

Finding an Optimal Team

Michał Okulewicz

Warsaw University of Technology,
 Faculty of Mathematics and Information Science,
 Koszykowa 75, 00-662 Warsaw, Poland
 M.Okulewicz@mini.pw.edu.pl

Abstract—This article proposes a metaheuristic optimization/social simulation approach to find the optimal team for a given type of the project. The quality of the team is assessed in a black-box optimization environment, where the optimized function acts as a metaphor of the project to be completed within the certain time limit (number of fitness function evaluations) and each fitness function evaluation is considered to be a metaphor of a unit task. The employees in a team are modeled according to the Belbin’s Team Roles and the Particle Swarm Optimization (PSO) is used as a teamwork framework algorithm, while Evolutionary Algorithm (EA) as an algorithm for controlling the set of Team Roles for team members and leaders. This approach has been tested in a scenario of a simulated self-organizing team, where each employee decides about his own actions. The results from the performed simulation suggest, that such teams perform best if their leader is one of the actual work-oriented roles. Additionally, some projects required significantly different set of roles than the average team, resulting in improvement of the specialized team’s performance over that of the average team.

I. INTRODUCTION

IN 1981 and 1988 Raymond Meredith Belbin proposed his Team Roles theory [1]. This theory identifies nine clusters of people’s behaviors expressed during the cooperation with other people. The clusters are build upon well-known psychometric factors and an individual’s temperament assessment. Although the initial way of assigning team roles for the people has been criticized [2] the theory has been commercially successful, resulting in emergence of the Belbin Associates (<http://www.belbin.com/>).

From the author’s personal team management experience, this theory has proven quite useful for choosing proper tasks assignment and making accurate tasks justification, especially in the lack of other motivation methods.

An automatic method of finding a group of people with such set of Team Roles, whom could efficiently complete a given type of project (easy/repeatable, with some obstacles but a known general method of approach, high-risk research and development etc.), would greatly lower the risk and cost of doing such project. Such method could be also combined with a methods of automatic project planning [3], [4] or reusing information from previous projects [5], providing a set of decision support systems for project managers. Although such an approach will always be prone to the errors resulting from the simplicity of behavior modeling, it is still useful because of the possibility of simulating a vast number of hypothetical situations [6] and observing the recurring patterns in the proposed team structure. Such a simple model allows for

capturing the general features of the team member behaviors with connection to the problem (e.g. analyzing the alternatives, following the state-of-the-art approach). Other applications of virtual agents simulation results used for improving business processes can be found in [7].

As the Particle Swarm Optimization (PSO) has been originally proposed as a social simulation framework, this article proposes a generalization of the PSO particle, which would act as a metaphor of an employee.

The rest of the paper is organized as follows. Section II describes the generalized PSO particle and the generalized PSO as the team simulation framework. Section III presents Evolutionary Algorithm (EA) as a method of changing team members and leaders Team Roles. Subsequently, section IV defines the Team Roles from the Belbin’s Theory. Sections V, VI and VII describe experiments setup and their results. Finally, section VIII concludes the paper.

II. PARTICLE SWARM OPTIMIZATION: TEAMWORK FRAMEWORK

As already mentioned, Particle Swarm Optimization (PSO) has been initially designed as an algorithm mimicking simple social behavior [8], quite similar to Reynold’s boids [9]. Also some recent research presents connection between complex network analysis and particle swarm behavior [10]. Therefore, it is well-suited to be interpreted as a sort of a multi-agent system providing a good framework for such simulated environment. Please note, that in the rest of the article the generalized PSO *particles* will be referred to as *employees*, if it will be more meaningful in the given context.

