Superlinear Scalability in Parallel Computing and Multi-Robot Systems: Shared Resources, Collaboration, and Network Topology

Heiko Hamann

April 12, 2018

Abstract

The uniting idea of both parallel computing and multi-robot systems is that having multiple processors or robots working on a task decreases the processing time. Typically we desire a linear speedup, that is, doubling the number of processing units halves the execution time. Sometimes superlinear scalability is observed in parallel computing systems and more frequently in multi-robot and swarm systems. Superlinearity means each individual processing unit gets more efficient by increasing the system size—a desired and rather counterintuitive phenomenon.

In an interdisciplinary approach, we compare abstract models of system performance from three different fields of research: parallel computing, multi-robot systems, and network science. We find agreement in the modeled universal properties of scalability and summarize our findings by formulating more generic interpretations of the observed phenomena. Our result is that scalability across fields can be interpreted as a tradeoff in three dimensions between too competitive and too cooperative processing schemes, too little information sharing and too much information sharing, while finding a balance between neither underusing nor depleting shared resources. We successfully verify our claims by two simple simulations of a multi-robot and a network system.

1 Introduction

Superlinear scalability is a desirable phenomenon in both parallel computing and multi-robot systems. It is counterintuitive because one seemingly receives a profit without paying for it. Also our experience of working together in human groups dominantly gives a different impression, as indicated, for example, by the Ringelmann effect. According to Ingham et al. [1] the Ringelmann effect implies a nonlinear decrease of individual performance with increasing group size. We call the improvement in speed of the task execution achieved by adding more processing units 'speedup': $S = T_1/T_p$, for latency T_1 of the smaller system (typically one processing unit) and latency T_p of the bigger system ($p > 1$ processing units); efficiency is $E = T_1/(pT_p)$. Scalability describes how far we can go in keep adding processing units p without getting $E < 1$.

As superlinear speedups seem special, they were frequently discussed and studied [2, 3]. There even exists a proof showing the impossibility of superlinear speedups but it assumes fixed problem size [4]. Superlinear speedups are rather infrequently observed in parallel computing (e.g., cache-size effects [5]) compared to rather frequent observations in multi-robot and swarm systems (e.g., inherently collaborative tasks [6]). When observed, superlinearity is often a discrete effect, such as a workpackage happening to fit into the processors cache [5] or a robot group being able to form a bucket brigade [7, 8]. Superlinear scalability has much potential that should be enough motivation to investigate it across different domains and to understand how one can provoke it.

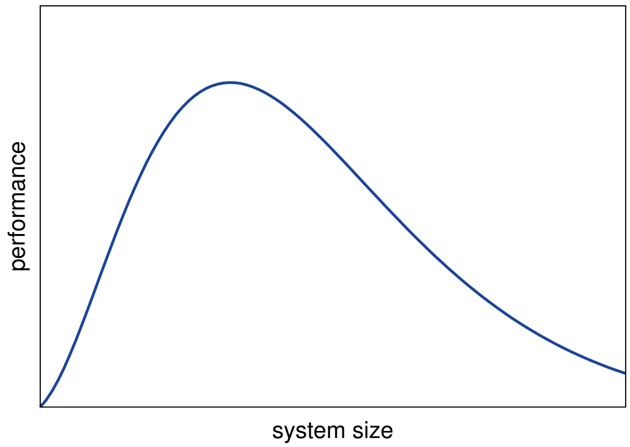


Figure 1: Generic diagram of system performance over system size for multi-robot systems, without units, function $P(N) = N^b \exp(cN)$ (eq. 1) depending on parameters $b > 1$ (profits from cooperation, N^b), $c < 0$ (overhead due to interference, $\exp(-N)$), and system size N [10, 11].

1.1 Superlinear performance in multi-robot systems

Superlinear performance increases are observed in multi-robot systems due to physical effects in tasks, such as pulling, passing gaps, and passing steps [9]. Analyzing the literature on multi-robot systems and, in particular, swarm robotics, one finds that plots of system performance over system size (number of robots) have similar features independent of the investigate task (see Fig. 1 as example).

