Multi-Objective Collective Search and Movement-based Metrics in Swarm Robotics

Sebastian Mai
Institute for Intelligent Cooperating Systems
Otto-von-Guericke-University, Magdeburg
sebastian.mai@ovgu.de

Heiner Zille
Institute for Intelligent Cooperating Systems
Otto-von-Guericke-University, Magdeburg
heiner.zille@ovgu.de

Christoph Steup
Institute for Intelligent Cooperating Systems
Otto-von-Guericke-University, Magdeburg
christoph.steup@ovgu.de

Sanaz Mostaghim
Institute for Intelligent Cooperating Systems
Otto-von-Guericke-University, Magdeburg
sanaz.mostaghim@ovgu.de

ABSTRACT

Particle Swarm Optimization is a well researched meta-heuristic for single- and multi-objective problems. It is based on the movement of particles, which enables its application to collective search in robotic applications. However, in the robotic context some assumptions regarding performance measurement of PSO-algorithms do not apply. Concerning energy and time, while reading a sensor is usually cheap, the movement is costly in robotic applications. Traditional methods do not take into account any cost metrics associated with the actual movement of the particles. Therefore, new metrics are required to understand how well a PSO-algorithm can perform as a collective search mechanism. This article proposes two metrics, the Normalized Movement Energy Cost and Normalized Movement Time Cost that enable researchers to analyze an algorithm's performance not just regarding the obtained solution quality, but also with respect to movement time and energy costs.

CCS CONCEPTS

• **Theory of computation → Bio-inspired optimization**; **Evolutionary algorithms**; • **Computing methodologies → Multi-agent systems**; *Robotic planning*; Multi-agent planning;

KEYWORDS

Multi-Objective PSO, Collective Search, Swarm Robotics, Movement Cost, Multi-Objective Optimization

ACM Reference Format:
Sebastian Mai, Heiner Zille, Christoph Steup, and Sanaz Mostaghim. 2019. Multi-Objective Collective Search and Movement-based Metrics in Swarm Robotics. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3321967

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3321967

1 PSO IN ROBOTIC SEARCH

Movement and positions of particles are explicitly modelled in the Particle Swarm Optimization (PSO) search heuristic. This property makes PSO a suitable mechanism for robotic applications (especially robotic search). In contrast to other population-based meta-heuristics like Evolutionary Algorithms, the search mechanism of PSO methods are inspired by the movement of swarms in nature. Furthermore, PSO methods can work in a decentralized manner and are robust and well proven in optimization [2]. However, a drawback of the existing PSO algorithms is that some of the assumptions made in optimization do not hold in the context of moving robotic swarms. In particular, the cost of the algorithm in optimization is usually associated to the computational cost of the function evaluations. In many applications however, reading a sensor-value is cheap in terms of energy-expenditure [3] and the used time [5]. In those applications the number of function evaluations is nearly meaningless as the time and energy cost associated to movement of the particles (i.e. robots) vastly exceeds the cost of the measurement [5]. The evaluation of the performance of PSO algorithms is usually done within the actual robotic system or a simulated robotic system with the focus on the performance in terms application specific goals, without considering additional costs of movements associated with the PSO search mechanism. By using an explicit metric to measure movement cost researchers can compare algorithms by the occurring cost and have a means to select an appropriate algorithm. Alternatively, the metric can be used to optimize the particle movement during runtime.

2 MOVEMENT ASSOCIATED COST-METRICS

Two different movement costs that often occur are those associated with *time* [5] and *energy* [1, 3] consumption. To approximate these costs we define two metrics in the following sections, namely the *Normalized Movement Energy Cost* (*NMEC*) and the *Normalized Movement Time Cost* (*NMTC*). The metrics can be used to quantify movement cost in a single algorithm run, time interval or update step. When a PSO algorithm is used as a control mechanism, the solution space in the (multi-objective) optimization problem corresponds to the movement space of the robot. Most robots will move either in 2D- or 3D-space, so two and three dimensional test problems are most relevant for an evaluation of algorithms with the movement cost metrics. In most applications the orientation of the robot is just as important as the position. The combined information of position and orientation is called the pose of a robot. A complete 3D-pose is uniquely defined by six variables. In addition to the space a robot moves in, the configuration of certain components of the robot is relevant to the correct execution of a task. This configuration space is especially relevant to the operation of robotic arms. However, the distance-based metrics are not as meaningful for higher dimensional robotic applications, as the movement cost to get from one point to another in the state space usually follows very complex, non-linear relationships. The cost for that movement can not be accurately represented by measuring the euclidean distance of the points any more. Still using higher dimensional test problems may yield useful insights for robotic applications. The used notation and the two metrics are defined in the following: N: size of the population, $\vec{x}_i^{(t)}$: position of individual i at iteration t, g: maximum number of generations.

