

Design and Robust Scheduling of Nano-Satellite Swarm for Synthetic Aperture Radar Applications

Chee Kiang Pang, Akash Kumar, Cher Hiang Goh, and Cao Vinh Le
Department of Electrical and Computer Engineering
National University of Singapore
Singapore, Singapore 117583
Email: {justinpang, akash, elegch, cvle}@nus.edu.sg

Abstract—This paper presents the design and robust scheduling of nano-satellite (nanosat) swarm for synthetic aperture radar (SAR) applications. Based on power budget and bandwidth limit of nanosats, the nanosats' form factor and swarm size are chosen to ensure requirements on ground resolution and signal-to-noise ratio. An energy-efficient and robust scheduling considering stochastic failures is proposed using scenario optimization with convexification. The effectiveness of our proposed scheduling approach is verified with mathematical rigor as well as extensive simulation results on a realistic SAR application using strip and spot modes.

I. INTRODUCTION

With recent advancements in consumer electronics, the ability to operate co-orbital nano-satellites (nanosats) in swarm platforms in the same orbit to replace the more costly and bulkier satellites opens new opportunities and challenges for space industries [1]. As such, power budget and bandwidth limit of nanosats must be carefully taken into account when designing a nanosat swarm. In addition, energy efficiency and reliability are highly essential scheduling objectives. Due to restrictive on-board computational capability, given nanosat specifications and mission details, it is essential for a nanosat swarm to be properly designed and scheduled prior to launch. This is opposite to the traditional synthetic aperture radar (SAR) applications performed by large satellites, where a built-in scheduler is able to receive the jobs submitted by users and perform scheduling online [4]–[6].

To the best of our knowledge, design of a nanosat swarm has only been proposed in [2], considering radar system design and revisit times for nanosat constellation on different orbits. Dynamic scheduling has been recently proposed for cooperative satellites. In particular, Pemberton and Greenwald identified the need for dynamic scheduling of imaging satellites in the presence of uncertain changes in the environment, desired jobs, as well as the availability of resources [7]. Liao and Yang formulated a satellite scheduling problem considering stochastic weather conditions and proposed a solution using a rolling horizon approach [8]. Wang *et al.* proposed a heuristic dispatching rules for dynamic scheduling problem of Earth observing satellites to deal with uncertain arrival of new jobs [4]. A satellite mission scheduling problem was studied

This work was supported in part by Singapore MOE AcRF Tier 1 Grant R-263-000-A52-112.

involving scheduling of jobs to be performed by a satellite in which new job requests can arrive stochastically [5]. A two-phase scheduling method is also proposed for the consideration of emergency jobs in earth observing satellites scheduling [6].

To the best of our knowledge, energy-efficient and robust scheduling of nanosat swarm considering stochastic failures has not been considered in state-of-the-art literature. In addition, the scheduling approaches of existing works were commonly proposed by a dispatching rule [4] and rescheduling [8][5]. The former has no baseline schedule and jobs are dispatched online based on a predefined criterion, the latter has a baseline schedule to be revised multiple times online. These approaches require online computation and are not suitable for nanosats.

In this paper, design and robust scheduling of a swarm of homogeneous nanosats is presented for SAR applications. Based on power budget and bandwidth limit of nanosats, the nanosats' form factor and swarm size are properly chosen to ensure minimum requirements on ground resolution and signal-to-noise ratio (SNR). An energy-efficient and robust scheduling approach considering stochastic failures is then proposed using convex scenario optimization (CSO). The key idea of CSO is to find a feasible solution to balance the energy-optimal schedules from individual scenarios of nanosat failures, in which occurrence probabilities of scenarios are formulated using Weibull reliability analysis. Our proposed CSO is applied to SAR applications in Singapore using strip and spot modes. The effectiveness of our proposed CSO is verified with mathematical rigour as well as extensive simulation results.

II. DESIGN OF NANOSAT SWARM BASED ON POWER BUDGET AND BANDWIDTH LIMIT

In this section, design of a nanosat swarm for SAR applications is presented, which includes the following steps: provision of specifications of nanosats, provision of mission details, and proposal of nanosats' form factor and swarm size.

A. Specifications of Nanosats

Nanosats are widely defined as artificial satellites with a wet mass between 1 and 10 kg (2.2 and 22 lb). The most common platform to build a nanosat is cubesat, which measures the

satellites in units of $10 \times 10 \times 10$ cm with a wet mass between 1 and 2 kg per cube.

