Guerrilla Performance Analysis for Robot Swarms: Degrees of Collaboration and Chains of Interference Events

Heiko Hamann[1][(✉)] , Till Aust[1], and Andreagiovanni Reina[2]

[1] Institute of Computer Engineering, University of Lübeck, Lübeck, Germany
hamann@iti.uni-luebeck.de
[2] Department of Computer Science, University of Sheffield, Sheffield, UK
a.reina@sheffield.ac.uk

Abstract. Scalability is a key feature of swarm robotics. Hence, measuring performance depending on swarm size is important to check the validity of the design. Performance diagrams have generic qualities across many different application scenarios. We summarize these findings and condense them in a practical performance analysis guide for swarm robotics. We introduce three general classes of performance: linear increase, saturation, and increase/decrease. As the performance diagrams may contain rich information about underlying processes, such as the degree of collaboration and chains of interference events in crowded situations, we discuss options for quickly devising hypotheses about the underlying robot behaviors. The validity of our performance analysis guide is then made plausible in a number of simple examples based on models and simulations.

1 Introduction

In a world of growing businesses and growing populations the question of how to collaborate effectively and how to form efficient groups is important. Groups that are too large can become inefficient as the cost needed by the group members to coordinate their actions is greater than the benefits the collaboration would bring. For example, rumor has it that Jeff Bezos limits group sizes by the amount its members can eat (so-called 'Two Pizza Rule' [33]) and Brooks's law says "adding manpower to a late software project makes it later" [5]. A scientific result is the Ringelmann effect describing the decreasing productivity of individuals with increasing group size [38]. However, certain systems can exploit collaboration at their advantage to obtain a superlinear increase in group performance, that is, the work completed by the group is more than the sum of work each individual could perform alone. Superlinear increase in group performance, commonly found in swarm robotics [18,30], can also be found in collaborating humans [41] and in distributed computing [9].

In engineered systems, collaboration between the units composing the system can be constrained by limited shared resources, for example, memory access in
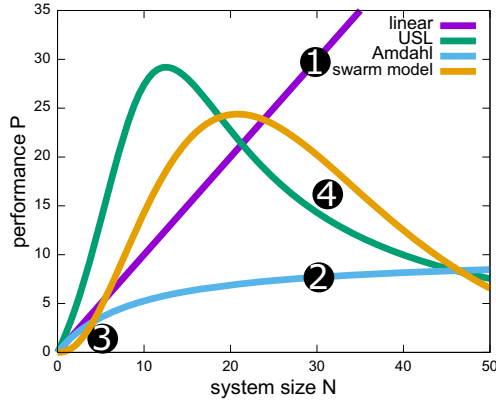
Fig. 1. Schematic plot of qualitatively different swarm performance $P(N)$ curves as a function of the system size N. For large swarm sizes, the performance can (i) increase linearly (purple line, zone ❶), (ii) saturate to a constant value (blue line, zone ❷), or (iii) initially increase (zone ❸ of green and orange lines) and subsequently decrease (zone ❹ of green and orange lines). In scalability analysis, constant increase corresponds to Gustafson's law[10], saturation corresponds to Amdahl's law [1], and increase/decrease to Gunther's universal scalability law (USL) [7]. We show two curves of increase/decrease: The initial phase (zone ❸) can show a slow (orange) or quick (green) increase of performance. The final phase, zone ❹, can show high order (green) or almost linear (orange) decrease. These visual cues can give insights about swarm behavior and efficient design. (Colour figure online)

computing [17] or physical space in swarm robotics [12]. The system performance varies when increasing system size or reducing resources. What is measured by swarm performance P depends on the particular application and scenario. Performance P is a quantification of a task-specific feature that is commonly agreed as a valid measure of success. For example, in foraging that can be the number of collected items [42], in emergent taxis the traveled distance of the swarm's barycenter [2], and in collective decision-making a combination of speed and accuracy [44]. A scalable system is supposed to work efficiently for different load and/or system size [29]. In swarm robotics, scalability of system size is supposed to be a common feature of a properly engineered swarm system [13]. However, robots have a physical body and their movement can interfere with others when the swarm density $\rho = N/A$ (number of robots N per area A) is high [11]. Increasing area A (shared resource) with swarm size N would keep the density ρ constant. This experiment design would provide no information gain about the system's scalability. Instead, we are interested in measuring swarm performance P over swarm density ρ. In most published experiments this, in turn, means measuring swarm performance P over system size N because usually the provided area A is constant.

A promising feature of robot swarms is that they can form an open system ('open swarms' [35]) that have potential for scalability in real time. That is,

robots can join and leave the swarm on demand depending on the needs of the moment [27]. In this type of systems, the robots can collectively adapt to varying swarm size (or densities) by updating their control parameters in real time [34, 45]. While in this work, we focus on swarms with constant size within an experiment, to engineer open swarms, scalability analysis is crucial to quantify the performance for varying system size. In fact, scalability analysis may reveal that adding more units is counterproductive and can instruct the swarm engineer on the most efficient way to react to real-time changes.