It was noticed that there is an optimal robot density, that is, how many robots should share the same area [12]. Initially the performance curve increases with increased number of robots but then first levels off and then decreases [13]. The most obvious shared resource in robotics is space. Adding robots to the system while keeping the provided area constant, the additional robots may generate more possibilities to cooperate but they may also physically interfere [14, 15]. So we identify a first tradeoff between options for collaborations and an increased overhead due to physical interference. Østergaard et al. [16] discuss the existence of a general multi-robot performance diagram with a focus on its peak performance:

> *We know that by varying the implementation of a given task, we can move the point of "maximum performance" and we can change the shapes of the curve on either side of it, but we cannot change the general shape of the graph.*

Examples of such performance diagrams are found across the literature, for example, in multi-robot foraging [17, 14, 18, 19, 20], collective decision making [21], the emergent taxis scenario [11, 22, 23], and aggregation behaviors [24, 25].

In previous works [11, 10], the author has proposed the following simple model of performance $P(N)$ in multi-robot and swarm systems:

$$P(N) = C(N)(I(N) - d) = a_1 N^b a_2 \exp(cN) \ , \tag{1}$$

for a cooperation function $C(N) = a_1 N^b$, an interference function $I(N) = a_2 \exp(cN) + d$, parameter $c < 0$, and scaling constants $a_1, a_2 > 0$, d for translation up/down. For $b > 1$ we have potentially superlinear scalability but interference is counteracting exponentially with $\exp(-N)$. This rather rough and abstract model was successfully fitted to a number of multi-robot scenarios [11, 10].

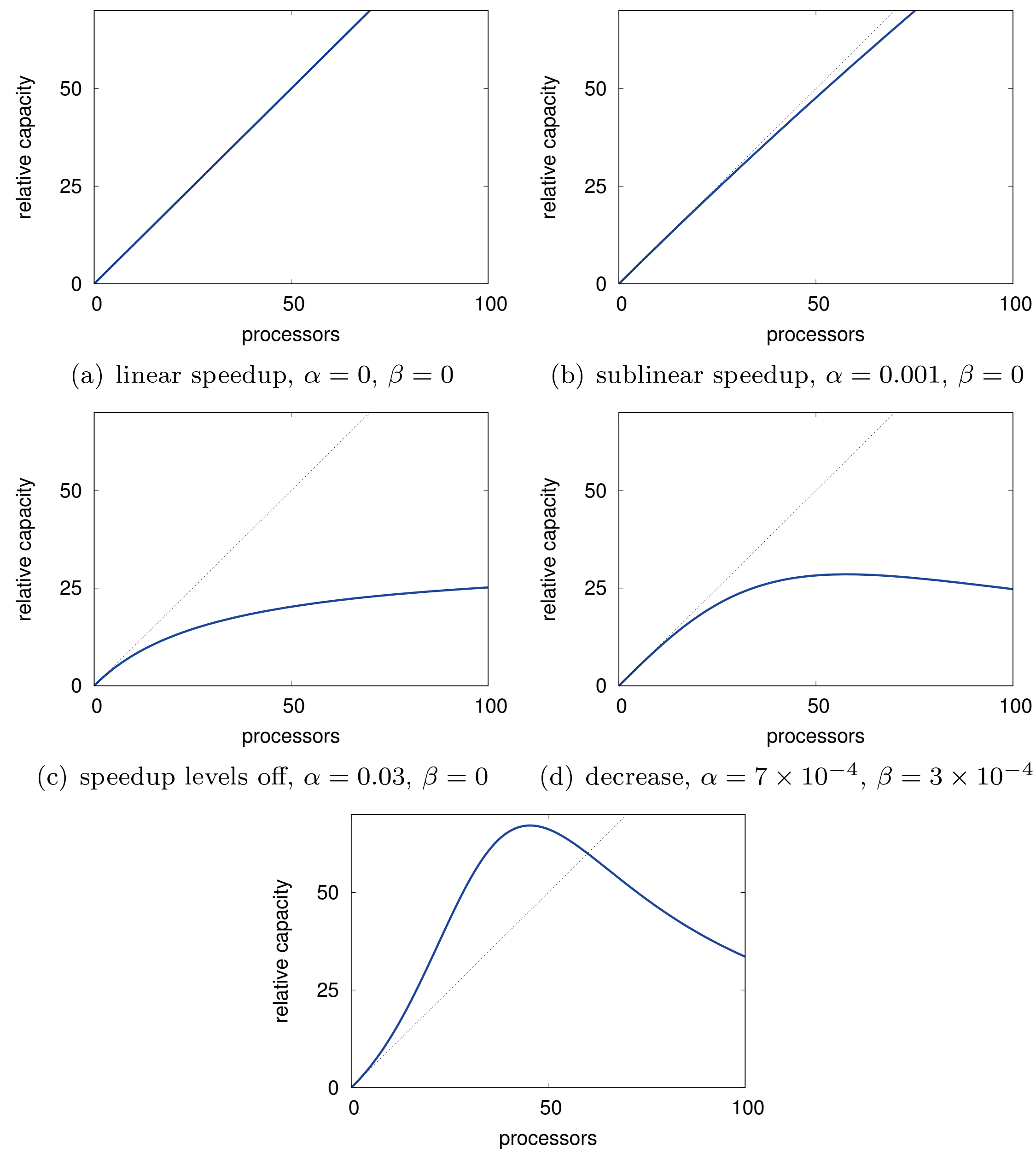
Out of the multi-robot domain, other systems are worth mentioning. Although they are much harder to measure, similar diagrams are also found for natural swarms, such as the hypothesis for per capita output in social wasps by Jeanne et al. [26]. The well-known 'fundamental diagram' of traffic flow is also similar but symmetric [27].

