Neurocomputing 149 (2015) 993–1002

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Planning the sports training sessions with the bat algorithm

Iztok Fister [a], Samo Rauter [b], Xin-She Yang [c], Karin Ljubič [d], Iztok Fister Jr [a,*]

[a] University of Maribor, Faculty of Electrical Engineering and Computer Science, Smetanova 17, 2000 Maribor, Slovenia
[b] University of Ljubljana, Faculty of Sport, Gortanova 22, 1000 Ljubljana, Slovenia
[c] School of Science and Technology, Middlesex University, United Kingdom
[d] University of Maribor, Faculty of Medicine, Taborska 8, 2000 Maribor, Slovenia

ARTICLE INFO

Article history:
Received 19 December 2013
Received in revised form
7 July 2014
Accepted 23 July 2014
Communicated by Swagatam Das
Available online 6 August 2014

Keywords:
Sport training
Training load
Bat algorithm
Swarm intelligence

ABSTRACT

Planning proper sports training has always been a very challenging task for coaches. In line with this, they need to have almost two special abilities: firstly, to have a lot of earlier experiences with sports training and secondly, to know the capability of their athletes very well. New ways in planning sports training have emerged with development of pervasive and mobile technologies. Recently, a GPS receiver is one of the most useful parts of each standard sports watch that enables athletes to track the duration of their sports activities and analyze them later on digital computers using GPS viewers. Most sport's watches are also capable of measuring an athlete's heart rate during activities. Both measures represent reliable data sources that can be used for planning the sports trainings by coaches. In this paper, we introduce a novel intelligent planning method for sports training sessions, where the training plans are generated on digital computers using the bat algorithm according to reliable data obtained from sports watches. Real-world experiments showed promising results that encouraged us to proceed with this research also in the future.

1. Introduction

Athletes today are having a lot of possibilities to improve their training performance and therefore can be better prepared for competitions. Nowadays, a big part of the sports life of every athlete is connected with training technologies [1]. These technologies are composed of:

- smart sport watches,
- power and cadence meters,
- different variants of heart rate monitors,
- different variants of music players,
- and many more.

Sport watches probably represent the most important pieces of these technologies. The first interest in sport watches arose in the middle of the 90s when the Finnish company Polar released very powerful sport watches with heart rate monitor and timer for measuring the duration of sport activities. These watches have the following functions:

- monitoring the total duration of a sports activity,
- watching current and maximum heart rates,
- monitoring current elevation and total ascent,
- current temperature,
- saving activities on sport watch.

Additionally, cyclists have also obtained some sensors which were put on the bikes and are capable of monitoring the speed. Furthermore, later variants of these watches provided connection with digital computers and in line with this, an online analysis of workouts.

These sports watches had been one of the more important training tools for every amateur and professional athlete in the past. Then, the training technology made a big step forward. An expansion of the GPS technology allowed many companies to develop sports watches with GPS receivers [2]. These watches have huge advantages over previous generations of watches, because they measure very precisely the characteristics data about trainings when athletes using the GPS receiver. As a result, runners and bicyclists do not need to use any special sensors for determining their speed, altitude and duration of activities. Currently, companies like Garmin, Polar and Suunto are making serious efforts to develop additional options for such watches in order to meet the needs of athletes worldwide.

* Corresponding author.
E-mail addresses: iztok.fister@uni-mb.si (I. Fister), samo.rauter@fsp.uni-lj.si (S. Rauter), x.yang@mdx.ac.uk (X.-S. Yang), karin.ljubic@uni-mb.si (K. Ljubič), iztok.fister2@uni-mb.si (I. Fister Jr).

http://dx.doi.org/10.1016/j.neucom.2014.07.034
0925-2312/

Certainly, connection with digital computers and online analysis of trainings are the greatest advance of these sport watches. For instance, Garmin Connect web service developed a majestic online training service, where users can analyze their workouts after their performed activities. The web service also encourages athletes to practice more in order to reach better results on the official competitions. On the other hand, the results of these workouts can also be exported into a digital computer in XML form and analyzed later.

An aim of this paper is to propose an intelligent planning for sports training using a digital computer on the basis of reliable data exported from sports watches in the form of XML activity files [3]. Essentially, two measures are important for this planning: a duration of activity and an average heart rate. Both measures are precisely measured using the sports watches which saves data about specific activities in their internal memory and can be exported onto a digital computer for the further analyses. On the basis of these exported data, the plan of sports training for specific athlete is performed using a bat algorithm.

The bat algorithm belongs to a class of Swarm Intelligence (SI) [4–6]. This algorithm arose in the year 2010, when Yang [7] created a new optimization algorithm inspired by the behavior of micro-bats that use a special mechanism called echolocation. Echolocation is used by bats for orientation and prey finding. The original bat algorithm was applied to various benchmark functions, where have achieved solid results. The convergence rate of this algorithm was improved in the study [8], where the authors hybridized the original bat algorithm with differential evolution strategies (HBA). In the study [9] the same authors dealt with hybridizing the bat algorithm using differential evolution strategies [11] and a random forests machine learning method (HBARF). The complete survey regarding bat algorithm can be found in [12].

The proposed algorithm for planning the sports training sessions is able to create the training plan for a given training period of a specific athlete. It starts with a set of base trainings characterized by different durations and average intensities determined by the average heart rate. The base training sessions are selected by the coaches on the basis of XML activity files obtained in the past. Thus, it is assumed that the long-duration training sessions are more suitable for the starting periods of training, while the more intensive short-duration training sessions need to be performed during the more matured period of training, when the athlete is already fit. Indeed, the number of fully intensive training sessions is kept to a minimum.

Although there are a lot of different commercial tools for tracking sports activities, as we know there is no system for the intelligent planning of sports training on a digital computer. Our proposed algorithm is therefore mainly devoted to coaches as a help tools for planning the optimal training period for the specific athlete preparing himself/herself for upcoming competitions. The results of this algorithm conform with the expectations of professional coaches and therefore open-up a lot of potentials for the further development.

The structure of this paper is as follows. Section 2 introduces the basics of a sport training. Here, the mathematical model is developed for planing the sports training. In Section 3, the bat algorithm for planning the sports training sessions is discussed in detail. Experiments and results are the subject of Section 4. Section 5 provides a conclusion, where our work is summarized and the directions for the further developments are outlined.