Our main motivation is that swarm performance curves $P(N)$ seem to possess generic qualities that appear across a wide collection of different swarm scenarios [11, 12, 14]. Our contribution is to summarize these findings here and to turn them into a practical performance analysis guide for swarm robotics. In this work, our access to understanding swarms is almost exclusively phenomonemological and macroscopic. Still, such findings can help to understand essential qualitative features of the swarm and to develop approaches to resolve performance issues. Although deriving microscopic properties (e.g., required behaviors of individual robots) from macroscopic properties is difficult [15, 36], we are able to indicate some micro-macro links that may even be generic. For example, we show how the macroscopic performance curve can indicate whether small or big groups of robots interact in beneficial or detrimental ways.

The term 'guerrilla' in the title is a tribute to Gunther [7] who wrote the renowned book 'Guerrilla Capacity Planning' [8] to provide industry managers with a simple framework for scalabilty planning. Here we let the term represent the rather practical and phenomenological top-down approach to performance in swarm robotics. We provide a practical guide to quickly understand fundamental scalability features of a studied swarm based on elementary insights about superficial characteristics of their $P(N)$ plot. We present three classes of performance: linear increase, saturation, and increase/decrease. Then we focus on the increase/decrease class and discuss how the performance curve can explain the relationship between collaboration and interference among the robots.

2   Three General Classes of Performance System Behavior

Analyzing the system performance $P(N)$ reveals three qualitatively different types of scalability classes: linear increase, saturation, and increase/decrease.

2.1   Linear Increase

If we observe a sustained trend of performance $P(N) \propto N$ up to large values of N (see purple line ❶ in Fig. 1), then we observe the scalability class of *linear increase*. This situation is advantageous as the swarm performance improves by increasing the number of robots. However, we should note that it cannot be considered the ideal case as also superlinear performance scaling can be observed in swarm robotics and computing systems [9, 12, 28], as represented by the rapid initial increase of the green curve in Fig. 1.

2.2 Saturation

We observe the *saturation* class when performance $P(N)$ approaches a maximum $P(N \to \infty) = s^*$ (see blue curve ❷ in Fig. 1). Therefore, such a regime has no performance peak and is equivalent to Amdahl's law [1] that was originally formulated to (pessimistically) describe the scalability of parallel computers. While Amdahl's law has demonstrated its applicability [8], we argue that this saturation scenario is rare in swarm robotics or ignores costs (see Sect. 2.4). Physical interference due to high robot densities usually has a significant impact on swarm performance causing an increase/decrease situation.

2.3 Increase/Decrease

In swarm robotics, the representative scalability class is *increase/decrease*, characterized by increasing performance for small N, a performance peak at a critical swarm size N_c, and decreasing performance for $N > N_c$. Performance $P(N < N_c)$ increases because robots efficiently collaborate or work in parallel to perform the task, and performance $P(N > N_c)$ decreases because robots interfere with each other.

Gunther [7,9] proposed the universal scalability law (USL) to describe this increase/decrease class as observed in computing. The USL is based on performance improvements S (speedup) for size N compared to the minimal system $N = 1$. The USL is

$$S(N) = \frac{P(N)}{P(1)} = \frac{N}{1 + \sigma(N-1) + \kappa N(N-1)} \ , \tag{1}$$

with parameter σ describing the influence of contention (e.g., queues for shared resources) and parameter κ describing a coherency delay (e.g., distributing and synchronizing information). The USL properly parameterized by σ and κ covers all scalability classes (*linear increase, saturation, increase/decrease*). In Fig. 1, the green line (labeled USL) gives an example for *increase/decrease*. Another model developed especially for the scalability analysis of robot swarms [11,12] is

$$P(N) = aN^b \exp(cN) \ , \tag{2}$$

for constants $a > 0$, $b > 0$, and $c < 0$. The function can be understood as a dichotomous pair of a term for potential of collaboration N^b and a term for interference $\exp(cN)$. In Fig. 1 the orange curve is an example of Eq. 2 in the *increase/decrease* regime (labeled 'swarm model').

2.4 Ambiguous Definition of Swarm Performance

In Sect. 1 we argued that the definition of swarm performance $P(N)$ for a particular application scenario should be an agreed measure of success. This introduces degrees of subjectivity in our scalability analysis and ultimately ambiguity in the observed results. While it seems unlikely that this can be resolved in a fully

generic way, we propose four simple guidelines of how to improve the scalability analysis and avoid common mistakes: constant task, full range, added cost, and marginal performance.

Constant Task. In any performance analysis, but specifically for large system size, when the performance curve keeps growing as $P(N) \propto N$ (see ❶ in Fig. 1) the swarm performance analysis practitioner should question if the performance has been measured on the same task T for any swarm size N. By adapting the task to large system sizes, the performance may not provide useful indications on the system's scalability as two parameters (size N and task T) have been changed at the same time. We recommend to consider as part of the task a constant working area $A \in T$. Increasing size N of a swarm on constant area has the effect of increasing swarm density $\rho = N/A$ that can increase physical interference among robots. Physical interference is expected to have a negative impact on the swarm performance $P(N)$. While we acknowledge that certain tasks—e.g., area coverage or movement-free tasks based on communication only—could exploit increased density to improve the performance [6], we argue that linear increase is a pathological case that should be carefully interpreted. As the performance should measure the completion of a fixed task, it could be expected that it would, at least, saturate for large sizes N.