1.2 Universal Scalability Law

There is a model for parallel processing performance in distributed systems by Gunther [28]. He calls it the Universal Scalability Law (USL). For a relative capacity $R(N)$ (i.e., X_N/X_1, for throughput X_N achieved using N processors and throughput X_1 for one processor) he defines

$$R(N) = \frac{N}{1 + \alpha((N-1) + \beta N(N-1))} \ , \tag{2}$$

for a coefficient α that gives the degree of contention (inference) in the system and coefficient β that gives the lack of coherency in the distributed data. Contention occurs because resources are shared. Whenever the capacity of a shared resource is used completely and another process requests to use that resource, then the process has to wait. Contention increases with increasing system size, while keeping resources at the same capacity. Lack of coherency occurs because processes, to a certain extent, operate locally. For example, they have local changes in their caches that are not immediately communicated to all other processes. Maintaining coherency is costly and the costs increase with increasing system size.



(a) linear speedup, $\alpha = 0$, $\beta = 0$

(b) sublinear speedup, $\alpha = 0.001$, $\beta = 0$

(c) speedup levels off, $\alpha = 0.03$, $\beta = 0$

(d) decrease, $\alpha = 7 \times 10^{-4}$, $\beta = 3 \times 10^{-4}$

(e) superlinear, $\alpha = -0.03$, $\beta = 5 \times 10^{-4}$

Figure 2: Universal Scalability Law following Gunther [28], four standard situations and superlinear speedup [5] depending on parameters α (degree of contention) and β (lack of coherency).

Gunther identifies four qualitatively different situations:

a. If contention and lack of coherency are negligible, then we get "equal bang for the buck" and have a linear speedup ($\alpha = 0$, $\beta = 0$, Fig. 2a).

b. If there is a cost for sharing resources in the form of contention, then we have a sublinear speedup ($\alpha > 0$, $\beta = 0$, Fig. 2b).

c. If there is an increased negative influence due to contention, then the speedup clearly levels off ($\alpha \gg 0$, $\beta = 0$, Fig. 2c).

d. If in addition there is also an increased influence of incoherence then there exists a peak speedup and for bigger system sizes the speedup decreases ($\alpha \gg 0$, $\beta > 0$, Fig. 2d).

In the original work of Gunther [28], superlinear performance increases are basically not allowed. In a more recent work [5], superlinear speedups are discussed and negative contention coefficients $\alpha < 0$ are allowed now (see Fig. 2e). While contention $\alpha > 0$ refers to capacity consumption due to sublinear scalability, $\alpha < 0$ refers to a capacity boost due to superlinear scalability. In parallel computing, superlinear speedups can occur due to some interplay between problem size per computing unit and available memory. For example, if the problem can be divided into pieces that fit completely into a CPU's cache, then one can observe a considerable speedup. In swarm robotics, superlinear performance increases occur due to qualitatively different collaboration modes that are accessible with increasing swarm size as in the bucket brigade example [7, 8] or when assembled swarm robots cross a hole in a team.

In the context of swarm robotics we can interpret contention as interference between robots due to shared resources, such as an entrance to a base station or generally space. Following this interpretation, the collision avoidance behavior between robots can be understood as a waiting loop because the shared resource *space* is currently not available. That is intuitive and similar to an airplane flying a holding pattern because the resource *runway* is currently in use and should certainly not be shared. Incoherence, in turn, can be interpreted as inconsistencies or overhead due to limited communication of information or due to imperfect synchrony.

While Gunther assumes that there cannot be a system-wide deadlock situation due to contention (speedup monotonically increases with increasing α), that could occur in a swarm robotics system. For example, the swarm density could be too high, such that all robots permanently try to avoid collisions resulting in zero performance.