Power supply of nanosats is provided by rechargeable battery packs, which are tailored to fit in cubesat platforms. 10% of space of nanosats is assumed to carry battery packs. For example, lithium polymer battery is one of the most popular batteries used for cubesats with typical energy density of 1.08 MJ/L. The power which can be supplied safely by battery packs is restricted due to thermal issue in removing the excess heat, *e.g.*, maximum dissipating rate was reported as 15 W for a 2U cubesat. In addition, it is assumed that battery charge/discharge efficiency is 100% and self-discharge rate is 0%.

Recharging of battery packs is commonly done using solar panels. Different materials can be used for solar panels. One approach is to use gallium arsenide thin film cells with energy density of 40 W/m² using multiple junction cells at high solar concentrations [10].

Let p^{\max} be the maximum power which can be supplied to a nanosat. The power budget of nanosats can be planned such that 60% of energy capacity is used for SAR operations and 40% is used for other processes. SAR operations and recharging of battery packs spend up to 5% and 45% of orbital cycle under sunlight, respectively, while other processes are carried out during 50% of orbital cycle in darkness.

Let f be the bandwidth of SAR. One of the common antennas used in nanosats is the patch antenna. The satellite must be facing the Earth during transmission. Two or more antennas may be required, which can take up a significant amount of space in the satellite's surface. They also have limited bandwidth, compared to other types of antennas. Typical value for antenna bandwidth of 3U cubesats is $f = 30$ MHz and is assumed to be scalable by form factors [2].

B. Mission Details

SAR jobs are commonly excuted under three modes, namely, strip, spot, and scan. Strip mode maintains a fixed pointing direction of the radar antenna broadside to the platform track. A strip map is formed in width by the swath of the SAR and follows the length contour of nanosats' ground track. On the contrary, spot mode is used for obtaining high resolution signals by steering the radar beam to keep the target within the beam for a longer time and thus form a longer synthetic aperture. Scan mode is omitted as it is too energy consuming for implementation on nanosats.

For SAR applications in countries near Equator like Singapore, it is worth noting that they are usually covered by the equatorial cloud bands. From month to month, a band of clouds girdles the Equator. This band of persistent clouds is called the intertropical convergence zone, the place where the easterly trade winds in the northern and southern hemispheres meet. Orbits with long revisit time have difficulty to see the ground during their pass, since large percentage of the ground is covered by cloud. As such, an equatorial low earth orbit (LEO) is selected for a short revisit time of 90 minutes.

Mission details also provide user requirements on ground resolution and SNR of SAR jobs. Ground resolution is defined as the minimum distance on the ground at which two object points can be imaged separately, while SNR is defined as the ratio between radar transmitted power versus antenna thermal noise.

C. Nanosat Swarm Design

Let us denote by $R = \{r_j\}_{j=1}^K$ and $V = \{v_i\}_{i=1}^Q$ the nanosat swarm and the set of mission jobs, respectively. The required bandwidths of job v_i at a desired ground resolution under strip mode and spot mode are

$$f_i^{\min} = \frac{c}{2\psi_i}, \quad f_i^{\min} = \frac{c}{2\psi_i \sin \alpha_i}, \quad (1)$$

respectively, where ψ_i denotes desired ground resolution of v_i and c is the speed of light. α_i denotes steering angle of radar beam of v_i .

The required power of of v_i at a desired ground resolution and SNR is calculated by

$$p_i^{\min} = \frac{\text{SNR}_i \times (4\pi)^3 H^2 k F T_a \psi_i}{G^2 \lambda^2 \sigma}, \quad (2)$$

where SNR_i represents the desired SNR of v_i . k , T_a , and H denote Boltzmann constant, antenna noise temperature, and orbital altitude, respectively. G is antenna gain and λ expresses radar beam wavelength. σ and F denote radar target cross section and amplifier stage noise figure.

We assume additivity in bandwidth and power requirements when using a nanosat swarm, *i.e.*,

$$\sum_{r_j \in R} p_{ij} \geq p_i^{\min}, \quad \sum_{r_j \in R} f_j \geq f_i^{\min}, \quad (3)$$

where p_{ij} and f_j denote power of r_j to perform v_i and bandwidth of r_j , respectively. p_{ij} and f_j can be obtained based on nanosat specifications.

As the nanosat swarm must satisfy f_i^{\min} and p_i^{\min} for all $v_i \in V$, let us define the following:

$$p_{\max}^{\min} \triangleq \max \{p_i^{\min}\}_{i=1}^Q, \quad f_{\max}^{\min} \triangleq \max \{f_i^{\min}\}_{i=1}^Q. \quad (4)$$

The minimum swarm size K^{\min} is computed for each form factor by

$$K^{\min} = \arg \min_k (kp^{\max} \geq p_{\max}^{\min} \wedge kf \geq f_{\max}^{\min}), \quad (5)$$

where \wedge denotes logical conjunction.