Full Range. Another typical shortcoming of performance analysis that could explain the observation of a linear increase of performance for large sizes N, is a short range of N. Considering only relatively small sizes of N would only show a partial picture of the system behavior. An incomplete scalability analysis could be harmful as the system behavior would not be fully understood. For example, in cleaning or object collection tasks, it is reasonable that performance saturates once dirty areas get scarce or most objects have been collected respectively.

Added Cost. A performance curve that does not decrease for large system sizes (e.g., see ❶ and ❷ in Fig. 1 for linear increase and saturation respectively) suggests minimal interference among robots. For example, in an area coverage task, the more robots are added to the swarm, the better the area gets covered until performance saturation is observed [31]. Scalability analysis should support the system designer in making decisions about the optimal swarm size in terms of its internal function and real-world factors, such as deployment cost. Hence, system performance $P(N)$ should be complemented with the cost of added units to select the 'best' swarm size N. In the above coverage task, the saturation of the performance puts the scalability analyst in a situation where performances of large swarms cannot be distinguished anymore (e.g., $P(N) \approx P(10^3 N)$). We would ignore effects of diminishing returns. In addition, one may be tempted to add more robots to increase redundancy and robustness (redundancy-induced robustness). The lack of any cost suggests that 'bigger is better' as there is no immediate negative impact of interference and performance P increases monotonically with N or interference may even be a feature [6].
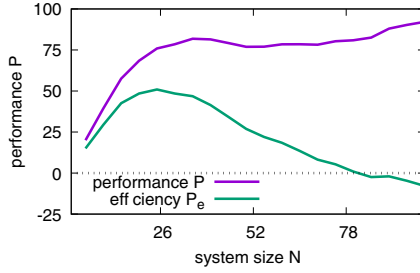
Fig. 2. Saturating performance P based on data from simulations of a coverage task by Özdemir et al. [31] and the efficiency measure $P_e = P(N) - C(N)$ for $C(N) = N$.

We recommend to always complement the study of swarm performance $P(N)$ with the study of cost $C(N)$, to analyze the efficiency $P_e(N) = P(N) - C(N)$, that can be more informative than $P(N)$ alone. Cost $C(N)$ should account for relevant aspects, such as economical (purchase of additional robots [40]) or logistic costs (covering the environment with robots would reduce the space for other type of activities). For example, this would allow to usefully balance the cost and benefits of redundancy-induced robustness. In Fig. 2b, we show the effect of adding a constant cost per unit $C(N) = c\,N$ to the performance data of an area coverage study by Özdemir et al. [31]. P_e shows a peak and can hence indicate an optimal swarm size N. A designer seeking robustness can quantify the decrease in efficiency and choose an appropriate swarm size N.

Marginal Performance. Another measure that can improve scalability analysis is marginal performance $P_m(N) = P(N) - P(N-1) = dP(N)/dN$. Considering added swarm performance per unit can help deciding the swarm size. The measure $P_m(N)$ can be particularly useful when compared with the marginal cost $C_m(N) = C(N) - C(N-1) = dC(N)/dN$. For $P_m(N) < C_m(N)$, adding robots to the system would decrease swarm performance. Similarly, one could consider the mean individual performance $I(N) = P(N)/N$. In a more holistic way, here the entire swarm shares the benefits of an added robot. Also in this case, the measure $I(N)$, that indicates the performance contribution of each robot, can be compared with the individual cost $I_c(N) = C(N)/N$ in order to appropriately scale the system.

3   From Eye-Catchers to a Practical Performance Analysis

The performance class that is most frequently observed in swarm robotics is *increase/decrease*. For this class we provide a guide how to quickly interpret $P(N)$ diagrams in terms of two features: shape of the curve for small system sizes (see ❸ in Fig. 1) and shape of the curve for large systems (see ❹ in Fig. 1).

3.1   Increase: Low- and High-Order Robot-Robot Collaboration

By looking at the initial phase of the performance curve (❸ in Fig. 1, for $N <$ 15), we can obtain indications of how much robot-robot collaboration is done to complete the task (cf. other, more sophisticated efforts to derive group sizes from macroscopic measurements [15]). A fast increase of $P(N)$ for smallest values $N \in \{1, 2, 3\}$, shows that a small swarm is already sufficient to complete at least parts of the task (e.g., green curve of Fig. 1). Instead, if the curve has a slow start and $P(N)$ shows a noticeable increase only for larger sizes N, it could indicate the necessity of robot-robot collaboration in larger groups. In most published swarm performance measurements, the initial increase of performance is approximately linear (fast increase). However, there are rare cases of published datasets showing a nonlinear (curved) and slow increase [26]. Note that we do not focus on distinguishing between super- and sub-linear performance increases, instead we try to understand when to expect linear and when nonlinear increases.