In order to allow *employees* to incorporate a different set of actions, modeling the behavior of an *employee* acting in a certain team role, the velocity update formula of the Standard PSO [11], has been generalized from choosing the best particle among particle’s neighbors to a various aggregation formulas (see Table I). This could result in a behaviors of particles like in Charged Particle Swarm Optimization [12] (in order to incorporate possibility of repulse [13] from certain individuals) and Civilization Algorithm / PSO with charisma (in order to incorporate possibility of special treatment of the information from the team leader):

$$\begin{aligned}
 v &= \omega v + \\
 & c_1 r_1 (x_{Best} - x) + \\
 & (min_{c_2} + c_2) r_2 (x_{NeighborAggregation} - x)
 \end{aligned} \tag{1}$$

$$P(X \in Neighbors(Y)) = \begin{cases} 1, & X = Y \\ \frac{p_{ts}(X) + p_{tl}(Y)}{2}, & Team(X) = Team(Y) \\ \frac{p_s(X) + p_l(Y)}{2}, & Team(X) \neq Team(Y) \\ \min(1, p_{ts}(X) + p_{tl}(Y)), & Team(X) = Team(Y) \wedge (Leader(X) \vee Leader(Y)) \\ \min(1, p_s(X) + p_l(Y)), & Team(X) \neq Team(Y) \wedge Leader(X) \wedge Leader(Y) \end{cases}$$

Fig. 1. The probabilities that X would be a neighbor of Y in a given iteration.

Where:

- v is a velocity vector of the particle,
- x is a current location of the particle,
- x_{Best} is a best location visited by the particle,
- $x_{NeighborAggregation}$ is a location computed (aggregated) on the base of information from all the neighbors of the particle (typically just the best location),

Additionally:

- ω is an inertia coefficient,
- c_1 is a local attraction factor,
- c_2 (with additional min_{c_2} bias) is a neighbor attraction (could be used as repulsion) factor.
- r_1, r_2 are vectors of uniformly distributed random variables.

The communication topology is a random one, with four probabilities defined for each role and a team leader modifier. The possibility of communication between two *employees* is established independently in each iteration, following the four probabilities:

- p_{ts} a probability of being neighbor of a teammate (speaking),
- p_{tl} a probability of neighboring a teammate (listening),
- p_s a probability of being neighbor of an employee from another team (speaking),
- p_l a probability of neighboring an employee from another team (listening).

In order to reduce the number of parameters of the simulation, 5 levels of probabilities have been introduced:

- 1) $p_{HIGH} = 0.8$
- 2) $p_{TYPICAL} = 0.5$
- 3) $p_{LOW} = 0.1$
- 4) $p_{VERY_LOW} = 0.01$
- 5) $p_{ZERO} = 0.0$

In addition, if a given employee is a leader he will have higher probabilities of communicating with other leaders and his team members. The detailed rules for computing probability of successful communication between two *employees* are presented in Fig. 1.

III. EVOLUTIONARY ALGORITHM: TEAM MANAGEMENT FRAMEWORK

For managing the teams an Evolutionary Algorithm (EA) has been used (thus creating a meta-optimization like algorithm, although used for creating a certain Team Roles set rather than just optimizing a function).

The **fitness function** for the EA is a median of the difference between the achieved value of the function and the target value of the function within the limited budget of function evaluations on 15 different instances of the same function. The fitness function for the team is computed from the best values achieved by the members of this team.

The **cross-over operator** of the EA switches the Team Role of the team leader from the first parent with the Team Role of the team leader from the randomly chosen second parent, thus creating the child team.

The **mutation operator** of the EA randomly changes a team member's Role into another Role.

The **selection operator** is based on the tournament selection between the parents and the offspring in order to maintain high diversity of the population. Please note, that the offspring could be result of the cross-over operation or mutation. In the case of cross-over randomly chosen child would compete with first of the parents. In the case of the mutation the cloned mutant will compete with the original individual.

IV. TEAM ROLES MODELS

In this section, a short description of each of the nine Team Roles will be given, summarized with the set of parameters (c_1, c_2 , neighbor aggregation method, communication probabilities) used for modeling behavior for each role (see Table I). Each role description starts with a short quote from the <http://www.belbin.com> website. For a detailed descriptions please refer to the works of the R.M.Belbin [14]. After that short description each agent would be described in terms of communication abilities and methods of processing the received information.

