

# Evolving team compositions by agent swapping

Paweł Lichocki, *Member, IEEE*, Steffen Wischmann, Laurent Keller, and Dario Floreano, *Senior Member, IEEE*

**Abstract**—Optimizing collective behavior in multiagent systems requires algorithms to find not only appropriate individual behaviors but also a suitable composition of agents within a team. Over the last two decades, evolutionary methods have been shown to be a promising approach for the design of agents and their compositions into teams. The choice of a crossover operator that facilitates the evolution of optimal team composition is recognized to be crucial, but so far it has never been thoroughly quantified. Here we highlight the limitations of two different crossover operators that exchange entire agents between teams: restricted agent swapping that exchanges only corresponding agents between teams and free agent swapping that allows an arbitrary exchange of agents. Our results show that restricted agent swapping suffers from premature convergence, whereas free agent swapping entails insufficient convergence. Consequently, in both cases the exploration and exploitation aspects of the evolutionary algorithm are not well balanced resulting in the evolution of suboptimal team compositions. To overcome this problem we propose to combine the two methods. Our approach first applies free agent swapping to explore the search space and then restricted agent swapping to exploit it. This mixed approach turns out to be a much more efficient strategy for the evolution of team compositions compared to either strategy alone. Our results suggest that such a mixed agent swapping algorithm should always be preferred whenever the optimal composition of individuals in a multiagent system is unknown.

**Index Terms**—Multiagent systems, evolutionary computation, team optimization, team composition, cooperation, crossover.

## I. INTRODUCTION

THE OPTIMIZATION of collective behavior displayed by teams of agents plays a crucial role in an increasing number of applications [1], [2], spanning from software agents (e.g., [3]–[5]) to robotics (e.g., [6]–[8]). Evolutionary computation has been advocated as an effective and promising strategy in this domain [9], [10]. An important question in this respect is the composition of the teams of agents. All agents from one team may either use the same control algorithm (genetically homogenous teams) or employ different ones (genetically heterogenous teams) [11]–[13]. Evolving homogenous teams does not differ conceptually from evolving single agents, because in both cases only one control algorithm is discovered [14]. In contrast, with heterogeneous teams a set of distinct control algorithms must be optimized simultaneously.

PL, SW and DF are with the Laboratory of Intelligent Systems (LIS), Ecole Polytechnique Fédérale de Lausanne (EPFL), Station 11, CH-1015 Lausanne, Switzerland, e-mail: pawel.lichocki@epfl.ch, dario.floreano@epfl.ch.

SW and LK are with Department of Ecology and Evolution (DEE), Biophore, University of Lausanne (UNIL), Dorigny, 1015 Lausanne, Switzerland

This work was supported by the Swiss National Science Foundation (grant number K-23K0-117914).

Manuscript received Month XX, XXXX; revised Month XX, XXXX.

Copyright (c) 2012 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

TABLE I

A SAMPLE OF APPROACHES IN THE EVOLUTION OF HETEROGENEOUS TEAMS USING INDIVIDUAL ENCODING AND TEAM ENCODING.

Individual encoding	
Agent grouping	References
(A) One team	[16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26]
(B) Many teams	[13], [27], [28], [29], [30], [31], [32], [33]
(C) Sub-populations	[11], [31], [34], [35], [36], [37], [38]
Team encoding	
Crossover	References
(D) Restricted agent recombination	[11], [14], [15], [16], [28], [33], [39], [40], [41], [42], [43], [44], [45], [46]
(E) Free agent recombination	[28], [40], [43], [46], [47]
(F) Restricted agent swapping	[39], [42]
(G) Free agent swapping	-

Consequently, the challenge is not only to find the optimal agents, but also the optimal composition of agents within a team [13]. Heterogenous teams are of growing interest in the evolutionary community, because they are expected to perform better than homogeneous teams in problems that require task specialization [12], [13], [15].

In heterogeneous team evolution, two genetic encodings can be used: (1) individual encoding, where a genome represents one agent and (2) team encoding, where a genome represents a whole team. Consequently, there are differences in an evolutionary algorithm used with individual and team encoding (Fig. 1). With individual encoding, one must decide on a method of grouping agents into temporary teams for purpose of performance evaluation (Fig. 1.A). With team encoding, one must choose a crossover operator that exchanges genetic material between teams (i.e., swap agents), and not only between agents (Fig. 1.B). Both issues are facets of the same challenge: How to (re)compose agents in teams, in order to facilitate the evolutionary search. This question has already been studied for individual encoding (Table I.A-C) but scarcely addressed for team encoding, although the body of work using team encoding is rich (Table I.D-G).

In team encoding a single genotype encodes the entire team, which makes it decomposable into parts corresponding to the agents. Thus, a crossover operator may exchange genetic material on two levels [14]. First, the crossover can recombine the genetic material between agents from the parenting teams. We refer to this process as *agent recombination* (Table I.D,E and Fig. 2, top row). Second, the crossover can swap entire agents between the parenting teams. We refer to this process as *agent swapping* (Table I.F,G and Fig. 2, bottom row). In contrast to agent recombination, agent swapping does not exchange genetic material between the agents. Consequently,

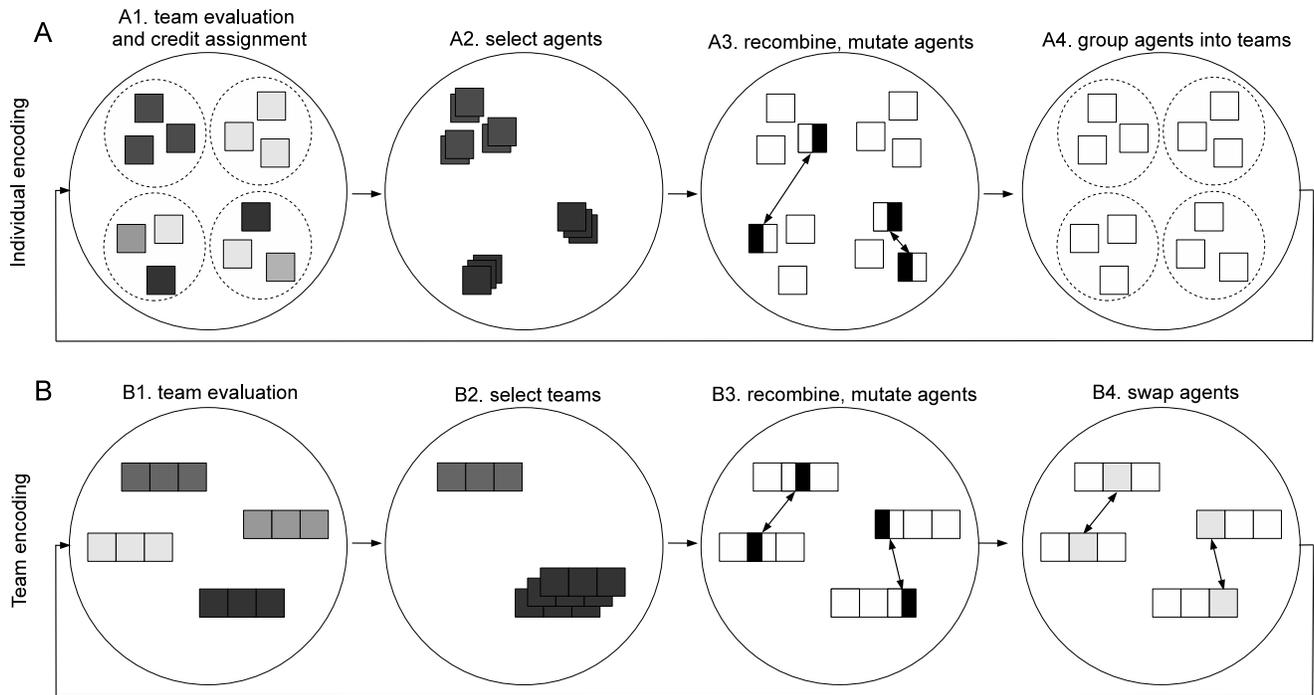


Fig. 1. The evolutionary algorithm cycle in the evolution of heterogeneous teams using (A) individual encoding and (B) team encoding. (A) Individual encoding: the evolutionary algorithm operates on a population of genotypes, each encoding one agent (depicted as squares). (A1) The agents are grouped into teams and their performance is evaluated together. Each agent must be assigned with a fitness value (the credit assignment problem [12], [13], [48]). A straightforward way of addressing this problem is to distribute the team's performance score equally among the team members (top two teams in A1). This is known as global reward [12], or team-level selection [13]. Alternatively, the agents can be assigned with different fitness values, proportional to their personal contribution into the team's performance score (bottom two teams in A1). This is known as local reward [12], or individual-level selection [13]. For simplicity, global and local rewards are illustrated together in one population, but typically only one kind of reward is used with all teams from the population. Next, (A2) the algorithm proceeds with the selection of the agents accordingly to their fitness values. In (A1) a darker color denotes higher fitness values, which translates into more copies of fitter agents (A2). Afterwards, (A3) the agents' genotypes are recombined or mutated. Finally, (A4) the agents are grouped into new teams before the performance evaluation and the algorithm starts over with (A1). (B) Team encoding: the evolutionary algorithm operates on a population of genotypes, each encoding one entire team (a square depicts a part of the team's genome corresponding to one agent). Consequently, there is no need for an extra step of grouping agents into teams and the algorithm begins at once with (B1) the evaluation of the teams' performance. There is no credit assignment problem, because team encoding implies that all agents in the team share the same fitness. Next, (B2) the teams are selected according to their fitness values. In (B1) a darker color denotes higher fitness values, which translates into more copies of fitter teams (B2). Then, (B3) the team's genotypes are subject to mutation or agent recombination. Finally, (B4) entire agents may be swapped between the teams. This concludes the cycle and the algorithm starts over with (B1). Note that swapping agents between teams in team encoding conceptually corresponds to grouping of agents into teams in individual encoding.

the purpose of agent recombination is to discover “good” agents, and the purpose of agent swapping is to discover “good” team compositions. In addition, one may consider a team to be an ordered sequence of agents. In such a case, the crossover may be *restricted* to act only on agents on corresponding positions in the parenting teams (Table I.D,F and Fig. 2, left). Or, it may be *free* to act on any agents from the parenting teams (Table I.E,G and Fig. 2, right).

In contrast to agent recombination (see, e.g., [28], [43], [46]), no attempts have been made to quantify the efficiency of agent swapping in the evolution of teams [12]. In particular, it has not been tested if and why using agent swapping leads, or not, to the evolution of optimal team compositions. Here, we experimentally compare restricted agent swapping (RAS) and free agent swapping (FAS) in a problem of finding the optimal team composition. We consider multiple agents that need to divide the labor in order to achieve top performance, i.e., the optimal team is composed of distinct groups of genetically identical agents. We focus on team encoding, which assumes team level of selection (Fig. 1) that has been advocated as an efficient strategy in the optimization of teams [13].

Consequently, agent interactions and their impact on individual selection pressures are out of scope of this study. Also, we focus on the evolution of team compositions, and not on the evolution of agents themselves. Thus, we mainly consider large teams of agents having small genomes, i.e., a valid single agent is relatively easy to evolve. We further elaborate on these assumptions in the Discussion section.

We highlight the limitations of both RAS and FAS and explain the conditions under which they fail to evolve teams displaying the optimal composition. These limitations are opposite to each other. With RAS the evolutionary algorithm suffers from premature convergence of the population, whereas with FAS it suffers from insufficient convergence of the population. Consequently, in both cases the exploration and exploitation aspects of the evolutionary algorithm are not well balanced. To overcome this problem we propose to combine the two methods, i.e., first use FAS to explore the search space, and then use RAS to exploit it. This mixed approach proves to be a more efficient strategy in the evolution of team compositions than restricted or free agent swapping alone. Finally, we also validated RAS and FAS on a problem of optimizing

decentralized controllers for task allocation and discussed our results in the context of other real-life applications.

## II. BACKGROUND

The differentiation into individual and team encoding resembles an old discussion on evolving rule-based systems [49]. In the approach taken by De Jong, dubbed “the Pitt approach”, a single individual encoded the entire rule set. In contrast, in the approach taken by Holland, dubbed “the Michigan approach”, a single individual encoded just a single decision rule, and the entire population corresponded to the rule set (see [49] and references therein for more details).

The approaches using individual encoding can be classified into three categories, according to how the agents are grouped into teams for the purpose of the performance evaluation. With the “one team” approach, all agents from the population are evaluated together, i.e., they compose one team (Table I.A). This method is often used with a continuously updated gene-pool (“steady-state evolution”). With the “many teams” approach, the agents are randomly grouped into many teams (Table I.B). This method is often used with separate gene-pools for subsequent generations (“generational evolution”). With the “sub-populations” approach there are separate sub-populations of agents (Table I.C). An individual is evaluated by teaming it up with individuals from other subpopulations. This method is known as cooperative co-evolution.

In individual encoding the teams are created ad hoc for the purpose of performance evaluation. Consequently, there are no genetic operators applied at the team level. In contrast, in team encoding we distinguish four qualitatively different types of crossover operators that exchange genetic material between the teams (Fig. 2). The crossover either *recombines* the agents from the parenting teams (Table I.D,E) or it *swaps* entire agents between the parenting teams (Table I.F,G). In addition, the crossover is either *restricted* to act only on agents on corresponding positions in the parenting teams (Table I.D,F), or it is *free* to act on any agents from the parenting teams (Table I.E,G). Note that agent swapping is in fact a special case of agent recombination, where the recombination points are always chosen at the beginning (or at the end) of the two genotypes’ parts that encode the parenting agents.