Both scalability functions described in Sect. 2.3 can represent both linear (fast) and nonlinear (slow) increase despite their simplicity. Interestingly, similar nonlinear system behaviors can be observed in models from not directly related fields, such as PT2 lag elements in control theory, or residence times in cascades of stirred-tank reactors (tanks in series) [23]. In both of these examples, sequences of events or higher order time-delays introduce the observed nonlinearity. Comparable effects emerge in robot swarms when several robots need to collaborate in order to perform the given task.

To support our above claims, we show two minimal examples in which observing the system performance curve for small system sizes (❸, $N < 10$, in Fig. 1) allows us to estimate the necessary amount of robot-robot interactions to complete the task. If robots can perform the task without any/much help from other robots, then the initial increase is steep and close to linear. We say that robots require low-order interactions. If robots require considerable help from other robots to perform the task, then the initial performance remains low for small sizes N and shows a curved (nonlinear) increase. We say that robots require high-order interactions. We give evidence for this conjecture through two simple analyses: a simple combinatorial argumentation and empirical observations in simulations of an abstract system inspired by the stick pulling experiment [18].

Our combinatorial consideration is based on the precondition for robot-robot collaboration: robots need to be in close proximity to each other. In swarm robotics, robot movement is often based on random motion [3]. We consider the probability that collaboration among k robots takes place as a stochastic event proportional to k and swarm density ρ. Assuming a simple grid environment where collaboration takes place between neighboring robots, we can derive the probability of having at least k robots in Moore neighborhoods, 3×3, of $m = 9$ cells. Swarm density ρ indicates the (independent) probability of finding a robot in a given cell. The probability Γ_k of finding at least k robots in a Moore neighborhood of $m = 9$ cells corresponds to $\Gamma_k = \sum_{i=k}^{m} \binom{m}{i} \rho^i (1-\rho)^{(m-i)}$. In Fig. 3a we show Γ_k as a function of ρ for $k \in \{1, 2, 3, 4\}$. As expected, the probability that at least k robots meet (our assumed precondition for collabo-

(a) $\Gamma_k = \sum_{i=k}^{m} \binom{m}{i} \rho^i (1-\rho)^{(m-i)}$

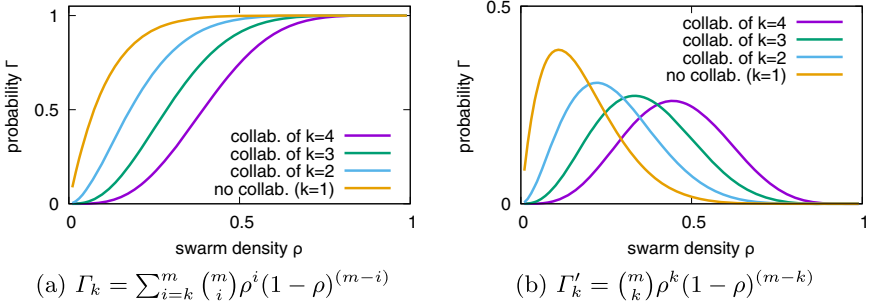(b) $\Gamma'_k = \binom{m}{k} \rho^k (1-\rho)^{(m-k)}$

Fig. 3. Combinatorial explanation of chances for collaboration, collaboration probability Γ_k for $k \in \{1, 2, 3, 4\}$ and neighborhood size $m = 9$ (Moore neighborhood), and swarm density ρ; two scenarios: (a) without and (b) with interference.

ration) decreases by increasing k. Looking at the initial part of the curves, for low density values, larger groups have a slow (nonlinear) increase. Instead, small groups (e.g., $k = 1$ or $k = 2$) have fast and almost linear increases. In Fig. 3a, we assume no interference between robots, thus values larger than k still allow for collaboration without overhead. In Fig. 3b, we assume that values larger than k would prohibit collaboration. The shown probability is $\Gamma'_k = \binom{m}{k} \rho^k (1-\rho)^{(m-k)}$. Despite the different shapes for high densities (see Sect. 3.2), the initial part shows the same type of shapes for varying k.
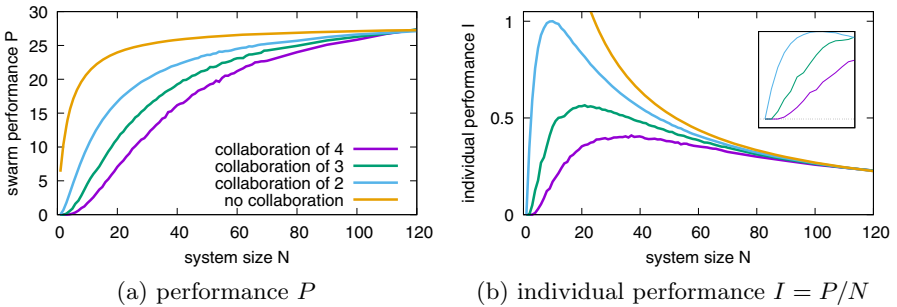

(a) performance P

(b) individual performance $I = P/N$

Fig. 4. Stick pulling scenario: (a) Swarm performance P and (b) individual performance I measured in an abstract stochastic model of the robot swarm stick pulling scenario where one (no collaboration), two, three, or four robots are required to pull a stick. Performance normalized to equalize for $P(N = 120)$. Inset shows $N \in \{1, 2, \ldots, 15\}$ for all performance maxima normalized to one.