A. Plant

The first Team Role to be identified was the Plant (PL). The role was so-called because one such individual was planted in each team. They tended to be highly creative and good at solving problems in unconventional ways. [15]

PL is an individual worker with unorthodox ideas, but possibly some communication issues. PL works in isolation and tries to find new (uncommon) ways to solve the problems within the project. As a particle it acts as a charged particle in CPSO algorithm, repulsing from the average location of the best known results, thus searching for completely new solutions.

TABLE I
SUMMARY OF TEAM ROLES PARAMETERS

Team role	c_1	min_{c_2}	c_2	Aggregation method	p_{ts}	p_{tl}	p_s	p_l
Plant	1.4	-1.4	1.4	Average location	LOW	TYPICAL	ZERO	ZERO
Monitor Evaluator	0.0	0.9	0.2	Promising and not explored cluster centers	TYPICAL	TYPICAL	ZERO	ZERO
Coordinator	0.0	0.9	0.2	Subsequent cluster centers	TYPICAL	TYPICAL	LOW	LOW
Resource Investigator	0.0	0.9	0.2	Max distance	HIGH	HIGH	TYPICAL	TYPICAL
Implementer	1.4	0.0	1.4	Best neighbour	TYPICAL	TYPICAL	ZERO	ZERO
Completer Finisher	0.0	0.9	0.2	Best neighbour	LOW	VERY_LOW	ZERO	ZERO
Teamworker	0.0	0.0	1.4	Average location	TYPICAL	TYPICAL	LOW	LOW
Shaper	1.4	0.0	1.4	Best neighbour	HIGH	HIGH	LOW	LOW
Specialist	0.0	0.9	0.2	Function approximation	TYPICAL	TYPICAL	ZERO	ZERO

B. Monitor Evaluator

The Monitor Evaluator (ME) was needed to provide a logical eye, make impartial judgements where required and to weigh up the team's options in a dispassionate way. [15]

ME greatest ability is to separate facts from opinions and assess situation without emotional biases. MEs focus on elaborating on all the plausible alternatives leading to achieving project's objectives.

As a particle, ME visits the locations which seem to be not explored enough, while having quite a good overall fitness function value. ME uses the UCB1 [16] approach in continuous problem by clustering the samples gathered by other particles, and computing the average value of the fitness function combined with the size (measured in number of samples) of the cluster. It uses such evaluation to choose which area should be explored, thus maintaining an exploitation-exploration balance. This behavior could also be looked upon as a sharing mechanism known from evolutionary approach, where fitness function quality is divided by the number of nearby specimen.

C. Coordinator

Co-ordinators (CO) were needed to focus on the team's objectives, draw out team members and delegate work appropriately. [15]

CO's abilities concentrate around proper work division and tasks delegation.

As a particle CO explores the area of each of the samples' clusters in a subsequent manner. CO behavior is closely related to the work of the ME, but CO focuses its attention on each of the clusters regardless of its average function value.

D. Resource Investigator

When the team was at risk of becoming isolated and inwardly-focused, Resource Investigators (RI) provided inside knowledge on the opposition and made sure that the team's idea would carry to the world outside the team. [15]

RI gathers information about other teams results (through high probability of communicating with the members of other teams).

As a particle, RI explores the promising areas which are the furthest from its current location.

E. Implementer

Implementers (IMP) were needed to plan a practical, workable strategy and carry it out as efficiently as possible. [15]

IMPs might be looked upon as the backbone member of a team. In the simulation, IMP acts as a standard PSO particle, which simulates following a most natural strategy balancing a choice between best external information and best personal experience.