III. ENERGY-EFFICIENT AND ROBUST SCHEDULING CONSIDERING STOCHASTIC FAILURES

In this section, standard notations are used. \mathbb{R} , \mathbb{R}_+ , and \mathbb{R}_+ are used to denote set of real numbers, set of nonnegative real numbers, and set of strictly positive real numbers, respectively. \mathbb{Z} is used to denote set of integer numbers. \mathbb{N} and \mathbb{N}^* denote the set of natural numbers and the set of non-zero natural numbers, respectively. $[a, b]$ is used to denote a range or interval defined by $\{a \leq x \leq b\}$.

TABLE I
NANOSAT SWARM DESIGN BASED ON BANDWIDTH AND POWER
REQUIREMENTS OF MISSION JOBS

Max bandwidth (f_{\max}^{\min}) MHz	SNR dB	Max power (p_{\max}^{\min}) W	Min swarm size (K^{\min})				
			1U	2U	3U	4U	5U
5	40	101.54	14	7	5	4	3
5	45	321.08	43	22	15	11	9
10	40	203.07	28	14	10	7	6
15	40	304.6	41	21	14	11	9
20	40	406.14	55	28	19	14	11
25	40	507.67	68	34	23	17	14
30	35	192.65	26	13	9	7	6
30	40	609.2	82	41	28	21	17
35	35	224.76	30	15	10	8	6
40	35	256.87	35	18	12	9	7
45	35	288.97	39	20	13	10	8
50	35	321.08	43	22	15	11	9

A. Problem Formulation

Given f_{\max}^{\min} and p_{\max}^{\min} , the mission manager first refers to Table I for references on the nanosats' form factor and K^{\min} . Depending on the number of failures that the mission manager wants the nanosat swarm to tolerate, the number of nanosats K is determined by

$$K = K^{\min} + N, \quad (6)$$

where N is the maximum number of failures under consideration. In such a case, new orbit insertion must be carried out before the occurrence of $(N + 1)^{\text{th}}$ failure, else the mission will fail.

The mission manager's next task is to plan the power levels for nanosats according to N . While being designed homogeneously, nanosats are subjected to a practical range of variation in power consumption, denoted by $[-\varepsilon\%, +\varepsilon\%]$, because of the lumped and undecoupled errors throughout many production phases.

p_i and p^n are used denote the nominal powers which the mission manager sets for v_i in the case of zero failure and for all jobs in the case of n failures, respectively. The actual realizations of p_i and p^n for r_j are p_{ij} and p_j^n , respectively, with variations defined by

$$p_{ij} \in [p_i(1 - \varepsilon\%), p_i(1 + \varepsilon\%)], \quad (7)$$

$$p_j^n \in [p^n(1 - \varepsilon\%), p^n(1 + \varepsilon\%)], \quad (8)$$

where $\{p_{ij}, p_j^n\} \in \mathbb{R}_+$, $n \in \mathbb{N}^+$ and $n \leq N$.

Let $d_i \in \mathbb{R}_+$ denote the processing time of v_i and $t^e \in \mathbb{R}_+$ denote the end time of mission. Based on mission details, d_i and t^e are known prior to launch. Decision variables are denoted by x_{ij} . $x_{ij} = 1$ if r_j is assigned to v_i , else $x_{ij} = 0$.

B. Weibull Reliability Analysis

In general, reliability of a device can be defined by the probability to execute its intended functions [12]. Reliability of satellite components and systems are commonly modelled by Weibull distribution. The reliability function of r_j is then

given by

$$R_j(t) = e^{-\left(\frac{t}{\alpha_j}\right)^\beta}, \quad (9)$$

for $t \geq 0$, and $R_j(t) = 1$ for $t < 0$, where $\beta \in \mathbb{R}_+$ is the shape parameter and $\alpha_j \in \mathbb{R}_+$ is the scale parameter.

