishment site, and would then replenish user agents 3 and 1.

Each time a user agent is replenished, the time that it was empty up until that point must be calculated. For each task replenishing a user agent, the time empty in a segment, $t_{segment,i}$, can be calculated as:

$$t_{segment,i} = \max(0, t_{b,i} - t_{e,i})$$

where

- $t_{b,i}$ is the time at which the replenishment agent began replenishing user agent, i
- $t_{e,i}$ is the time when user agent, i , will fully deplete its resources

Once the schedule has been executed, the time empty for each user agent must be calculated based on when it was

last replenished. So for each user agent, the time empty at the end of the schedule, $t_{end,i}$ is:

$$t_{end,i} = \max(0, t_{max} - t_{e,i})$$

where t_{max} is the time that the schedule will finish being executed. For a soft deadline, the cost of the schedule is then defined as:

$$c = \frac{\sum_{i=0}^n t_{segment,i} + \sum_{i=0}^n t_{end,i}}{nt_{max}} \quad (1)$$

where n is the total number of user agents. The nt_{max} term normalises the cost such that $0 \leq c \leq 1$. A hard deadline would be represented by a cost proportional to the number of user agents that ran out of the resource. For each user agent, the time till empty can be calculated as:

$$t_{e,i} = \frac{u_i}{r_i} \quad (2)$$

In most cases, $u_i = u_{i,max}$. However, if the replenishment agent did not have sufficient resources to completely replenish the user agent previously, then u_i will be lower than $u_{i,max}$.

The time until a user agent is replenished depends on when the replenishment agent last finished servicing a user agent. Let $t_{f,i}$ be the time that the replenishment agent has finished replenishing user agent i , has packed up and is ready to travel to the location of the next task. Then the time that user agent $i+1$ is replenished, $t_{b,i+1}$, will depend on $t_{f,i}$:

$$t_{b,i+1} = t_{f,i} + \frac{d_{a,i}}{v_a} + t_{s,a} \quad (3)$$

Then $t_{f,i+1}$ can be calculated as:

$$t_{f,i+1} = t_{b,i+1} + t_{r,i+1} + t_{p,a} \quad (4)$$

$t_{r,i}$ is the time that it takes to replenish user agent i and depends on the current resource level, the resource usage rate and the replenishment rate:

$$t_{r,i} = \frac{u_{i,max} - u_i}{f_a} + \frac{t_{r,i} r_i}{f_a} = \frac{u_{i,max} - u_i}{f_a - r_i}$$

However, if the replenishment agent does not have sufficient resources to fully replenish the user agent, then the time to replenish the user agent will be dependent on the resource level of the replenishment agent:

$$t_{r,i} = \min\left(\frac{u_a}{f_a}, \frac{u_{i,max} - u_i}{f_a - r_i}\right) \quad (5)$$

The resource level in user agent, i , after being replenished, $u_i^{t_1}$, will be:

$$u_i^{t_1} = \min(u_{i,max}, u_i^{t_0} + t_{r,i}(f_a - r_i))$$

where $u_i^{t_0}$ is the resource level before being replenished. The resource level in the replenishment agent will then be:

$$u_a^{t_1} = \max(0, u_a^{t_0} - t_{r,i} f_a)$$

For an 'r' task, the total time $t_{f,a}$ will include the travel time to the replenishment site, setup and pickup times

of the replenishment site, and the time to replenish the replenishment agent:

$$t_{f,a} = t_{f,i} + \frac{d_{a,r}}{v_a} + t_{s,r} + \frac{u_{a,max} - u_a}{f_r} + t_{p,r} \quad (6)$$

C. Stochastic Equations

As noted previously, a number of the parameters of the system are stochastic in practice, and a deterministic cost calculation does not capture the inherent risk of a particular schedule. Using just the expected values of the parameters, the cost for two different schedules may be zero. Once uncertainty is taken into account, one schedule could be found to be more risky than the other.

The uncertainty can be incorporated by calculating an expected cost using the probability distributions of the parameters. The most straight-forward method to do this is a Monte Carlo cost analysis where the deterministic cost calculation is run multiple times using values sampled from the probability distributions. The quality of the cost estimate will improve on average as more samples are used. However, using a large number of samples has a significant impact on computation time when running a schedule optimisation.