2. Sports training

According to the definition the sports training is a process built on scientific and pedagogical principles, which affects the performance of an athlete using planned and systematic training sessions, thus allowing him/her to strive for the highest achievements [13]. The ultimate effect of the supplied process of sports training can be seen in an athlete's improved form, the increased capacity of his/her body or/and in the worst case, over-training. In relation to the expected competitive performance, the sport form can be described as a phenomenon of short-term increased capacity by the athlete. An athlete's achieved sport form in an expected competition means that his process of sport training was the most effective. The efficient process of sports training is also described as a qualitative approach that optimizes an athlete's time dedicated to training. Namely, the frequency of sports exercise is always in conflict with its intensity. In other words, a large amounts of time spent exercising could not be carried out very intensively.

Most athletes use heart rate monitors for measuring the sports training intensity. The time duration (TD) measured by a stop-watch and the intensity of the workout measured by a heart rate monitor (HR) are the simple metrics for monitoring the difficulty of sport training. Banister [14] has made step forward towards smart training analysis during training sessions using the method "TRAining IMpulse" (TRIMP) for quantifying the training load. Nowadays, power meters are more often used by bicyclists for evaluating the difficulty of sport training intensity instead of the physiological parameter HR. In line with this, Friel [10] for this purpose proposed a measure Training Stress Score (TSS) as a way of expressing the difficulty from a training session regarding workload. The measure can be expressed as a product of the intensity and duration necessary to accomplish the workout.

2.1. Mathematical definition of the problem

The problem of planning the sports training can be mathematically defined as follows. Let us assume a set of base training sessions $T=\{t_1,\ldots,t_n\}$ characteristics for certain athlete. Each base training session is specified by the athlete's average heart rate $\overline{HR}_i$ and the time duration TD_i, which determine the profit rate as $p_i=c_i\cdot\overline{HR}_i/TD_i$, where $c_i=\overline{HR}_i/\overline{HR}$ and $\overline{HR}=1/n\cdot\sum_{j=1}^{n}\overline{HR}_j$. Here, a coefficient c_i is used to normalize the values of $\overline{HR}_i$. The $\overline{HR}_i$ is measured by minutes (min), while the TD_i by beats per minutes (BpM), counting the number of heart beats per minutes. The task of the optimal planning the sports training is to find the integer vector $\mathbf{y}=\{y_1,\ldots,y_n\}$ determining the training plan such that the error rate (er) is minimized, in other words

$$er^*=\min|K-hr|, \tag{1}$$

where K the intensity factor that prescribed the maximal heart rate reached during the training period, and hr is defined as follows:

$$hr=\frac{1}{n}\sum_{i=1}^{n}\overline{HR}_i\cdot y_i\le K,\quad y_i\in[1,n]. \tag{2}$$

Note that hr denotes the calculated heart rate hr obtained by a specific training plan $\mathbf{y}$. In practice, coaches would like that the calculated heart rate hr is as close to the prescribed intensity factor K as possible. As a result, when the ideal solution is found the calculated heart rate hr matches value of the intensity factor. In that case, the error rate hr must be zero.

Additionally, a training effort value ev is calculated for each training plan that estimates an average effort needed by an athlete to finish it. The training effort value is expressed as

$$ev=\max\sum_{i=1}^{n}p_i\cdot y_i. \tag{3}$$

The vector $\mathbf{y}$ in Eq. (3) counts the number of the same training sessions y_i, where the efficiency of i-th training session depends on the intensity $\overline{HR}_i$ and time duration TD_i. Both metrics are inversely related. This means that the intensity of training decreases when the time duration increases, and vice versa. In Eq. (3), this relationship is captured by introducing the profit rate p_i.

In summary, training sessions of less intensity and longer duration are preferred by this equation. Here, coaches are interested in how to determine a sensible number of base training sessions in order to bring the athlete in the full form. Note that the average training effort is indirectly regulated by the intensity factor K. When the athlete is near to optimal form the parameter K is set below the average heart rate of trainings included into the base training set $\overline{HR}$, in other words, $K < \overline{HR}$, where $\overline{HR} = \sum_{i=1}^{n} \overline{HR}_i$.

3. Swarm intelligence and the bat algorithm

The hardest optimization problems (also NP-hard problems [15]) cannot be solved exactly because of their time as well as space complexity. Therefore, meta-heuristic algorithms have been more popular became to solve these problems approximately. Usually, these algorithms take an inspiration for their operation from nature. There are two main categories of nature-inspired algorithms, i.e., evolutionary algorithms [16] and swarm intelligence [5]. The former imitates a Darwin's evolutionary theory [17], where the fitter individuals have more chances to survive in a struggle for survival, while for the latter, an inspiration presents on the first look simple creatures that are able to perform some inherent actions. However, together in group, these creatures i.e., social insects and animals, are suitable to perform complex tasks, e.g. building magnificent nests by termites. Mainly, some important swarm intelligence algorithms are as follows: Particle Swarm Optimization (PSO) [18], Artificial Bee Colony (ABC) [19–21], Firefly Algorithm (FA) [22,23], Ant Colony Optimization (ACO) [24].

The original bat algorithm is one of the younger members of the swarm intelligence. It was developed in 2010 by Yang [7]. The inspiration for his work was micro-bats and their special mechanism named echolocation. Echolocation is a mechanism uses by bats for orientation and finding their preys. Bats are not the only creatures using such a mechanism.

The original bat algorithm is population based, where each individual represents the candidate solution. The candidate solutions are represented as vectors $\mathbf{x_i} = (x_{i1}, \ldots, x_{in})^T$ for $i = 1 \ldots Np$ with real-valued elements $\{x_{ij}\}$, where each elements can capture values from interval $x_{lb} \ldots x_{ub}$. Thus, x_{lb} and x_{ub} denote the corresponding lower and upper bounds, while the population size is determined by Np parameter.

Algorithm 1. Original bat algorithm.