In state-of-the-art literature, several authors have recently considered costs of usage of machines, *i.e.*, a machine assigned more workload degrades faster and incurs more costs. Analogously, a dynamic Weibull reliability of r_j is proposed herein such that the costs of usage of nanosats are included. The usage of r_j is

$$s_j = \sum_{v_i \in V} x_{ij}, \quad (10)$$

and α is modified by $\alpha_j(s_j)$, where $\alpha_j(\cdot)$ is a strictly monotonically decreasing function such that $\alpha_j : \mathbb{N} \rightarrow \mathbb{R}_+$. For simplicity, $\alpha_j(s_j)$ is chosen by

$$\alpha_j(s_j) = \frac{\delta_1}{s_j + \delta_2}, \quad (11)$$

where $\{\delta_1, \delta_2\} \in \mathbb{R}_+$. β , δ_1 , δ_2 can be determined based on offline testing or commercial failure database of satellites such as SpaceTraks.

C. Scenario Optimization

A set of scenarios is constructed, each of which represents a distinctive combination of failures. Let ω denote a scenario and $\Omega_{R,n}$ denotes a superset of n -combination sets of R . For example, $\Omega_{R,2} = \{\{r_1, r_2\}, \{r_1, r_3\}, \{r_2, r_3\}\}$ for $R = \{r_1, r_2, r_3\}$. Let $\omega \subset \Omega$, where Ω denotes the set of possible failure scenarios, then clearly

$$\Omega = \bigcup_{n=1}^N \Omega_{R,n} \cup \omega_0, \quad (12)$$

where ω_0 denote the scenario of zero failure. Obviously, $N \leq K$.

Let us introduce an additional decision variable $\gamma \in \mathbb{R}$ and $0 \leq \gamma \leq 1$ to denote the worst reliability of all nanosats during the mission. Let π_ω denote the occurrence probability of scenario ω , then π_ω is

$$\pi_\omega = \begin{cases} \gamma^{K-n}(1 - \gamma)^n & \forall \omega \subset \Omega_{R,n}, \\ \gamma^K & \omega = \omega_0. \end{cases} \quad (13)$$

Thus, our proposed SO problem is

$$\text{Min } g(x_{ij}; \gamma) = \Xi + \Theta \quad (14)$$

$$\text{s.t.}, R_j(t^e; s_j) > \gamma, \forall r_j \in R, \quad (15)$$

$$\sum_{r_j \in R \setminus \omega} x_{ij} f_j \geq f_i^{\min}, \forall v_i \in V, \forall \omega \subset \Omega, \quad (16)$$

$$\sum_{r_j \in R} x_{ij} p_{ij} \geq p_i^{\min}, \forall v_i \in V, \quad (17)$$

$$\sum_{r_j \in R \setminus \omega} x_{ij} p_j^n \geq p_i^{\min}, \forall v_i \in V, \forall \omega \subset \Omega_{R,n}, \forall n, \quad (18)$$

$$0 \leq \gamma \leq 1, x_{ij} \in \{0, 1\}. \quad (19)$$

$g(x_{ij}; \gamma)$ consists of two costs, namely, the energy cost of scenario ω_0

$$\Xi = \sum_{v_i \in V} \sum_{r_j \in R} \pi_{\omega_0} x_{ij} p_{ij} d_i, \quad (20)$$

and the energy cost of other scenarios

$$\Theta = \sum_{n=1}^N \sum_{\omega \subset \Omega_{R,n}} \sum_{v_i \in V} \sum_{r_j \in R \setminus \omega} \pi_{\omega} x_{ij} p_j^n d_i. \quad (21)$$

(15) indicates worst reliability of all nanosats throughout the mission. (16) specifies the minimum bandwidth requirements, while (17) and (18) indicate the minimum power requirements. Finally, the bounds of decision variables are given in (19).

D. Convexity and Convexification

SO problem includes two nonlinear terms, namely, $g(x_{ij}; \gamma)$ and $R_j(t^e; s_j)$. While it is straightforward to show that $R_j(t^e; s_j)$ is convex, the convexity of $g(x_{ij}; \gamma)$ requires further analysis. We have the following result.

Theorem 1: $g(x_{ij}; \gamma)$ is non-convex on \mathbb{R}_+^{Q+1} for any $\{K, Q, N\} \in \mathbb{N}$.

To convexify $g(x_{ij}; \gamma)$, the following results are needed [11]:

Lemma 1: $\rho e^{r_1 x_1 + r_2 x_2 + \dots + r_n x_n}$ is convex on \mathbb{R}^n if $\rho > 0$ and $\mathbf{r} \in \mathbb{R}^n$.

Lemma 2: $\rho x^p y^q$ is convex on \mathbb{R}_+^2 if $\rho < 0$ and $p + q \leq 1$.

