

# Performance of insect-inspired self-organized task allocation mechanisms

Franziska Kluegl<sup>\*</sup>

<sup>\*</sup>Artificial Intelligence, University of Würzburg  
Am Hubland, D 97074 Würzburg  
kluegl@informatik.uni-wuerzburg.de

Anna Dornhaus<sup>†</sup>

<sup>†</sup>Ecology and Evolutionary Biology, University of Arizona  
P.O. Box 210088, Tucson, AZ 85721s  
dornhaus@email.arizona.edu

## Abstract

Who conducts which task in a group or team? This is one of the central questions for any form of division of labour, in social insects as well as in multi-agent systems. Real social insect division of labour achieves the efficiency, solution quality, robustness or flexibility that artificial systems aim for. The “superorganism” of ants or bees still is able to maintain its function even if complete task groups are taken away. Through evolution, the decision rules of every single individual ant are optimised in its environmental context and in coordination with the others. However, it is not trivial to simply transfer task allocation mechanisms found in social insects to engineering multi-agent systems like swarm robots. One obstacle is that there are several models of insect task allocation, yet from biological point of view it is not clear which social insect species applies what mechanism and why. This was the motivation behind our study, where we compare the performance of different task allocation mechanisms under different conditions and constraints.

Performance is tested in an abstract model where a number of  $A$  agents have to perform  $T$  tasks. Every task  $t$  is characterised by one of  $Y$  types and a dynamic value  $r_t$  that increases in every timestep and denotes the urgency of the task  $t$ . Execution of task  $t$  by agent  $a$  means that  $a$  reduces  $r_t$  by an amount dependent on ability,  $f_a$ . In the model, the basic decision mechanism is threshold-based, meaning that tasks are executed if their urgency exceeds a specific threshold. Each agent has distinct thresholds for different task types. Based on different configurations of this threshold vector we could model caste-like systems (each agent only executes one task type), purely stimulus-based systems (no threshold is active) or different randomly distributed thresholds. We varied number of agents, their abilities, number of tasks and number of task types. We compared the performance of the different threshold configurations with completely random and optimal centralised task allocation. Also, additional aspects of decision making have been tested that show how constraints on the individuals reasoning capabilities affect the performance of these task allocation mechanisms. For example, we have varied the task selection mechanism such that agents can either only assess urgency of one task per evaluation cycle or assess all tasks that they perceive; we have also varied whether they can take the current urgency or its derivatives into account when selecting the next task. Analogous tests were made for the perception abilities concerning when to terminate task execution.

Overall, we could show that of the distributed task selection algorithms, random choice of the next task works surprisingly well in many situations. A fixed allocation (a ‘caste’ system) performs worse than the alternatives in most cases. If agents are able to compare several task objects at no cost, variable task thresholds in different agents improve collective performance as they improve selectivity and concentration on the most urgent tasks.