The concept of restricted (also called fixed in [28]) and free (also called unfixed in [28] and interpositional in [40]) agent recombination was introduced independently by Luke and Spector [43] and by Haynes and Sen [45]. Agent recombination was called inner crossover in [14]. Restricted agent swapping (RAS), called team transformation in [39], was introduced by Luke et al. [42] and by Andre and Teller [39]. Free agent swapping (FAS) has not been studied directly yet (Table I.G). Note that evolutionary algorithms with individual encoding, random grouping of agents into many teams and global reward yield high resemblance to team encoding with free agent swapping (Fig. 1). This includes some of the work referenced in Table I.B, i.e., [13], [28], [30]–[33].

Haynes and Sen [45] noted that RAS may be implemented as *uniform* crossover (i.e., swapping bits on corresponding positions between two genotypes with some probability), with

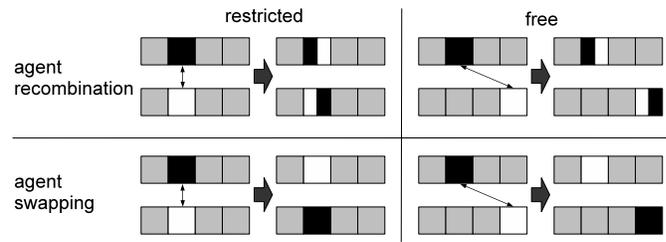


Fig. 2. Applying four crossover operators on the genotype of two teams of four agents each. Black and white segments of teams’ genotypes correspond to agents taking part in the crossover. Grey segments of teams’ genotypes correspond to agents not taking part in the crossover. The crossover can either *recombine* the genetic material from two agents from parenting teams (top row), or *swap* entire agents between parenting teams (bottom row). In addition, the crossover may be *restricted* to act only on agents on corresponding positions in parenting teams (left column). Or, it may be *free* to act on any agents from parenting teams (right column).

the difference that instead of bits it swaps entire agents. Here, we analogously note that FAS may be implemented as *shuffle-uniform* crossover (i.e., swapping bits on shuffled positions between two genotypes with some probability), with the difference that instead of bits it swaps entire agents. The family of uniform crossovers was introduced by Syswerda [50] and analytically studied by Eshelman et al. [51] and De Jong and Spears [52], but not in the context of team evolution. In addition, Miconi [14] and Nakashima et al. [53] used a simple 1-point or 2-point crossover to recombine the teams’ genotypes. This approach leans itself to the category of restricted crossovers. During the crossover, it usually swaps entire agents between teams, with the exception of the agents that happen to be placed on the crossover cutting points. The 1-point and 2-point crossovers, and generally  $n$ -point crossover, have been shown to have less exploratory power than uniform crossover [51].

A few attempts have been made to quantify the efficiency of agent recombination for the evolution of teams, but the studies are inconclusive [12]. Some authors advocate using restricted agent recombination [43], [46] and some are proponents of free agent recombination [40]. The problem of evolving team compositions has been addressed only for genetic programming by Hara and Nagao [41] (automatically defined groups) and by Bongard [15] (the Legion system). In these two approaches both the agents and their partitioning into separate sub-teams are evolved together. These approaches are promising, but designed for the tree representation typical for genetic programming. In addition, authors considered only a small number of evolving sub-teams (maximum number of sub-teams: 6 in [41] and 3 in [15]). Importantly, there is no study that quantifies the efficiency of agent swapping for the evolution of teams.

Operators similar to RAS and FAS have been proposed in different fields of evolutionary computation. Agent swapping is equivalent to root crossover in genetic programming, which swaps whole trees instead of subtrees [42]. Also in genetic programming, Koza studied operators altering the architecture of a multi-part program, which he used with automatic defined functions [54]. These operators included branch duplication

and deletion, which in the context of evolving team composition would translate into agent deletion and duplication, respectively. In gene expression programming, Ferreira proposed gene recombination and gene transposition [55]. Gene recombination swaps entire genes between the parenting genotypes, and thus is equivalent to restricted agent swapping. Gene transposition overwrites one gene with a copy of another gene, and thus is equivalent to agent deletion and duplication performed jointly. Finally, in the field of evolutionary strategies, Sebagg and Schoenauer proposed mutation by imitation [56]. With imitation the probability of an allele's mutation depends on whether the allele is the same or different as the corresponding alleles in some of the best/worst individuals in the population. Consequently, mutation by imitation might be considered a sort of restricted recombination.

### III. EXPERIMENTAL METHOD

#### A. Restricted and free agent swapping

We evolve teams of agents using team encoding and study restricted agent swapping (RAS, Fig. 3, left) and free agent swapping (FAS, Fig. 3, right). We implement RAS as a uniform crossover [50], which exchanges the  $i$ th agent from the first team with the  $i$ th agent from the second team, with probability  $p$  for each agent. In an additional experiment we tested RAS implemented as a 1-point and 2-point crossover [Appendix A].

We implement FAS as a shuffle-uniform crossover [50], which exchanges the  $i$ th agent from the first team with the  $S(i)$ th agent from the second team, with probability  $p$  for each agent.  $S$  is a random permutation of integers from 1 to  $M$ , where  $M$  is the number of agents in the team. RAS may be considered a specific case of FAS where the sequence  $S$  is set to 1, 2, ...,  $M$ . For sake of simplicity we consider a situation with only two parenting teams, but the operators can scale to any number of parenting teams.

In addition, we theoretically investigate the connections between RAS, FAS and other operators that alter the team composition inspired by architecture-changing operators [54] used in genetic programming and gene expression programming [Appendix H].

#### B. Problem formulation

We experimentally compare RAS and FAS in the evolution of agent teams facing a problem of finding the optimal team composition (i.e., the division of a team into groups of identical agents). Our aim is to mimic a situation when a team needs to display a certain composition of agents in order to achieve top performance [57]. For example, multiple robots may be more efficient if they compose distinct groups, each focusing on a different task [58]. In biology, this process is known as division of labor (see [59], [60] and [61] for comprehensive reviews). For example, in many species of honey bees and ants, some workers forage for food, others care for the brood, others perform maintenance work in the nest, etc. [62]–[65].

We consider a team consisting of  $M$  agents. There are  $D$  distinct types of agents. Thus, there are overall  $D^M$  different

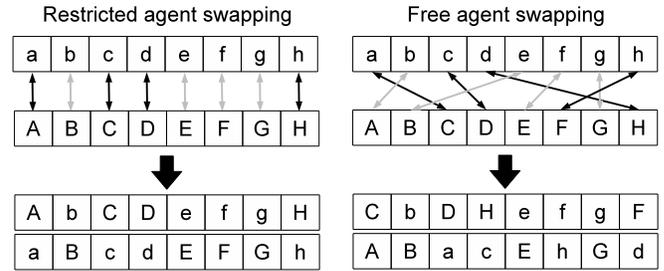


Fig. 3. Restricted agent swapping (left) and free agent swapping (right) applied on two teams of eight agents. In the presented example the probability of agent swapping was  $p = 0.5$ , black lines denote the exchanged agents and the grey lines denote the agents that happen to be kept in their original team. The random permutation used to reshuffle the agents before the swap in (b) is (3, 1, 4, 8, 2, 5, 7, 6).

teams possible. We assume that the optimal team contains  $K$  distinct agents, each repeating  $R$  times (for simplicity we assume  $M = K \cdot R$ ). Thus, the optimal composition of a team is defined as “ $K$  groups of  $R$  agents”, where the agents between the groups are different and the agents within a group are identical. The optimal team is homogeneous for  $K = 1$  and  $R = M$ . The optimal team is heterogeneous for  $K = M$  and  $R = 1$ . And, the optimal team is hybrid [12] (also called partially heterogeneous [13]) for  $1 < K < M$  and  $1 < R < M$ .

For a real-life problem the optimal team composition, and consequently the values of  $K$  and  $R$  are unknown. They are discovered by means of artificial evolution, driven by a fitness function  $F$  that measures team performance for a given problem. It should be expected that team performance is correlated with the composition of the team. Thus, we set  $K$  and  $R$  a priori and we define a fitness function  $f$  that directly depends on the proportion of “proper” agents in each of the  $K$  groups  $f = \sum_{j=1}^K \min(\frac{R}{M}, x_j)$ . The value  $x_j$  is the fraction of agents from a team that belong to the  $j$ th group and operator  $\min(a, b)$  takes a value  $a$  if  $a < b$ , and  $b$  otherwise.

For example, consider three agents: A, B and C. Let  $(x, y, z)$  denote the number of agents A, B and C, respectively, in a team. The team size is set to six (i.e.,  $x + y + z = 6$ ), and the optimal team consists of three agents A and three agents B (i.e., (3, 3, 0)). Exactly one team (0, 0, 6) has the lowest performance  $f = 0$ . Exactly one team (3, 3, 0) has the maximal performance  $f = 1$ . And, for instance, teams (1, 1, 4), (0, 2, 4) and (2, 0, 4) all have performance  $f = 0.33$ ; teams (6, 0, 0), (3, 0, 3) and (1, 2, 3) all have performance  $f = 0.5$ ; and teams (2, 2, 2), (1, 5, 0) and (1, 3, 2) all have performance  $f = 0.67$ .

The fitness function  $f$ , although not directly applicable to real-life problems, lets us to test the efficiency of RAS and FAS in the evolution of team composition under controlled conditions. The advantage of our approach is the ease of generating instances of various complexities and sizes. This enables us to perform systematic studies and thus draw statistically significant conclusions. We believe that the presented formulation abstracts well the core properties of many problems when one needs to optimize the team composition in a multi-agent system. We further elaborate on this issue in the Discussion section. In addition, we show the applicability of

our results by validating the efficiency of RAS and FAS in the evolution of decentralized controllers in a task-allocation problem [Appendix G].

### C. Evolutionary experiments

We evolve teams of agents in three *treatments*: 1) using RAS for all generations of the evolutionary algorithm, 2) using FAS for all generations of the evolutionary algorithm and 3) using FAS for the first half, and RAS for the second half, of all generations of the evolutionary algorithm. We also investigated alternative ways of combining FAS and RAS [Appendix F].

We compare RAS and FAS under three *conditions*, where the optimal team is composed of 1000, 100 and 10 groups of 1, 10 and 100 identical agents, respectively ( $K \in \{1000, 100, 10\}$  and  $R \in \{1, 10, 100\}$ ). Overall, this makes a total of 9 *experimental lines* (3 treatments  $\times$  3 conditions). Each experimental line is replicated 10 times. In all numerical experiments we use populations of 1000 teams of  $M = 1000$  agents each. Population size and team sizes are kept constant across generations. The number of all distinct types of agents is set to  $D = 10000$ . Each evolutionary run lasts for 1000 generations, with the exception of an additional experiment, where the number of generations is set to 2000 [see Appendix G for more details].

The software testbed has been implemented with the help of ECJ framework [66]. The numerical experiments have been run on the Pleiades cluster at EPFL, Lausanne.

### D. Genetic architecture, selection and reproduction

A team's genotype consists of 1000 alleles (one allele per agent), which are integers from 1 to 10000. At the first generation of each evolutionary run, each of  $1000 \times 1000$  alleles is independently set to a random integer value between 1 and 10000 with uniform distribution. Teams are evaluated in the collaborative task (i.e., how similar the team's composition and the optimal composition are) and assigned a performance (see Section III-B). To construct the 1000 teams of the following generation we select 500 times two teams. Each parent is independently selected from the current population using tournament selection with tournament size set to 2, with the exception of an additional experiment, where tournament size is set to 7 [Appendix D]. The two selected teams are reorganized with FAS or RAS, which results in two new teams that are added to the next generation's population. We use RAS and FAS with the probability of exchanging the agents between two teams set to  $p = 0.5$  for each position in a team, with the exception of an additional experiment, where  $p = 0.2$  is used [Appendix B]. Note that  $p = 0.5$  is the highest possible value, because swapping agents between teams A and B with probability  $0.5 < p \leq 1$  is equivalent to swapping agents between teams B and A with probability  $1 - p$ . The newly added teams are not subject to mutation, with the exception of an additional experiment, where each allele is independently set to a random integer value between 1 and 10000 (uniform distribution) with a probability 0.001 [Appendix C]. We do not use mutations in the main experiments for two reasons. First, our intention is to investigate the evolutionary dynamics

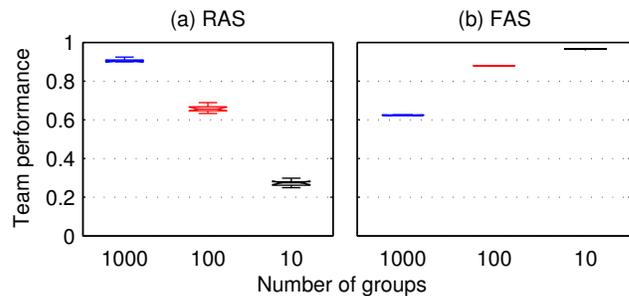


Fig. 4. Box and Whisker plots showing the mean performance of 1000 teams evolved with (a) restricted agent swapping and (b) free agent swapping (10 replicates). Teams consisted of 1000 agents, which needed to display three different compositions (1000, 100 and 10 groups of identical agents) in order for the team to achieve the optimal performance.

of RAS and FAS. Thus, to get clearer results on the effects of agent swapping, we do not use the mutation. Second, with 10000 different agents, the population of  $1000 \times 1000$  agents already contains each agent on average 100 times. Thus, the introduction of the innovative genetic material during evolution should not be necessary, if the agent swapping does efficiently compose optimal teams. Finally, one of the 1000 new teams of the following generation is randomly chosen, discarded and replaced by an exact copy of the best team from the current generation (i.e., elitism of size 1).

