# How Many Bits Does a Swarm Need? Adaptive-Precision Task Allocation for Resource-Constrained Robots

Eashwar Sathyamurthy

***Abstract*— Multi-robot systems must decide which robot should perform which task. These decisions are typically based on numerical utility values transmitted by robots to a coordinator, and sending each utility using 32 bits precision can be costly for miniature and resource-constrained robots. We study how much numerical precision is actually needed to make good task-allocation decisions. We propose Margin-Triggered Bit Refinement (MTBR), which starts with low-precision utilities and requests additional bits only when more information is needed to distinguish between competing assignments. Across the tested swarm sizes, four-bit utilities preserve more than 98.8% of the 32-bit reference welfare while reducing the utility payload by 87.5%. Although the exact robot–task pairing may change, the resulting assignment achieves nearly the same overall value. MTBR further improves the welfare–utility-payload trade-off by directing additional precision only to utilities involved in ambiguous competing assignments and consistently outperforms matched-payload refinement baselines. By concentrating additional precision on the decisions that need it, MTBR provides a resource-aware task-allocation strategy for miniature robot swarms operating under tight communication, computation, and energy constraints.**

***Keywords* —** Multi-robot task allocation, miniature robots, resource-constrained autonomy, communication-efficient coordination, adaptive precision, low-power robotics.

## I. Introduction

Multi-robot task allocation (MRTA) is the problem of assigning robots to tasks based on factors such as task value, travel cost, and robot capability. These factors are typically combined into real-valued utilities, and the goal is to maximize their total value [1], [2]. Distributed auctions and consensus methods reduce reliance on a centralized coordinator [3], [4], while communication-aware MRTA studies when and where robots should exchange information to coordinate their task assignments [5]. More recent work considers whether agents should communicate when their information is stale or asymmetric [6]. We ask a different question: if each robot already has a local utility value, how many bits of that value must it reveal to the coordinator to resolve competing task assignments globally? Building on adaptive quantization in distributed optimization [7], we introduce a method that progressively refines utility values only when more precision is needed to resolve task assignments in multi-robot systems.

This focus on precision matters because the numerical representation of a utility is not free on very small robots. In software, a utility may be stored using a 32- or 64-bit IEEE 754 floating-point representation [8], whereas MRTA formulations typically treat it simply as a real number. This distinction can matter in miniature swarms, where communication and onboard resources are limited. Kilobots, for example, use an 8-MHz microcontroller and short-range infrared communication [10], [11], while insect-scale RoboBees face severe constraints on mass, power, and onboard computation [9], [12]. If many robots repeatedly send utilities to a coordinator, transmitting more precision than needed adds communication without necessarily changing the resulting assignment. In our setting, the coordinator is simply an external node that receives utility values from the robots and uses them to determine the task assignment; it does not need to be physically onboard any miniature robot.

Reducing the number of transmitted bits requires representing utility values at lower precision, which introduces quantization error. The important question is therefore not whether quantization changes the assignment, but whether that change meaningfully affects its quality.The coordinator evaluates each assignment by its welfare, defined as the sum of utilities of the selected robot–task pairs. When several assignments have nearly equal welfare, quantization may select a different assignment with little effect on mission performance. In such cases, spending more bits to reproduce the full-precision assignment provides little benefit. More precision is valuable only when uncertainty in the utilities could lead to a meaningfully worse assignment

We make three contributions. First, we show how communicated precision affects both welfare loss and whether the same assignment as the full-precision solution is recovered. We derive a worst-case bound on the effect of utility errors and a margin-based condition for exact recovery. Second, we introduce Margin-Triggered Bit Refinement (MTBR), which concentrates residual bits on the edges responsible for ambiguity. Third, we evaluate fixed and adaptive precision on synthetic spatial MRTA instances, measuring welfare, exact recovery, and communication. The study is a communication-design analysis rather than a hardware demonstration; its relevance to insect-scale autonomy is that it identifies which numerical information a constrained coordination layer actually needs.