2 Unified interpretation across fields of research

Both the simple multi-robot performance model (eq. 1) and the Universal Scalability Law (eq. 2) are phenomenological macroscopic models, that is, they are not derived from elementary microscopic features that could be tracked back to concrete procedures and behaviors of robots and processing units. Hence, also their interpretation and specifically the interpretation of individual mathematical terms are abstract considerations that make the chosen function more plausible and understandable but they are also subject to speculation. For example, Gunther's assumption that coefficient α corresponds to contention is a widely applicable concept. However, his assumption that coefficient β corresponds to the lack of coherency is much more specific. Similarly, in the simple multi-robot model the assumed exponential increase of interference is a rather strong assumption. We follow that both models allow or even invite reinterpretations and have potential to be generalized. In the following, we make a number of assumptions of how this can be done, which are then verified in Sec. 3.

We argue that interacting entities in parallel computing, multi-robot systems, and networks are facing tradeoffs in three categories: (R) utilization of shared resources, (I) information flow, and (C) degree of collaboration. A typical system has to deal with several instances from one or more of these categories (e.g., two shared resources and one type of collaboration). They are usually mutually dependent, which is the cause of their complexity. For example, maximizing the utilization of a resource $r_1 \in R$ may be necessary to maximize a type of collaboration $c \in C$, but before r_1 is fully utilized another resource $r_2 \in R$ is already depleted causing overhead.

In Fig. 3 we give a schematic overview of our interpretation. We separate the interval of system size $N \in \{1, 2, \ldots\}$ into three regions. First, the region of underused resource, too little collaboration, and too little information flow (left-hand side in Fig. 3). Second, the region of optimally balanced tradeoffs corresponding to optimal achievable performance (middle part in Fig. 3). Third, the region of depleted resources, too much collaboration, and too intensive information exchange (right-hand side in Fig. 3).

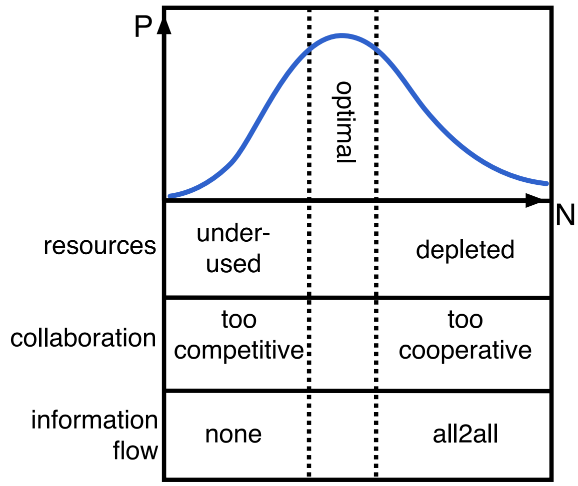


Figure 3: Schema of how to interpret scalability challenges across fields

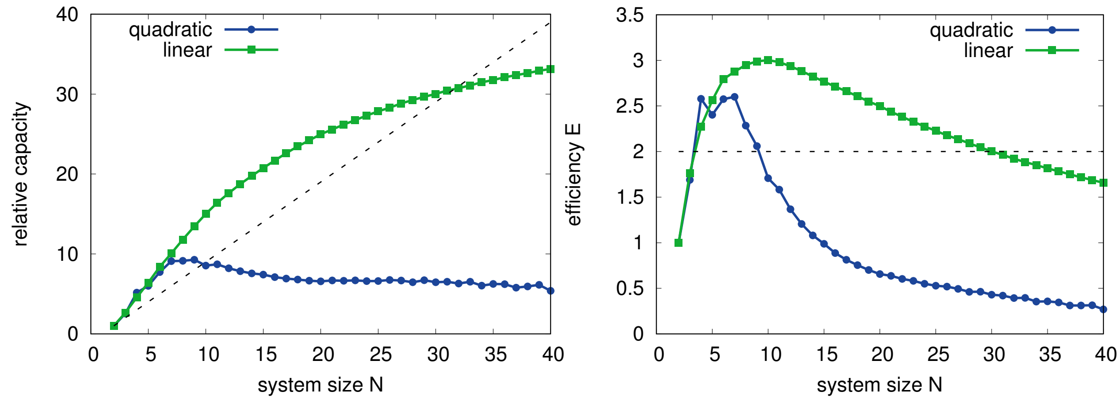
While it is intuitive to understand that depleting resources is disadvantageous and creates overhead (e.g., long queues, interference), it is maybe less intuitive to understand why there can be too much of collaboration or too much information flow. For the degree of collaboration, we distinguish between a competitive approach without or with little collaboration and a cooperative approach with a high degree of cooperation. In parallel computing, we can relate that to the distinction between competition parallelization (a parallel race to solve the same problem with different methods) and partitioning parallelization (standard approach to parallelization). Without collaboration all processing units work on their own and create a competitive environment. With a maximal degree of collaboration all processing units cooperate and may, hence, work on too similar potential solutions to the problem. Similarly, for the information flow we can avoid any exchange of information or share information all-to-all. Without sharing any information we may fall back to a purely competitive approach but in a multi-robot setting this could still be a useful parallelization, for example, of a cleaning task. With all-to-all communication we may loose diversity in the solution approaches and end up with a homogeneous approach where each processing unit basically processes the same workpackages.