### E. Statistical analysis

To compare the teams evolved with RAS and FAS we average, for each experimental line and replicate, team performance over 1000 teams (10 replicates) at generation 1000. We report also the best team performance from 1000 teams (10 replicates) [Appendix E]. We explain the differences in team performance in terms of variation between teams that RAS and FAS introduced into the population. To this aim, we calculated the standard deviation of teams' performance in a population for each of the 1000 generations. In particular, we compared the convergence time (i.e., the number of generations until the measured standard deviation reached zero) and the final variation level between the teams (i.e., the value of the measured standard deviation at generation 1000). To explain the differences in variation between teams, we compared the proportion of corresponding positions that in all teams contained only agents that could not become members of the optimal team. And, we compared the proportion of agents in teams that could become members of the optimal team, averaged over 1000 teams. The last two characteristics were calculated every 25 generations, due to high demand for resources of these calculations (i.e., CPU and disk space). Statistical significance within multiple experimental lines was determined with Kruskal-Wallis test (nonparametric one-way analysis of variance). Statistical significance between a pair of experimental lines was determined with Wilcoxon test (rank sum test for equal medians).

## IV. RESULTS

We analyze the first two treatments (RAS and FAS) in Sections IV.A and IV.B, and the third treatment (mixed

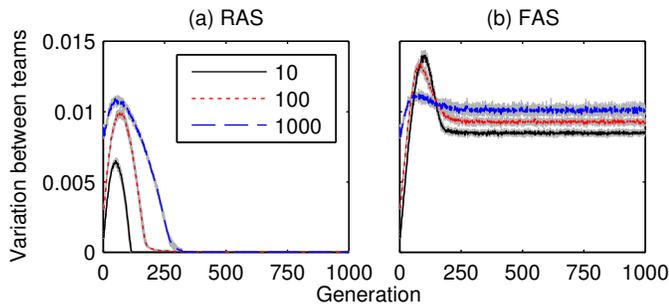


Fig. 5. Mean variation ( $\pm$  s.d. in grey) between the 1000 teams in a population (10 replicates), measured for all 1000 generations. The variation between the teams was quantified with the standard deviation of teams' performance in a population for (a) restricted agent swapping and (b) free agent swapping, in each of the three conditions (10, 100 and 1000 groups of identical agents in the optimal composition).

FAS/RAS) in Section IV.C.

#### A. Restricted and free agent swapping

There were important differences in team performance between the three conditions (1000, 100 and 10 groups) at the 1000th generation for both treatments (RAS: Fig. 4a, Kruskal-Wallis test,  $df = 2$ ,  $p < 0.001$ ; FAS: Fig. 4b, Kruskal-Wallis test,  $df = 2$ ,  $p < 0.001$ ). With RAS, the highest team performance was for 1000 groups (100 groups:  $-27.4\%$ ; 10 groups:  $-69.8\%$ ; three pairwise Wilcoxon tests  $df = 9$ ,  $p < 0.001$ ). In contrast, with FAS the highest team performance was for 10 groups (100 groups:  $-9\%$ ; 1000 groups:  $-35.4\%$ ; three pairwise Wilcoxon tests  $df = 9$ ,  $p < 0.001$ ).

The performance difference between conditions in treatments with RAS and FAS was caused by convergence issues, different for each of the two treatments. With RAS, the performance difference was associated with the convergence time, i.e., the number of generations until the standard deviation of team performance in the population reached zero (Fig. 5a, mean  $\pm$  s.d. generations for 1000 groups:  $334.3 \pm 14.6$ ; 100 groups:  $294.2 \pm 36.7$ ; 10 groups:  $113.5 \pm 3.1$ ; Kruskal-Wallis test,  $df = 2$ ,  $p < 0.001$ ; three pairwise Wilcoxon tests  $df = 9$ ,  $p < 0.01$ ). Therefore, the evolutionary algorithm using RAS suffered from "premature convergence". In contrast, with FAS the performance difference was associated with the variation between the evolved teams, i.e., the value of the standard deviation of team performance in the population at generation 1000 (Fig. 5b; Kruskal-Wallis test,  $df = 2$ ,  $p < 0.001$ ; three pairwise Wilcoxon tests  $df = 9$ ,  $p < 0.001$ ). Therefore, the evolutionary algorithm using FAS suffered from "insufficient convergence".

With RAS, premature convergence was detrimental to team performance, because it led to the disappearance of agents required in the optimal team on corresponding positions in teams across the entire population (Fig. 6a, top row). This was not the case for treatments with FAS (Fig. 6b, top row). Consequently, in treatments with RAS the teams contained agents that could not become members of the optimal team (Fig. 6a, bottom row). With FAS, insufficient convergence

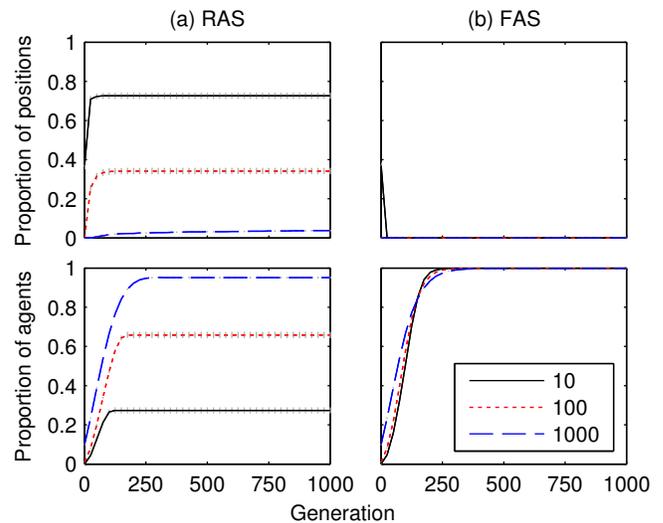


Fig. 6. Top: mean proportion ( $\pm$  s.d. in grey) of corresponding positions that contained in all 1000 teams only the agents that could not become members of the optimal team. Bottom: mean proportion ( $\pm$  s.d. in grey) of agents in a team that could become members of the optimal team, averaged over 1000 teams. The populations were analyzed every 25 generations over 10 replicates for (a) restricted agent swapping and (b) free agent swapping, in each of the three conditions (10, 100 and 1000 groups of identical agents in the optimal composition).

was detrimental to team performance, because it prevented the formation of the optimal composition. This is supported by the fact that team performance stayed low, even though the teams contained only the agents required in the optimal team (Fig. 6b, bottom row). We illustrate this process with a simple thought experiment. Consider a population consisting of two identical teams each containing agents A and B (in that order). In such conditions RAS may swap agent A with agent A only, and agent B with agent B only. This does not change team compositions in the population, and thus does not affect the teams' performance. In contrast, FAS may at some point swap agent A with B, which would result in one team containing both agents A, and the second team containing both agents B. Consequently, FAS may destroy favorable team compositions, even when entire population contains only optimal teams.

We performed a sensitivity analysis to see how our results were affected by: lower probability of agent swapping, by mutations and by higher selection pressure. Lower probability of agent swapping had a small detrimental effect on team performance in both treatments, and was more marked for RAS [Appendix B]. In treatments with RAS, using the mutation counterbalanced to some extent premature convergence (compare Fig. 5 and Fig. C2), which led to an increase in team performance. In contrast, with FAS, the mutation introduced additional undesirable variation between the teams, and thus had a weak detrimental effect on the performance [Appendix C]. In treatments with RAS, stronger selection increased the detrimental effect of premature convergence on team performance. With FAS, stronger selection did not overcome the problem of insufficient convergence and had no effect on the performance of the evolved teams [Appendix D].

We tested alternative implementations of RAS, i.e., 1-

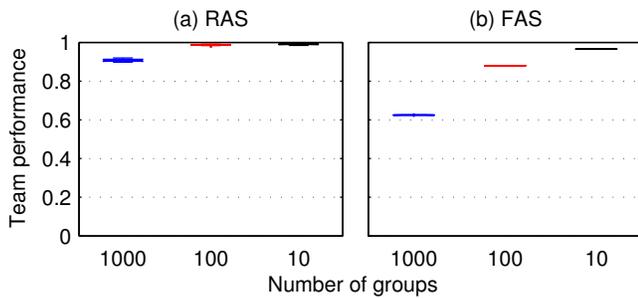


Fig. 7. Box and Whisker plots showing the mean performance of 1000 teams evolved with (a) restricted agent swapping and (b) free agent swapping (10 replicates). Teams consisted of 1000 agents, which needed to display three different compositions (10, 100 and 1000 groups of identical agents) in order for the team to achieve the optimal performance. The populations were initialized randomly with a bias. For each of the three conditions and for both treatments, the initial population always contained on average 10% of agents that could become members of the optimal team.

point and 2-point crossover. The results indicate that uniform crossover is better suited for restricted agent swapping than  $n$ -point crossover [Appendix A]. Here, we presented the mean team performance in the population (Fig. 4, 7). We report also the best team performance in the population (averaged over 10 replicates), which shows not to be qualitatively different from the mean team performance [Appendix E].

### B. Restricted and free agent swapping - disparities in the initial population

In the previous experiments, the agents in the teams were initialized randomly with uniform distribution (i.e., each initialized agent was set to be one of the 10000 possible agents with probability 0.0001). Consequently, the proportion of agents in the initial population that could become members of the optimal team varied between conditions. In the first condition with 1000 groups, the initial population contained on average 10% of agents that could become members of the optimal team. In the second condition with 100 groups, the initial population contained 1% of such agents. And in the third condition with 10 groups, the initial population contained only 0.1% of such agents. We tested if these disparities in the initial population had influence on the difference in team performance between conditions for both treatments. To this aim, we performed a control experiment in which the populations were initialized randomly with a bias. For each of the three conditions, the initial population always contained on average 10% of agents that could become members of the optimal team.

In treatments with RAS and without disparities in the initial population, there was an important increase in team performance for conditions with 10 and 100 groups in the optimal composition (compare Fig. 4a and Fig. 7a; two Wilcoxon tests,  $df = 9$ ,  $p < 0.001$ ). By contrast, there was no significant change over the 1000 generations for 1000 groups (compare Fig. 4a and Fig. 7a; Wilcoxon test  $df = 9$ ,  $p = 0.47$ ). This suggests that with RAS team performance primarily depended on the number of agents that could become members of the optimal team. The performance of the evolved teams was higher with more such agents in the initial population.

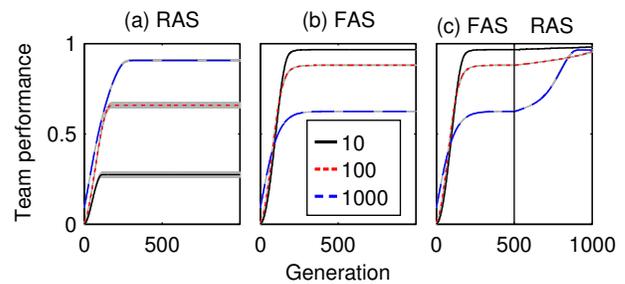


Fig. 8. Mean performance ( $\pm$  s.d. in grey) of 1000 teams evolved with (a) restricted agent swapping, (b) free agent swapping and (c) mixed FAS/RAS (10 replicates). In (c) the teams were evolved with free agent swapping in generations from 1 to 500 (inclusive), and with restricted agent swapping in generations from 501 to 1000. The black, solid vertical line at generation 500 marks the transition from using FAS to using RAS. Teams consisted of 1000 agents, which needed to display three different compositions (10, 100 and 1000 groups of identical agents) in order for the team to achieve the optimal performance.

In treatments with FAS, in contrast to RAS, there were no important differences in performance of teams evolved with and without disparities in the initial population for each of the three conditions (compare Fig. 4b and Fig. 7b, three pairwise Wilcoxon tests  $df = 9$ ,  $p > 0.3$ ). This suggests that with FAS team performance primarily depended on the optimal composition. The performance of the evolved teams was higher with a lower number of groups in the optimal team.

### C. Combining free and restricted agent swapping

In the treatments with RAS and FAS, the exploration and exploitation aspects of the evolutionary algorithm were not well balanced. With RAS the evolutionary algorithm suffered from premature convergence, whereas with FAS it suffered from insufficient convergence. Consequently, in both cases only suboptimal solutions have evolved (Fig. 8ab). In order to overcome the limitations of both RAS and FAS, we combined the two swapping methods in a complementary way: for the first 500 generations FAS was used, and for the next 500 generations RAS was used. We expected this would allow to efficiently explore the search space first and then to exploit it. We tested the efficiency of the mixed approach in the evolution of team composition for the three conditions (1000, 100 and 10 groups in the optimal composition).

There were important differences in team performance between the treatments (RAS, FAS, FAS/RAS) at the 1000th generation for each of the three conditions (Fig. 8, three Kruskal-Wallis tests,  $df = 2$ ,  $p < 0.001$ ). For each of the three conditions, the performance was higher with FAS/RAS than with RAS alone (Fig. 8ac, three Wilcoxon tests  $df = 9$ ,  $p < 0.001$ ) and than with FAS alone (Fig. 8bc, three Wilcoxon tests  $df = 9$ ,  $p < 0.001$ ). This was because in treatments with the mixed FAS/RAS approach, using FAS for the first 500 generations resulted in the abundance of agents that could become members of the optimal team (Fig. 6b, bottom row). These agents could then be efficiently used to compose optimal teams using RAS in the following 500 generations. In addition, for the first 500 generations there was high variation between the evolving teams typical for FAS, which was then decreased

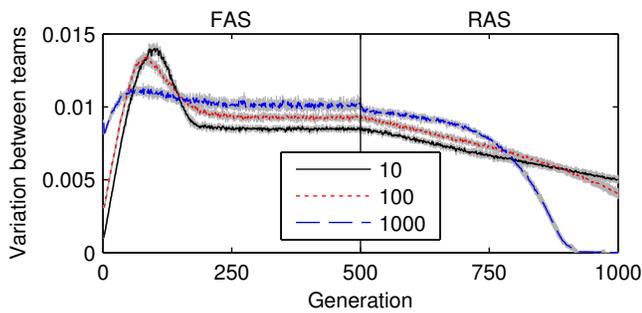


Fig. 9. Mean variation ( $\pm$  s.d. in grey) between the 1000 teams in a population, quantified with the standard deviation of teams' performance in a population. The teams were evolved in three different conditions (10, 100 and 1000 groups of identical agents in the optimal composition) with free agent swapping in generations from 1 to 500 (inclusive), and with restricted agent swapping in generations from 501 to 1000 (10 replicates). The black, solid vertical line at generation 500 marks the transition from using FAS to using RAS.

by applying RAS (Fig. 9). Overall, this mixed approach overcame the convergence issues of both FAS and RAS.