## II. Problem Formulation and Main Insight

Consider $N$ robots and $N$ tasks. Let $u_{ij} \in [u_{\min}, u_{\max}]$ denote robot $i$'s utility for task $j$, with range $W = u_{\max} - u_{\min}$. A one-to-one assignment is represented by a permutation matrix $\mathbf{X} \in \mathcal{X}$, where $x_{ij} = 1$ if robot $i$ is assigned

Eashwar Sathyamurthy is an independent researcher. He received the Ph.D. degree in Mechanical Engineering from the University of Maryland, College Park, MD, USA, in 2026. Email: eashwar.sathyamurthy@gmail.com.

task $j$. Its welfare, or total utility, is

$$J(\mathbf{X}) = \langle \mathbf{U}, \mathbf{X} \rangle = \sum_{i=1}^{N} \sum_{j=1}^{N} u_{ij} x_{ij}. \tag{1}$$

Using the original utilities, the optimal assignment is $\mathbf{X}^* = \arg\max_{\mathbf{X} \in \mathcal{X}} J(\mathbf{X})$. To model finite communication, each utility is encoded using $B$ bits, allowing it to be represented by one of $2^B$ evenly spaced values between $u_{\min}$ and $u_{\max}$. The quantizer $Q_B$ rounds the original utility to the nearest of these values. Here, $B$ denotes the communication bit budget of a uniform quantizer, not an IEEE 754 floating-point format. For example, if $B = 2$ and utilities $u_{ij}$ lie in $[0, 1]$, the four possible values are $0$, $1/3$, $2/3$, and $1$. Since these $2^B$ values include both endpoints, they create $2^B - 1$ equal gaps across the range $W = u_{\max} - u_{\min}$. Each gap therefore has width $W/(2^B - 1)$. Because a utility is rounded to the nearest level, the largest possible error occurs halfway between two adjacent levels and is half this spacing:

$$\epsilon_B = \max_u |Q_B(u) - u| = \frac{W}{2(2^B - 1)}. \tag{2}$$

The coordinator uses the quantized utility matrix $Q_B(\mathbf{U})$ obtained from all robots to compute

$$\hat{\mathbf{X}}_B = \arg\max_{\mathbf{X} \in \mathcal{X}} \langle Q_B(\mathbf{U}), \mathbf{X} \rangle, \tag{3}$$

where $\hat{\mathbf{X}}_B$ is the assignment selected using $B$-bit utility values. We evaluate this assignment in two ways. First, its welfare loss is measured using the original, unquantized utilities, with normalized welfare regret defined as

$$R(B) = J(\mathbf{X}^*) - J(\hat{\mathbf{X}}_B), \qquad R_{\%}(B) = 100\, \frac{R(B)}{J(\mathbf{X}^*)}. \tag{4}$$

Second, exact recovery checks whether the quantized solution reproduces the full-precision assignment, i.e., $\hat{\mathbf{X}}_B = \mathbf{X}^*$. We report exact recovery separately because a different assignment may still achieve nearly the same welfare.

For empirical evaluation, robots and tasks are placed in a two-dimensional workspace, with utilities combining task priority and normalized Euclidean travel cost. Utilities are normalized to a shared a priori range, and full $N \times N$ utility matrices are used unless stated otherwise. The same linear assignment solver is used for both the full-precision and quantized problems.

## III. Precision Bounds and Adaptive Refinement

We now relate the $B$-bit utility representation to the two measures above: welfare loss and exact assignment recovery. Let $\mathbf{E}_B = Q_B(\mathbf{U}) - \mathbf{U}$ denote the quantization error, so every utility changes by at most $\epsilon_B$ and $|\mathbf{E}_B|_\infty \le \epsilon_B$. If $k$ robots receive different tasks in $\hat{\mathbf{X}}_B$ and $\mathbf{X}^*$, then $k = \frac{1}{2}|\hat{\mathbf{X}}_B - \mathbf{X}^*|_1$ because each changed robot contributes one removed assignment and one added assignment.

**Proposition 1 (rate–regret envelope).** If each quantized utility differs from its original value by at most $\epsilon_B$, then

$$R(B) \le 2k\epsilon_B \le 2N\epsilon_B = \frac{NW}{2^B - 1}. \tag{5}$$

Each changed robot replaces one assignment edge with another, and each edge can shift by at most $\epsilon_B$. Thus the welfare difference can change by at most $2\epsilon_B$ per changed robot, giving $R(B) \le 2k\epsilon_B$. Since $k \le N$, the remaining bounds follow directly. This is a worst-case bound; the actual regret can be much smaller.