Our second argumentation is based on a minimal simulation of a simplistic abstract model inspired by the stick pulling scenario [18] that was published before [12]. We have a swarm of N robots and $M = 20$ stick sites containing one stick each. In the original experiment, collaboration of $k = 2$ robots is

required to successfully pull a stick. We test four cases with $k \in \{1, 2, 3, 4\}$, where $k = 1$ means no collaboration required and $k = 4$ means four robots are required to pull one stick. Robots commute randomly between stick sites and their arrival times are modeled in time steps by commute times $\tau(N) = N + \xi$ (i.e., linearly proportional to swarm size N) for a noise term $\xi \in \{0, 1, 2\}$. Robots wait at stick sites for up to seven time steps until they give up and leave to a randomly chosen stick site. All robots are initialized to the commute state with uniformly distributed arrival times $\tau \in \{0, 1, \ldots, N-1\}$. We simulate for swarm sizes $N \in \{1, 2, \ldots, 120\}$, for 1,000 time steps each, with constant stick site number $M = 20$, and 10^4 independent runs for each N. The results are shown in Fig. 4. Figure 4a shows the normalized swarm performance P for required collaboration $k \in \{1, 2, 3, 4\}$. Swarm performance P saturates because commute times τ scale linearly with swarm size N. The nonlinear effect of higher-order collaboration ($k = 3$ and $k = 4$) is obvious. Figure 4b shows the normalized individual performance $I = P/N$. Again showing the nonlinear effect of higher-order collaboration. In addition, we also see qualitative differences in the curves for high swarm sizes $N > 30$: curved for $k = 1$ and almost linear for $k = 4$.

3.2   Decrease: Low- and High-Order Robot-Robot Interaction

Now we study the decrease in swarm performance for sizes N bigger than the swarm size N_c for peak performance (see ❹ in Fig. 1). In published works reporting swarm performance diagrams, the plot of $P(N)$ is sometimes almost linear [16,25], sometimes slightly curved [22,43,47], and sometimes curved [18,39,42] for sizes $N > N_c$. For example, Llenas et al. [25] report performance plots with graceful linear degradation for a foraging scenario. The underlying simulation of Kilobots was simplified, temporarily small clusters formed that dissolved quickly, and traffic lanes were formed. Hence, most collision avoidance actions were of first order, that is, robots made a transition to collision avoidance but didn't trigger collision avoidance in others. This is similar to traffic models were a linear decrease is assumed classically, for example in the Lighthill–Whitham–Richards (LWR) model [24]. The traffic is assumed to be fully synchronized with strong serial dependencies due to lanes (1-d space) for system size N_c. If system size is further increased, traffic is disturbed, and for too crowded systems traffic jams emerge. In swarm robotics the situation is more complex as space is 2-d and it is unknown which robots in collision avoidance state may trigger collision avoidance in others. Another analogy are transport systems [19]. There viscosity or mechanical impedance increases nonlinearly with concentration (cf. interference in Eq. 2). For robots that translates to number of collision avoidance events.

Performance for big sizes (for $N > 20$ as seen at ❹ in Fig. 1) is our focus now. If robots interfering with each other manage to resolve the interference (e.g., by avoidance movements) and return to productive mode quickly, then the performance decrease is low and close to linear. We say they show low-order interference. If robots by trying to resolve interference, trigger cascades of collision avoidance, then the performance decrease is steep and curved. We say

they show high-order interference. To support our claims, we present empirical evidence based on a simulation. The main idea of this experiment is to control the number of collision-avoidance events that a robot triggers. We define as first order interference the collision avoidance that is triggered by two robots moving close to each other. During *collision avoidance* (CA), the robots perform a set of maneuvers to avoid physical collision. If during the execution of this set of avoidance maneuvers, the robot triggers collision avoidance in another robot, we define it as second order interference. Therefore, when these robots performing collision avoidance (in state CA) trigger another j^{th} robot, such event corresponds to the jth order interference, for j robots involved. This is related to the basic reproduction number R_0 in the SIR epidemic model [20], where R_0 defines the average number of infections that each infected individuals causes. Considering R_0 the average number of collision avoidance events that each robot in state CA triggers, we have that with $R_0 = 1$ each robot in state CA 'infects' one other robot with the 'collision-avoidance disease.' With $R_0 > 1$ each robot in state CA triggers more than one collision avoidance, its growth is exponential, and the resulting decrease of performance $P(N)$ is nonlinear.