We fixed the moment of switch from FAS to RAS at the middle of the evolutionary process (500th generation). But if needed one may envision a dynamic switch between the agent swapping methods, which should be performed as soon as the variation between teams stabilizes (here around the 250th generation, Fig. 9). In addition, we tested alternative implementations of combining RAS and FAS, by applying both operators simultaneously but with different and varying probabilities of agent swapping. The results indicate that RAS and FAS should be used exclusively in order to balance the exploration and exploitation aspects of the evolutionary algorithm [Appendix F].

## V. DISCUSSION

We considered a situation when an optimal team consisted of equally sized groups of identical agents, but the results allow us to speculate how the restricted and free agent swapping compare in the situation when groups differ in sizes. In treatments with RAS, team performance depended on the contents of the initial population, and not on the actual optimal composition. Consequently, limitations and analysis presented in this paper for RAS should also hold for optimal team composition with groups of different sizes. This is because no link should be expected between the number of agents of a specific type in the initial population and in the optimal composition.

In contrast, with FAS, team performance depended on the optimal team composition. FAS introduced a variation between team compositions, which was detrimental to team performance. The level of this variation depended on the condition. For example, the variation was the highest, and hence team performance was the lowest, when the optimal team was composed of many small groups (here 1000 groups of 1 agent). This was because, with FAS, it is more probable for a team to deviate from the optimum when the optimal composition consists of small groups than when it consists of big groups. We illustrate this property with a simple thought

experiment. Consider two teams of 4 agents, both displaying the same optimal composition that consists of four groups of 1 agent. There are 16 possible free agent swaps between the two teams. Only 4 of them, the ones that happen on the corresponding positions, maintain the optimal composition in the teams. In contrast, consider two teams of 4 agents, both displaying the same optimal composition that consists of two groups of 2 agents. Again, there are 16 possible free agent swaps between the two teams. But now there are 8 swaps that maintain the optimal composition in the teams. Therefore, the performance of teams evolved with FAS depends on the size of groups in the optimal composition. But it does not depend on the fact if these groups are of equal size, or not. Consequently, limitations and analysis presented in this paper for FAS should also hold for optimal team composition with groups of different sizes.

Several general guidelines on applying RAS and FAS to real-life problems can be drawn. First, we compared RAS and FAS on a problem of evolving team compositions, whose formulation was highly general. The part of a team's genotype that corresponded to a single agent was rather simplistic, i.e., a vector of one (main experiments) to a few numbers (additional experiments in [Appendix G]). Consequently, our results are directly applicable whenever the goal is to optimize just few control parameters per agent, and not to create entire controllers for each agent from scratch. This is the case for many real-life applications, when the problem is often decomposed into two parts: the one solved manually and the one solved automatically. For example, in behavioral robotics, agents perform behaviors that have been implemented manually [67]–[69]. In bio-inspired scheduling, agents allocate themselves to tasks which they handle by calling existing routines [70]–[73]. In real-time strategy games, units perform predefined actions or entire sequences of actions [74], [75]. In ant colony optimization the agents traverse the edges of a graph according to established algorithms [2], [76]. In all these cases, the agents choose behaviors, tasks, actions and edges based on the values of a few control parameters that can be optimized automatically (see e.g., [77], [78]). In conclusion, our problem's formulation, although general, very well conforms to practical applications, as diverse as robotics, task allocation, video games and hyper-heuristics (i.e., searching in the space of heuristics).

Three implementation issues related to RAS and FAS should be discussed. First, we considered the agent swapping operators to work on teams having linear genotypes (in contrast to, e.g., a tree-based representation popular in genetic programming). Both RAS and FAS can be easily adapted to other teams' representations. In principle, FAS requires teams to be multi-sets of agents, i.e., sets in which elements may repeat. Whereas RAS requires teams to be sequences of agents. This is not much of a limitation, because one can always create a sequence by imposing an ordering on a multi-set. Note, however, that the ordering must remain constant during the evolution or, at least, it must change in the same way for all teams in the population. Otherwise, the agents could change their positions due to variations in the ordering. Consequently, there would be no restrictions on agents keeping their positions

in teams and using RAS would make little sense.

Second, we evolved teams of constant size, and thus we used genotypes of constant length. Both RAS and FAS could be adapted to variable-length genotypes by restraining the operators to act only on common parts of the genotypes. Alternatively, one could use agent deletion and duplication which alters the team compositions similarly to FAS [Appendix H].

Third, for the mixed FAS/RAS operator we considered also alternative implementations. We applied both operators simultaneously, but with different and varying probabilities of agent swapping. The results discourage such an approach and suggest that RAS and FAS should be used exclusively. This is due to the disruptive character of FAS, which is strongly marked even for low probabilities of agent swapping [Appendix F].

We focused on the evolution of team compositions using team encoding, i.e., when a single genotype encodes all individuals from one team. Team encoding implicitly assumes team level of selection (i.e., global reward, see Fig. 1). In contrast to individual level of selection, team level of selection omits pathologies caused by competition between team members (e.g., evolution of cheaters [79]–[81]), which could decrease the overall team performance. Therefore, team level of selection, and consequently team encoding, are advocated when the goal is to optimize the overall team’s performance [13]. Studying inter-agent interactions usually requires a game-theoretic perspective and is an interesting subject in itself (see, e.g., [82]), but was not in the scope of this paper.

Nevertheless, our results are also relevant for evolutionary algorithms using individual encoding (e.g., [13], [28], [30]–[33]). This is because, the converging character of RAS and the disruptive character of FAS are general properties of these operators. Thus, premature and insufficient convergence would be to some extent marked with RAS and FAS, respectively, regardless of the encoding and the level of selection used. For example, with individual encoding, in order to assess performance the agents are grouped into teams, often at random (see e.g., [13], [32], [83]). Random grouping in individual encoding introduces a constant variation between compositions of evolving teams, similarly to FAS in team encoding. Consequently, random grouping shares the limitations of FAS, and may hamper the evolutionary process whenever high level of genetic specialization between agents is required.

It should be noted that the evidence from our study has three limitations, which translate into three directions of future research. First, we assumed that a single change in team composition perfectly translates into a corresponding change in team performance. It might be desirable to consider epistatic and noisy fitness functions (see e.g., [84]). We expect FAS to drive the evolution towards optimum even then, because of the highly explorative nature of this method. We supported this claim by validating agent swapping operators in a stochastic and dynamic problem of decentralized task-allocation [Appendix G]. In this practical application, we showed that the teams evolved with FAS had higher performance than the teams evolved with RAS, probably due to low level of agent specialization that was required to solve the problem. This result is consistent with our previous analyses and shows that

the conclusions are valid also for more complex problems.

Second, we assumed a situation with no locally optimal team compositions. In practice, the globally optimal team composition might be hidden in a part of the landscape with low average payoff (i.e., deceptive fitness landscape [85]). Such a fitness landscape could hamper the evolution of optimal teams, similarly to other problems being solved with evolutionary algorithms [86], [87]. Fortunately, evolutionary algorithms frequently turn out to be excellent heuristics for most deceptive fitness landscapes [85].

Third, we focused on teams consisting of agents that did not evolve themselves. This allowed us to decouple the effects of agent recombination and agent swapping and to directly study the evolution of team compositions. We also validated RAS and FAS including mutations. This did not affect the conclusions drawn [Appendices C and G]. Note that developing the methods that efficiently evolve large teams of complex genotypes for multi-agent systems remains a big challenge. So far, other authors have focused on the evolution of complex controllers for small teams of agents (e.g., GP trees [43], [45] and neural networks [13], [32]), and have overlooked the evolution of team compositions (with the exception of [15], [41], which however considers only simple teams of up to 6 groups). Consequently, our investigation of evolving team compositions fills the gap and hopefully will help to address other challenging questions on optimizing multiple agents.

## VI. CONCLUSION

In this paper, we were able to quantify for the first time the efficiency of restricted and free agent swapping in the evolution of team compositions using team encoding. The analysis and comparison between restricted and free agent swapping revealed the limitations of both approaches. Our study highlights and explains the convergence issues, which were detrimental to team performance. Using restricted agent swapping resulted in an efficient evolution of team compositions, only if the population contained enough agents that could become members of the optimal team (but regardless of the optimal composition). In contrast, using free agent swapping resulted in an efficient evolution of team compositions, only if optimal composition imposed a low requirement on genetic specialization of the agents (but regardless of the contents of initial population). In order to overcome these limitations, the results strongly suggest to use a combination of the two methods of agent swapping, which were so far only used in isolation. The mixed approach balances the exploration and exploitation aspects of the evolutionary algorithm and experimentally proves to be a superior strategy in the evolution of team compositions.

## APPENDIX A

### 1-POINT AND 2-POINT CROSSOVER

Miconi [14] and Nakashima et al. [53] used 1-point and 2-point crossover to recombine the teams’ genotypes. The  $n$ -point crossover might be considered as a competitive way to implement RAS. It has been shown that  $n$ -point crossover has lower exploratory power than uniform crossover [51], [52].

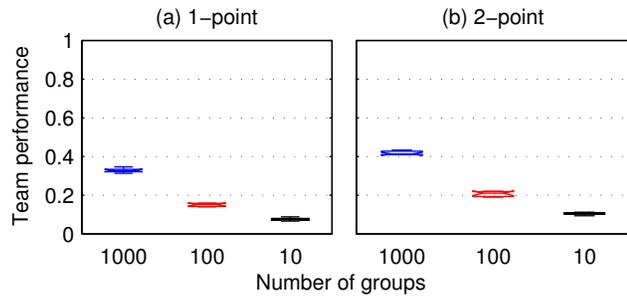


Fig. A1. Box and Whisker plots showing the mean performance of 1000 teams evolved with (a) 1-point crossover and (b) 2-point crossover (10 replicates). Teams consisted of 1000 agents, which needed to display three different compositions (10, 100 and 1000 groups of identical agents) in order for the team to achieve the optimal performance.

Also, teams are sets of agents (formally multi-sets) and not sequences of agents, thus the disruptive nature of the uniform crossover (i.e., breaking the schemata) [52] is not necessarily an undesired feature. These two facts suggest that uniform crossover [45], [50] is a better choice than  $n$ -point crossover in the implementation of RAS. We found support for this claim, showing that in treatments with RAS implemented as 1- and 2-point crossover team performance was lower (Fig. A1) than in treatments with RAS implemented as uniform crossover. This performance drop was associated with faster convergence (Fig. A2) with 1- and 2- point than uniform crossover. Consequently, the proportion of corresponding positions that in all teams contained only agents that could not become members of the optimal team was higher (Fig. A3, top row) with 1- and 2-point than uniform crossover. And, the proportion of agents in teams that could become members of the optimal team was lower (Fig. A3, bottom row) with 1- and 2-point than uniform crossover.

In treatments with 1- and 2-point crossovers, to construct the 1000 teams of the following generation we 500 times selected two teams (tournament size set to 2). Then, instead of reorganizing the two teams with uniform crossover, the teams' genotypes were crossed-over with 1-point or 2-point crossovers. With 1-point crossover, one random locus  $1 \leq L \leq 1000$  was chosen and the parenting teams exchanged all agents on positions from 1 to  $L$ . With 2-point crossover, two random loci  $1 \leq L_1 \leq L_2 \leq 1000$  were chosen and the parenting teams exchanged all agents on positions from  $L_1$  to  $L_2$ . Other settings were the same as in the *Main article* [see *Experimental method*].

## APPENDIX B

### EFFECTS OF AGENT SWAPPING WITH PROBABILITY $p = 0.2$

To test the sensitivity of the results to different probability of swapping the agents between the parenting teams we performed an additional experiment with this probability set to 0.2. Other settings were the same as in the *Main article* [see *Experimental method*]. We found that the control experiment is in a reasonably good agreement with the results reported in the main text, with respect to team performance (Fig. B1), the variation between teams (Fig. B2), the proportion of corresponding positions that in all teams contained only

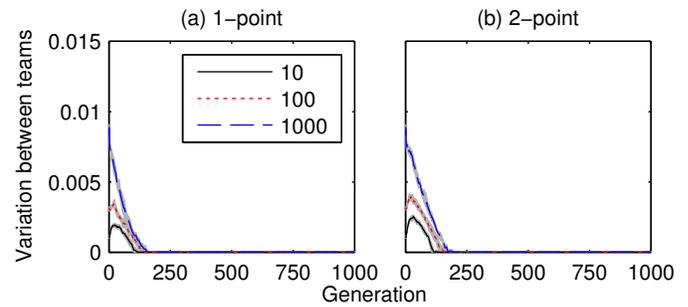


Fig. A2. Mean variation ( $\pm$  s.d. in grey) between the 1000 teams in a population (10 replicates), measured for all 1000 generations. The variation between the teams was quantified with the standard deviation of teams' performance in a population for (a) 1-point crossover and (b) 2-point crossover, in each of the three conditions (10, 100 and 1000 groups of identical agents in the optimal composition).