A faster method of calculating the expected cost is to explicitly calculate the probability distribution for the time empty at each step of the schedule and calculate the expected cost from these probability distributions. The time that the user agent is empty is given by:

$$T_{segment,i} = T_{b,i} - T_{e,i}$$

As the time empty can only be ≥ 0 , the expected value of the time empty is given by:

$$E[T_{segment,i}] = \int_0^\infty xp(T_{segment,i}) dx \quad (7)$$

where $p(T_{segment,i})$ denotes the probability density function of $T_{segment,i}$. Similarly at the end of the schedule:

$$T_{end,i} = T_{max} - T_{e,i}$$

$$E[T_{end,i}] = \int_0^\infty xp(T_{end,i}) dx$$

Therefore the expected cost of the schedule for a soft deadline is:

$$E[C] = \frac{\sum_{i=0}^n E[T_{segment,i}] + \sum_{i=0}^n E[T_{end,i}]}{nE[T_{max}]} \quad (8)$$

where $E[T_{max}]$ is given by the mean value of the distribution $p(T_{max})$. For a hard deadline, the cost would reflect the probability of the user agents running empty rather than the total time that they are empty.

Most of the following equations are similar to the deterministic equations from Section III-B, using probability distributions rather than explicit values. Equation (2) becomes:

$$T_{e,i} = \frac{U_i}{R_i}$$

Equation (3) becomes:

$$T_{b,i+1} = T_{f,i} + \frac{d_{a,i}}{V_a} + T_{s,a}$$

Let $T_{m,i+1}$ be the time that the user agent is finished being replenished:

$$T_{m,i+1} = T_{b,i+1} + T_{r,i+1} \quad (9)$$

Using (9), (4) becomes:

$$T_{f,i+1} = T_{m,i+1} + T_{p,a} \quad (10)$$

Equation (5) cannot be simply modified like the previous equations as taking the minimum between two probability distributions would lead to a discontinuous probability distribution. Therefore, a series of steps are required to calculate the time that it takes to replenish the user agent. Firstly, calculate the resource level of the user agent at time $T_{b,i}^{t_1}$:

$$U_i^{t_1} = U_i^{t_0} - R_i(T_{b,i}^{t_1} - T_{m,i}^{t_0})$$

where $T_{m,i}^{t_0}$ is the time that the previous replenishment of the user agent was finished at. Depending on the choice of probability distribution, some of the distribution will extend into the negative part of the domain. However, a negative level of resource does not make sense, so steps should be taken to minimise the probability of a negative value. Similarly, a resource level greater than the maximum resource level is impossible and should be accounted for. Section IV-E discusses a possible adjustment technique. A # will be used to denote variables that have been adjusted to match the physical constraints.

The next step is to calculate the time it will take for the replenishment agent to replenish the user agent assuming that the replenishment agent has sufficient resource levels to achieve this, $T_{r,i}$:

$$T_{r,i} = \frac{u_{i,max} - U_i^{t_1\#}}{F_a - R_i}$$

Taking into account the quantity of resource used by the user agent during the replenishment, the amount of resource to be replenished, Q_i , is given by:

$$Q_i = u_{i,max} - U_i^{t_1\#} + T_{r,i}R_i$$

However, the replenishment agent may not have sufficient resources to fully replenish the user agent, so Q_i must be adjusted against U_a . The actual replenishment time can then be calculated as:

$$T_{r,i} = \frac{Q_i\#}{F_a}$$

The resource level of the user agent after being replenished, $U_i^{t_2}$, is given after adjustment against $u_{i,max}$ by:

$$U_i^{t_2} = (U_i^{t_1\#} + U_a - T_{r,i}R_i)\#$$

U_a is used instead of $T_{r,i}F_a$ to ensure that the adjusted distribution has an appropriate shape. The resource level of the replenishment agent is then given by:

$$U_a^{t_2} = (U_a^{t_1} - Q_i\#)\#$$

Finally, (6) becomes:

$$T_{f,a} = T_{f,i} + \frac{d_{a,r}}{V_a} + T_{s,r} + \frac{u_{a,max} - U_a}{F_r} + T_{p,r} \quad (11)$$