3 Results

To verify and further investigate these interpretations, we study two example scenarios. In the stick pulling scenario, we investigate the tradeoff between properly exploiting shared resources while not depleting them. In the parallel optimization scenario, we investigate the tradeoff between intensifying collaboration but not loosing too much diversity.

3.1 Stick pulling: shared resources and collaboration

We investigate the tradeoff between not depleting resources while creating sufficiently many opportunities for collaboration in the well-known stick pulling task [6]. A group of robots equipped with grippers is supposed to collect sticks. The sticks are found standing upright in holes. The sticks are too long as if a single robot could remove them from the hole in one grip. Instead robots have to cooperate. A first robot does the first grip and removes the stick half-way. A second robot then grips the stick and removes it completely from the hole. The task is interesting as for an efficient solution a proper balance of the robot number relative to the number of sticks and the provided area is required as well as an optimized waiting time (for how long should the robot wait for support after the first grip).



(a) system performance

(b) efficiency (based on data as shown left)

Figure 4: Stick pulling experiment, relative capacity and efficiency $E = T_2/(NT_N)$ over system size N, averaged over 5000 repetitions

To make our point here, we restrict ourselves to a simplified, non-embodied model. We only model that $N \in \{2, 3, \ldots, 20\}$ robots are randomly distributed among $M = 20$ stick sites, wait there for a defined waiting time $w_{\text{atStick}} = 7$ [discrete time steps], and the commute time T between sites is also modeled. We scale the commute time T with the system size N in two variants. First, we scale it linearly

$$T_l(N) = N + \xi, \tag{3}$$

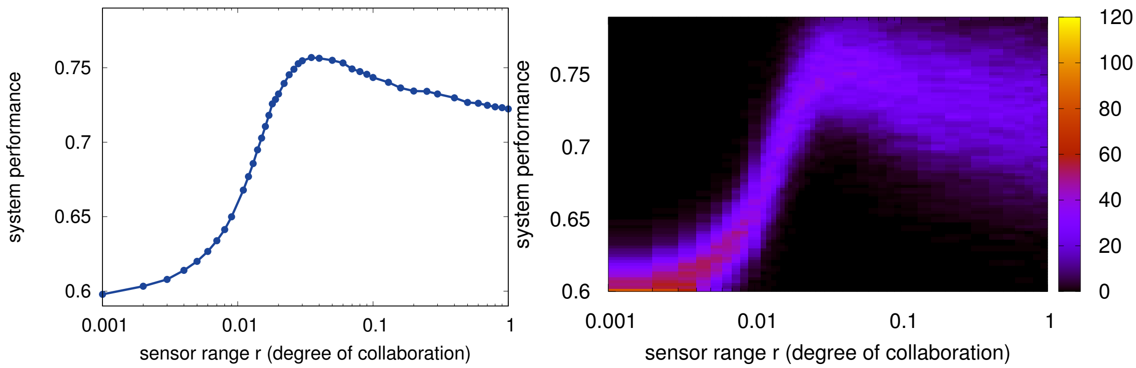
for a random number $\xi \in \{0, 1, 2\}$. Second, we scale it quadratically

$$T_q(N) = cN^2 + \xi, \tag{4}$$

with an arbitrary constant $c = 0.12$ to scale T_q to intervals comparable to T_l and again a random number $\xi \in \{0, 1, 2\}$. The underlying idea is that with increased system size there is more traffic, robots physically interfere, and have delays due to collision avoidance behaviors. The following simulations are separated in two sets where we either use T_l or T_q to calculate how long a robot has to travel from any stick site to any other stick site.