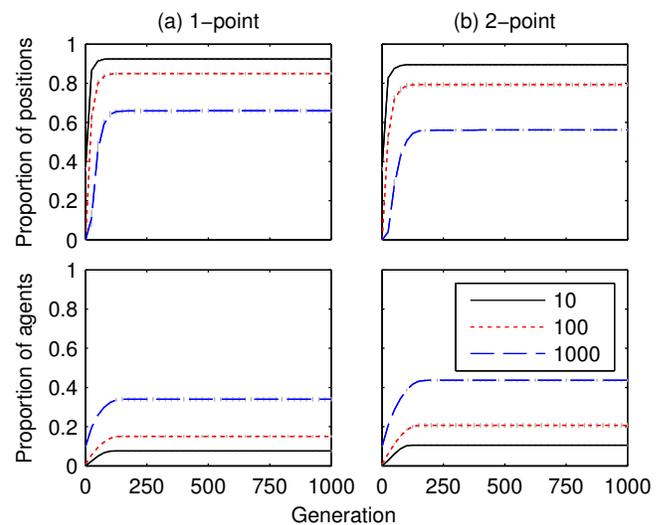


Fig. A3. Top: mean proportion ( $\pm$  s.d. in grey) of corresponding positions that contained in all 1000 teams only the agents that could not become members of the optimal team. Bottom: mean proportion ( $\pm$  s.d. in grey) of agents in a team that could become members of the optimal team, averaged over 1000 teams. The populations were analyzed every 25 generations over 10 replicates for (a) 1-point crossover and (b) 2-point crossover, in each of the three conditions (10, 100 and 1000 groups of identical agents in the optimal composition).

agents that could not become members of the optimal team (Fig. B3, top row) and the proportion of agents in teams that could become members of the optimal team (Fig. B3, bottom row).

## APPENDIX C

### EFFECTS OF MUTATION

With RAS the evolutionary algorithm suffered from premature convergence. Therefore, using mutation with RAS should have a positive effect on the exploration aspect of the evolutionary algorithm, and, thus, translate into an increase in team performance. In contrast, with FAS the evolutionary algorithm suffered from insufficient convergence. Consequently, using mutation with FAS should have a negative effect on the exploitation aspect of the evolutionary algorithm, and, thus, translate into a decrease in team performance. We found

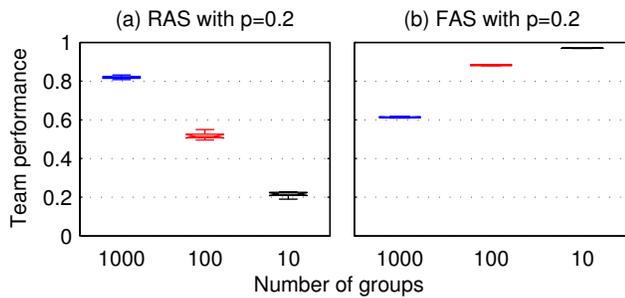


Fig. B1. Box and Whisker plots showing the mean performance of 1000 teams evolved with (a) restricted agent swapping and (b) free agent swapping (10 replicates). Teams consisted of 1000 agents, which needed to display three different compositions (10, 100 and 1000 groups of identical agents) in order for the team to achieve the optimal performance. For both treatments (RAS and FAS), the agents were swapped with probability 0.2.

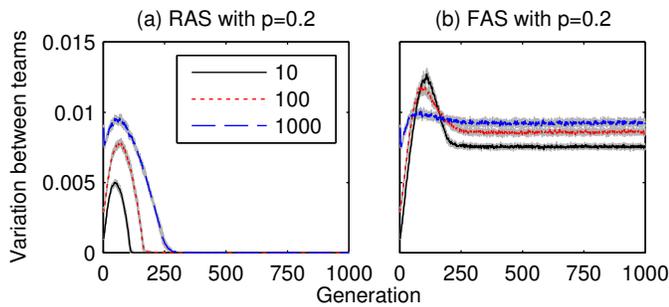


Fig. B2. Mean variation ( $\pm$  s.d. in grey) between the 1000 teams in a population (10 replicates), measured for all 1000 generations. The variation between the teams was quantified with the standard deviation of teams' performance in a population for (a) restricted agent swapping and (b) free agent swapping, in each of the three conditions (10, 100 and 1000 groups of identical agents in the optimal composition). For both treatments (RAS and FAS), the agents were swapped with probability 0.2.

support for both claims in an additional experiment were the teams' genotypes were subject to mutation, i.e., each allele of the newly added teams to the population was randomly set to a value between 1 and 10000 with a probability 0.001. Other settings were the same as in the *Main article* [see *Experimental method*].

In treatments with RAS, using the mutation counterbalanced to some extent premature convergence (Fig. C2a) and its negative effect on team compositions, with respect to the proportion of corresponding positions that contained in all teams only agents that could not become members of the optimal team (Fig. C3a, top row) and to the proportion of agents in teams that could become members of the optimal team (Fig. C3a, bottom row). This translated into a higher team performance (Fig. C1a) than in treatments with RAS without mutation. In contrast, in treatments with FAS the mutation increased the variation between the teams (Fig. C2b), which had a detrimental effect on evolving team compositions (Fig. C3b, top row and Fig. C3b, bottom row) and, consequently, on team performance (Fig. C1b).

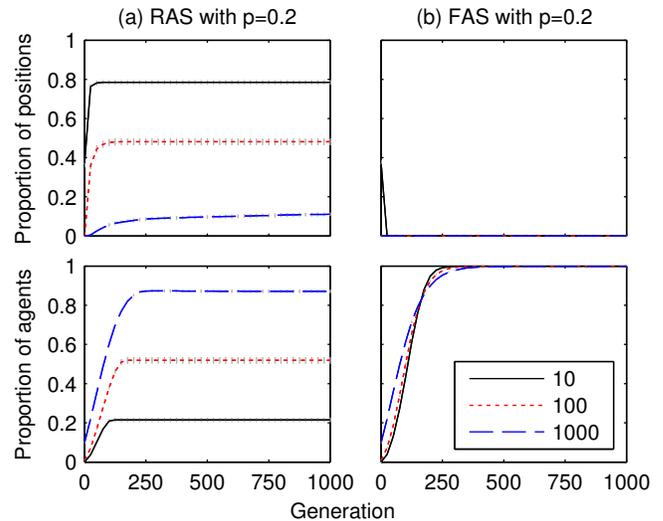


Fig. B3. Top: mean proportion ( $\pm$  s.d. in grey) of corresponding positions that contained in all 1000 teams only the agents that could not become members of the optimal team. Bottom: mean proportion ( $\pm$  s.d. in grey) of agents in a team that could become members of the optimal team, averaged over 1000 teams. The populations were analyzed every 25 generations over 10 replicates for (a) restricted agent swapping and (b) free agent swapping, in each of the three conditions (10, 100 and 1000 groups of identical agents in the optimal composition). For both treatments (RAS and FAS), the agents were swapped with probability 0.2.

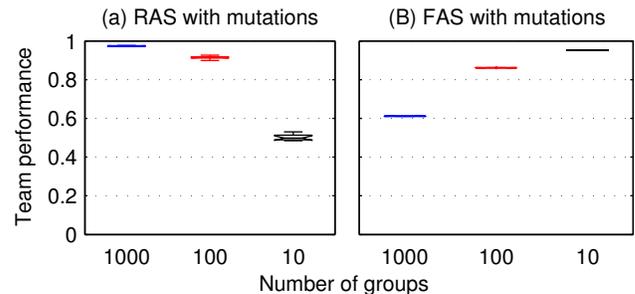


Fig. C1. Box and Whisker plots showing the mean performance of 1000 teams evolved with (a) restricted agent swapping and (b) free agent swapping (10 replicates). Teams consisted of 1000 agents, which needed to display three different compositions (10, 100 and 1000 groups of identical agents) in order for the team to achieve the optimal performance. For both treatments (RAS and FAS), each allele in a team's genotype was mutated with probability 0.001.

#### APPENDIX D EFFECTS OF SELECTION PRESSURE

We performed a sensitivity analysis and we investigated how our results were affected by increasing the selection pressure. To this aim we used a tournament size set to 7. Other settings were the same as in the *Main article* [see *Experimental method*]. We found that with RAS, stronger selection increased premature convergence (Fig. D2a), which translated into lower team performance (Fig. D1a) than in treatments with tournament size set to 2 (see *Results* in the *Main text*). This was because the increased premature convergence lead to higher proportion of corresponding positions that in all teams contained only agents that could not become members of the optimal team (Fig. D3a, top row) and to lower proportion of

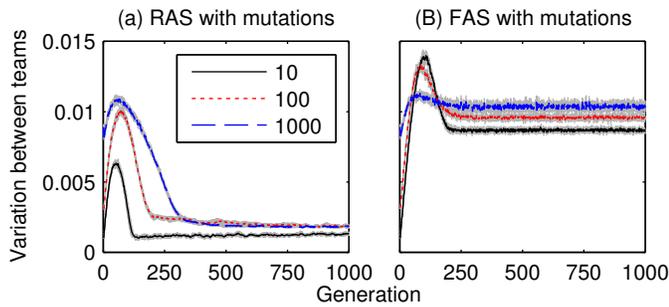


Fig. C2. Mean variation ( $\pm$  s.d. in grey) between the 1000 teams in a population (10 replicates), measured for all 1000 generations. The variation between the teams was quantified with the standard deviation of teams' performance in a population for (a) restricted agent swapping and (b) free agent swapping, in each of the three conditions (10, 100 and 1000 groups of identical agents in the optimal composition). For both treatments (RAS and FAS), each allele in a team's genotype was mutated with probability 0.001.

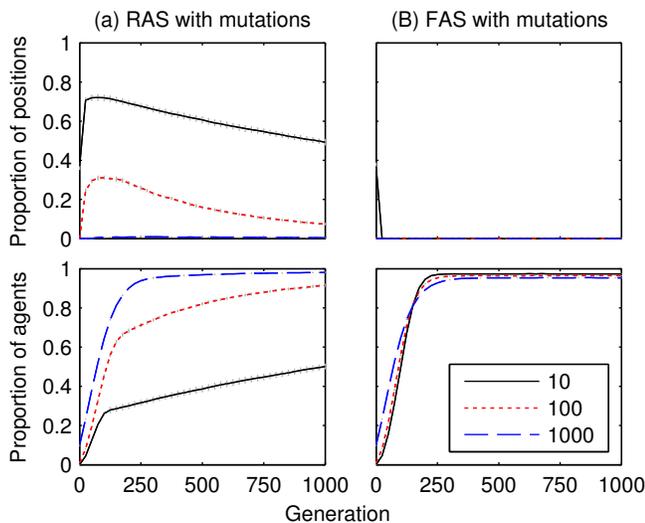


Fig. C3. Top: mean proportion ( $\pm$  s.d. in grey) of corresponding positions that contained in all 1000 teams only the agents that could not become members of the optimal team. Bottom: mean proportion ( $\pm$  s.d. in grey) of agents in a team that could become members of the optimal team, averaged over 1000 teams. The populations were analyzed every 25 generations over 10 replicates for (a) restricted agent swapping and (b) free agent swapping, in each of the three conditions (10, 100 and 1000 groups of identical agents in the optimal composition). For both treatments (RAS and FAS), each allele in a team's genotype was mutated with probability 0.001.

agents in teams that could become members of the optimal team (Fig. D3a, bottom row).

In contrast, with FAS, stronger selection had no effect on the performance of the evolved teams (Fig. D1b), because it did not overcome the problem of insufficient convergence (Fig. D2b). Consequently, the results for treatments with FAS with the two strengths of the selection pressure are in the agreement with respect to the proportion of corresponding positions that in all teams contained agents that could not become members of the optimal team (Fig. D3b, top row) and the proportion of agents in a team that could become members of the optimal team (Fig. D3b, bottom row).

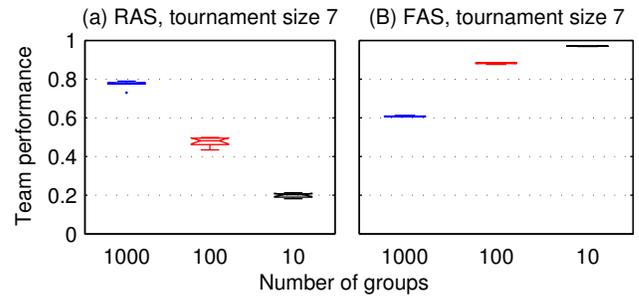


Fig. D1. Box and Whisker plots showing the mean performance of 1000 teams evolved with (a) restricted agent swapping and (b) free agent swapping (10 replicates). Teams consisted of 1000 agents, which needed to display three different compositions (10, 100 and 1000 groups of identical agents) in order for the team to achieve the optimal performance. For both treatments (RAS and FAS), the tournament size in the tournament selection was set to 7.

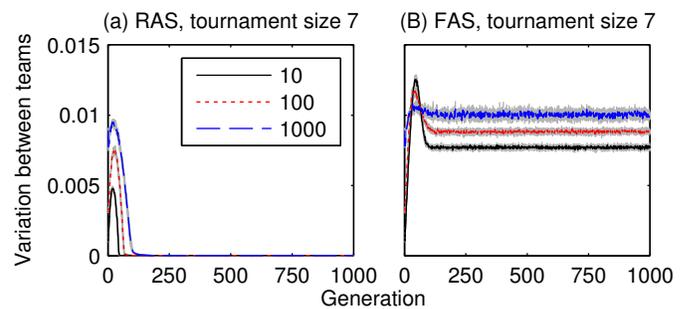


Fig. D2. Mean variation ( $\pm$  s.d. in grey) between the 1000 teams in a population (10 replicates), measured for all 1000 generations. The variation between the teams was quantified with the standard deviation of teams' performance in a population for (a) restricted agent swapping and (b) free agent swapping, in each of the three conditions (10, 100 and 1000 groups of identical agents in the optimal composition). For both treatments (RAS and FAS), the tournament size in the tournament selection was set to 7.