F. Completer Finisher

Completer Finishers (CF) were most effectively used at the end of a task, to "polish" and scrutinise the work for errors, subjecting it to the highest standards of quality control. [15]

CF likes to work on a task until it is properly finished. Therefore CF's listening ability is low, as CF is concentrated on his work (on the other hand, CF provides information about the progress). As a particle, CF is attracted to the best neighbor location, but due to low probability of being informed about a new location it explores one area for a longer period of time.

Due to ω factor it is expected to oscillate around a promising location (until finding a better one) with the smaller steps at each iteration, therefore acting similar to a Variable Neighborhood Search algorithm.

G. Teamworker

Teamworkers (TW) helped the team to gel, using their versatility to identify the work required and complete it on behalf of the team. [15]

TW tries to help people with their work. Although TW can communicate easily, such employee will not necessarily engage in a work related conversation.

As a particle TW is attracted by all of its neighbours working like a particle in Fully Informed PSO algorithm [17].

H. Shaper

Challenging individuals, known as Shapers (SH), provided the necessary drive to ensure that the team kept moving and did not lose focus or momentum. [15]

SH tries to finish project as fast as possible (possibly even at the cost of its quality). SHs like to influence the way other

TABLE II
AVERAGE FREQUENCY OF THE ROLES WHILE ACTING AS A TEAM MEMBER AND TEAM LEADER

Function	CF	CO	IMP	ME	PL	RI	SH	SP	TW
Team leader	0.45	0.04	0.11	0.03	0.03	0.05	0.12	0.08	0.08
Team member	0.23	0.09	0.12	0.09	0.08	0.09	0.10	0.11	0.09

TABLE III
AVERAGE FREQUENCY OF THE ROLES WHILE ACTING AS A TEAM LEADER FOR DIFFERENT FITNESS FUNCTIONS. THE FUNCTIONS ARE DIVIDED INTO 5 GROUPS: SEPARABLE, WITH LOW OR MODERATE CONDITIONING, WITH HIGH CONDITIONING, MULTI-MODAL WITH GLOBAL STRUCTURE, MULTI-MODAL WITH WEAK GLOBAL STRUCTURE [18]

Optimized function	CF	CO	IMP	ME	PL	RI	SH	SP	TW
f1	0.24	0.12	0.10	0.10	0.00	0.06	0.06	0.30	0.02
f2	0.38	0.00	0.20	0.00	0.00	0.02	0.30	0.10	0.00
f3	0.30	0.02	0.08	0.00	0.04	0.08	0.28	0.10	0.10
f4	0.42	0.02	0.14	0.00	0.08	0.02	0.30	0.02	0.00
f5	0.12	0.00	0.16	0.00	0.00	0.10	0.52	0.08	0.02
f6	0.72	0.00	0.10	0.00	0.00	0.00	0.18	0.00	0.00
f7	0.48	0.00	0.14	0.00	0.00	0.00	0.18	0.10	0.10
f8	0.76	0.02	0.04	0.12	0.00	0.00	0.00	0.04	0.02
f9	0.54	0.00	0.02	0.08	0.08	0.14	0.06	0.08	0.00
f10	0.60	0.16	0.06	0.00	0.00	0.00	0.00	0.18	0.00
f11	0.46	0.06	0.00	0.00	0.02	0.04	0.00	0.16	0.26
f12	0.46	0.12	0.22	0.02	0.00	0.00	0.16	0.00	0.02
f13	0.50	0.10	0.12	0.00	0.10	0.04	0.00	0.00	0.14
f14	0.52	0.00	0.02	0.02	0.12	0.10	0.16	0.00	0.06
f15	0.54	0.00	0.14	0.00	0.00	0.02	0.08	0.10	0.12
f16	0.70	0.02	0.18	0.00	0.00	0.00	0.00	0.10	0.00
f17	0.50	0.00	0.00	0.08	0.10	0.02	0.04	0.22	0.04
f18	0.56	0.08	0.08	0.02	0.02	0.14	0.00	0.00	0.10
f19	0.06	0.00	0.22	0.10	0.04	0.06	0.00	0.02	0.50
f20	0.42	0.00	0.22	0.00	0.10	0.00	0.20	0.00	0.06
f21	0.46	0.06	0.14	0.06	0.06	0.04	0.00	0.12	0.06
f22	0.26	0.18	0.18	0.00	0.00	0.04	0.18	0.14	0.02
f23	0.40	0.00	0.04	0.08	0.00	0.26	0.00	0.04	0.18
f24	0.51	0.00	0.16	0.04	0.00	0.00	0.16	0.00	0.13

people work (if SH thinks, that the project would benefit from this).