Input: Bat population $\mathbf{x_i} = (x_{i1}, \ldots, x_{in})^T$ for $i = 1 \ldots Np$, *MAX_FE*.
Output: The best solution $\mathbf{x}_{best}$ and its corresponding value $f_{min} \min(f(\mathbf{x}_i))$.
1: init_bat();
2: *eval* = evaluate_the_new_population;
3: f_{min} = find_the_best_solution($\mathbf{x}_{best}$); {initialization}
4: **while** termination_condition_not_meet **do**
5: **for** $i = 1$ **to** Np **do**
6: $\mathbf{z}$ = generate_new_solution ($\mathbf{x}_i$);
7: **if** rand(0, 1) $> r_i$ **then**
8: $\mathbf{z}$ = improve_the_best_solution ($\mathbf{x}_{best}$)
9: **end if** {local search step}
10: f_{new} = evaluate_the_new_solution($\mathbf{z}$);
11: $eval = eval + 1$;
12: **if** $f_{new} \le f_i$ **and** $N(0,1) < A_i$ **then**
13: $\mathbf{x}_i = \mathbf{z}$; $f_i = f_{new}$;
14: **end if** {save the best solution conditionally}
15: f_{min} = find_the_best_solution($\mathbf{x}_{best}$);
16: **end for**
17: **end while**

The pseudo-code of the original bat algorithm is illustrated in Algorithm 1, where bats' behavior is captured within the fitness function of the problem to be solved. The original bat algorithm consists of the following components [25]:

- *initialization* (lines 1–3): initializing the algorithm parameters, generating the initial population, evaluating this, and finally, determining the best solution $\mathbf{x}_{best}$ in the population,
- *generate_the_new_solution* (line 6): moving the virtual bats in the search space according to the physical rules of bat echolocation,
- *local_search_step* (lines 7–9): improving the best solution using random walk heuristic [26],
- *evaluate_the_new_solution* (line 10): evaluating the new solution,
- *save_the_best_solution_conditionally* (lines 12–14): saving the new best solution under some probability A_i similar to simulated annealing [27],
- *find_the_best_solution* (line 15): finding the current best solution.

Initialization of the bat population is performed randomly. Then, each candidate solution is evaluated. Note that *evaluate_the_new_population* calculates the fitness for each virtual bat in the population using *evaluate_the_new_ solution* function. Generating the new solutions is performed according to the following equations:

$$
\begin{aligned}
Q_i^{(t)} &= Q_{min} + (Q_{max} - Q_{min}) N(0,1), \\
\mathbf{v}_i^{(t+1)} &= \mathbf{v}_i^t + (\mathbf{x}_i^t - \mathbf{best}) Q_i^{(t)}, \\
\mathbf{x}_i^{(t+1)} &= \mathbf{x}_i^{(t)} + \mathbf{v}_i^{(t+1)},
\end{aligned}
\tag{4}
$$

where $N(0,1)$ is a random number drawn from a Gaussian distribution with zero mean and a standard deviation of one. A random walk heuristic [26] implemented in the function *improve_the_best_solution* modifies the current best solution according to the equation:

$$
\mathbf{x^{(t)}} = \mathbf{best} + \epsilon A_i^{(t)} N(0,1), \tag{5}
$$

where $N(0,1)$ denotes the random number drawn from a Gaussian distribution with zero mean and a standard deviation of one, ϵ being the scaling factor, and $A_i^{(t)}$ the loudness. A local search is launched with the probability of pulse rate r_i. As already stated, the probability of accepting the new best solution in the component *save_the_best_solution_conditionally* depends on loudness A_i. Actually, the original BA algorithm is controlled by two algorithm parameters: the pulse rate r_i and the loudness A_i. Typically, the rate of pulse emission r_i increases and the loudness A_i decreases when the population draws nearer to the local optimum. Both characteristics imitate natural bats, where the rate of pulse emission increases and the loudness decreases when a bat finds a prey. Mathematically, these characteristics are captured using the following equations:

$$
A_i^{(t+1)} = \alpha A_i^{(t)}, \quad r_i^{(t)} = r_i^{(0)} [1 - \exp(-\gamma \epsilon)], \tag{6}
$$

where α and γ are constants. Actually, the α parameter plays a similar role as a cooling factor in the simulated annealing algorithm that controls the convergence rate of this algorithm.

3.1. Modified bat algorithm for the planning sports training sessions

In an early stage of the emergence, the original bat algorithm has been applied primarily to continuous optimization problems [28–32]. Recently, some tries to use this algorithm also by solving the discrete optimization problems that have been emerged [33,34]. A problem of planning the sports training sessions is a typical discrete problem that had never been solved using the computer technology before.

Although many other evolutionary [35–39,11] and swarm intelligence algorithms [20,21,23] could be applied for solving the mentioned problem, in this paper, the bat algorithm is used essentially because of its simplicity. In fact, this algorithm demands only few control parameters and it exposes a low time complexity solving the real-world problems.

Some modifications of the original bat algorithm are needed in order to prepare it for the planning sports training sessions. That is, the candidate solutions in the original bat algorithm are represented as real-valued vectors $\mathbf{x}$, while the problem of planning the sports training sessions demands an integer vectors $\mathbf{y}$ denoting the number of training sessions. Therefore, some mapping between representation of solution in real-valued search space to the solution in a problem space is needed, where it is represented as an integer vector. In the proposed modified bat algorithm (MBA), this mapping is performed as follows.

A candidate solution in the proposed modified bat algorithm (MBA) is represented using the real-valued vector $\mathbf{x}_i = \{x_i1, \ldots, x_in\}^T$ for $i = 1 \ldots n$ with elements $x_{ij} \in [0, 1]$, same as the original bat algorithm. The evaluation function is calculated as follows. At first, the permutation of base trainings $\boldsymbol{\pi}_i = \{\pi_{i1}, \ldots, \pi_{in}\}$ is mapped from the vector $\mathbf{x}_i$ such that the following relation is valid:

$$x_{i\pi_{i1}} < x_{i\pi_{i2}} < \ldots < x_{i\pi_{in}}. \tag{7}$$

Then, the number of base training sessions $y_i = \{y_{i1}, \ldots, y_{in}\}^T$ is calculated according to the equation:

$$y_{ij} = \lceil \frac{(n - \pi_{ij}) \cdot m}{N} \rceil, \quad \text{subject to} \quad \sum_{i=1}^{n} y_i = m, \tag{8}$$

where $N = n \cdot (n+1)/2$ counts the sum of a sequence of numbers from $1 \ldots n$, $(n - \pi_{ij})$ denotes weights, and m determines the number of training days (also a training period). Thus, the sum of all base training sessions must be equal to the training period m. Furthermore, all base trainings must be considered in each training plan. Finally, the training effort value is determined for the candidate solution according to Eq. (3).