$f(x_{ij}; \gamma)$ is non-convex as it contains some products of decision variables in the form $\rho \gamma^n x$ (the subscripts of x_{ij} is dropped for brevity), where $\rho \in \mathbb{R}$. First, γ can be approximated by a set of discrete values $\{\gamma_k\}_{k=1}^M$, $\gamma_k \in \mathbb{R}$ and $0 \leq \gamma_k \leq 1$, such that

$$\gamma = \sum_{k=1}^M \xi_k \gamma_k \text{ and } \sum_{k=1}^M \xi_k = 1. \quad (22)$$

x can be represented by $\{x_1, x_2\} = \{0, 1\}$, such that

$$x = \zeta_1 x_1 + \zeta_2 x_2 \text{ and } \zeta_1 + \zeta_2 = 1. \quad (23)$$

If $\rho > 0$, then $\rho \gamma^n x$ is convexified by introducing additional variables X and Γ such that

$$x = e^X \text{ and } \gamma = e^\Gamma, \quad (24)$$

then the following equivalent convex representation is obtained

$$\text{Min } \rho \gamma^n x \quad (25)$$

$$\Leftrightarrow \text{Min } \rho e^{n\Gamma + X} \quad (26)$$

$$\text{s.t.}, \Gamma = \sum_{k=1}^M \xi_k \ln(\gamma_k), \quad (27)$$

$$\sum_{k=1}^M \xi_k = 1, \quad (28)$$

$$X = \zeta_1 \ln(x_1) + \zeta_2 \ln(x_2), \quad (29)$$

$$\zeta_1 + \zeta_2 = 1, \quad (30)$$

$$\{\xi_k, \zeta_k\} \in \{0, 1\}. \quad (31)$$

If $\rho < 0$, then $\rho \gamma^n x$ is convexified by introducing additional variables X and Γ such that

$$x = X^{\frac{1}{1+n}} \text{ and } \gamma = \Gamma^{\frac{1}{1+n}}, \quad (32)$$

then the following equivalent convex representation is obtained

$$\text{Min } \rho \gamma^n x \quad (33)$$

$$\Leftrightarrow \text{Min } \rho \Gamma^{\frac{n}{1+n}} X^{\frac{1}{1+n}} \quad (34)$$

$$\text{s.t.}, \Gamma = \sum_{k=1}^M \xi_k \gamma_k^{1+n}, \quad (35)$$

$$\sum_{k=1}^M \xi_k = 1, \quad (36)$$

$$X = \zeta_1 x_1^{1+n} + \zeta_2 x_2^{1+n}, \quad (37)$$

$$\zeta_1 + \zeta_2 = 1, \quad (38)$$

$$\{\xi_k, \zeta_k\} \in \{0, 1\}. \quad (39)$$

CSO problem can be solved by convex optimization which is widely available in commercial solvers. The cutting-plane algorithm is used herein.

IV. SIMULATION RESULTS

Simulations are carried out to verify the feasibility of our proposed CSO. For evaluation, CSO are compared to two general rescheduling algorithms, namely, partial rescheduling (PR) and complete or full rescheduling (CR) [13]. The basic idea of CR is to prepare an baseline schedule without anticipation of failures. Once a failure happens, the existing schedule will be revised online. While having the same baseline schedule, PR revises schedules online to minimize schedule instability instead of scheduling objectives of baseline schedule.

A. Singapore SAR Application

A SAR application in Singapore is selected for simulations. The mission is to taking M consecutive SAR signals of three specific locations using both strip and spot modes. In particular, the set of mission jobs includes taking M strip signals from east to west of Singapore, M spot signals of Kent Ridge campus, National University of Singapore (NUS), and M spot signals of Singapore Changi international airport.

As Singapore locates just 1° north of the Equator which is usually covered by the equatorial cloud bands, an Equatorial LEO is chosen with $H = 700$ km, inclination angle $\approx 0^\circ$, and revisit time ≈ 90 min. Based on revisit time and areas of target locations, processing times of jobs can be calculated by $d_i = 9, 27, 29$ s for $v_i \in V_1, V_2, V_3$, respectively.

$p_i^{\min} = 30$ MHz $\forall v_i \in V$. $\text{SNR}_i = 30, 35$ dB for $v_i \in V_2 \cup V_3$ and $v_i \in V_1$, respectively. As such, $p_{\max}^{\min} = 30$ and $p_{\max}^{\max} = 192.65$ accordingly. Based to Table I, the mission manager has different options to build the nanosat swarm. Let the nanosats' form factor be chosen as 3U, as such $K^{\min} = 9$. Let $N \in [1, 5]$ and $\varepsilon \in \{\pm 7.5, \pm 2.5\}$. Table II details the power levels of nanosats. p_{ij} and p_j^n are generated using uniform distribution. With regard to Weibull reliability, $\beta = 1.5$, $\delta_1 = 2.06 \times 10^6$, and $\delta_2 = 100$.