As a particle SH acts exactly as IMP, but will have higher communication probability, which will speed up the convergence of the particles, thus resulting in finishing work earlier (although possibly in a much worse than optimal location).

I. Specialist

It was only after the initial research had been completed that the ninth Team Role, Specialist (SP) emerged. In the real world, the value of an individual with in-depth knowledge of a key area came to be recognized as yet another essential team contribution. [15]

SP takes pride in being an expert and does not perform very well in cooperation, therefore as a particle SP observes the gathered samples and builds a set of linear models approximating promising areas of the optimized functions and then explores the area near the peak of approximating square functions.

V. ASSESSMENT ENVIRONMENT

As an assessment environment the GECCO (since 2009) and CEC (since 2015) BlackBox Optimization Benchmark set

[18] has been used. It consists of 24 functions divided into 5 different categories, which would allow to have comparison for a different project difficulty levels.

This way, additional objective information on the quality of the team's performance has been obtained by comparing it with standard optimization algorithms (although the research does not focus on this part).

VI. TESTS

The training of the teams has been done with 10 repetitions of the experiment with 9 teams with 10 teammembers each with 1000 PSO iterations and 20 EA iterations on the set of all 24 benchmark functions (15 instances of each function).

The results of the training provided a frequency of each role acting as a team leader or a team member for each of the functions in the final population within the EA algorithm. The specialized team's were tested against the team generated from the average frequency of roles from all the functions.

VII. RESULTS

The results of the average roles frequency found within the training phase are presented in Table II. It can be seen that the **Completer Finisher** (CO) has been chosen as the most frequent Team Role for both the team member and the team

TABLE IV

AVERAGE FREQUENCY OF THE ROLES WHILE ACTING AS A TEAM MEMBER FOR DIFFERENT FITNESS FUNCTIONS. THE FUNCTIONS ARE DIVIDED INTO 5 GROUPS: SEPARABLE, WITH LOW OR MODERATE CONDITIONING, WITH HIGH CONDITIONING, MULTI-MODAL WITH GLOBAL STRUCTURE, MULTI-MODAL WITH WEAK GLOBAL STRUCTURE [18]