We use the Webots simulation environment [46] for our experiments on interference. The simulated robot is the Thymio II [37] operating as a swarm of size N in a 2 m × 2 m arena. We simulate a simple multi-robot navigation task. The arena has four bases (north, south, east, west). The robots' goal is to reach the respective opposite base (e.g., from north to south and vice versa). At the beginning of each run, we randomly distribute $N = 1$ to $N = 55$ robots (at least 10 cm away from any wall) depending on the density we want to test. Then the robots do a random walk until they touch a base. They use it then as their reference base (from where they started) to set the vis-a-vis base as their next target (north and south, east and west). When a robot detects an obstacle (wall or robot) or its target base, it turns in a random direction for a random time, and moves straight again. When they touch the target, the performance counter is increased by one (and the new target is the opposite base). One run takes 6 simulated minutes. We do two types of simulation runs. In full avoidance runs, all robots follow the standard procedure and remain in the arena at all times. In first order avoidance runs, we limit the effect of interference by limiting collision avoidance triggering cascades. A robot in state CA that triggers a transition to CA in another robot is allowed to trigger only this one CA event but is then temporarily removed with probability P_{remove} for the time it stays in state CA. Once its CA behavior has been completed, it is put back into the arena if the spot is empty; otherwise it is put back later once the spot is empty. We vary probability $P_{\text{remove}} \in \{0.4, 0.7, 1\}$ where $P_{\text{remove}} = 1$ means the robot is always removed once it has triggered CA in another robot.
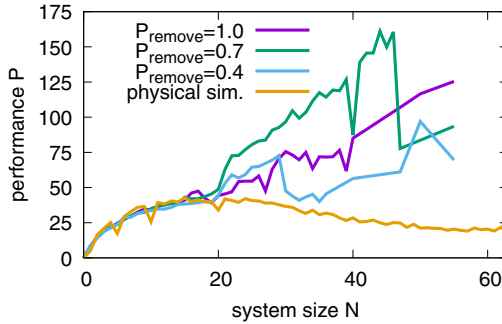
Fig. 5. Interference in swarms of N robots: simple navigation task, with probability $P_{\mathrm{remove}} \in \{0.4, 0.7, 1\}$ we remove robots that triggered collision avoidance of others ($n = 100$ independent simulation runs per data point).

The results are shown in Fig. 5.[1] The data is noisy despite the invested computational effort of more than 200 CPU days. However, an overall trend can be noticed. With P_{remove} the swarm density is regulated by removing robots that cannot be put back into the arena immediately because their spot is taken. Due to a nontrivial and not further discussed interplay of effects, the setup $P_{\mathrm{remove}} = 1$ is better for $N \geq 47$ but outperformed by setup $P_{\mathrm{remove}} = 0.7$ for $20 < N < 47$. Either robots are taken out too quickly slowing down their travels ($P_{\mathrm{remove}} = 1$, $N < 47$) or robots are left in the system increasing collision avoidance events ($P_{\mathrm{remove}} = 0.7$, $N \geq 47$). Hence, independent of the task a robot swarm can artificially be pushed from increase/decrease to the saturation scenario. This is similar to other approaches where simulated physics (embodied systems not allowing to pass through other bodies) was turned on/off [39,42]. Also behaviors in ants mitigate overcrowded situations to avoid the increase/decrease situation in favor of a saturation scenario [4,21,32]. With our experiment we investigated the impact of interference on performance by modulating probability P_{remove}.

4   Conclusion

We have given a practical guide to analyze swarm performance and scalability. Swarm performance plots contain rich information about underlying processes. The left part of the swarm performance plot can give hints on the level of collaboration necessary to solve the task. The right part of the plot is a reflection of the ratio between marginal cost and performance. Performance scales in qualitatively different ways depending on the task. Tasks that are not limited by physical interference (e.g., area coverage) show no collapse of performance for increased swarm sizes. However, usually physical interference has a negative effect in a variety of tasks. These qualitative differences vanish once we a apply a benefit-cost analysis (BCA) that reveals the relation between the marginal performance

[1] See http://doi.org/10.5281/zenodo.3947822 for videos, screenshot, and data.

(added swarm performance of an added robot) and the relative marginal cost. An important design choice is about the redundancy-induced robustness. Swarm robotics is commonly assumed to be robust to failures because of its high degree of redundancy. In a homogeneous swarm, robots are exchangeable and serve as mutual replacements. Through BCA and marginal cost/performance analysis the designer can make a more informed choice to balance the efficiency-robustness tradeoff. Following our practical ('guerrilla') performance analysis guide allows swarm scalability analysts to quickly formulate hypotheses about the underlying system behaviors and consequently to speedup the design and studies in swarm robotics.