We next ask when quantization preserves the full-precision assignment itself. For any competing assignment $\mathbf{X}$, define its welfare gap from the optimum as $\Delta(\mathbf{X}) = J(\mathbf{X}^*) - J(\mathbf{X})$, and let $k = \frac{1}{2}|\mathbf{X}^* - \mathbf{X}|_1$ be the number of robots assigned differently.

**Proposition 2 (global margin certificate).** The unique optimum $\mathbf{X}^*$ remains uniquely optimal under every entrywise perturbation satisfying $|\mathbf{E}|_\infty \le \epsilon_B$ if and only if

$$\Delta(\mathbf{X}) > 2k\epsilon_B, \quad \forall \mathbf{X} \neq \mathbf{X}^*. \tag{6}$$

Here, $\Delta(\mathbf{X})$ is how much better $\mathbf{X}^*$ is than a competing assignment $\mathbf{X}$, while $2k\epsilon_B$ is the largest amount bounded utility errors could reduce that advantage. Thus, exact recovery is guaranteed when every competing assignment remains worse even under the worst allowed perturbation.

For a fixed deterministic quantizer, (6) is therefore a sufficient certificate. Let $\Delta = \min_{\mathbf{X} \neq \mathbf{X}^*} \Delta(\mathbf{X})$ be the smallest welfare gap. Since $k \le N$, the conservative condition $\Delta > 2N\epsilon_B$ guarantees the certificate for every competitor. Substituting $\epsilon_B = W/[2(2^B - 1)]$ gives

$$\Delta > \frac{NW}{2^B - 1},$$

which yields the sufficient bit budget

$$B_{\text{suff}} = \left\lfloor \log_2 \left( 1 + \frac{NW}{\Delta} \right) \right\rfloor + 1. \tag{7}$$

The floor-plus-one form preserves the strict inequality. Checking only two-robot swaps is not sufficient because the closest competing assignment may involve several robots. Hence, additional precision is mainly needed when globally competing assignments have small welfare gaps.

### A. Margin-Triggered Bit Refinement (MTBR)

Proposition 2 shows that more precision is needed when the best assignment and a competitor have a small welfare gap. MTBR uses this idea adaptively: robots first send coarse utility information, and additional bits are requested only when the coordinator cannot yet determine the winner. Unlike the fixed endpoint quantizer used above, MTBR uses nested dyadic intervals so that each new bit halves the remaining uncertainty. For a normalized utility $u_{ij} \in [0, 1]$,

$$\begin{aligned} I_B(u_{ij}) &= \left[ q_{ij}^{(B)} 2^{-B}, \left( q_{ij}^{(B)} + 1 \right) 2^{-B} \right], \\ q_{ij}^{(B)} &= \min \left\{ \left\lfloor 2^B u_{ij} \right\rfloor, 2^B - 1 \right\}. \end{aligned} \tag{8}$$

where $q_{ij}^{(B)} \in 0, \ldots, 2^B - 1$ is the index of the interval containing $u_{ij}$.

For example, suppose each robot is initially allowed to send only $B = 2$ bits for each robot–task utility. After normalization of utilities $u_{ij}$ to $[0, 1]$, those two bits identify one