Optimized function	CF	CO	IMP	ME	PL	RI	SH	SP	TW
f1	0.13	0.08	0.11	0.09	0.07	0.09	0.09	0.27	0.07
f2	0.25	0.06	0.16	0.08	0.07	0.06	0.18	0.08	0.07
f3	0.26	0.07	0.11	0.10	0.08	0.10	0.12	0.09	0.07
f4	0.22	0.09	0.14	0.08	0.08	0.09	0.15	0.08	0.07
f5	0.15	0.07	0.13	0.08	0.10	0.09	0.19	0.13	0.07
f6	0.29	0.07	0.14	0.08	0.10	0.06	0.14	0.07	0.05
f7	0.26	0.09	0.10	0.08	0.07	0.08	0.12	0.10	0.10
f8	0.32	0.06	0.11	0.12	0.08	0.07	0.08	0.09	0.08
f9	0.29	0.09	0.11	0.08	0.09	0.07	0.10	0.09	0.08
f10	0.31	0.10	0.09	0.09	0.08	0.07	0.06	0.12	0.09
f11	0.27	0.10	0.10	0.08	0.09	0.09	0.08	0.11	0.09
f12	0.22	0.10	0.15	0.07	0.07	0.09	0.10	0.10	0.08
f13	0.26	0.09	0.15	0.08	0.08	0.10	0.05	0.10	0.09
f14	0.27	0.07	0.09	0.09	0.10	0.10	0.10	0.08	0.09
f15	0.26	0.09	0.13	0.08	0.09	0.10	0.05	0.11	0.08
f16	0.23	0.09	0.12	0.08	0.07	0.11	0.10	0.10	0.10
f17	0.21	0.09	0.12	0.12	0.12	0.09	0.08	0.10	0.07
f18	0.24	0.10	0.12	0.07	0.08	0.11	0.08	0.10	0.10
f19	0.18	0.09	0.13	0.08	0.06	0.06	0.10	0.12	0.19
f20	0.19	0.08	0.11	0.08	0.11	0.09	0.14	0.09	0.10
f21	0.24	0.09	0.11	0.06	0.08	0.09	0.09	0.14	0.09
f22	0.18	0.13	0.13	0.09	0.07	0.10	0.11	0.11	0.09
f23	0.19	0.07	0.12	0.07	0.08	0.14	0.09	0.08	0.16
f24	0.22	0.09	0.13	0.08	0.05	0.09	0.11	0.08	0.14

Specialized teams

Average team

PSO with LM

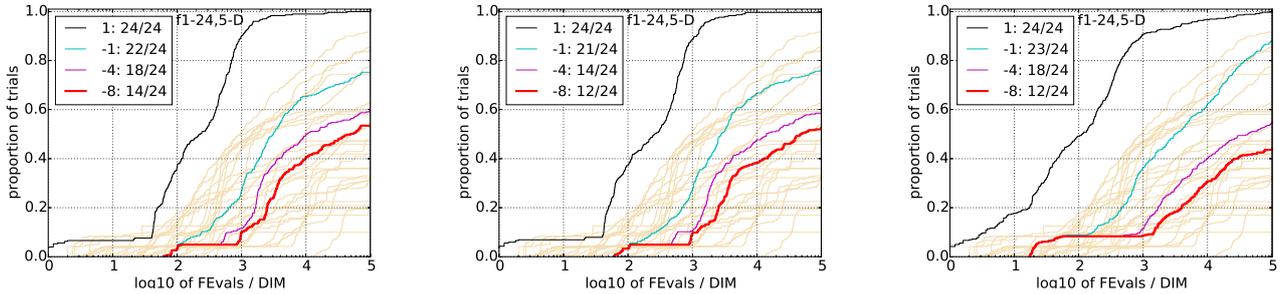


Fig. 2. Figures presenting the performance of the teams on a 5 dimension set of benchmark functions in terms of the fraction of runs (vertical axis) reaching certain optimization targets within the logarithm of certain number of function evaluations (horizontal axis). Additionally, a performance of a PSO algorithm enhanced with a linear model (approximating the optimized function with parabolas) is given.

leader. In the case of the team leaders the **Shapers** (SH) and **Implementers** (IMP) were chosen more frequently than the other 6 Team Roles, while in the case of the team members there seems to be no other significant distinction between average occurrences in the Team Roles except for the already mentioned CF. As can be seen from Tables III and IV the roles in the teams differ between the functions and the leader is always chosen among the most frequent of them.

The performance of the teams constructed from the average frequency of roles vs. specialized frequency of roles vs. PSO algorithm enhanced with a linear model with square function approximation is presented on Fig. 2. Although the performance in terms of the fraction of tries reaching 10^1 , 10^{-1} , 10^{-4} and 10^{-8} targets is practically indistinguishable, on the other hand the number of types of functions for which

certain optimization targets have been reached is higher for the specialized teams.