TABLE II
POWER LEVELS OF NANOSATS FOR SIMULATION CASES

N	K	p_i^+	p_i^{++}	p^1	p^2	p^3	p^4	p^5
1	10	6.09	19.26	21.40	-	-	-	-
2	11	5.53	17.51	19.26	21.40	-	-	-
3	12	5.07	16.05	17.51	19.26	21.40	-	-
4	13	4.68	14.81	16.05	17.51	19.26	21.40	-
5	14	4.35	13.76	14.81	16.05	17.51	19.26	21.40

⁺ $v_i \in V_2 \cup V_3$

⁺⁺ $v_i \in V_1$

B. Performance Metrics

The performance metrics used to evaluate CSO, CR, and PR are mean total energy consumption (MTEC), mean time to mission failure (MTMF), and mean time between failure (MTBF). The temporal instants of failures are obtained by sampling $R_j(t; s_j)$.

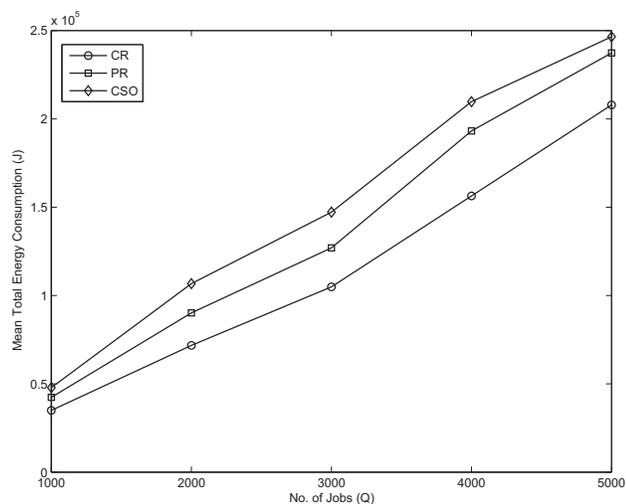


Fig. 1. MTEC at $\epsilon = 7.5$.

MTEC is computed by assuming that all scheduling algorithms are able to finish the mission, *i.e.*, only N failures occur. Q is varied by $Q \in [1000, 5000]$. Missions containing 1000 and 5000 jobs last for 125 and 625 days, respectively. On the contrary, $Q = 5000$ is fixed when computing MTMF and MTBF so that the mission is sufficient long for $N + 1$ failures to occur. MTMF is defined as the mean time from stating of the mission to the $(N + 1)^{\text{th}}$ failure instant. MTBF is the mean time from n^{th} to $(n + 1)^{\text{th}}$ failure instants, where $n \leq N - 1$.

C. Results and Discussions

This computational test is carried on MATLAB. For each simulation case, three performance metrics are obtained after 100 test runs.

Figs. 1 and 2 illustrate the performance of all algorithms in terms of MTEC. It is observed that MTEC increases for all algorithm with the increase of Q . For all cases, CSO achieves comparable performance with PR, while CR yields the lowest MTEC. This result can be attributed to the fact that CR scarifies

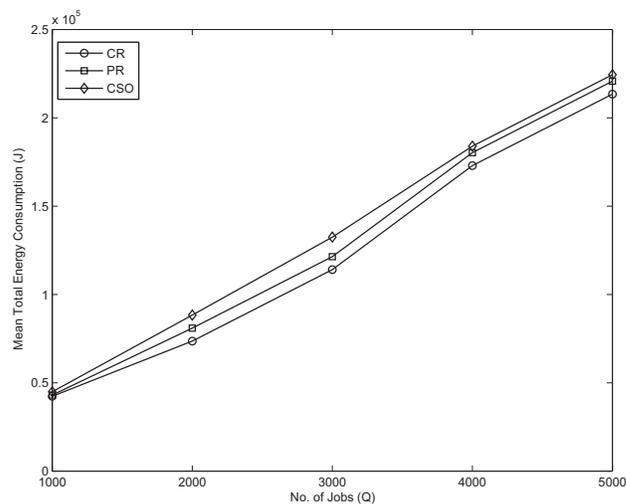


Fig. 2. MTEC at $\epsilon = 2.5$.