An example of mapping the candidate solution $\mathbf{x}_i$ via permutation of the base training sessions $\boldsymbol{\pi}_i$ to the final training plan $\mathbf{y}_i$ for a training period of the 30 days is illustrated in Table 1.

Let us suppose the base training set with five base trainings and profits $\mathbf{p} = \{1.31, 0.43, 0.52, 4.48, 1.91\}$. Then, the training effort value for the corresponding training plan $\mathbf{y}_i$ is determined by $ev = 68.90$.

3.2. Repairing the infeasible solutions

Usually, candidate solutions that are constructed according to Eq. (1) are infeasible, i.e., they violate a condition that limits the calculated heart rate hr of constructed training plan to be below the intensity factor K. In other words, each constructed solution must satisfy the condition $hr \le K$. When the infeasible solution is detected it is repaired [40] in order to become feasible. An algorithm illustrated in Algorithm 2 is applied to MBA for planning the sports training sessions to handle the infeasible solutions.

Table 1
Candidate solution mapping.

	Dimension j				
Elements i	1	2	3	4	5
Candidate solution $\mathbf{x}_i$	0.30	0.51	0.11	0.98	0.77
Permutation $\boldsymbol{\pi}_i$	4	3	5	1	2
Training plan $\mathbf{y}_i$	4	6	2	10	8

Algorithm 2. MBA repairing algorithm.

Input: hr, $\mathbf{y}$ - infeasible solution.
Output: hr, $\mathbf{y}$ - feasible solution, if $hr \le K$.
1: $t = 0$; $tl = \emptyset$; $max\_k = 0$;
2: **while** $t < d$ **and** $hr > K$ **do**
3: $i = \text{rand}(1, d) \wedge i \notin tl$;
4: **for** j = 1 **to** d **do**
5: $dif_j = \text{try\_swap}(\mathbf{y}, i, j)$;
6: **if** $dif_j > max\_k$ **then**
7: $max\_k = dif_j$;
8: $k = j$;
9: **end if**
10: **end for**
11: **if** $k \ge 0$ **then**
12: $\text{do\_swap}(\mathbf{y}, i, k)$;
13: $hr = hr - max\_k$;
14: **end if**
15: $tl = tl \cup i$;
16: $t = t + 1$;
17: **end while**
18: **return** hr;

An input of MBA repairing algorithm (Algorithm 2) represents the infeasible solution $\mathbf{y}$ with its calculated heart rate. An output consists of the same variables, but the repaired candidate solution can be either the feasible when the condition $hr \le K$ is satisfied or infeasible when the satisfaction condition is not met.

The algorithm consists of two loops. The outer loop (lines 2–17) is executed until all of the i-th base trainings are exhausted or the satisfaction condition is met. In the inner loop (lines 4–10), all of the j-th base trainings are selected one by one. Then, a pair of base training (i,j) is constructed and the difference of the average heart rates between the original and the modified permutation obtained by swapping the i-th and j-th base trainings in the original permutation π is identified. As a result, the inner loop is terminated when the best move (i,k) having the maximal difference max_k is obtained. This move is then applied to the original permutation.

4. Experiments and results

The aim of our experimental work was to show that the algorithm for planning the sports training is capable to create plans for sports training sessions of the similar quality as those of the coaches. The experiments relied on the data obtained during the real training process by an amateur cyclist over more than four years. During this period, more than 1000 files were produced using the sport watches and transferred from them to the computer in the form of XML activity files. Planning the number of base training sessions was performed in a period of 2 months, in which the specific athlete has prepared for the cycling National Championship.

Ten base training sessions characteristic for the specific athlete were selected by the coaches. Thus, the task of coaches was also to prescribe the everyday training schedule in such a way that all the base training sessions from the plan would be included. At the end of the training period, it was expected that the athlete would be fit and prepared for the competition.

Four experiments were conducted in order to check the suitability of the proposed method as follows:

- constructing the training plans,
- investigating the influence of the fitness function evaluations,
- determining where the optimization makes sense,
- identifying the impact of the fitness function evaluations on the convergence of results,
- comparing the results of the MBA with the other well known evolutionary and swarm intelligence algorithms.

The parameter setting of the MBA algorithm during the experiments was as follows. The size of base training set was set at $n=10$, while the training period lasted $m=55$ days (2 months with 5–6 rest days). The population size was limited to $Np=40$. The algorithm's parameters pulse rate and loudness were fixed at the values $r=0.2$ and $A=0.8$, while the frequency rates were limited by their minimum and maximum values drawn in the interval $Q \in [0.0, 2.0]$. The problem variables occupied values in the interval $x_i \in [0, 1]$. The algorithm was run 25 times. As a termination condition, the numbers of fitness function evaluations (FEs) were used. These values of control parameters were set after extensive testing. The task of the planning was to find the training plan with the error rate zero and the maximum training effort value.

The MBA algorithm is stochastic and therefore, the different results are obtained in each run. In order to measure this phenomenon, the standard statistical measures, like minimum, maximum and average values together with the median, and standard deviation of error rates were obtained over 25 independent runs. On the other hand, decreasing or increasing the intensity factor K over specific bounds causes that the solution cannot be found in some runs when. However, the results of the MBA can be improved by repairing infeasible solutions.

In the remainder of this paper, creating the base training set is presented in detail. Then, the results of the experiments are illustrated. This section concludes with a discussion.

4.1. Creating the base training set

The base training set was created from the several characteristic XML activity files which were created for the specific athlete in the past. Although the numbers of these XML activity files had increased enormously over the time, the coaches selected only ten files in order to complete a base training set. The files in this set were identified by their own identification numbers, and were distinguished from each other according to the time duration and the average heart rate. Additionally, an appropriate profit value was assigned to each file that was used in the planning process.

The format of XML activity files is standardized and therefore, compatible with the major of sports application that are available on different web portals and mobile devices [41]. Using these sports application, we can analyze the data obtained by our training, which provide a unique form of virtual communication, virtual competitions or even help us in sports training advertising.

The activities saved in XML files are also portable between many web services and portals. Algorithm 3 illustrates a small part of the sample activity. In this piece of code, the activity is defined and then the statistics data about it are summarized.