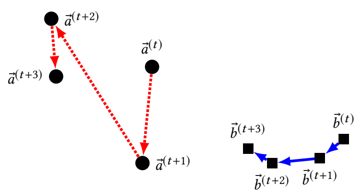

Figure 1: Particle a has a longer path than particle b. The relative motion between t and $t+3$ is the same for both particles.

2.1 Normalized Movement Energy Cost

We assume that energy expenditure of each particle is roughly proportional to the travelled distance. Thus we define the energy-cost of the movement during an algorithm run via the normalized sum of the total traveled distances. More precisely, we first calculate the sum of the travelled distances in each generation for all particles. To compare experiments with different swarm sizes and different lengths, we then normalized this sum by dividing it by the number of particles as well as the number of generations. The normalization enables the approximation of the Energy cost of a single algorithm step. Afterwards, it can be compared to the average cost of the run or a specific time interval. We call this value *Normalized Movement Energy Cost* (*NMEC*), as shown in Equation (1).

$$NMEC = \frac{\sum_t \sum_i \|\vec{x}_i^{(t+1)} - \vec{x}_i^{(t)}\|}{N \cdot g} \tag{1}$$

This metric is especially relevant for robots where the energy cost of movement is high in comparison to the energy that is continuously spend for computation or measurement and the energy supply is limited. Effectively the *NMEC* corresponds to the average distance each particle travels per time-step. In Fig. 1 particle a clearly travels further than particle b, hence $NMEC(a) > NMEC(b)$. The *NMEC* is a very intuitive measure for movement cost and may be the more important of the two metrics.

2.2 Normalized Movement Time Cost

The usual implementation of a PSO computes all new locations of the particles, then moves all the particles to the new locations, and then continues with the next iteration. In a robotic implementation of PSO all robots need to arrive at their new location before the next PSO-generation starts, because most PSO algorithms work in a synchronous manner. The synchronisation cost can be illustrated by the particles in Fig. 1: In case both particles a and b belong to the same population particle b always arrives at the new location before particle a. As a consequence, particle b would spend a significant time waiting before the next generation of the PSO algorithm. This synchronization step blocks the progression of the whole swarm, as it depends on the arrival of the last robot. Hence, the time taken for one generation of the algorithm is associated with the maximum distance traveled by the particles in that iteration. Based on this assumption, we define the Normalized Movement Time Cost (*NMTC*) as the sum of longest distances travelled by any particle in the swarm in each generation. This is equivalent to the average maximum distance traveled per generation, as shown in Equation (2). Note that the synchronisation time can be saved by using asynchronous updates, in this case the *NMTC* looses its meaning (at least as a measure for time cost).

$$NMTC = \frac{\sum_t \max_i \|\vec{x}_i^{(t+1)} - \vec{x}_i^{(t)}\|}{g} \tag{2}$$

3 DISCUSSION

The two proposed metrics are idealized models of a real robot's energy or time consumption. Due to the multitude of different robot modalities the *NMTC* and *NMEC* can not be interpreted as an accurate measure for energy or time cost (an accurate movement model for a specific robot is likely much more complex and less interpretable). However, both measures can give valuable insight into the movement behaviour of an algorithm that translates to the performance of a robot using the behaviour. The metrics can either be used to compare different algorithms in terms of their movement efficiency, to choose an appropriate algorithm for a given application, or to optimize the movement during runtime. For the latter use case we were able to modify SMPSO [4] to save large amounts of movement cost without losses in solution quality in terms of function evaluations with very simple adaptations (swapping particle indices to minimize movement cost of the current time step). We assume that a lot can be learned from understanding the movement behaviour in PSO algorithms. On the one hand movement is a major cost factor in robotic applications, on the other hand PSO uses many random movements and measuring the movements could be beneficial to gain a deeper insight into PSO performance in certain problems.

REFERENCES

[1] Palina Bartashevich, Doreen Koerte, and Sanaz Mostaghim. 2017. Energy-saving Decision Making for Aerial Swarms: PSO-based Navigation in Vector Fields. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, Honolulu, HI, USA, 1–8.

[2] James M Hereford, Michael Siebold, and Shannon Nichols. 2007. Using the particle swarm optimization algorithm for robotic search applications. In *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE*. IEEE, Honolulu, HI, USA, 53–59.

[3] Sanaz Mostaghim, Christoph Steup, and Fabian Witt. 2016. Energy Aware Particle Swarm Optimization as Search Mechanism for Aerial Micro-Robots. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, Athens,Greece, 1–7.

[4] Antonio J. Nebro, Juan J. Durillo, José Garcia-Nieto, Carlos A. Coello, Francisco Luna, and Enrique Alba. 2009. SMPSO: A New PSO-based Metaheuristic for Multi-Objective Optimization. In *2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making(MCDM)*. IEEE, Nashville,TI,USA, 66–73.

[5] Jim Pugh and Alcherio Martinoli. 2007. Inspiring and Modeling Multi-Robot Search with Particle Swarm Optimization. In *2007 IEEE Swarm Intelligence Symposium*. IEEE, Honolulu, HI, USA, 332–339.