References

1. Amdahl, G.M.: Validity of the single processor approach to achieving large scale computing capabilities. In: AFIPS Conference Proceedings, pp. 483–485. ACM (1967)
2. Bjerknes, J.D., Winfield, A., Melhuish, C.: An analysis of emergent taxis in a wireless connected swarm of mobile robots. In: Shi, Y., Dorigo, M. (eds.) IEEE Swarm Intelligence Symposium, pp. 45–52. IEEE Press, Los Alamitos (2007)
3. Dimidov, C., Oriolo, G., Trianni, V.: Random walks in swarm robotics: an experiment with kilobots. In: Dorigo, M., et al. (eds.) ANTS 2016. LNCS, vol. 9882, pp. 185–196. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44427-7_16
4. Dussutour, A., Fourcassié, V., Helbing, D., Deneubourg, J.L.: Optimal traffic organization in ants under crowded conditions. Nature **428**, 70–73 (2004)
5. Frederick, P., Brooks, J.: The Mythical Man-Month. Addison-Wesley, Boston (1995)
6. Goldberg, D., Matarić, M.J.: Interference as a tool for designing and evaluating multi-robot controllers. In: Kuipers, B.J., Webber, B. (eds.) Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI 1997), pp. 637–642. MIT Press, Cambridge (1997)
7. Gunther, N.J.: A simple capacity model of massively parallel transaction systems. In: CMG National Conference, pp. 1035–1044 (1993)
8. Gunther, N.J.: Guerrilla Capacity Planning. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-31010-5
9. Gunther, N.J., Puglia, P., Tomasette, K.: Hadoop super-linear scalability: the perpetual motion of parallel performance. ACM Queue **13**(5), 46–55 (2015)
10. Gustafson, J.L.: Reevaluating Amdahl's law. Commun. ACM **31**(5), 532–533 (1988). https://doi.org/10.1145/42411.42415
11. Hamann, H.: Towards swarm calculus: urn models of collective decisions and universal properties of swarm performance. Swarm Intell. **7**(2–3), 145–172 (2013). https://doi.org/10.1007/s11721-013-0080-0
12. Hamann, H.: Superlinear scalability in parallel computing and multi-robot systems: shared resources, collaboration, and network topology. In: Berekovic, M., Buchty, R., Hamann, H., Koch, D., Pionteck, T. (eds.) ARCS 2018. LNCS, vol. 10793, pp. 31–42. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-77610-1_3
13. Swarm Robotics: A Formal Approach. Lecture Notes in Computer Science. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-74528-2_7
14. Hamann, H., Reina, A.: Scalability in computing and robotics. arXiv, June 2020. https://arxiv.org/abs/2006.04969

15. Hamann, H., Valentini, G., Khaluf, Y., Dorigo, M.: Derivation of a micro-macro link for collective decision-making systems. In: Bartz-Beielstein, T., Branke, J., Filipič, B., Smith, J. (eds.) PPSN 2014. LNCS, vol. 8672, pp. 181–190. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10762-2_18

16. Hayes, A.T.: How many robots? Group size and efficiency in collective search tasks. In: Asama, H., Arai, T., Fukuda, T., Hasegawa T. (eds.) Distributed Autonomous Robotic Systems, vol. 5, pp. 289–298. Springer, Tokyo (2002). https://doi.org/10.1007/978-4-431-65941-9_29

17. Hill, M.D.: What is scalability? ACM SIGARCH Comput. Archit. News **18**(4), 18–21 (1990)

18. Ijspeert, A.J., Martinoli, A., Billard, A., Gambardella, L.M.: Collaboration through the exploitation of local interactions in autonomous collective robotics: the stick pulling experiment. Auton. Robots **11**, 149–171 (2001). https://doi.org/10.1023/A:1011227210047

19. Jensen, K.H., Kim, W., Holbrook, N.M., Bush, J.W.M.: Optimal concentrations in transport systems. J. Roy. Soc. Interface **10**(83), 20130138 (2013)

20. Keeling, M.J., Rohani, P.: Modeling Infectious Diseases in Humans and Animals. Princeton University Press, Princeton (2011)

21. Laure-Anne, P., Sebastien, M., Jacques, G., Buhl, J., Audrey, D.: Experimental investigation of ant traffic under crowded conditions. eLife **8**, e48945 (2019)

22. Lerman, K., Galstyan, A.: Mathematical model of foraging in a group of robots: effect of interference. Auton. Robots **13**, 127–141 (2002)

23. Levenspiel, O.: Chemical reaction engineering. Ind. Eng. Chem. Res. **38**(11), 4140–4143 (1999)

24. Lighthill, M.J., Whitham, G.B.: On kinematic waves. II. A theory of traffic flow on long crowded roads. Proc. Roy. Soc. London **A229**(1178), 317–345 (1955)

25. Font Llenas, A., Talamali, M.S., Xu, X., Marshall, J.A.R., Reina, A.: Quality-sensitive Foraging by a robot swarm through virtual pheromone trails. In: Dorigo, M., Birattari, M., Blum, C., Christensen, A.L., Reina, A., Trianni, V. (eds.) ANTS 2018. LNCS, vol. 11172, pp. 135–149. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00533-7_11