## APPENDIX E

### BEST TEAM PERFORMANCE IN THE POPULATION

We report here the performance of the best team at generation 1000th over 10 replicates (Fig. E1) and compare it with mean team performance. To this aim we calculated over 10 replicates the mean  $\pm$  s.d. ratio  $r$  between the value of mean performance in the population and the value of the best performance. There were no differences between mean and best team performance for treatments with RAS ( $r = 1 \pm 0$  for all three conditions). There were small differences between mean and best team performance for treatments with FAS ( $r$  for 1000 groups:  $0.931 \pm 0.004$ ; 100 groups:  $0.955 \pm 0.003$ ; 10 groups:  $0.971 \pm 0.001$ ). These differences between mean and best team performance correspond well to the variation between teams at the 1000th generation (0 for RAS, and approx. 0.01 for FAS, see Fig. 5).

## APPENDIX F

### ALTERNATIVE METHODS OF COMBINING RAS AND FAS

In the main text we combined FAS with RAS by applying them from generation 1 to 500 and from generation 501 to 1000, respectively. Here, we investigated two alternative methods that apply both operators together, but with different

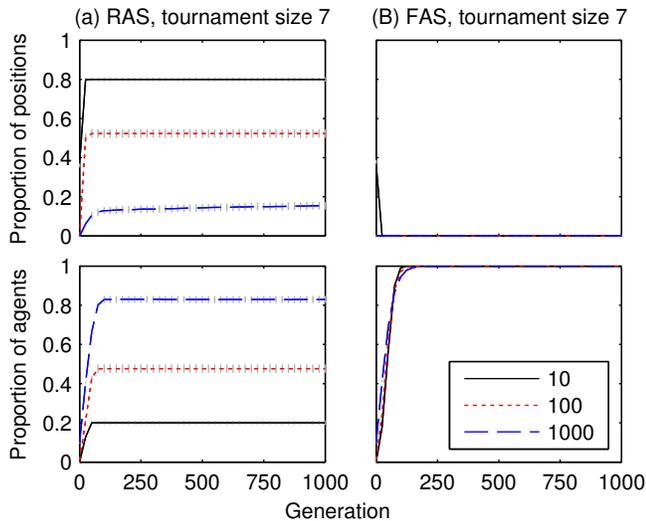


Fig. D3. Top: mean proportion ( $\pm$  s.d. in grey) of corresponding positions that contained in all 1000 teams only the agents that could not become members of the optimal team. Bottom: mean proportion ( $\pm$  s.d. in grey) of agents in a team that could become members of the optimal team, averaged over 1000 teams. The populations were analyzed every 25 generations over 10 replicates for (a) restricted agent swapping and (b) free agent swapping, in each of the three conditions (10, 100 and 1000 groups of identical agents in the optimal composition). For both treatments (RAS and FAS), the tournament size in the tournament selection was set to 7.

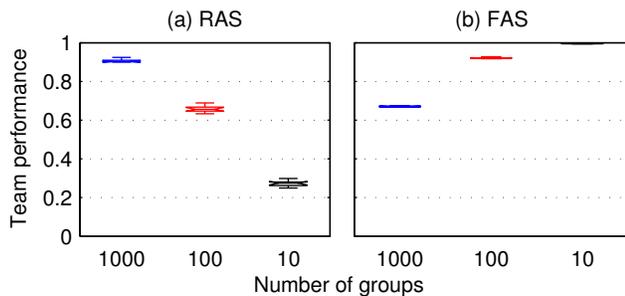


Fig. E1. Box and Whisker plots showing the best performance from 1000 teams evolved with (a) restricted agent swapping and (b) free agent swapping (10 replicates). Teams consisted of 1000 agents, which needed to display three different compositions (10, 100 and 1000 groups of identical agents) in order for the team to achieve the optimal performance.

probabilities of agent swapping (Fig. F1, top row). With the first method, the probability of agent swapping with FAS ( $p_{FAS}$ ) was set to = 0.5 at generation 1 and decreased with a constant rate to 0 at generation 1000. With the second method, the  $p_{FAS}$  was set to 0.5 from generation 1 to 250, then it decreased with a constant rate to 0 until the generation 750, and was fixed to 0 until the generation 1000. In both methods, the probability of agent swapping with RAS ( $p_{RAS}$ ) was set to  $0.5 - p_{FAS}$ . Other settings were the same as in the *Main article* [see *Experimental method*].

The results for both alternative methods were discouraging (Fig. F1, middle row), because, the disruptive character of FAS was strongly marked even for low probabilities of agent swapping. Consequently, the first alternative method did not converge (Fig. F1a, bottom row), and the second alternative method started converging only after  $p_{FAS}$  decreased to 0

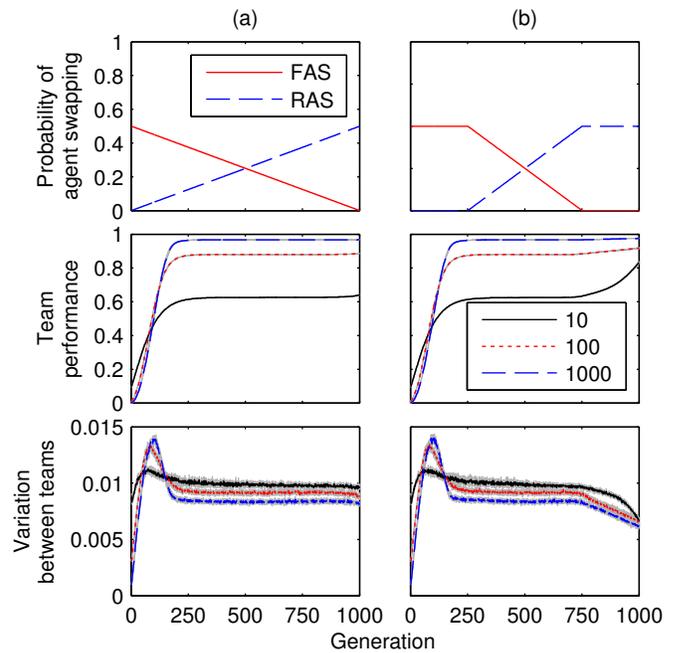


Fig. F1. Top row: the probabilities of agent swapping with two alternative methods of combining RAS and FAS, (a) and (b) respectively. Middle row: mean performance ( $\pm$  s.d. in grey) of 1000 teams. Teams consisted of 1000 agents, which needed to display three different compositions (10, 100 and 1000 groups of identical agents) in order for the team to achieve the optimal performance. Bottom row: mean variation ( $\pm$  s.d. in grey) between the 1000 teams. The variation between the teams was quantified with the standard deviation of teams' performance in a population. Both the mean performance and the mean variation were measured for all 1000 generations (10 replicates).

(Fig. F1b, bottom row). Therefore, in order to balance well the exploration and exploitation aspects in the evolution of team compositions one should use FAS and RAS exclusively.

## APPENDIX G

### APPLICATION OF RAS AND FAS TO THE EVOLUTION OF DECENTRALIZED CONTROLLERS FOR TASK ALLOCATION

**Methods.** We validated the agent swapping operators (RAS and FAS) in a complex problem of evolving decentralized controllers for task allocation. We considered a team composed of 1000 agents which allocated themselves to two distinct tasks using thresholds based models [70]–[73]. Agents receive information of the team's needs via commonly perceived stimuli. Then, agents respond to team's needs based on the values of their internal thresholds. Here, we optimized two different versions of the response thresholds models. First, with deterministic response threshold model (DTM, [88], [89]) every agent had two thresholds corresponding to each of the two tasks. An agent performed the task with the highest positive difference between the stimulus and its own corresponding response threshold, or remained idle if both of its thresholds were higher than the stimuli. If the difference between the stimulus and the agent's corresponding response thresholds was the same for all tasks, one of them was randomly chosen and performed by the agent. Second, with the extended response threshold model (ETM) [78], every agent had two thresholds corresponding to each of the two tasks

and two weights corresponding to each of the two stimuli. An agent performed the task with the highest positive difference between the weighted stimulus and its own corresponding response threshold, or remained idle if both of its thresholds were higher than the weighted stimuli. If the difference between the weighted stimulus and the agent's corresponding response threshold was the same for all tasks, one of them was randomly chosen and performed by the agent.

To quantify the teams' efficiency in task allocation we used a stochastic agent-based simulation to model a situation in which agents had to perform two distinct tasks [78], [90]. Our aim was to mimic situations with two vital tasks such as foraging and regulation. A team consisted of 1000 agents placed in an environment with an infinite number of two types of items, foraging and regulatory. The team's lifespan was divided into 100 time-steps. At the beginning of each time-step, an agent was presented with two task stimuli, one for the foraging items and the other for the regulatory items. If there were no items in the base, the corresponding stimulus was set to its maximal intensity, which was 1. Otherwise, the intensity of the stimulus for each task was inversely proportional to the number of corresponding items in the base. The foraging stimulus at time-step  $t$  was equal to  $1 - 10^{-4} \cdot a^F(t-1)$ , where  $a^F(t-1)$  is the number of foraging items accumulated in the base at time step  $t-1$ . The regulatory stimulus at time-step  $t$  was equal to  $1 - 5 \cdot 10^{-3} \cdot a^R(t-1)$ , where  $a^R(t-1)$  is the number of regulatory items accumulated in the base at time step  $t-1$ . At each time step, every agent performed the chosen task (or stayed idle) according to the task allocation mechanism (DTM, ETM) considered in the experiment. At each time step, an agent had a probability of 0.1 to successfully collect one item corresponding to the task performed. At each time-step, the number of foraged items in the base were depleted by ten items if they were not depleted in the previous time-step, or otherwise they were depleted by ten items with the probability of  $\frac{1}{2}$ . The same procedure was applied to independently deplete the regulatory items.

The team performance directly depended on the number of collected foraging items, but these were counted only when the number of regulatory items in the base was within predefined bounds (140–160 items). At the first time-step of a simulation, there were no items of the foraging and regulatory tasks in the base. The team performance  $f$  was calculated by adding the partial performance obtained at each time-step, with  $f = \sum_{t=1}^{100} f(t)$  where the team performance at each time-step ( $f(t)$ ) was quantified as the number of items of the foraging task collected when the number of items of the regulatory task present in the base was between 140 and 160:  $f(t) = b(t) \cdot g^F(t)$  where  $b(t) = 1$  if  $140 \leq g^R(i) \leq 160$  and  $b(t) = 0$  otherwise,  $g^F(t)$  represents the number of items foraged at time-step  $t$  and  $g^R(t)$  the number of items being regulated within the base at time-step  $t$ . Thus, if teams performed well in only one of the two tasks, their fitness was low. We normalized the resulting fitness values by 10000, which is the expected amount of foraging items collected if all 1000 agents were foraging for all 100 time-steps with the probability of success equal to 0.1.

In experiments with DTM and ETM, each team had a

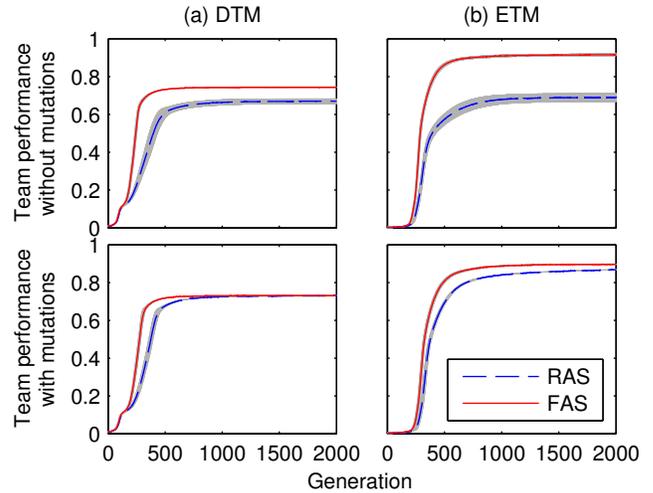


Fig. G1. Mean performance ( $\pm$  s.d. in grey) of 1000 teams evolved with restricted agent swapping (blue) and free agent swapping (red) for two treatments: (a) deterministic response thresholds model and (b) extended response thresholds model (30 replicates). Teams consisted of 1000 agents, which needed to dynamically self-allocate to two different tasks in order for the team to achieve the optimal performance.

genome consisting of 1000 parts, which corresponded to the agents. With DTM, each of these 1000 genome's parts consisted of two thresholds, both ranging from  $-1$  to  $+1$  (8-bit encoding, 256 possible values with a resolution of  $\frac{1}{128}$ ). With ETM, each of the 1000 genome's parts consisted of two thresholds and two weights, all ranging from  $-1$  to  $+1$  (8-bit encoding, 256 possible values with a resolution of  $\frac{1}{128}$ ).

We performed 2000 generations of artificial selection in thirty independent replicates for each of the two models (DTM and ETM), for two treatments (RAS and FAS), and in two conditions (without and with mutations). Overall there were  $2 \times 2 \times 2 = 8$  experimental lines. In the experiments with mutations each allele of the newly added teams to the population was randomly set to a value between  $-1$  to  $+1$  (8-bit encoding, 256 possible values with a resolution of  $\frac{1}{128}$ ) with a probability 0.001. Other settings were the same as in the *Main article* [see *Experimental method*].