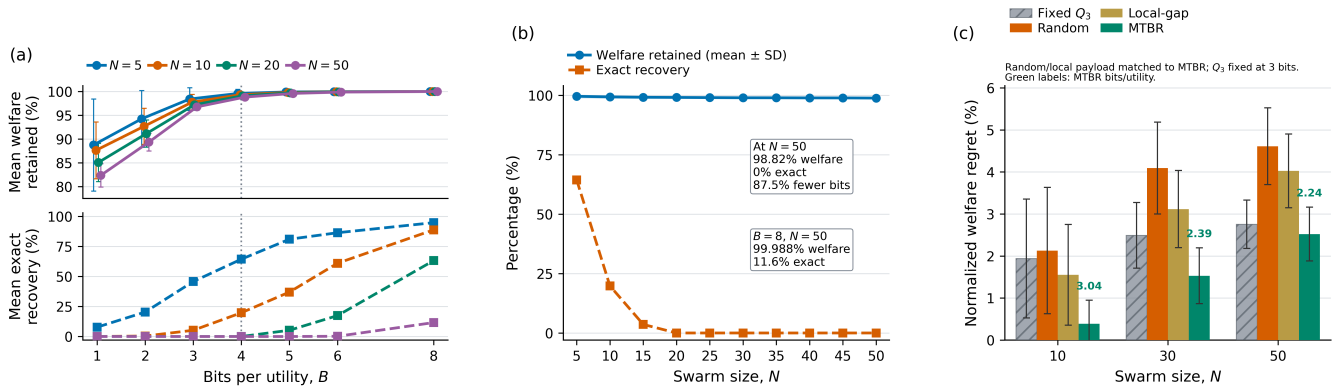



Fig. 1. **Low precision preserves task-allocation value, and selective refinement improves it further. (a)** Across representative swarm sizes, assignment welfare reaches near-reference levels with only a few bits per utility, while exact recovery of the $B = 32$ reference assignment requires much higher precision. **(b)** With four-bit utilities, welfare remains above 98.8% of the $B = 32$ reference for all tested swarm sizes while using 87.5% less utility payload. **(c)** MTBR achieves lower normalized welfare regret ($R_{\%}(B)$) than Random and Local-gap at the same utility payload, and also outperforms fixed three-bit quantization ($Q_3$). Error bars show $\pm 1$ sample SD, and green labels indicate the mean MTBR bits per utility.

of four intervals in $I_B$: $00 \to [0, 0.25]$, $01 \to [0.25, 0.50]$, $10 \to [0.50, 0.75]$, and $11 \to [0.75, 1]$. If robot 1 values task 1 at $u_{11} = 0.68$, it sends 10, so the coordinator knows only that $u_{11} \in [0.50, 0.75]$. If this uncertainty affects the assignment, MTBR requests one more bit; with $B = 3$, the message becomes 101, narrowing the same utility to $[0.625, 0.75]$.

The coordinator first uses the midpoint of each received interval as the current utility estimate and solves the MRTA problem. Suppose $u_{11} = 0.68$, $u_{12} = 0.62$, $u_{21} = 0.61$, and $u_{22} = 0.82$, the $B = 2$ intervals are $[0.50, 0.75]$, $[0.50, 0.75]$, $[0.50, 0.75]$, and $[0.75, 1]$, respectively. Their midpoints select $\hat{\mathbf{X}} = (1, 1), (2, 2)$.

The coordinator then asks whether this assignment must remain optimal for any utility values within these intervals. It tests the worst case for $\hat{\mathbf{X}}$ by setting its selected utilities to their lower bounds, $u_{11} = 0.50$ and $u_{22} = 0.75$, and all competing utilities to their upper bounds, $u_{12} = u_{21} = 0.75$. These values form the adversarial matrix $\mathbf{U}^{\text{adv}}$. Let $\mathbf{X}^{\text{adv}}$ be the best assignment other than $\hat{\mathbf{X}}$ under this matrix. The candidate is certified if

$$\sum_{(i,j)\in\hat{\mathbf{X}}} \underline{u}_{ij} - \langle \mathbf{U}^{\text{adv}}, \mathbf{X}^{\text{adv}} \rangle > 0. \tag{9}$$

Thus, the lowest possible welfare of $\hat{\mathbf{X}}$ must exceed the highest possible welfare of its strongest competitor.

Here, $\mathbf{X}^{\text{adv}} = (1, 2), (2, 1)$ has welfare $0.75+0.75 = 1.50$, while the lower-bound welfare of $\hat{\mathbf{X}}$ is $0.50+0.75 = 1.25$, so the test fails. MTBR therefore requests one additional bit for the four utilities on which the two assignments differ. At $B = 3$, their intervals narrow enough that the lower-bound welfare of $\hat{\mathbf{X}}$ becomes $0.625 + 0.75 = 1.375$, while the competing assignment can reach at most $0.625 + 0.625 = 1.25$; hence $\hat{\mathbf{X}}$ is certified. In larger MRTA problems, utilities shared by both assignments need no refinement because they contribute equally to both. The coordinator repeats this solve–test–refine cycle until certification or the bit budget is exhausted.