Algorithm 3. Activity in XML.

```
〈 Activity Sport="Biking"〉
 〈 Id 〉 2011-08-02T07:26:02.000Z 〈/Id 〉
 〈 Lap StartTime="2011-08-02T07:26:02.000Z"〉
  〈 TotalTimeSeconds 〉 6000.06 〈/TotalTimeSeconds 〉
  〈 DistanceMeters 〉 47329.76 〈/DistanceMeters 〉
  〈 MaximumSpeed 〉 16.06399917602539 〈/MaximumSpeed 〉
  〈 Calories 〉 1547 〈/Calories 〉
  〈 AverageHeartRateBpm 〉
   〈 Value 〉 130 〈/Value 〉
  〈/AverageHeartRateBpm 〉
  〈 MaximumHeartRateBpm 〉
   〈 Value 〉 171 〈/Value 〉
  〈/MaximumHeartRateBpm 〉
```

Each activity starts with an identification of the sport discipline where it was tracked. In XML notation, this data is embraced within the 〈 Activity Sport 〉 tags. Essentially, the following data are important for this study: time duration in seconds, identified within 〈 TotalTimeSeconds 〉 and 〈/Total TimeSeconds 〉 tags, and average heart rate within 〈 AverageHeartRateBpm 〉 and 〈/AverageHeartRateBpm 〉 tags.

Algorithm 4. Parsed data from XML.

01	100.00	130
02	353.88	151
03	257.30	133
04	29.68	133
05	66.45	127
06	61.30	168
07	215.13	156
08	273.25	141
09	176.63	126
10	144.76	148

In order to obtain an extracted data, a parsing of the XML activity file is needed. During our experiments, a parser written in Python was developed. For instance, Algorithm 4 presents a short example of the extracted data, where the first number denotes the identification number of the base training, the second is the total duration time of this activity in minutes, and the third the average heart rate.

Finally, the entire base training set as proposed by coaches is presented in Table 2, where in addition to parsed data, the profit values are presented.

Note that the average time duration $\overline{TD}$, average heart rates $\overline{HR}$, and average profit values $\overline{p}$ of all base training sessions are also presented in the last row of the table. Essentially, the presented trainings can be divided into long- and short-duration. The former have profit values $p_i \le 1.0$, while the latter $p_i > 1.0$. For example,

Table 2
The base training set.

Base training	Duration (min)	Average heart rate	Profits values
1	100.00	130.00	1.19
2	353.88	151.00	0.46
3	257.30	133.00	0.49
4	29.68	133.00	4.21
5	66.45	127.00	1.72
6	61.30	168.00	3.26
7	215.13	156.00	0.80
8	273.25	141.00	0.52
9	176.63	126.00	0.63
10	144.76	148.00	1.06
Average	167.84	141.30	1.43

the bicycle training 6 lasting 1 h with an average heart rate 168 and profit value of 2.74 belongs to short-duration training. In average, the bicycle training lasts nearly 3 h with an average heart rate of 141.30.

4.2. Construction of the training plans

An aim of this experiment was to generate the feasible training plans by varying the intensity factor in interval $K \in [136, 144]$ with steps of one BpM. As a result, nine different training plans were constructed. The observed interval of intensity factors was not selected accidentally, because the instances with $K \ll \overline{HR}$ are under constrained, while the instances with $K \gg \overline{HR}$ are over constrained and therefore harder for solving as showed during extensive experiments. The algorithm was terminated after $FEs = 50{,}000$ fitness function evaluations.

In order to identify, how the intensity factor K influences the construction of the training plans, the factor is varied in the interval $K\ in [136, 144]$. Thus, the average heart rate of trainings in base training set $\overline{HR} \approx 141$ was observed as well. The results of this experiment are illustrated in Table 3, where the proposed number of base training sessions (BT) obtained by mentioned values of intensity factors K can be seen. Note that the quality of training plan expressed by the training effort value (ev) is also included into the table. It applies here, the higher the value of ev, the more effort is needed by the athlete to finish the training plan.

From Table 3, it can be seen that the MBA algorithm prefers the long-duration base trainings in the construction of the training plans. In line with this, the short-duration trainings, like 4 and 6 are rarely proposed by the process. On the other hand, the MBA algorithm obtains a lot of solutions (i.e., base training plans) with $er = 0$ and different effort values ev for each intensity factor K. Although the solutions with maximum effort values are presented in the table, coaches have a whole range of base trainings at their disposal. Therefore, the right training plan is not difficult to be selected by coaches for athletes with different requirements.

4.3. Influence of the fitness function evaluations

An influence of the fitness function evaluations was investigated in this experiment. Therefore, the numerical values obtained by planning the MBA algorithm for sports training sessions in short-term ($FEs = 500$), medium-term ($FEs = 5{,}000$), and long-term ($FEs = 50{,}000$) runs. Data accumulated over 25 runs are presented in Table 4 according to standard statistical measures, like minimum (min), maximum (max), average (avg), median ($median$) and standard deviation ($stdev$) of the error rate. The mentioned values of the various termination conditions were determined after extensive tests.

Table 3
Predicted number of trainings.

BT/K	136	137	138	139	140	141	142	143	144
1	4	4	5	3	6	4	4	5	2
2	5	6	3	10	10	10	10	6	9
3	10	7	10	9	9	9	9	8	10
4	6	3	6	8	5	5	1	2	6
5	9	10	2	6	7	2	8	4	4
6	1	2	1	2	2	1	3	3	5
7	2	1	4	1	8	8	7	7	7
8	8	9	9	4	3	3	6	9	3
9	7	8	8	5	4	7	2	1	1
10	3	5	7	7	1	6	5	10	8
ev	71.56	72.24	73.99	71.80	74.40	78.98	78.40	75.30	71.93

Table 4
The results of the planning sports sessions by the MBA algorithm.