Each robot can be in one of $M + 1 = 21$ states. In state s_0 a robot is currently commuting. In state s_i with $i > 0$ a robot is currently positioned at stick site i and waits for help. In addition, each robot has a current waiting time w that represents for how long a robot has been waiting already. Once at least two robots meet at a stick site at the same time step, we say they instantaneously remove the stick, they immediately start to commute to another, randomly selected site, and the stick is put back ready to be removed again. System performance is measured in the total number of removed sticks over the full duration of an experiment (1000 time steps). Each experiment setting was repeated 5000 times.

The results are shown in Fig. 4. First, Fig. 4(a) gives the relative capacity, that is, the total number of pulled sticks normalized by the performance for system size $N = 2$, over system size N for linearly scaled commute time T_l (green) and quadratically scaled commute time T_q (blue; standard deviation for linear commute time: $N = 2$, 2.16; $N = 10$, 7.5; $N = 15$, 8.1; $N = 20$, 8.6; $N = 30$, 9.1; $N = 40$, 9.5; standard deviation for quadratic commute time: $N = 2$, 2.6; $N = 10$, 6.8; $N = 15$, 6.4; $N = 20$, 6.2; $N = 30$, 5.4; $N = 40$, 4.4). For the linear commute times, the system performance improves with increasing system size for all tested system sizes. For the quadratic commute times, the system performance decreases starting with $N = 8$. As expected, the quadratic scaling increases the commute times much faster (despite discount factor $c = 0.12$). The dashed line gives the linear scaling. The relative capacity is superlinear for $N < 10$ (quadratic commute times) and for $N < 32$ (linear). Fig. 4(b) gives the efficiency normalized with the mean execution time observed for system size $N = 2$. Both efficiencies decrease for too big system sizes. The efficiencies shown in Fig. 4(b) correspond to case **d.** in Gunther's USL ($\alpha \gg 0$, $\beta > 0$).



(a) system performance

(b) histogram of system performance

Figure 5: Parallel optimization experiment, system performance, note logarithmic scale on the horizontal axis (left: mean, right: histogram) over sensor range r, averaged over 1000 repetitions

3.2 Parallel optimization: network topologies and information flow

Here, we follow a network model described by Lazer and Friedman [29] but extend it to random geometric graphs instead of predefined network topologies and small-world networks. Random geometric graphs are more closely related to typical setups in multi-robot systems. Initially we place robots in a 2-d plane (i.e., a point process), which is here the unit square. Hence, each robot has a position x. We say, all robots have a given sensor range r. A considered robot has an edge to another robot if that robot is within range r, that is, for robot positions x_1 and x_2 we test the Euclidean distance $|x_1 - x_2| < r$.

The task is to solve an optimization problem in parallel. The problem is generated using Kauffman's so-called NK model [30]. That is a standard technique to generate test problems with rugged fitness landscapes, for example, in evolutionary computation [31]. For details about the optimization problem see Lazer and Friedman [29]. Each robot (or processing unit) could, in principle, try to solve the problem on its own. That is actually also what a robot needs to do if it happens to have no neighbors in the geometric graph. The idea is, however, that the robots cooperate and share information about the optimization problem. Neighboring robots can compare their current best solutions, the robot with the worse solution can replace it with the other robot's better solution, and continue to optimize the problem starting from there.

The network model iterates over the following procedure. Each of the $N = 100$ robots checks whether a neighbor has currently a better solution. If yes, it replaces its own current solution with the best solution of its neighbors. If not, the robot does a local search, that is, a brute force approach to improve its current solution by checking small changes of it. This is iterated for 20 time steps. We test different sensor ranges $r \in [0.001, 1]$ and each experiment setting is repeated 1000 times.

In Fig. 5 we give the results. In these plots we use a logarithmic scale for the horizontal axis, which does not change the shape qualitatively (e.g., in comparison to Fig. 1). Fig. 5(a) gives the mean performance (i.e., best solution in the multi-robot system) averaged over 1000 repetitions of the experiment setting. Fig. 5(b) gives a histogram of the same data set and indicates that there is rather little variance. The sensor range r determines how many neighbors a robot has in average. The number of neighbors determines how often a robot adapts solutions from other robots instead of doing a local search to optimize the problem. Having more neighbors helps to gather better solutions but if each robot has many neighbors the overall system reduces its potential for exploring the problem's search space.