**Results.** There were important differences in team performance between the two treatments (RAS and FAS), under both conditions (without and with mutations), and for both models (DTM and ETM). The team performance was higher with FAS than with RAS for both models, when mutations were not used (Fig. G1, top row; mean performance  $\pm$  s.d. at generation 2000 with DTM and without mutations, FAS:  $(7438 \pm 6) \cdot 10^{-4}$ , RAS:  $(6698 \pm 141) \cdot 10^{-4}$ ; with ETM and without mutations, FAS:  $(9161 \pm 9) \cdot 10^{-4}$ , RAS:  $(6695 \pm 225) \cdot 10^{-4}$ ; both Wilcoxon tests,  $df = 29$ ,  $p < 0.001$ ). With DTM and with mutations, there was a significant difference in performance between the two treatments, however it was very small (Fig. G1a, bottom row; mean performance  $\pm$  s.d. at generation 2000, FAS:  $(7304 \pm 7) \cdot 10^{-4}$ , RAS:  $(7298 \pm 10) \cdot 10^{-4}$ ; Wilcoxon test,  $df = 29$ ,  $p < 0.01$ ). With ETM and with mutations, the performance was higher with FAS than with RAS (Fig. G1a, bottom row; mean performance  $\pm$  s.d. at generation

2000, FAS:  $(8966 \pm 17) \cdot 10^{-4}$ , RAS:  $(8672 \pm 22) \cdot 10^{-4}$ ; Wilcoxon test,  $df = 29$ ,  $p < 0.001$ ). The results confirm our observations that using mutations with RAS may to some extent overcome premature convergence [Appendix C]. Nevertheless, for the evolution of more complex agents (here ETM) FAS remained superior to RAS. Moreover, in all cases FAS led to a faster evolution of teams displaying the highest performance (Fig. G1). Note that the performance differences between the DTM and ETM are due to limitations in task switching implicitly present in the formulation of the DTM (see [78] for more details).

We chose this setup because it has been used previously [78], [90]. We are aware that every specific problem has its own constraints and limitations. For example, in the scenario used here there are two tasks, thus the optimal team consists probably of a few large groups of agents of the same type, e.g., regulators and foragers. This explains the observed higher performance in treatments with FAS than with RAS (in the main experiment we showed that using FAS is a more efficient strategy of evolving team compositions when the optimal team has a rather low level of agent heterogeneity). Thus, we believe that the results presented in this Appendix support our conclusions from the main text and show that they are also valid for more complex problems.

#### APPENDIX H

##### THE FAMILY OF OPERATORS ALTERING TEAM COMPOSITION

Interesting connections between RAS, FAS and other team composition altering operators exist. First note that FAS swaps agents on permuted positions, in contrast to RAS which swaps agents on corresponding positions. Consequently, one can implement FAS as a composition of RAS and an operator that permutes the agents' order in a team, i.e., samples agents without repetitions.

A natural next step is to consider a situation when the agents in a team are sampled with repetitions, because this allows for a more straightforward implementation of the agents' sampling operator. Let a parenting team have  $M$  agents. In order to construct an offspring team, the agent sampling with repetition (ASWR) picks uniformly at random  $M$  agents with repetitions from the parenting team and puts them in the offspring team. Consider agent  $A$  in the parenting team. The number of its copies in the offspring team follows a binomial distribution with  $M$  trials and the probability of success being  $\frac{1}{M}$ . Thus, if the size of a team is sufficiently large ( $M > 20$ ) one can approximate the binomial distribution by Poisson distribution with the coefficient  $\lambda = 1$ . Consequently, after applying ASWR on a parenting team, each agent will be present in the offspring team  $k$  times with probability  $\frac{1}{e \cdot k!}$ , where  $k = 0, 1, 2, \dots, M$ . Thus, ASWR implements a sort of simultaneous agents' deletion and duplication (or rather "multiplication").

Interestingly, operators that delete and duplicate parts of genomes have been already proposed. In genetic programming there is branch deletion and duplication [54]. And, in gene expression programming there is gene transfer [55]. Gene

transfer is equivalent to branch "substitution", i.e., having two branches  $A$  and  $B$ , delete  $A$  and duplicate  $B$  in place of  $A$ . In the context of evolving team compositions an analogous agent transfer operator might be proposed. It should have a similar impact on the evolutionary dynamics as FAS. This is because agent transfer would be in essence similar to agent sampling, which in turn is a subcomponent of FAS. Note however, that agent duplication, deletion and transfer are applied to a single parenting team. Consequently, they should be considered as mutations, rather than cross-over.

#### ACKNOWLEDGMENT

Authors thank Andrea Maesani, Pavan Ramdya, and Krzysztof Krawiec for useful comments on the manuscript.

#### REFERENCES

- [1] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm intelligence: from natural to artificial systems*. New York: Oxford University Press, 1999.
- [2] —, "Inspiration for optimization from social insect behaviour," *Nature*, vol. 406, no. 6791, pp. 39–42, 2000.
- [3] C. Reynolds, "An evolved, vision-based behavioral model of coordinated group motion," in *Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, J. Meyer, H. Roitblat, and S. Wilson, Eds. Cambridge, MA: MIT Press, 1993, pp. 384–392.
- [4] W. Xiang and H. Lee, "Ant colony intelligence in multi-agent dynamic manufacturing scheduling," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 1, pp. 73–85, 2008.
- [5] G. Mathews, H. Durrant-Whyte, and M. Prokopenko, "Decentralised decision making in heterogeneous teams using anonymous optimisation," *Robotics and Autonomous Systems*, vol. 57, no. 3, pp. 310–320, 2009.
- [6] M. Quinn, L. Smith, G. Mayley, and P. Husbands, "Evolving controllers for a homogeneous system of physical robots: Structured cooperation with minimal sensors," *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 361, no. 1811, pp. 2321–2343, 2003.
- [7] M. Dorigo, V. Trianni, E. Şahin, R. Groß, T. Labelle, G. Baldassarre, S. Nolfi, J. Deneubourg, F. Mondada, D. Floreano *et al.*, "Evolving self-organizing behaviors for a swarm-bot," *Autonomous Robots*, vol. 17, no. 2, pp. 223–245, 2004.
- [8] C. Parker and H. Zhang, "Cooperative decision-making in decentralized multiple-robot systems: The best-of-n problem," *Mechatronics, IEEE/ASME Transactions on*, vol. 14, no. 2, pp. 240–251, 2009.
- [9] G. Baldassarre, S. Nolfi, and D. Parisi, "Evolving mobile robots able to display collective behaviors," *Artificial Life*, vol. 9, no. 3, pp. 255–267, 2003.
- [10] S. Nolfi and D. Floreano, *Evolutionary Robotics: The Biology, Intelligence, and Technology*. Cambridge, MA: MIT Press, 2000.
- [11] H. Iba, "Emergent cooperation for multiple agents using genetic programming," in *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*, V. H.-M., W. Ebeling, I. Rechenberger, and H.-P. Schwefel, Eds. Springer-Verlag, 1996, pp. 32–41.
- [12] L. Panait and S. Luke, "Cooperative multi-agent learning: The state of the art," *Autonomous Agents and Multi-Agent Systems*, vol. 11, no. 3, pp. 387–434, 2005.
- [13] M. Waibel, L. Keller, and D. Floreano, "Genetic team composition and level of selection in the evolution of cooperation," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 648–660, 2009.
- [14] T. Miconi, "When evolving populations is better than coevolving individuals: the blind mice problem," in *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, G. Gottlob and T. Walsh, Eds. San Francisco, CA: Morgan Kaufmann Publishers Inc., 2003, pp. 647–652.
- [15] J. Bongard, "The legion system: A novel approach to evolving heterogeneity for collective problem solving," in *Proceedings of the European Conference on Genetic Programming (EuroGP 2000)*, ser. Lecture Notes on Computer Science, R. Poli, W. Banzhaf, L. W.B., J. Miller, P. Nordin, and F. T.C., Eds., vol. 1802. Springer-Verlag, 2000, pp. 16–28.
- [16] T. Miconi, "A collective genetic algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001)*, L. Spector, E. Goodman, A. Wu, L. W.B., V. H.-M., M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. Garzon, and E. Burke, Eds. Morgan Kaufmann, 2001.

- [17] G. Werner and M. Dyer, "Evolution of communication in artificial organisms," in *Proceedings of the Second International Conference on Artificial Life*, ser. SFI Studies in the Sciences of Complexity, C. Langton, C. Taylor, J. Farmer, and S. Rasmussen, Eds., vol. X. Addison-Wesley, 1991, pp. 659–687.
- [18] S. Ficici, R. Watson, and J. Pollack, "Embodied evolution: A response to challenges in evolutionary robotics," in *Proceedings of the 8th European Workshop on Learning Robots*, ser. Lecture Notes in Artificial Intelligence, J. Wyatt and D. J., Eds., vol. 1812. Springer, 1999, pp. 14–22.
- [19] K. Stanley, B. Bryant, and R. Miikkulainen, "Real-time neuroevolution in the nero video game," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 6, pp. 653–668, 2005.
- [20] R. Watson, S. Ficici, and J. Pollack, "Embodied evolution: Distributing an evolutionary algorithm in a population of robots," *Robotics and Autonomous Systems*, vol. 39, no. 1, pp. 1–18, 2002.
- [21] L. Spector, J. Klein, C. Perry, and M. Feinstein, "Emergence of collective behavior in evolving populations of flying agents," *Genetic Programming and Evolvable Machines*, vol. 6, no. 1, pp. 111–125, 2005.
- [22] C. Ward, F. Gobet, and G. Kendall, "Evolving collective behavior in an artificial ecology," *Artificial Life*, vol. 7, no. 2, pp. 191–209, 2001.
- [23] E. Simoes and D. Barone, "Predation: An approach to improving the evolution of real robots with a distributed evolutionary controller," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2002)*, vol. 1. IEEE, 2002, pp. 664–669.
- [24] R. Bianco and S. Nolfi, "Toward open-ended evolutionary robotics: evolving elementary robotic units able to self-assemble and self-reproduce," *Connection Science*, vol. 16, no. 4, pp. 227–248, 2004.
- [25] A. Agah and G. Bekey, "Phylogenetic and ontogenetic learning in a colony of interacting robots," *Autonomous Robots*, vol. 4, no. 1, pp. 85–100, 1997.
- [26] A. Cangelosi and D. Parisi, "The emergence of a 'language' in an evolving population of neural networks," *Connection Science*, vol. 10, no. 2, pp. 83–97, 1998.
- [27] M. Quinn, "A comparison of approaches to the evolution of homogeneous multi-robot teams," in *Proceedings of the Congress on Evolutionary Computation (CEC 2001)*, vol. 1. IEEE, 2001, pp. 128–135.
- [28] Y. Suzuki and T. Arita, "A comprehensive evaluation of the methods for evolving a cooperative team," *Artificial Life and Robotics*, vol. 10, no. 2, pp. 157–161, 2006.
- [29] A. Agah and K. Tanie, "Robots playing to win: evolutionary soccer strategies," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 1997)*, vol. 1. IEEE, 1997, pp. 632–637.
- [30] L. Bull, "Evolutionary computing in multi-agent environments: Operators," in *Proceedings of the Seventh International Conference on Evolutionary Programming (EP98)*, ser. Lecture Notes in Computer Science, V. Porto, N. Saravanan, D. Waagen, and A. Eiben, Eds., vol. 1447. Springer, 1998, pp. 370–377.
- [31] A. Eiben, G. Nitschke, and M. Schut, "Collective specialization for evolutionary design of a multi-robot system," *Swarm Robotics*, pp. 189–205, 2007.
- [32] D. Floreano, S. Mitri, S. Magnenat, and L. Keller, "Evolutionary conditions for the emergence of communication in robots," *Current Biology*, vol. 17, no. 6, pp. 514–519, 2007.
- [33] T. Soule and R. Heckendorn, "Improving performance and cooperation in multi-agent systems," in *Genetic Programming Theory and Practice V*, ser. Genetic and Evolutionary Computation, R. Riolo, T. Soule, and B. Worzel, Eds. Springer, 2008, pp. 221–237.
- [34] M. Potter and K. De Jong, "A cooperative coevolutionary approach to function optimization," in *Proceedings of the Third International Conference on Parallel Problem Solving from Nature (PPSN III)*, ser. Lecture Notes in Computer Science, Y. Davidor, H.-P. Schwefel, and R. Männer, Eds., vol. 866. Springer, 1994, pp. 249–257.
- [35] M. A. Potter and D. V. De Jong, "Cooperative coevolution: An architecture for evolving coadapted subcomponents," *Evolutionary Computation*, vol. 8, pp. 1–29, 2000.
- [36] R. Wiegand, W. Liles, and K. De Jong, "An empirical analysis of collaboration methods in cooperative coevolutionary algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001)*, 2001, pp. 1235–1242.
- [37] M. Mirolli and D. Parisi, "How can we explain the emergence of a language that benefits the hearer but not the speaker?" *Connection Science*, vol. 17, no. 3, pp. 307–324, 2005.
- [38] L. Panait, S. Luke, and R. Wiegand, "Biasing coevolutionary search for optimal multiagent behaviors," *Evolutionary Computation, IEEE Transactions on*, vol. 10, no. 6, pp. 629–645, 2006.
- [39] D. Andre and A. Teller, "Evolving team darwin united," in *RoboCup-98: Robot Soccer World Cup II*, ser. Lecture Notes on Computer Science, M. Asada and H. Kitano, Eds., vol. 1604. Springer Verlag, 1999, pp. 346–351.
- [40] M. Brameier and W. Banzhaf, "Evolving teams of predictors with linear genetic programming," *Genetic Programming and Evolvable Machines*, vol. 2, no. 4, pp. 381–407, 2001.
- [41] A. Hara and T. Nagao, "Emergence of cooperative behavior using adg: Automatically defined groups," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 1999)*, W. Banzhaf, J. Daida, A. Eiben, M. Garzon, V. Honavar, M. Jakiela, and R. Smith, Eds. Morgan Kaufmann, 1999, pp. 1038–1046.
- [42] S. Luke, C. Hohn, J. Farris, G. Jackson, and J. Hendler, "Co-evolving soccer softbot team coordination with genetic programming," in *RoboCup-97: Robot soccer world cup I*. Springer, 1998, pp. 398–411.
- [43] S. Luke and L. Spector, "Evolving teamwork and coordination with genetic programming," in *Proceedings of the First Annual Conference on Genetic Programming*, J. Koza, D. Goldberg, D. Fogel, and R. Riolo, Eds. Cambridge, MA: MIT Press, 1996, pp. 150–156.
- [44] H. Botee and E. Bonabeau, "Evolving ant colony optimization," *Complex Systems*, vol. 1, no. 2, pp. 149–159, 1998.
- [45] T. Haynes and S. Sen, "Co-adaptation in a team," *International Journal of Computational Intelligence and Organizations*, vol. 1, no. 4, pp. 1–20, 1996.
- [46] —, "Crossover operators for evolving a team," in *Proceedings of the Second Annual Conference on Genetic Programming*, J. Koza, B. Deb, M. Dorigo, D. Fogel, M. Garzon, H. Iba, and R. Riolo, Eds. Stanford University, CA: Morgan Kaufmann, 1997, pp. 162–167.
- [47] A. Agogino, K. Stanley, and R. Miikkulainen, "Online interactive neuro-evolution," *Neural Processing Letters*, vol. 11, no. 1, pp. 29–38, 2000.
- [48] J. Grefenstette, "Credit assignment in rule discovery systems based on genetic algorithms," *Machine Learning*, vol. 3, no. 2, pp. 225–245, 1988.
- [49] K. De Jong, "Learning with genetic algorithms: An overview," *Machine Learning*, vol. 3, pp. 121–138, 1988.
- [50] G. Syswerda, "Uniform crossover in genetic algorithms," in *Proceedings of the Third International Conference on Genetic Algorithms (ICGA 1989)*, J. Schaffer, Ed. Morgan Kaufmann, 1989, pp. 2–9.
- [51] L. Eshelman, R. Caruana, and J. Schaffer, "Biases in the crossover landscape," in *Proceedings of the Third International Conference on Genetic Algorithms (ICGA 1989)*, J. Schaffer, Ed. Morgan Kaufmann Publishers Inc., 1989, pp. 10–19.
- [52] K. De Jong and W. Spears, "A formal analysis of the role of multi-point crossover in genetic algorithms," *Annals of Mathematics and Artificial Intelligence*, vol. 5, no. 1, pp. 1–26, 1992.
- [53] T. Nakashima, M. Takatani, M. Udo, H. Ishibuchi, and M. Nii, "Performance evaluation of an evolutionary method for robocup soccer strategies," in *RoboCup 2005: Robot Soccer World Cup IX*, ser. Lecture Notes in Artificial Intelligence, A. Bredendfeld, A. Jacoff, I. Noda, and Y. Takahashi, Eds. Springer, 2006, pp. 616–623.
- [54] J. Koza, "Evolving the architecture of a multi-part program in genetic programming using architecture-altering operations," in *Evolutionary Programming IV: Proceedings of the Fourth Annual Conference on Evolutionary Programming*, ser. A Bradford Book: Complex Adaptive Systems series, J. R. McDonnell, R. G. Reynolds, and D. B. Fogel, Eds. Cambridge, MA: The MIT Press, 1995, pp. 695–718.
- [55] C. Ferreira, "Gene expression programming: A new adaptive algorithm for solving problems," *Complex Systems*, vol. 13, no. 2, pp. 87–129, 2001.
- [56] M. Sebag and M. Schoenauer, "Mutation by imitation in boolean evolution strategies," in *Proceedings of the 4th Conference on Parallel Problem Solving from Nature (PPSN IV)*, ser. Lectures Notes in Computer Science, H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, Eds., vol. 1141. Springer, 1996, pp. 356–365.
- [57] S. Berman, A. Halasz, M. Hsieh, and V. Kumar, "Optimized stochastic policies for task allocation in swarms of robots," *IEEE Transactions on Robotics*, pp. 927–937, 2009.
- [58] T. Dahl, M. Mataric, and G. Sukhatme, "Multi-robot task allocation through vacancy chain scheduling," *Robotics and Autonomous Systems*, vol. 57, no. 6-7, pp. 674–687, 2009.
- [59] S. Beshers and J. Fewell, "Models of Division of Labor in Social Insects," *Annual Review of Entomology*, vol. 46, no. 1, pp. 413–440, 2001.
- [60] C. Smith, A. Toth, A. Suarez, and G. Robinson, "Genetic and genomic analyses of the division of labour in insect societies," *Nature Reviews Genetics*, vol. 9, no. 10, pp. 735–748, 2008.