K	FEs	Min	Max	Avg	Median	Stdev
135	500	0.0000	0.1455	0.0242	0.0182	0.0354
	5000	0.0000	0.0000	0.0000	0.0000	0.0000
	50,000	0.0000	0.0000	0.0000	0.0000	0.0000
136	500	0.0000	0.1091	0.0341	0.0364	0.0273
	5000	0.0000	0.0000	0.0000	0.0000	0.0000
	50,000	0.0000	0.0000	0.0000	0.0000	0.0000
137	500	0.0000	0.1273	0.0438	0.0364	0.0347
	5000	0.0000	0.0000	0.0000	0.0000	0.0000
	50,000	0.0000	0.0000	0.0000	0.0000	0.0000
138	500	0.0000	0.2182	0.0674	0.0364	0.0637
	5000	0.0000	0.0000	0.0000	0.0000	0.0000
	50,000	0.0000	0.0000	0.0000	0.0000	0.0000
139	500	0.0182	0.8000	0.1363	0.0545	0.1750
	5000	0.0000	0.0182	0.0023	0.0000	0.0060
	50,000	0.0000	0.0000	0.0000	0.0000	0.0000
140	500	0.0000	2.7636	0.2232	0.0545	0.5560
	5000	0.0000	0.0182	0.0045	0.0000	0.0079
	50,000	0.0000	0.0000	0.000	0.0000	0.0000
141	500	0.0182	3.8909	0.7532	0.3091	1.0288
	5000	0.0000	0.0182	0.0023	0.0000	0.0060
	50,000	0.0000	0.0000	0.0000	0.0000	0.0000
142	500	0.0000	4.2364	1.2904	0.7091	1.4199
	5000	0.0000	0.0182	0.0053	0.0000	0.0083
	50,000	0.0000	0.0182	0.0008	0.0000	0.0036
143	500	0.0182	5.0909	1.3390	0.2545	1.6826
	5000	0.0000	0.0182	0.0053	0.0000	0.0083
	50,000	0.0000	0.0182	0.0015	0.0000	0.0050
144	500	0.0545	6.6909	3.1648	3.2000	2.0559
	5000	0.0000	0.0182	0.0083	0.0000	0.0092
	50,000	0.0000	0.0182	0.0053	0.0000	0.0083

The results of the experiment show almost four characteristics according to

- the minimum: the results are not dependent on the duration of run critical. This means that the MBA algorithm is capable of a quick convergence.
- the average: the more quality solutions can be obtained by the long-term runs, in average. The similar conclusion is valid also by measures, like maximum and median.
- the standard deviation: these values decrease when the number of fitness function evaluations increase.

In summary, the wide specter of solutions can be obtained by coaches using the MBA algorithm for planning the sports sessions. However, the proper selection between the proposed training plans depends on the athlete's current form and on the strictness of the coaches.

4.4. Where the optimization makes sense

In this experiment, effects of the intensity factor K on the results according to the average training effort ev and calculated heart rate hr were investigated. In line with this, border values of intensity factors K need to be indicated, where the problem becomes under- and over-constrained. Thus, the intensity factor was varied in the interval $K \in [132, 151]$ BpM in steps of one by both measures. The results of planning aggregated over 25 independent runs are presented in Fig. 1 during the short-term, medium-term, and long-term runs.

This figure is divided into two diagrams. The former illustrates the results according to the calculated heart rate, while the latter according to the training effort value. As can be seen from the diagram in Fig. 1a, all three lines representing different runs match a reference line denoting the linear incrementing of the intensity factor K in the interval $K \in [135, 147]$. These matches indicates that

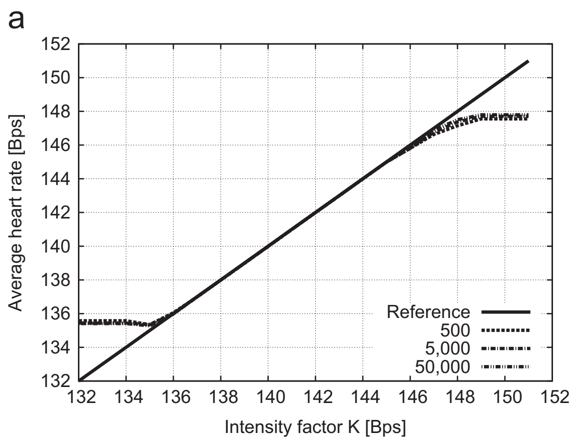


Fig. 1. Analyzing the results of planning the sports trainings: (a) calculated heart rate (hr) and (b) training effort value (ev).

the MBA found the correct solutions. Unfortunately, below this interval the problem is under-constrained, while above this the problem is over-constrained. That is, in these regions algorithms cannot find any solution. Furthermore, values of hr did not improve any more. As a result, the optimization makes sense only in the interval $K \in [135, 147]$.

On the other hand, Fig. 1b shows that the effort value ev in the interval, where the optimization make sense depends on the intensity factor K. Interestingly, the presented line in the diagram describes a step function, whose peaks and valleys occur at random. A small increase in intensity factor K may cause a large increase in effort value and vice versa. For example, an training effort value by $K=143$ is $ev=63$ in long-term run, while this is increased to 67 when the K is increase by 1. This behavior of effort value may be very useful for coaches that may adapt the appropriate training plans according to the athlete's current form.

4.5. Convergence graphs

An influence of the fitness function evaluations on the results of the MBA algorithm for planning the sports training sessions is the best illustrated in so named convergence graphs. The convergence graphs present the motion of the best and average fitness values during each generation. In our case, we analyzed the three motions obtained according to the different duration of runs, where the training plans by intensity value $K=138$ were constructed. The results of each motion was averaged over 25 independent runs are illustrated in Fig. 2 dividing into three diagrams. Note that the results are presented in logarithmic scale.

From these diagrams, it can be seen that the convergence of the MBA algorithm is rapid as presented by lines denoting the best fitness value. In contrast, the population diversity is lost gradually as can be seen by the lines denoting the average fitness values.

4.6. Comparative study

In order to investigate the quality of the results obtained with the MBA algorithm, a comparative study was conducted. In this study, the results of the MBA algorithm were compared with the results of the other evolutionary and swarm intelligence algorithms, like DE and PSO as well as deterministic algorithm DET. In line with this, the intensity factor was varied in interval $K \in [132, 151]$ in steps of one. As a result, 20 instances of problem are obtained capturing the wide region of instances in the neighborhood of the average heart rate $\overline{HR}=141$. All mentioned algorithms used the same mapping of candidate solutions in the real-valued search space to the integer represented solution in the problem space, and the repairing algorithm as presented in Algorithm 2. Deterministic algorithm obtains a permutation of base trainings according to profit values p_i. In each run, it produces only one solution.