4 Discussion and Conclusion

The stick pulling experiment clearly indicates the tradeoff between creating chances to collaborate and ensuring that resources are not depleted. Each stick site can be seen as a resource that needs to be populated, because if they are underused the system performance suffers. However, the resource 'space' between the sites is limited. Hence, we find an optimum system size that balances the use of stick sites and space.

The parallel optimization experiment indicates the tradeoff between collaborating while not loosing too much diversity. If there is no collaboration, each robot independently tries to optimize the problem. If each robot is connected to every other robot, then the search is not parallelized anymore but all robots investigate the same problem instances in parallel. There is clearly an optimum between sharing some information (a medium information flow through the system) and sharing too much information.

Gunther's interpretation of his Universal Scaling Law speaks of contention (i.e., overhead in sharing resources) and lack of coherence (e.g., as in cache hierarchies). While contention can be easily identified in the multi-robot setup (e.g., for linearly scaled commuting times in the stick pulling scenario, see Fig. 4(b)), a correspondence to 'lack of coherency' is difficult to be identified. Instead we see two contradicting uses of shared resources in the stick pulling scenario. One resource is supposed to be populated to increase profit (stick sites) but the other resource is already depleted and creates overheads (space). In the parallel optimization scenario, there is also no lack of coherence but instead a too intensive communication that then crucially reduces exploration in the system.

Superlinearity seems more frequent in multi-robot systems and swarm systems probably mainly due to physical effects. In tasks, such as collectively pulling a heavy object and passing a gap or a steep hill, one or a few robots basically achieve zero performance (they cannot pull the object at all due to friction, they can just not pass the gap or the hill) but once a certain threshold N_c of system size $N > N_c$ is reached the performance increases rapidly. Superlinearity as seen in the stick pulling scenario, however, is more subtle and less easily connected directly to such a single cause. Obviously it is the interplay of not underusing one resource while not depleting another.

As mentioned above, the generic swarm performance curve (Fig. 1) is observed frequently. Hence, we follow that the above described phenomena observed in the two investigated scenarios must also be frequent. It is encouraging to see that similar phenomena emerge in such different domains as multi-robot system, networks, and parallel computing. This is a clear indicator that universal models across all fields must exist.

References

[1] Ingham, A.G., Levinger, G., Graves, J., Peckham, V.: The Ringelmann effect: Studies of group size and group performance. Journal of Experimental Social Psychology **10**(4) (1974) 371–384

[2] Gustafson, J.L.: Fixed time, tiered memory, and superlinear speedup. In: Proceedings of the Fifth Distributed Memory Computing Conference (DMCC5). (1990) 1255–1260

[3] Helmbold, D.P., McDowell, C.E.: Modelling speedup (n) greater than n. IEEE Transactions on Parallel and Distributed Systems **1**(2) (1990) 250–256

[4] Faber, V., Lubeck, O.M., White Jr., A.B.: Superlinear speedup of an efficient sequential algorithm is not possible. Parallel Computing **3**(3) (1986) 259–260

[5] Gunther, N.J., Puglia, P., Tomasette, K.: Hadoop super-linear scalability: The perpetual motion of parallel performance. ACM Queue **13**(5) (2015) 46–55

[6] 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. Autonomous Robots **11** (2001) 149–171

[7] Lein, A., Vaughan, R.T.: Adaptive multi-robot bucket brigade foraging. Artificial Life **11** (2008) 337

[8] Pini, G., Brutschy, A., Birattari, M., Dorigo, M.: Interference reduction through task partitioning in a robotic swarm. In: Sixth International Conference on Informatics in Control, Automation and Robotics–ICINCO. (2009) 52–59

[9] Mondada, F., Bonani, M., Guignard, A., Magnenat, S., Studer, C., Floreano, D.: Superlinear physical performances in a SWARM-BOT. In Capcarrere, M.S., ed.: Proc. of the 8th European Conference on Artificial Life (ECAL). Volume 3630 of LNCS., Berlin, Germany, Springer (2005) 282–291

[10] Hamann, H.: Towards swarm calculus: Universal properties of swarm performance and collective decisions. In Dorigo, M., Birattari, M., Blum, C., Christensen, A.L., Engelbrecht, A.P., Groß, R., Stützle, T., eds.: Swarm Intelligence: 8th International Conference, ANTS 2012. Volume 7461 of LNCS., Berlin, Germany, Springer (2012) 168–179 **(best paper award)**.