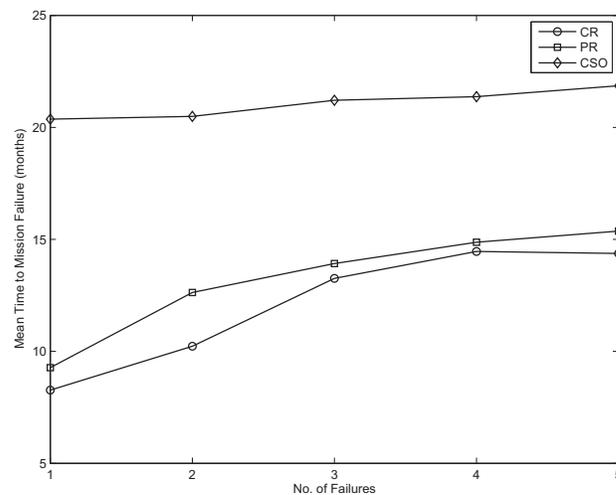


Fig. 3. MTMF at $\epsilon = 7.5$.

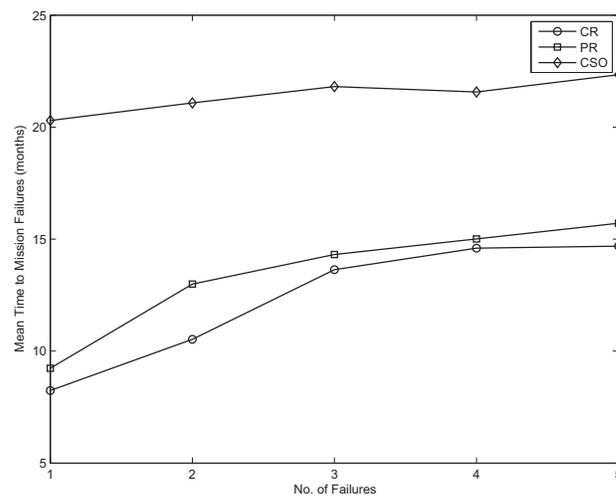


Fig. 4. MTMF at $\epsilon = 2.5$.

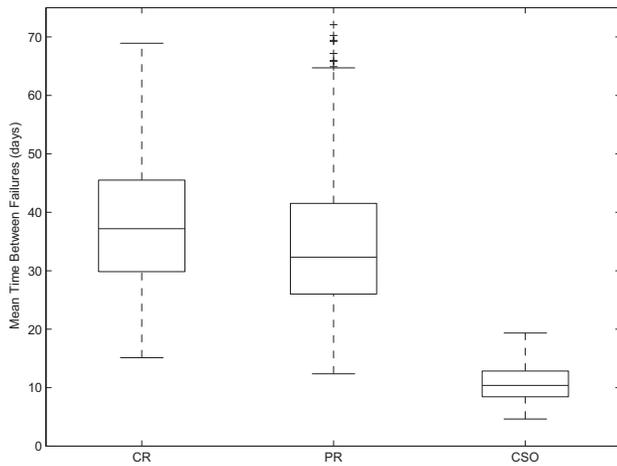


Fig. 5. MTBF at $\varepsilon = 7.5$.

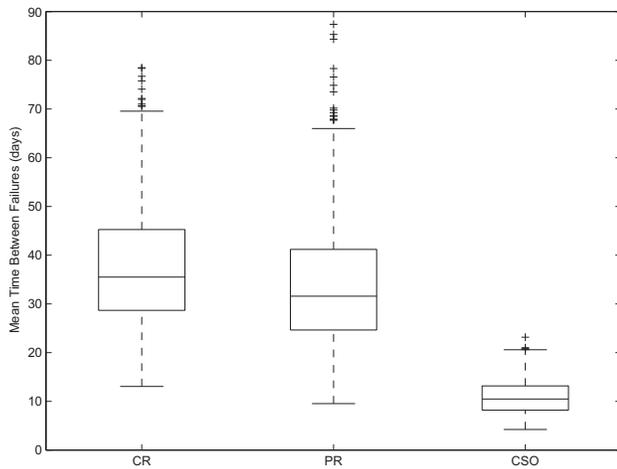


Fig. 6. MTBF at $\varepsilon = 2.5$.

both computational effort and schedule stability for MTEC. It also can be seen that the gap in MTEC between CSO and CR is reduced when ε gets smaller. As such, the performance of CSO in terms of MTEC can be significantly improved by careful design and manufacture of nanosats.