On the other hand, DE and PSO are population-based algorithms. While DE belongs to a class of evolutionary algorithms, PSO is member of swarm intelligence. Additionally, the DE control parameters were configured as follows. The amplification factor of the difference vector was set as $F=0.9$, and the crossover control parameter as $CR=0.5$. A setup of the PSO control parameters were following. Acceleration constants were $c_1=c_2=1.0$, while an inertia weights was calculated according to equation $w=(U-L)\cdot \text{rand}(0,1)+L$, where U and L denotes an upper and lower bound respectively, while rand$(0,1)$ is the random number drawn from interval $[0,1]$. The population based algorithms in the experiment (i.e., MBA, DE, and PSO) used the same population size $Np=40$ and terminated after $FEs=10{,}000$ function evaluations. These algorithms were run 25 times.

The results of this experiment according to measures, like error rate er, training effort value ev, and time complexity tc in seconds are illustrated in Table 5. Because of the paper limitation, only 10 instances of intensity factors, i.e., $K \in [135, 144]$ can be viewed in this table.

As can be seen from Table 5, DE and PSO solved all the instances of intensity factor K with $er=0$, while the MBA algorithm solved exactly (i.e., with $er=0$) instances from $K=135$ to $K=139$. The results of planning the instances with $K \geq 140$ became too hard for the MBA algorithm. Unfortunately, the deterministic algorithm did not solve any instance exactly. According to measure ev, it can be seen that the highest values have been obtained by the PSO algorithm, while the lowest by the DET, in general. The DE algorithm gained slightly higher values of ev, in average.

In order to evaluate the quality of the results statistically, Friedman tests [42,43] were conducted to compare the average ranks of the compared algorithms. Thus, a null-hypothesis is placed to state: two algorithms are equivalent and therefore, their ranks should be equal. When the null-hypothesis is rejected, the Bonferroni–Dunn test [44] is performed. In this test, the critical difference is calculated between the average ranks of those two algorithms. If the statistical difference is higher than the critical difference, the algorithms are significantly different.

Two Friedman tests were performed regarding data obtained by optimizing 20 instances according to two measures for each of four algorithms. The tests were conducted at the significance level 0.05. The results of the Friedman non-parametric test can be seen in Fig. 3 that is divided into two diagrams. Each diagram shows the ranks and confidence intervals (critical differences) for the

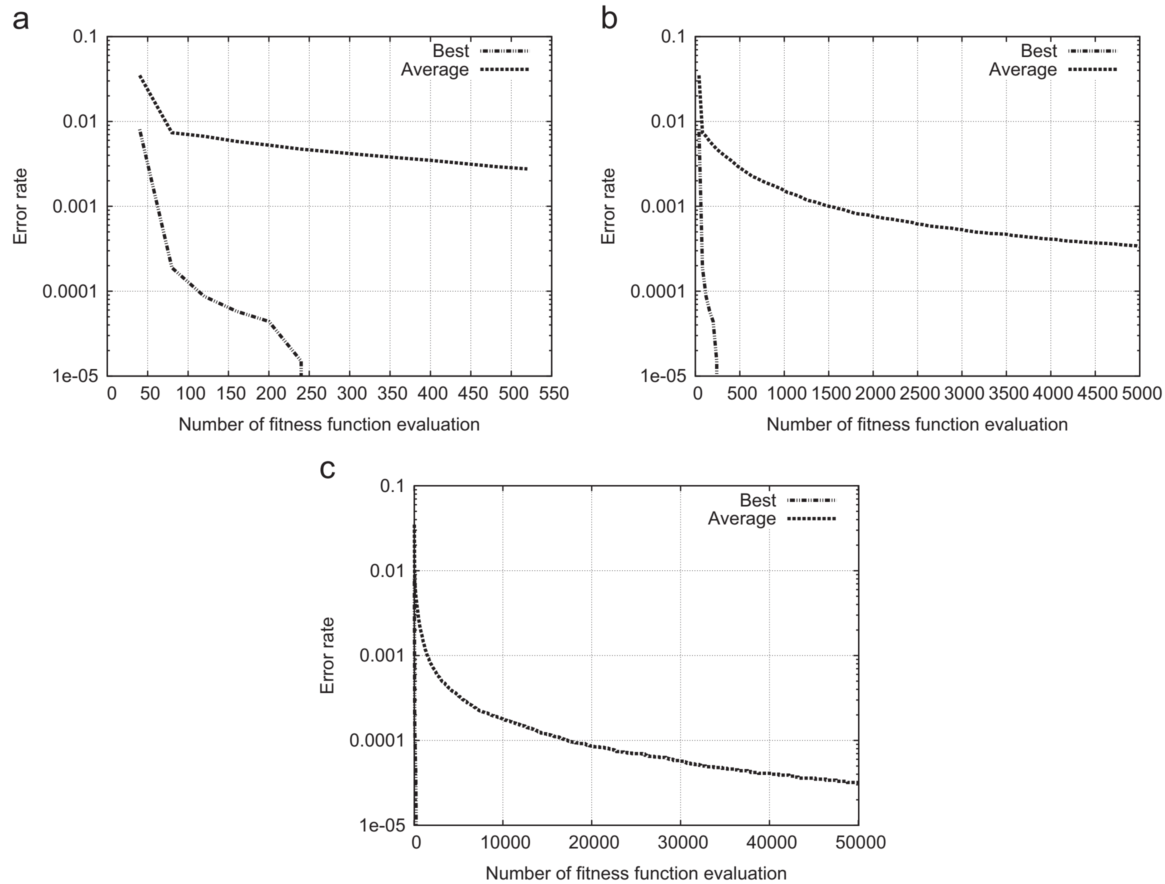


Fig. 2. Convergence graphs: (a) $FEs=500$; (b) $FEs=5000$; and (c) $FEs=50,000$.

Table 5
Comparing various algorithms.