## IV. Experimental Evaluation

We evaluate spatial MRTA problems with an equal number of robots and tasks, $N \in \{5, 10, 15, \ldots, 50\}$. In each instance, robots and tasks are placed in a two-dimensional workspace, and each robot–task utility combines task priority with normalized Euclidean travel cost. For every swarm size, we generate 500 instances and evaluate each at fixed precisions $B \in \{1, 2, 3, 4, 5, 6, 8, 16, 32\}$. A separate set of 300 instances per swarm size is used to evaluate adaptive refinement and its comparison methods. At every precision, the coordinator solves the resulting utility matrix using the same linear-sum assignment solver [13]. The uniformly quantized $B = 32$ solution serves as the empirical reference for both welfare and assignment identity.

The comparisons separate two questions: how much precision is needed, and where additional precision should be spent. Fixed $B$-bit quantization gives every robot–task utility the same number of bits and establishes the basic precision–performance tradeoff. For adaptive refinement, Random distributes the same additional utility payload as MTBR among randomly selected entries, while Local-gap assigns it to robots whose two best local task choices are closest. MTBR instead directs those bits to utilities involved in the strongest competing global assignment. Fixed $Q_3$ provides a simple one-round, three-bit comparison near the communication level used by MTBR.

### A. Mission Value Is Preserved Before Assignment Identity

Figure 1(a) examines how communication precision affects assignment quality at representative swarm sizes. As $B$ increases, welfare rapidly approaches the the $B = 32$ reference, whereas exact recovery of the same robot–task permutation improves much more slowly. The separation is especially clear at $B = 4$: welfare is already near 99% across all shown swarm sizes, even though the probability of reproducing the exact reference assignment decreases sharply

as $N$ grows. Increasing precision continues to improve exact recovery, but provides comparatively little additional welfare.

Figure 1(b) isolates this scaling effect by fixing $B = 4$ and evaluating every swarm size from $N = 5$ to 50. Welfare retained decreases only from 99.56% to 98.82%, while exact recovery falls from 64.4% at $N = 5$ to zero from $N = 20$ onward. This is not a loss of mission quality: at $N = 50$ the 4-bit assignment still retains 98.82% of reference welfare while using 87.5% less utility payload than the $B = 32$ representation. Even at $B = 8$, $N = 50$ retains 99.988% of welfare but exactly reproduces the reference assignment in only 11.6% of instances. Together, these results suggest that uniformly increasing precision is inefficient: most mission value is already recovered at low bit depth, while additional bits are mainly needed to resolve close competing assignments.

### *B. Adaptive Refinement Targets Global Ambiguity*

Figure 1(c) asks whether adaptive refinement benefits from choosing *which* utilities receive additional bits. Random and Local-gap use exactly the same utility payload as MTBR, but place the extra bits randomly or according to local task ambiguity, respectively. At $N = 10$, all three use 3.04 bits per utility on average, yet MTBR reduces regret to $0.387\% \pm 0.564\%$, compared with $1.554\% \pm 1.198\%$ for Local-gap and $2.130\% \pm 1.500\%$ for Random. This advantage persists as the assignment grows: at $N = 50$, MTBR has $2.523\% \pm 0.640\%$ regret, versus $4.024\% \pm 0.876\%$ and $4.612\% \pm 0.912\%$ for Local-gap and Random, while using only 2.24 bits per utility on average. Thus, the improvement comes from directing precision toward globally competing assignments rather than simply transmitting more bits.

Fixed $Q_3$ provides a complementary one-round baseline. Unlike Random and Local-gap, it is not payload matched to MTBR at larger $N$: it always assigns three bits to every utility. MTBR nevertheless has lower regret at every tested size, although the gap narrows with scale; at $N = 50$, regret is 2.523% for MTBR and 2.755% for $Q_3$. This selectivity comes at the cost of interaction, since MTBR's request–reply rounds use 4.54× the modeled network traffic of $Q_3$ at $N = 10$ and 1.90× at $N = 50$. This shows that MTBR benefits from refining utilities involved in global assignment competition, while fixed low precision remains simpler when extra communication rounds are costly.