VIII. CONCLUSIONS

The proposed simulation showed the bias for team leaders towards the focused on the goal and hard-working team roles: Completer Finisher, Shaper, Implementer, Specialist and Teamworker. The results suggest that such leaders might lead to a best performance of a self-organizing team (as in the test framework each *employee* decided on its own about the task SP is going to perform). Additionally, from the average team members frequencies, the necessity for the representative of each of the roles among the team members has been stated. Some of the results, definitely still need a closer examination as they might be a result of a too simple implementation

(especially of the Monitor Evaluator and Coordinator roles). Also lack of discrimination between reported and real best position might lead to small or improper impact on the teamwork by the Monitor Evaluator, Coordinator and Resource Investigator roles. It proved beneficial not only to find a proper balance of roles and most likely roles to act as a team leaders, but also to find teams with that balance specialized for the given problems, as the specialized teams performed slightly better than the average teams (see the number of functions for which the optimization goals were achieved on the Fig. 2).

In addition it is important to observe, that the proposed algorithm achieved quite good results as an optimization algorithm. It has been able to find the optimum value with difference from the target optimum lower than 10^{-1} in case of almost all the functions and lower than 10^{-8} in case of more than the half of them, achieving better results for the hardest 10^{-8} goal than a reference enhanced PSO. It should also be noted, that the algorithm has been tuned on the basis of performance with around 4.25^{10} fitness function evaluations budget, which can be observed in comparison to PSO performance (see Fig. 2).

Future work

The future work should consist of building a more advanced simulation environment. It would be beneficial for the actions and interactions of the artificial agents actions to be related to the actions needed to complete some abstract project which can be later executed by human agents team, constructed according to the simulations results.

Additionally, the *employees* models might take into the account such possibilities as:

- migration of employees within teams set (instead of just changing into another role),
- conflicts between the team roles approach to work (resulting in losing team members and the information gathered by them),
- reporting different achieved results in order to direct other *employees* to search given locations,
- *employees* being a probabilities vector of acting as a given role rather than just one team role.

The studies on the test environment itself might include:

- using asynchronous particles with time limit instead of a synchronized swarm with iterations budget bound,
- managing the team roles with another type of fitness function (taking into account cooperation of teams).