Figs. 3 and 4 show the performance of all algorithms in terms of MTMF. MTMF is an important measure mission reliability which basically specifies a deadline for a new orbit insertion to ensure the mission to be completed. It can be seen that CSO yields significantly longer MTMF as compared to CR and PR for all values of ε . The reason is that CR and PR tend to overuse and therefore degrade the most energy-efficient nanosats first. On the contrary, CSO balances utilization among nanosats to guarantee that all nanosats operate in a safe region above γ reliability threshold. The mission is hence scheduled such that all nanosats degrade gradually and simultaneously.

In addition to MTMF, MTBF is also of importance for the mission of a nanosat swarm. A shorter MTBF allows multiple nanosats to be replaced at the same time with fewer

orbit insertions, which cuts costs. Figs. 5 and 6 give the performance of all algorithms in terms of MTBF, where CSO yields significantly shorter MTMF as compared to CR and PR for all values of ε . It is worth noting that all failures using CSO occur toward the end of mission, while failures using CR and PR tend to occur in the middle of mission.

V. CONCLUSION

In this paper, design and robust scheduling of a nanosat swarm for SAR applications were studied. Design of nanosat swarm was first proposed based on power budget and bandwidth limit to ensure requirements on ground resolution and SNR. Energy-efficient and robust scheduling considering stochastic failures was proposed using CSO. The effectiveness of our proposed CSO was verified with mathematical rigor as well as a realistic SAR application in Singapore using both strip and spot modes. Our extensive simulation results showed that CSO achieved acceptable MTEC but while yielded significant improvements in MTMF and MTBF as compared to related works in current literature. Our future works include formation control and path-planning for a nanosat swarm using multi-agent and graph theory.

REFERENCES

- [1] C. J. M. Verhoeven, M. J. Bentum, G. L. E. Monna, J. Rotteveel, and J. Guo, "On the origin of satellite swarms," *Acta Astronautica*, vol. 68, nos. 7–8, pp. 1392–1395, April–May 2011.
- [2] S. Engelen, M. van den Oever, P. Mahapatra, P. Sundaramoorthy, E. Gill, R. Meijer and C. Verhoeven, "NanoSAR – Case study of synthetic aperture radar for nano-satellites," in *Proceedings of the 63rd International Astronautical Congress*, pp. 1–6, Naples, Italy, October 1–5, 2012.
- [3] D. Del Corso, C. Passerone, L. Reyneri, C. Sansoe, S. Speretta, and M. Tranchero, "Design of a university nano-satellite: The PiCPoT case," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 3, pp. 1985–2007, July 2011.
- [4] J. Wang, J. Li, and Y. Tan, "Study on heuristic algorithm for dynamic scheduling problem of earth observing satellites," in *Proceedings of the 8th IEEE International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, pp. 9–14, Qingdao, China, July 30–August 1, 2007.
- [5] B. Sun, W. Wang, X. Xie, and Q. Qin, "Satellite mission scheduling based on genetic algorithm," *Kybernetes*, vol. 39, no. 8, pp. 1255–1261, 2010.
- [6] G. Wu, J. Liu, M. Ma, D. Qiu, "A two-phase scheduling method with the consideration of task clustering for earth observing satellites," *Computers & Operations Research*, vol. 40, no. 7, pp. 1884–1894, July 2013.
- [7] J. C. Pemberton and L. G. Greenwald, "On the need for dynamic scheduling of imaging satellites," *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, vol. 34, no. 1, pp. 165–171, 2002.
- [8] D. Y. Liao and Y. T. Yang, "Imaging order scheduling of an earth observation satellite" *IEEE Transactions on Systems, Man and Cybernetics—Part C: Applications and Reviews*, vol. 37, no. 5, September 2007.
- [9] R. Dembo, "Scenario optimisation," *Annals of Operations Research*, vol. 30, no. 1, pp. 63–80, 1991.
- [10] GomSpace. *NanoPower P110 Series Solar Panels* [Online]. Available <http://gomspace.com/documents/GS-DS-P110-1.0.pdf>.
- [11] A. W. Roberts and D. E. Varberg. *Convex functions*. New York City, NY, US: Academic Press, 1973.
- [12] C. K. Pang, F. L. Lewis, T. H. Lee, and Z. Y. Dong, *Intelligent Diagnosis and Prognosis of Industrial Networked Systems*. Boca Raton, FL, USA: CRC Press, Taylor and Francis Group, 2011.
- [13] J. Kuster, *Providing Decision Support in the Operative Management of Process Disruptions*. Berlin, Germany: GITO-Verlag, 2008.