## V. Discussion and Conclusion

The main result is that high numerical precision is not required to obtain a high-value task allocation. Across the tested swarm sizes, four-bit utilities preserve nearly all of the $B = 32$ reference welfare even when the exact robot–task permutation changes. This distinction matters because exact recovery becomes increasingly difficult as the assignment grows, while the loss in mission value remains small. The useful target is therefore not always to reproduce the full-precision assignment, but to preserve the quality of the decision.

MTBR builds on this observation by adding precision only when coarse utilities leave competing assignments unresolved. At matched payload, it consistently outperforms random and local-gap refinement, showing that global assignment competition provides a useful signal for deciding where extra bits should go. The benefit is not free: MTBR requires additional request–reply rounds and therefore more protocol traffic. A fixed low-bit scheme remains simpler and more scalable when communication rounds or latency are costly, while MTBR is more appropriate when selective refinement is affordable.

The current study is limited to synthetic spatial MRTA problems evaluated in a centralized, coordinator-based allocation framework. The robots provide local utility information, but a single coordinator computes the assignment and requests any additional refinement. Communication cost is modeled in bits rather than measured energy or latency, and the experiments do not include sensing uncertainty, changing tasks, distributed consensus, or hardware measurements on miniature robots. These are necessary steps before claiming platform-level savings or fully decentralized swarm applicability. The practical lesson from the present results is simpler: communicate enough precision to make a good assignment decision, and spend additional bits only when the remaining uncertainty can meaningfully change that decision.

## References


[1] Gerkey, B. P., & Matarić, M. J. (2004). A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research, 23*(9), 939–954.

[2] Korsah, G. A., Stentz, A., & Dias, M. B. (2013). A comprehensive taxonomy for multi-robot task allocation. *The International Journal of Robotics Research, 32*(12), 1495–1512.

[3] Bertsekas, D. P. (1988). The auction algorithm: A distributed relaxation method for the assignment problem. *Annals of Operations Research, 14*, 105–123.

[4] Choi, H.-L., Brunet, L., & How, J. P. (2009). Consensus-based decentralized auctions for robust task allocation. *IEEE Transactions on Robotics, 25*(4), 912–926.

[5] Otte, M., Kuhlman, M. J., & Sofge, D. (2020). Auctions for multi-robot task allocation in communication limited environments. *Autonomous Robots, 44*(3–4), 547–584.

[6] Zhao, C., Li, M., Zhu, X., & Zhu, C. (2026). Assignment-consistent dynamic multi-UAV task allocation: Communication-efficient dynamic multi-unmanned aerial vehicle task allocation under stale and asymmetric information. *Drones, 10*(7), 523.

[7] Rikos, A. I., Jiang, W., Charalambous, T., & Johansson, K. H. (2024). Distributed optimization with finite bit adaptive quantization for efficient communication and precision enhancement. In *2024 IEEE 63rd Conference on Decision and Control (CDC)* (pp. 2531–2537). IEEE.

[8] IEEE. (2019). *IEEE standard for floating-point arithmetic* (IEEE Std 754-2019, Revision of IEEE Std 754-2008).

[9] Wood, R., Nagpal, R., & Wei, G.-Y. (2013). Flight of the RoboBees. *Scientific American, 308*(3), 60–65.

[10] Rubenstein, M., Ahler, C., & Nagpal, R. (2012). Kilobot: A low cost scalable robot system for collective behaviors. In *2012 IEEE International Conference on Robotics and Automation* (pp. 3293–3298). IEEE.

[11] Rubenstein, M., Cornejo, A., & Nagpal, R. (2014). Programmable self-assembly in a thousand-robot swarm. *Science, 345*(6198), 795–799.

[12] Ma, K. Y., Chirarattananon, P., Fuller, S. B., & Wood, R. J. (2013). Controlled flight of a biologically inspired, insect-scale robot. *Science, 340*(6132), 603–607.

[13] Virtanen, P., Gommers, R., Oliphant, T. E., et al. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods, 17*(3), 261–272.