IV. APPROXIMATION METHODS

It is assumed that the stochastic parameters can be described by independent probability distributions. While any probability distribution is valid, in this paper the parameters have been modelled with Gaussian distributions as these are relatively easy to work with. Additions and subtractions of Gaussian distributed variables result in a Gaussian distribution, whereas other operations such as division, multiplication and inversion result in non-Gaussian distributions. To facilitate the propagation of the distributions throughout the objective function, it is convenient to approximate the non-Gaussian distributions as Gaussian distributions. The following subsections discuss these approximation methods. It should be noted that the probability distributions for quantities such as the time that the user agent was last replenished at and the time taken to replenish the user agent are not independent and there will be some covariance between them. However, for this paper the covariance between all calculated distributions has been assumed to be zero. It will be demonstrated later in this paper that these are acceptable approximations and assumptions.

A. Inverse Gaussian Distribution

An approximation of the probability distribution resulting from the inverse of a Gaussian distributed variable was not found in the examined literature. Therefore, an approximation was formulated through a numerical study. Consider:

$$I = \frac{c}{G}$$

where I is an inverse Gaussian distributed variable, G is a Gaussian distributed variable described by mean, μ_G , and standard deviation, σ_G , and c is a constant. The distribution of I is then approximately described by a Gaussian distribution with mean, μ_I , and standard deviation, σ_I , where:

$$\mu_I = \frac{C}{\mu_G} \quad \sigma_I = \frac{\sigma_G C}{\mu_G^2} \quad (12)$$

Fig. 1 shows the inverse Gaussian distribution, the resultant approximation and the residual error between the two distributions for an initial μ_G/σ_G ratio of 20. The Kullback-Leibler (KL) Divergence [17] for various ratios is shown in Fig. 2. The KL Divergence is a measure of the difference between two distributions, and as it nears 0, the approximation approaches the actual distribution. Fig. 2 shows that the approximation improves as the μ_G/σ_G ratio increases.

B. Gaussian Ratio Distribution

A method for approximating a ratio of Gaussian distributed variables is given in [18]. A ratio distribution, R :

$$R = \frac{E}{F}$$

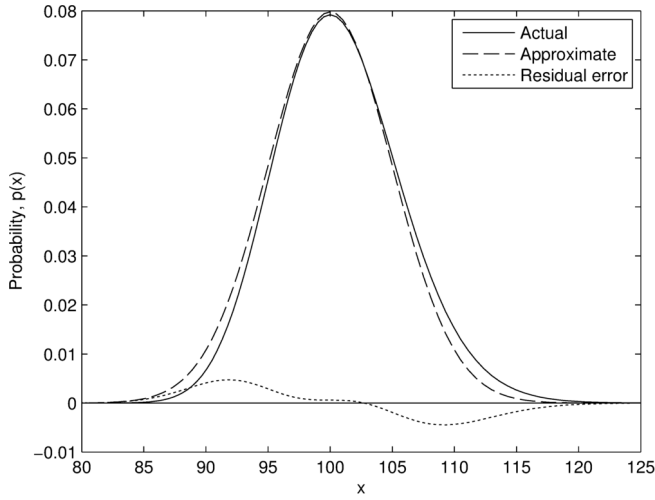


Fig. 1. Inverse Gaussian Distribution and Approximation for $\mu_G/\sigma_G = 20$