[11] Hamann, H.: Towards swarm calculus: Urn models of collective decisions and universal properties of swarm performance. Swarm Intelligence **7**(2-3) (2013) 145–172

[12] Schneider-Fontán, M., Matarić, M.J.: A study of territoriality: The role of critical mass in adaptive task division. In Maes, P., Wilson, S.W., Matarić, M.J., eds.: From animals to animats IV, MIT Press (1996) 553–561

[13] Arkin, R.C., Balch, T., Nitz, E.: Communication of behavioral state in multi-agent retrieval tasks. In Book, W., Luh, J., eds.: IEEE Conference on Robotics and Automation. Volume 3., Los Alamitos, CA, IEEE Press (1993) 588–594

[14] Lerman, K., Galstyan, A.: Mathematical model of foraging in a group of robots: Effect of interference. Autonomous Robots **13** (2002) 127–141

[15] Goldberg, D., Matarić, M.J.: Interference as a tool for designing and evaluating multi-robot controllers. In Kuipers, B.J., Webber, B., eds.: Proc. of the Fourteenth National Conference on Artificial Intelligence (AAAI'97), Cambridge, MA, MIT Press (1997) 637–642

[16] Østergaard, E.H., Sukhatme, G.S., Matarić, M.J.: Emergent bucket brigading: a simple mechanisms for improving performance in multi-robot constrained-space foraging tasks. In André, E., Sen, S., Frasson, C., Müller, J.P., eds.: Proceedings of the fifth international conference on Autonomous agents (AGENTS'01), New York, NY, USA, ACM (2001) 29–35

[17] Beckers, R., Holland, O.E., Deneubourg, J.L.: From local actions to global tasks: Stigmergy and collective robotics. In: Artificial life IV. (1994) 189–197

[18] Lerman, K., Martinoli, A., Galstyan, A.: A review of probabilistic macroscopic models for swarm robotic systems. In Şahin, E., Spears, W.M., eds.: Swarm Robotics - SAB 2004 International Workshop. Volume 3342 of LNCS. Springer, Berlin, Germany (2005) 143–152

[19] Khaluf, Y., Birattari, M., Rammig, F.: Probabilistic analysis of long-term swarm performance under spatial interferences. In Dediu, A.H., Martín-Vide, C., Truthe, B., Vega-Rodríguez, M.A., eds.: Proc of Theory and Practice of Natural Computing, Berlin, Heidelberg, Springer (2013) 121–132

[20] Brutschy, A., Pini, G., Pinciroli, C., Birattari, M., Dorigo, M.: Self-organized task allocation to sequentially interdependent tasks in swarm robotics. Autonomous Agents and Multi-Agent Systems **28**(1) (2014) 101–125

[21] Hamann, H., Schmickl, T., Wörn, H., Crailsheim, K.: Analysis of emergent symmetry breaking in collective decision making. Neural Computing & Applications **21**(2) (March 2012) 207–218

[22] Nembrini, J., Winfield, A.F.T., Melhuish, C.: Minimalist coherent swarming of wireless networked autonomous mobile robots. In Hallam, B., Floreano, D., Hallam, J., Hayes, G., Meyer, J.A., eds.: Proceedings of the seventh international conference on simulation of adaptive behavior on From animals to animats, Cambridge, MA, USA, MIT Press (2002) 373–382

[23] 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, Los Alamitos, CA, IEEE Press (2007) 45–52

[24] Meister, T., Thenius, R., Kengyel, D., Schmickl, T.: Cooperation of two different swarms controlled by BEECLUST algorithm. In: Mathematical Models for the Living Systems and Life Sciences (ECAL). (2013) 1124–1125

[25] Hamann, H.: Modeling and investigation of robot swarms. Master's thesis, University of Stuttgart, Germany (2006)

[26] Jeanne, R.L., Nordheim, E.V.: Productivity in a social wasp: per capita output increases with swarm size. Behavioral Ecology **7**(1) (1996) 43–48

[27] Lighthill, M.J., Whitham, G.B.: On kinematic waves. II. A theory of traffic flow on long crowded roads. Proceedings of the Royal Society of London **A229**(1178) (May 1955) 317–345

[28] Gunther, N.J.: A simple capacity model of massively parallel transaction systems. In: CMG National Conf. (1993) 1035–1044

[29] Lazer, D., Friedman, A.: The network structure of exploration and exploitation. Administrative Science Quarterly **52** (2007) 667–694

[30] Kauffman, S.A., Levin, S.: Towards a general theory of adaptive walks on rugged landscapes. Journal of Theoretical Biology **128**(1) (1987) 11–45

[31] Eiben, Á.E., Smith, J.E.: Introduction to Evolutionary Computing. Natural Computing Series. Springer (2003)