REFERENCES

- [1] R. M. Belbin, *Management teams: why they succeed or fail*, 1st ed. Routledge, 1981.
- [2] A. Furnham, H. Steele, and D. Pendleton, "A psychometric assessment of the Belbin Team-Role Self-Perception Inventory," *Journal of Occupational and Organizational Psychology*, vol. 66, no. 3, pp. 245–257, 1993. doi: 10.1111/j.2044-8325.1993.tb00535.x. [Online]. Available: <http://dx.doi.org/10.1111/j.2044-8325.1993.tb00535.x>
- [3] K. Walędzik, J. Mańdziuk, and S. Zadrozny, "Proactive and reactive risk-aware project scheduling," in *Computational Intelligence for Human-like Intelligence (CIHLI), 2014 IEEE Symposium on*. IEEE, 2014. doi: 10.1109/CIHLI.2014.7013392 pp. 94–101. [Online]. Available: <http://dx.doi.org/10.1109/CIHLI.2014.7013392>
- [4] —, "Risk-aware project scheduling for projects with varied risk levels," in *Computational Intelligence, 2015 IEEE Symposium Series on*. IEEE, 2015. doi: 10.1109/SSCI.2015.231 pp. 1642–1649. [Online]. Available: <http://dx.doi.org/10.1109/SSCI.2015.231>
- [5] Ł. Osuszek and S. Stanek, "Case based reasoning as an improvement of decision making and case processing in adaptive case management systems," in *Position Papers of the 2015 Federated Conference on Computer Science and Information Systems*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. Maciaszek, and M. Paprzycki, Eds., vol. 6. PTI, 2015. doi: 10.15439/2015F61 pp. 217–223. [Online]. Available: <http://dx.doi.org/10.15439/2015F61>
- [6] K. M. Carley, "Computational organizational science and organizational engineering," *Simulation Modelling Practice and Theory*, vol. 10, no. 57, pp. 253 – 269, 2002. doi: [http://dx.doi.org/10.1016/S1569-190X\(02\)00119-3](http://dx.doi.org/10.1016/S1569-190X(02)00119-3) Organisational Processes. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1569190X02001193>
- [7] M. Żytniewski, A. Sołtysik, A. Sołtysik-Piorunkiewicz, and B. Kopka, "Modeling of software agents' societies in knowledge-based organizations. the results of the study," in *Proceedings of the 2015 Federated Conference on Computer Science and Information Systems*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. Maciaszek, and M. Paprzycki, Eds., vol. 5. IEEE, 2015. doi: 10.15439/2015F216 pp. 1603–1610. [Online]. Available: <http://dx.doi.org/10.15439/2015F216>
- [8] R. C. Eberhart, J. Kennedy *et al.*, "A new optimizer using particle swarm theory," in *Proceedings of the sixth international symposium on micro machine and human science*, vol. 1. New York, NY, 1995. doi: 10.1109/MHS.1995.494215 pp. 39–43. [Online]. Available: <http://dx.doi.org/10.1109/MHS.1995.494215>
- [9] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *ACM SIGGRAPH computer graphics*, vol. 21, no. 4. ACM, 1987. doi: 10.1145/37401.37406 pp. 25–34. [Online]. Available: <http://dx.doi.org/10.1145/37401.37406>
- [10] R. Šenkerik, M. Pluháček, A. Viktorin, and J. Janošík, "On the application of complex network analysis for metaheuristics," in *7th BIOMA Conference*, 2016, pp. 201–213. [Online]. Available: <http://bioma.ijs.si/proceedings/2016/14%20-%20On%20the%20Application%20of%20Complex%20Network%20Analysis%20for%20Metaheuristics.pdf>
- [11] M. Clerc, "Standard Particle Swarm Optimization. From 2006 to 2011," 09 2012. [Online]. Available: http://clerc.maurice.free.fr/ps0/SPSO_descriptions.pdf
- [12] T. M. Blackwell and P. J. Bentley, "Dynamic search with charged swarms," in *Proceedings of the Genetic and Evolutionary Computation Conference*, ser. GECCO '02. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002. ISBN 1-55860-878-8 pp. 19–26. [Online]. Available: <http://dl.acm.org/citation.cfm?id=646205.682961>
- [13] B. L. J. Zhang, C. Fan and F. Shi, "An Improved Particle Swarm Optimization Based on Repulsion Factor," *Open Journal of Applied Sciences*, vol. 2, no. 4B, pp. 112–115, 2012. doi: 10.4236/ojapps.2012.24B027. [Online]. Available: <http://dx.doi.org/10.4236/ojapps.2012.24B027>
- [14] R. M. Belbin, *Team roles at work*, 2nd ed. Routledge, 2012.
- [15] BELBIN Associates, "Belbin Team Roles," 05 2015. [Online]. Available: <http://www.belbin.com/>
- [16] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002. doi: 10.1023/A:1013689704352. [Online]. Available: <http://dx.doi.org/10.1023/A:1013689704352>
- [17] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: simpler, maybe better," *Evolutionary Computation, IEEE Transactions on*, vol. 8, no. 3, pp. 204–210, 2004. doi: 10.1109/TEVC.2004.826074. [Online]. Available: <http://dx.doi.org/10.1109/TEVC.2004.826074>
- [18] A. Auger, N. Hansen, M. Schoenauer, O. A. Elhara, A. Atamna, D. Brockhoff, A. Liefooghe, T.-D. Tran, R. Datta, R. L. Riche, E. Touboul, X. Bay, X. Delorme, D. Fongang-Fongang, H. Mohammadi, D. Villanueva, G. Rudolph, M. Preuss, H. Trautmann, O. Mersmann, B. Bischl, and T. Wagner, "Comparing Continuous Optimisers: COCO," 01 2015. [Online]. Available: <http://coco.gforge.inria.fr/>