- [61] A. Duarte, F. Weissing, I. Pen, and L. Keller, "An evolutionary perspective on self-organized division of labor in social insects," *Annual Review of Ecology, Evolution and Systematics*, vol. 42, no. 1, in press.
- [62] G. Robinson, "Regulation of division of labor in insect societies," *Annual Review of Entomology*, vol. 37, no. 1, pp. 637–665, 1992.
- [63] D. Gordon, "The organization of work in social insect colonies," *Nature*, vol. 380, no. 6570, pp. 121–124, 1996.
- [64] J. Fewell, "Social insect networks," *Science*, vol. 301, no. 5641, p. 1867, 2003.
- [65] E. Robinson, O. Feinerman, and N. Franks, "Flexible task allocation and the organization of work in ants," *Proceedings of the Royal Society B: Biological Sciences*, vol. 276, no. 1677, p. 4373, 2009.
- [66] S. Luke, *A User Manual for the ECJ Evolutionary Computation Library*, 2010.
- [67] R. Brooks, "New approaches to robotics," *Science*, vol. 253, no. 5025, p. 1227, 1991.
- [68] —, "Intelligence without representation," *Artificial intelligence*, vol. 47, no. 1-3, pp. 139–159, 1991.
- [69] T. Balch and R. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926–939, 1998.
- [70] M. Campos, E. Bonabeau, G. Theraulaz, and J. Deneubourg, "Dynamic scheduling and division of labor in social insects," *Adaptive Behavior*, vol. 8, no. 2, pp. 83–95, 2000.
- [71] C. Jones and M. Mataric, "Adaptive division of labor in large-scale minimalist multi-robot systems," in *Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, vol. 2. IEEE, 2003, pp. 1969–1974.
- [72] M. Mataric, G. Sukhatme, and E. Østergaard, "Multi-robot task allocation in uncertain environments," *Autonomous Robots*, vol. 14, no. 2, pp. 255–263, 2003.
- [73] W. Agassounon and A. Martinoli, "Efficiency and robustness of threshold-based distributed allocation algorithms in multi-agent systems," in *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 3*. New York, NY, USA: ACM, 2002, pp. 1090–1097.
- [74] P. Lichocki, K. Krawiec, and W. Jaśkowski, "Evolving teams of cooperating agents for real-time strategy game," in *Proceedings of the EvoWorkshops 2009 on Applications of Evolutionary Computing*, ser. Lecture Notes in Computer Science, M. Giacobini, A. Brabazon, S. Cagnoni, G. Di Caro, A. Ekárt, A. Esparcia-Alcázar, M. Farooq, A. Fink, and P. Machado, Eds., vol. 5484. Springer, 2009, pp. 333–342.
- [75] M. Preuss, N. Beume, H. Danielsiek, T. Hein, B. Naujoks, N. Piatkowski, R. Ster, A. Thom, and S. Wessing, "Towards intelligent team composition and maneuvering in real-time strategy games," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 2, pp. 82–98, 2010.
- [76] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.
- [77] T. White, B. Pagurek, and F. Oppacher, "Asga: Improving the ant system by integration with genetic algorithms," in *Proceedings of the Third Annual Conference on Genetic Programming*, J. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. Fogel, M. Garzon, D. Goldberg, H. Iba, and R. Riolo, Eds. San Francisco, CA: Morgan Kaufmann, 1998, pp. 610–617.
- [78] P. Lichocki, D. Tarapore, L. Keller, and D. Floreano, "Neural networks as mechanisms to regulate division of labor," *The American Naturalist*, vol. 179, no. 3, pp. 391–400, 2012.
- [79] S. Goings, J. Clune, C. Ofria, and R. Pennock, "Kin selection: The rise and fall of kin-cheaters," in *Artificial Life IX: Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems*, J. Pollack, M. Bedau, P. Husbands, T. Ikegami, and R. Watson, Eds. Cambridge, MA: The MIT Press, 2004, pp. 303–308.
- [80] S. Mitri, D. Floreano, and L. Keller, "The evolution of information suppression in communicating robots with conflicting interests," *Proceedings of the National Academy of Sciences*, vol. 106, no. 37, pp. 15 786–15 790, 2009.
- [81] M. Waibel, D. Floreano, and L. Keller, "A quantitative test of hamilton's rule for the evolution of altruism," *PLoS Biology*, vol. 9, no. 5, p. e1000615, 2011.
- [82] Y. Shoham and K. Leyton-Brown, *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge: Cambridge University Press, 2009.
- [83] M. Waibel, D. Floreano, S. Magnenat, and L. Keller, "Division of labour and colony efficiency in social insects: effects of interactions between genetic architecture, colony kin structure and rate of perturbations," *Proceedings of the Royal Society B: Biological Sciences*, vol. 273, no. 1595, pp. 1815–1823, 2006.
- [84] D. Goldberg, K. Deb, and J. Clark, "Genetic algorithms, noise, and the sizing of populations," *Complex Systems*, vol. 6, pp. 333–362, 1991.
- [85] K. De Jong, "Genetic algorithms are not function optimizers," in *Proceedings of the Second Workshop on Foundations of Genetic Algorithms*, D. Whitley, Ed., 1993, pp. 5–17.
- [86] L. Whitley, "Fundamental principles of deception in genetic search," in *Proceedings of the First Workshop on Foundations of Genetic Algorithms*, G. Rawlins, Ed. Morgan Kaufmann, 1991, pp. 221–241.
- [87] J. Horn and D. Goldberg, "Genetic algorithm difficulty and the modality of fitness landscapes," in *Proceedings of the Third Workshop on Foundations of Genetic Algorithms*, L. Whitley and V. M.D., Eds. Morgan Kaufmann, 1994, pp. 243–269.
- [88] R. Page Jr and S. Mitchell, "Self-organization and the evolution of division of labor," *Apidologie*, vol. 29, pp. 171–190, 1998.
- [89] E. Bonabeau, G. Theraulaz, and J. Deneubourg, "Quantitative study of the fixed threshold model for the regulation of division of labour in insect societies," *Proc R Soc B*, vol. 263, pp. 1565–1569, 1996.
- [90] D. Tarapore, D. Floreano, and L. Keller, "Task-dependent influence of genetic architecture and mating frequency on division of labour in social insect societies," *Behavioral Ecology and Sociobiology*, vol. 64, no. 4, pp. 675–684, 2010.



**Pawel Lichocki** is pursuing his Ph.D. degree in computer, communication and information sciences working as a Doctoral Assistant in the Laboratory of Intelligent Systems, School of Engineering Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland. He received a B.Sc. (2007) and M.Sc. (2008) in Computer Science from Poznan University of Technology, with specialization in Intelligent Decisions Support Systems. His current research interest include evolution of natural and artificial societies (social insects, teams of robots, etc.) and bio-inspired control of multiagent systems.



**Steffen Wischmann** is a joint postdoctoral fellow at the Laboratory of Intelligent Systems, School of Engineering, Ecole Polytechnique Federale de Lausanne and at the Department of Ecology and Evolution, University of Lausanne. He received his M.Sc. in Biology from the Saarland University (Germany, 2003) and his PhD in Theoretical Biology from the University of Bonn (Germany, 2008). His main research interests include the proximate and ultimate mechanisms driving the evolution of intra- and interspecific social behavior, evolutionary

robotics, and information theoretic approaches to understand communication and cooperation.



**Laurent Keller** received the B.Sc. degree in Biology, M.Sc. degree in Biology and Ph.D. degree in Zoology from University of Lausanne, Lausanne, Switzerland, in 1983, 1985, and 1989, respectively. He is a Professor of Ecology and Evolution, and Head of the Department of Ecology and Evolution, Biophore, University of Lausanne, Lausanne, Switzerland. His research interests include the principles governing the evolution of animal societies and the ecological and evolutionary consequences of social life. In addition to publishing several research papers, he has edited the books *Queen Number and Sociality in Insects* in 1993 and *Levels of Selection in Evolution* in 1999. Dr. Keller was awarded the E. O. Wilson Naturalist Award in 2005.



**Dario Floreano** (SM05) received an M.A. in visual psychophysics at the University of Trieste in 1988, an M.Sc. in Neural Computation from the University of Stirling in 1992, and a PhD in Cognitive Systems and Robotics from University of Trieste in 1995. He is an Associate Professor and the Director of the Laboratory of Intelligent Systems, School of Engineering, Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland. His research interest includes the analysis and synthesis of bio-inspired adaptive systems, such as biomimetic robotics, neuromorphic engineering, and artificial evolution. He has published two books on these topics: *Evolutionary Robotics* with Stefano Nolfi (Cambridge, MA: MIT Press, 2000) and *Bio-Inspired Artificial Intelligence* with Claudio Mattiussi (Cambridge, MA: MIT Press, 2008).