26. Mateo, D., Kuan, Y.K., Bouffanais, R.: Effect of correlations in swarms on collective response. Sci. Rep. **7**, 10388 (2017). https://doi.org/10.1038/s41598-017-09830-w

27. Mayya, S., Pierpaoli, P., Egerstedt, M.: Voluntary retreat for decentralized interference reduction in robot swarms. In: International Conference on Robotics and Automation (ICRA), pp. 9667–9673, May 2019. https://doi.org/10.1109/ICRA.2019.8794124

28. Mondada, F., Gambardella, L.M., Floreano, D., Nolfi, S., Deneubourg, J.L., Dorigo, M.: The cooperation of swarm-bots: physical interactions in collective robotics. IEEE Robot. Autom. Mag. **12**(2), 21–28 (2005)

29. Neuman, B.C.: Scale in distributed systems. In: Readings in Distributed Computing Systems. IEEE Computer Society Press (1994)

30. O'Grady, R., Gross, R., Christensen, A.L., Mondada, F., Bonani, M., Dorigo, M.: Performance benefits of self-assembly in a swarm-bot. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2381–2387, October 2007. https://doi.org/10.1109/IROS.2007.4399424

31. Özdemir, A., Gauci, M., Kolling, A., Hall, M.D., Groß, R.: Spatial coverage without computation. In: International Conference on Robotics and Automation (ICRA), pp. 9674–9680. IEEE (2019)

32. Poissonnier, L.A., Motsch, S., Gautrais, J., Buhl, J., Dussutour, A.: Experimental investigation of ant traffic under crowded conditions. eLife **8**, e48945 (2019). https://doi.org/10.7554/eLife.48945
33. Pratt, E.L.: Virtual teams in very small classes. Virtual Teamwork 91 (2010)
34. Rausch, I., Reina, A., Simoens, P., Khaluf, Y.: Coherent collective behaviour emerging from decentralised balancing of social feedback and noise. Swarm Intell. 321–345 (2019). https://doi.org/10.1007/s11721-019-00173-y
35. Reina, A.: Robot teams stay safe with blockchains. Nat. Mach. Intell. **2**, 240–241 (2020). https://doi.org/10.1038/s42256-020-0178-1
36. Reina, A., Miletitch, R., Dorigo, M., Trianni, V.: A quantitative micro-macro link for collective decisions: the shortest path discovery/selection example. Swarm Intell. **9**(2–3), 75–102 (2015)
37. Riedo, F., Chevalier, M., Magnenat, S., Mondada, F.: Thymio II, a robot that grows wiser with children. In: IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO 2013), pp. 187–193. IEEE (2013)
38. Ringelmann, M.: Recherches sur les moteurs animés: Travail de l'homme. Annales de l'Institut National Agronomique, 2nd series **12**, 1–40 (1913)
39. Rosenfeld, A., Kaminka, G.A., Kraus, S.: A study of scalability properties in robotic teams. In: Scerri, P., Vincent, R., Mailler, R. (eds.) Coordination of Large-Scale Multiagent Systems, pp. 27–51. Springer, Boston (2006). https://doi.org/10.1007/0-387-27972-5_2
40. Salman, M., Ligot, A., Birattari, M.: Concurrent design of control software and configuration of hardware for robot swarms under economic constraints. PeerJ Comput. Sci. **5**, e221 (2019). https://doi.org/10.7717/peerj-cs.221
41. Sornette, D., Maillart, T., Ghezzi, G.: How much is the whole really more than the sum of its parts? $1 \boxplus 1 = 2.5$: superlinear productivity in collective group actions. PLOS ONE **9**(8), 1–15 (2014). https://doi.org/10.1371/journal.pone.0103023
42. Talamali, M.S., Bose, T., Haire, M., Xu, X., Marshall, J.A.R., Reina, A.: Sophisticated collective foraging with minimalist agents: a swarm robotics test. Swarm Intell. **14**(1), 25–56 (2019). https://doi.org/10.1007/s11721-019-00176-9
43. Trianni, V., Groß, R., Labella, T.H., Şahin, E., Dorigo, M.: Evolving aggregation behaviors in a swarm of robots. In: Banzhaf, W., Ziegler, J., Christaller, T., Dittrich, P., Kim, J.T. (eds.) ECAL 2003. LNCS (LNAI), vol. 2801, pp. 865–874. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39432-7_93
44. Valentini, G.: Achieving Consensus in Robot Swarms. SCI, vol. 706. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-53609-5
45. Wahby, M., Petzold, J., Eschke, C., Schmickl, T., Hamann, H.: Collective change detection: adaptivity to dynamic swarm densities and light conditions in robot swarms. Artif. Life Conf. Proc. **31**, 642–649 (2019). https://doi.org/10.1162/isal_a_00233
46. Webots: version r2020a by Cyberbotics Ltd. (2020). https://cyberbotics.com
47. Zahadat, P., Hofstadler, D.N.: Toward a theory of collective resource distribution: a study of a dynamic morphogenesis controller. Swarm Intell. (1), 347–380 (2019). https://doi.org/10.1007/s11721-019-00174-x