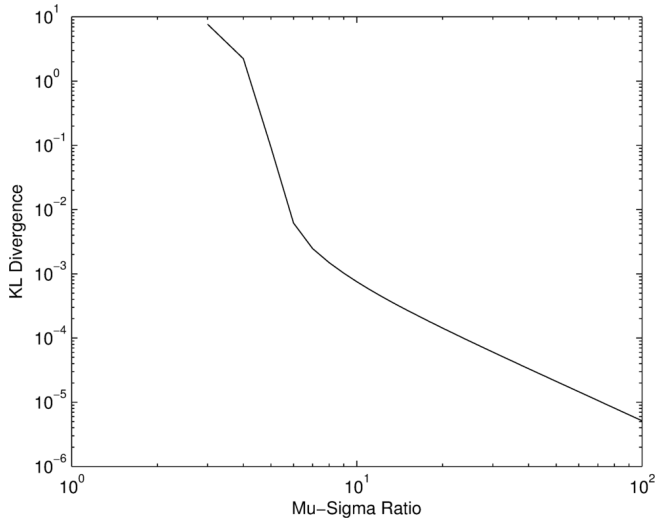


Fig. 2. KL Divergence for the Inverse Gaussian Approximation

with $\mu_E, \sigma_E, \mu_F, \sigma_F$ and $\rho = 0$ can be approximated with a Gaussian distribution where:

$$r = \frac{\sigma_F}{\sigma_E}, a = \frac{\mu_E}{\sigma_E} \text{ and } b = \frac{\mu_F}{\sigma_F}$$

$$\mu_R = \frac{a}{r(1.01b - 0.2713)} \quad (13)$$

$$\sigma_R = \frac{1}{r} \sqrt{\frac{a^2 + 1}{b^2 + 0.108b - 3.795} - r^2 \mu_R^2} \quad (14)$$

This approximation is only valid for $a < 2.5, b > 4$ [18]. As $a \rightarrow \infty$, the ratio distribution approaches the inverse Gaussian distribution and can be approximated as per section IV-A. For situations where $a \geq 2.5$, the inverse Gaussian approximation method is used with E treated as a constant, $e = \mu_E$.

C. Gaussian Product Distribution

An approximation to the product of two Gaussian distributed variables is presented in [19]. For a Gaussian product

distribution, M :

$$M = EF$$

Then:

$$\mu_M = \mu_E \mu_F \quad (15)$$

$$\sigma_M = \sigma_E^2 \sigma_F^2 (1 + \delta_E^2 + \delta_F^2) \quad (16)$$

where

$$\delta_x = \frac{\mu_x}{\sigma_x}$$

The authors note that the approximation improves as δ_E and δ_F become large.

D. Expected Value

Equation (7) gives the expected time empty for part of a schedule. Assuming the expected time empty is a Gaussian distribution with μ and σ , this becomes:

$$E[G] = \int_0^{\infty} \frac{x}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx$$

Solving the definite integral gives:

$$E[G] = \frac{\mu}{2} (1 + \text{erf}(\frac{\mu}{\sigma\sqrt{2}})) + \frac{\sigma}{\sqrt{2\pi}} e^{-\frac{\mu^2}{2\sigma^2}} \quad (17)$$

where erf is the Gauss error function.

E. Adjusting Distributions

Gaussian distributions have a domain of $(-\infty, \infty)$, and therefore there is some probability that a sampled value will be outside the physical limits. When considering quantities such as resource level for example, a negative value is not physically possible. Similarly, the maximum value is constrained by the capacity of the agent. A simple method was devised that shifts the mean and adjusts the standard deviation such that the majority of the distribution is within the logical bounds. If the distribution is within 3 standard deviations of 0, then:

- $\mu^\#$ is calculated using (17)
- $\sigma^\# = \mu^\# / 3$

By keeping the mean 3 standard deviations away from 0, the probability of a negative value is 0.13%. As an example, consider a future resource level which has been calculated as $\mu = -2$ and $\sigma = 10$. As shown by the shaded section in Fig. 3, 42.07% of the probability currently lies in the positive domain. The expected positive value of this distribution is calculated using (17) as 3.069. Therefore, the adjusted parameters are:

$$\mu^\# = 3.069 \text{ and } \sigma^\# = \mu^\# / 3 = 1.023$$

The adjusted distribution is shown in Fig. 3. As the initial distribution moves further into the negative part of the domain, the adjusted values will both approach 0. The following steps should be used to adjust against the maximum level of the resource:

- Subtract the resource level from the maximum resource level

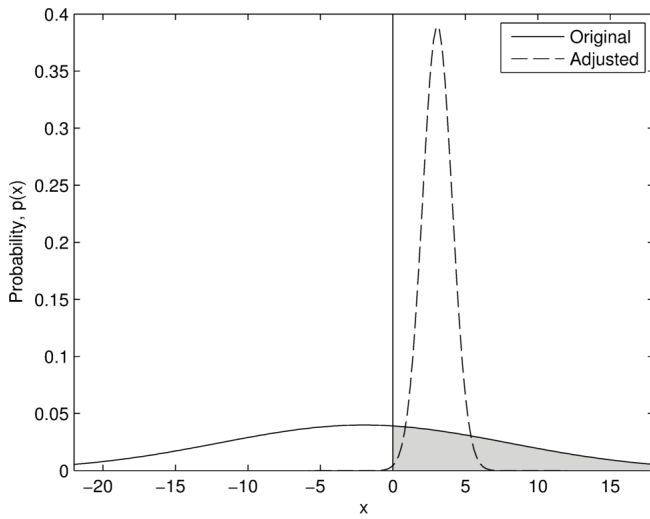


Fig. 3. Original and Adjusted Distributions

- Perform the adjustment on the resultant distribution
- Subtract the adjusted distribution from the maximum resource level

Note that this is just one possible method for adjusting the distribution. Using other distributions such as the Gamma distribution which does not go below 0 may provide a better approximation. However, the difficulty of working with both Gaussian and non-Gaussian distributions then arises.

V. COMPUTATIONAL STUDY

To test the relative performance of the objective functions, a Monte Carlo analysis was performed. For each randomly generated schedule, the expected cost was calculated using both a Monte Carlo approach with the equations detailed in Section III-B, and an analytical approach using the equations detailed in Section III-C combined with the approximation methods detailed in Section IV. Each Monte Carlo cost was calculated using 1000 samples from each of the parameter's probability distributions as a compromise between accuracy and computation time. This trade-off is shown in Fig. 4. The scenario consisted of a single replenishment agent, a homogeneous set of 6 user agents with soft deadlines, and a single replenishment point. The distances between the user agents and the replenishment point are shown in Fig. 5. The replenishment agent starts at the replenishment point. The parameters of the agents and the replenishment point are detailed in Tables I, II and III.

TABLE I
REPLENISHMENT AGENT PARAMETERS

Parameter	Mean	Standard Deviation
u_{max} (units)	5000	0
F (units/s)	10	0.5
T_s (s)	60	20
T_p (s)	20	5
V (m/s)	15	0.5

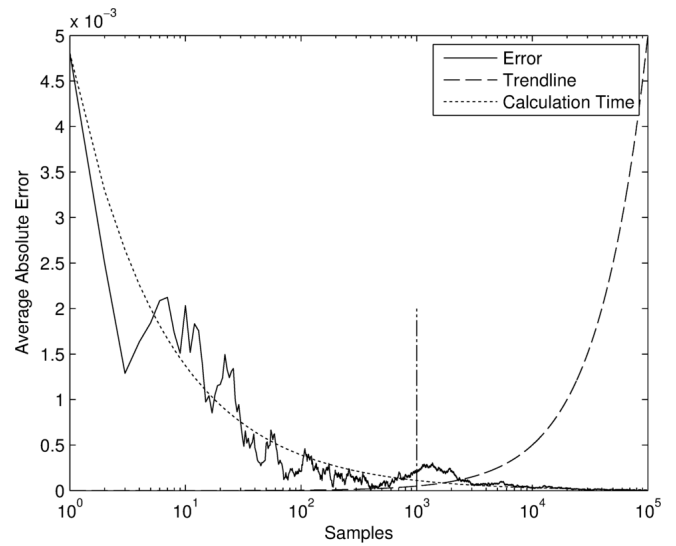


Fig. 4. Average absolute error versus Monte Carlo samples. Calculation time shown as a guide. 1000 sample point marked.

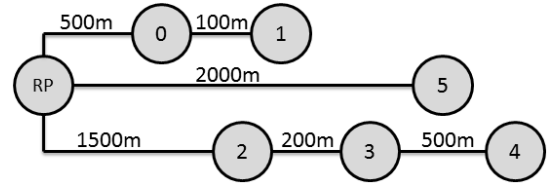


Fig. 5. Layout showing the location of the Replenishment Point (RP) and User Agents

A random selection of 90,000 schedules of between 5 and 10 tasks was generated with the only limitation being that consecutive tasks could not be the same. A schedule of [0,1,0,1,2] would be valid, while a schedule of [0,0,1,1,2] would not. The sample of 90,000 schedules represents around 0.1% of the possible schedules, and this number was chosen to limit the run time of the Monte Carlo analysis. One third of the schedules were run with all agents starting at empty, one third with all agents starting at half capacity, and the final third with all agents starting at full capacity. Fig. 6 shows that the analytical objective function provides a close approximation to the Monte Carlo generated cost. A

TABLE II
USER AGENT PARAMETERS

Parameter	Mean	Standard Deviation
u_{max} (units)	1000	0
R (units/s)	0.5	0.05

TABLE III
REPLENISHMENT POINT PARAMETERS

Parameter	Mean	Standard Deviation
T_s (s)	30	10
T_p (s)	10	1
F (units/s)	20	1

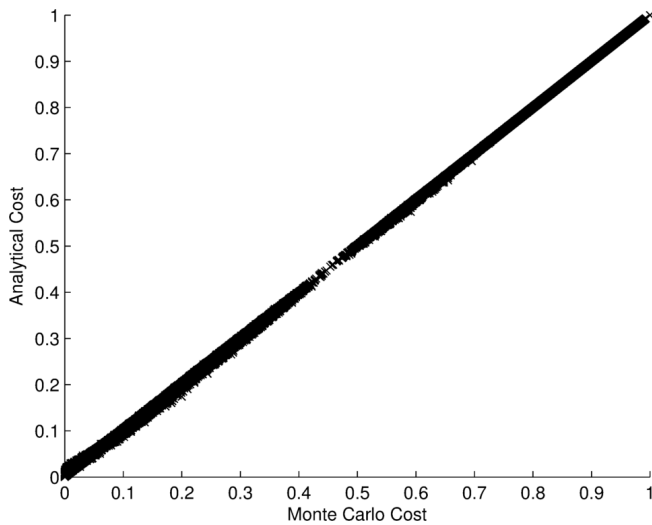


Fig. 6. Analytical Cost versus Monte Carlo Cost

linear trend line was fitted to the data with an equation of $y = 0.9983x + 0.0013$ and an R^2 value of 0.99996. (An R^2 value close to 1 indicates a good fit.)

For each of the initial conditions, the mean and standard deviation of the analytical costs minus the Monte Carlo costs is given in Table IV. Fig. 7 shows the differences plotted against the Monte Carlo cost. The error generally decreases as the cost approaches 1. Some of this error can be attributed to the use of only 1000 iterations for the Monte Carlo cost function. As shown in Fig. 4, this error is on average around 0.5×10^{-3} . The rest of the error is likely due to the approximations and assumptions from Section IV that were used to calculate the analytical cost. For example, the assumption that the probability distributions are independent is not necessarily valid. However, based on these results, the effects are arguably negligible.

While these measures show the consistency between the two methods, the primary role of the objective function is to discriminate between good and bad schedules. To test the accuracy of the analytical objective function, over 5 billion comparisons were made between schedules. For two schedules A and B, a correct result is recorded if both the Monte Carlo method and analytical method agree on which of A and B is better. Comparisons where the Monte Carlo cost for A and B were both equal to 0 were not included as the analytical method will return a very small but non-zero number for these schedules. The comparisons were run on the results from each set of initial conditions individually

TABLE IV
ANALYTICAL COST MINUS MONTE CARLO COST FOR 6 USER AGENTS

Initial Conditions	Mean ($\times 10^{-3}$)	Standard Deviation ($\times 10^{-3}$)
Empty	0.02	0.56
Half full	-0.10	2.79
Full	2.02	2.72

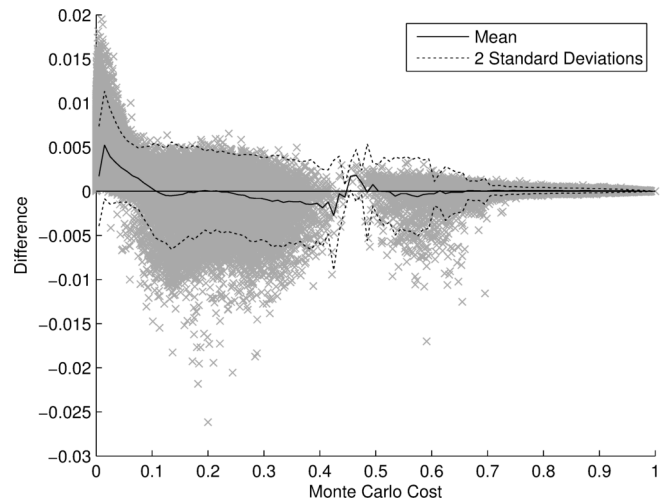


Fig. 7. Analytical Cost minus Monte Carlo Cost versus Monte Carlo Cost for 6 User Agents

as well as on the entire set of results. The results of these comparisons are given in Table V.

The weighted average accuracy using the individual results is 99.00%. The accuracy when the initial conditions are full is lower than for the other initial conditions. Table VI shows that the mean difference in cost between schedules for the incorrect comparisons is very small in comparison to the overall mean difference for the full initial conditions. While it isn't 100% accurate, for the cases in which it produces incorrect results the cost differences are possibly small enough that the schedules could be considered effectively identical.

If all of the initial conditions are considered together, the accuracy is 99.66%. This is a slightly unfair comparison as the initial conditions significantly influence the cost. However, the result highlights the discrimination accuracy of the method, in particular for cases of more extreme cost variation.

A similar study was performed on a larger scenario

TABLE V
COMPARISON ACCURACY FOR 6 USER AGENTS

Initial Conditions	Number of Comparisons	Correct
Empty	446129609	99.97%
Half full	449984570	99.05%
Full	342711946	97.65%
All	3938386683	99.66%

TABLE VI
MEAN DIFFERENCE BETWEEN SOLUTIONS FOR FULL INITIAL CONDITIONS

	Monte Carlo ($\times 10^{-3}$)	Analytical ($\times 10^{-3}$)
All comparisons	27.51	29.60
Incorrect comparisons	1.02	1.80

involving a heterogeneous set of 20 user agents and schedule lengths of between 16 and 20 tasks. Table VII shows the error between the analytical cost and Monte Carlo cost. With the increased number of user agents and tasks, the mean error has increased slightly while the standard deviation has overall decreased. Table VIII shows that the comparison accuracy for the empty, half full and full cases has decreased by at most 0.81%. The weighted average accuracy is 98.8%, while the overall accuracy has not changed from 99.66%.

The main advantage of the analytical objective function is the computation time. Compared to the Monte Carlo method, the analytical method can be several orders of magnitude faster. Using a schedule with 10 tasks as an example, the Monte Carlo method using 1000 samples took 665ms to calculate the cost, while the analytical method took only 2ms. These times were calculated using the same hardware and both methods were implemented in Python by the same programmer.

VI. CONCLUSIONS

This paper introduced a framework for the SCAR scenario. Methods for approximating Gaussian probability distributions were discussed and a new method of approximating the inverse Gaussian distribution was presented. These were used to quickly calculate an analytical expected cost of a schedule. A computational study between the analytical objective function and a Monte Carlo generated expected cost showed that the analytical objective function was over 99% accurate while computing several orders of magnitude faster in comparison to the Monte Carlo method.

In future research, the new objective function will be used in conjunction with combinatorial optimisation techniques such as simulated annealing and branch and bound to improve the quality of results and speed of computation. Future work could include accounting for the covariance

between distributions and performing a sensitivity analysis on the analytical method. Possible extensions to this work include using and collecting multiple resources such as fuel and data at the same time, moving the operational areas of the user agents at uncertain rates, and considering multiple replenishment agents.

REFERENCES

- [1] Z. Jin, T. Shima, and C. Schumacher, "Optimal scheduling for refueling multiple autonomous aerial vehicles," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 682–693, Aug. 2006.
- [2] Z. Jin, T. Shima, and C. Schumacher, "Scheduling and sequence reshuffle for autonomous aerial refueling of multiple UAVs," *2006 American Control Conference*, p. 6 pp., 2006.
- [3] S. Kaplan and G. Rabadi, "Exact and heuristic algorithms for the aerial refueling parallel machine scheduling problem with due date-to-deadline window and ready times," *Computers & Industrial Engineering*, vol. 62, no. 1, pp. 276–285, Feb. 2012.
- [4] S. Kaplan and G. Rabadi, "Simulated annealing and metaheuristic for randomized priority search algorithms for the aerial refuelling parallel machine scheduling problem with due date-to-deadline windows and release times," *Engineering Optimization*, vol. 45, no. 1, pp. 67–87, 2013.
- [5] H. Shen and P. Tsiotras, "Optimal Scheduling for Servicing Multiple Satellites," in *AIAA/AAS Astrodynamics Specialists Conference*, 2002, pp. 5–8.
- [6] P. Tsiotras and A. De Nailly, "A Comparison Between Peer-to-Peer and Single-Spacecraft Refueling Strategies," in *Infotech at Aerospace*, Crystal City, District of Columbia, 2005.
- [7] N. Mathew, S. Smith, and S. Waslander, "A graph-based approach to multi-robot rendezvous for recharging in persistent tasks," in *IEEE Int. Conf. on Robotics and Automation*, 2013, pp. 3482–3487.
- [8] O. Tekdas and V. Isler, "Using mobile robots to harvest data from sensor fields," *Wireless Communications, IEEE*, vol. 16, no. 1, pp. 22–28, 2009.
- [9] D. Bhadauria, O. Tekdas, and V. Isler, "Robotic data mules for collecting data over sparse sensor fields," *Journal of Field Robotics*, vol. 28, no. 3, pp. 388–404, 2011.
- [10] O. Tekdas, D. Bhadauria, and V. Isler, "Efficient data collection from wireless nodes under the two-ring communication model," *The International Journal of Robotics Research*, vol. 31, no. 6, pp. 774–784, Mar. 2012.
- [11] S. Kuindersma, R. Grupen, and A. Barto, "Variational Bayesian optimization for runtime risksensitive control," *Robotics: Science and Systems*, 2012.
- [12] M. Bassett, J. Pekny, and G. Reklaitis, "Using detailed scheduling to obtain realistic operating policies for a batch processing facility," *Industrial Engineering and Chemical Research*, vol. 36, pp. 1717–1726, 1997.
- [13] X. Lin, S. L. Janak, and C. a. Floudas, "A new robust optimization approach for scheduling under uncertainty," *Computers & Chemical Engineering*, vol. 28, no. 6-7, pp. 1069–1085, June 2004.
- [14] M. G. Ierapetritou and E. N. Pistikopoulos, "Global optimization for stochastic planning, scheduling and design problems," in *Global Optimization in Engineering Design*, 1996, pp. 231–288.
- [15] S. Orçun, I. K. Altinel, and O. Hortaçsu, "Scheduling of batch processes with operational uncertainties," *Computers & chemical engineering*, vol. 20, no. 96, pp. 2–7, 1996.
- [16] D. Bertsimas and M. Sim, "Robust discrete optimization and network flows," *Mathematical Programming*, vol. 98, no. 1-3, pp. 49–71, Sept. 2003.
- [17] S. Kullback and R. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, 1951.
- [18] G. Marsaglia, "Ratios of normal variables," *Journal of Statistical Software*, vol. 16, no. 4, 2006.
- [19] R. Ware and F. Lad, "Approximating the distribution for sums of products of normal variables," The University of Queensland, Tech. Rep. 1978, 2003.

TABLE VII

ANALYTICAL COST MINUS MONTE CARLO COST FOR 20 USER AGENTS

Initial Conditions	Mean ($\times 10^{-3}$)	Standard Deviation ($\times 10^{-3}$)
Empty	-0.40	1.31
Half full	-5.62	2.78
Full	0.40	1.31

TABLE VIII

COMPARISON ACCURACY FOR 20 USER AGENTS

Initial Conditions	Number of Comparisons	Correct
Empty	12497500	99.90%
Half full	12497500	98.65%
Full	5351411	96.84%
All	105342630	99.66%