K	Measures	DET	MBA	DE	PSO
135	*er*	0.1273	0.0000	0.0000	0.0000
	ev	63.0316	61.7948	62.4247	62.4856
	tc	0.00	2.85	2.34	3.80
136	*er*	0.5273	0.0000	0.0000	0.0000
	ev	57.2807	63.9139	65.1461	64.1193
	tc	0.00	2.45	2.06	3.01
137	*er*	0.2364	0.0000	0.0000	0.0000
	ev	55.1045	63.6534	62.7466	64.3801
	tc	0.00	2.34	1.92	2.73
138	*er*	0.5091	0.0000	0.0000	0.0000
	ev	50.6650	63.0621	63.3742	65.0272
	tc	0.00	2.16	1.82	2.44
139	*er*	1.5091	0.0000	0.0000	0.0000
	ev	50.6650	64.5781	65.6398	65.5509
	tc	0.00	2.07	1.75	2.42
140	*er*	2.5091	0.0008	0.0000	0.0000
	ev	50.6650	64.8648	64.9766	63.9104
	tc	0.00	1.97	1.64	2.35
141	*er*	0.2364	0.0008	0.0000	0.0000
	ev	45.5031	65.8986	67.2025	68.0683
	tc	0.00	1.84	1.62	2.22
142	*er*	0.6909	0.0023	0.0000	0.0000
	ev	45.4803	67.1543	64.2022	66.2178
	tc	0.00	1.82	1.51	2.12
143	*er*	1.6909	0.0333	0.0000	0.0000
	ev	45.4803	65.1553	62.1186	64.9102
	tc	0.00	1.78	1.57	2.11
144	*er*	2.6909	0.0977	0.0000	0.0000
	ev	45.4803	64.8997	64.1220	64.9602
	tc	0.00	1.76	1.45	2.08

algorithms under consideration with regard to the dimensions of the functions. Note that the significant difference between two algorithms is observed if their confidence intervals denoted as thickened lines in Fig. 3 do not overlap.

Fig. 3a shows that the results of PSO, DE and MBA outperformed the results of DET algorithm according to error rate. The PSO and DE are better than the MBA, but this advantage is not significant. As a result, the MBA algorithm can obtain the results of enough quality that can successfully be used for planning the sports training. The situation does not provide any consider change when the results are compared according to training effort value (Fig. 3b). Also here, the results of DET algorithm are significantly outperformed by the results of PSO, DE and MBA.

However, it is worth pointing out that this conclusion is based on the application of planning the sports training sessions, further detailed parametric studies may draw slightly different conclusions. Therefore, we do not intent to say that one algorithm is always better than others. This is consistent with the No free lunch theorem [45].

4.7. Discussion

What kind of training sessions, when and how many times it needs to be performed? In the past, answer to these questions was mainly reserved to the domain of coaches? However, it could be different in the near future. The results of our study had shown how some kind of artificial intelligence can be used for modifying and streamlining the work of coaches and athletes. Moreover, these training plans could in the future be determined automatically according to the history of training sessions, especially in the

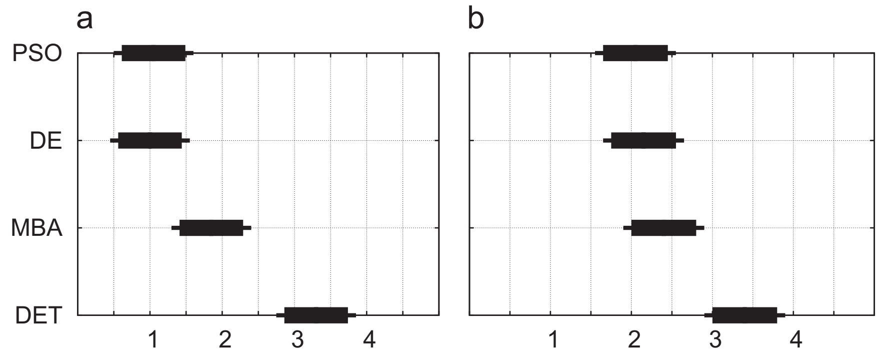


Fig. 3. Results of the Friedman non-parametric test: (a) error rate (er) and (b) training effort value (ev).

case of the quantifying each type of training session in the next training period.

The prepared algorithm is not a replacement for the coaches. In contrast, it serves as a help tool for them. Using this algorithm, the coaches can automate their manual work by prescribing the training plans for specific athletes everyday. In line with this, they can be more focused on the monitoring the activities of their athletes, further improving their forms, and repairing any errors which may arise during training.

5. Conclusion

A modern technology adopts all aspects of human life. This could not be resisted either to sport domain. Today, we cannot imagine any sports training session without technology aids. For instance, sport watches are capable of measuring the average heart rate as achieved by athlete during the sport activity (training), duration time of it, and even detailed data about his/her position during the training. The detailed path of movement, the average speed and configuration of terrain can be reconstructed from these data. Usually, these data are saved on the sport watch after activity and can be exported in form of XML activity file on a personal computer for the further analysis.

Especially useful, data in XML activity files are appropriate to coaches for indication of the athlete's current form and determination of the further training sessions in order to reach his/her full form for the specific incoming competition. Manual analysis of these files in order to find the required amount of base training sessions demands from the coach a lot of efforts, especially, when he/she is responsible for more athletes.

In line with this, the MBA algorithm is proposed as an intelligent planner for sports training sessions. It is capable of forecasting the training plans for specific athlete on the basis of exported XML activity files created by an athlete. The training plan comprises the number of base training sessions for the specified training period intended for the same athlete. Thus, the original bat algorithm is modified in order to solve the planning for the sports training. The proposed MBA algorithm was applied for planning the cycle training of specific athlete based on ten base training sessions. The obtained results showed that the predicted training plans comply to the high standard of cycle coaches.

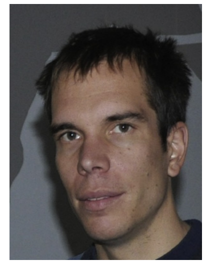
In the future work, we would like to direct our attention on analysis of XML activity files using the data mining methods. As a result, characteristics of particular trainings should be extracted. Then, the most characteristics trainings according to duration time and average heart rates may be selected in order to create the base training set automatically. On the other hand, the data mining methods could be used to monitor a progress of an athlete during training period. Finally, developing the swarm multi-population in order to simulate the cooperative coevolution during the SI search process as proposed in [46] seems to be a very promising idea for the future work.

References

[1] S. Poslad, Ubiquitous Computing: Smart Device, Environment, and Interactions, John Wiley & Sons Ltd., Chichester, UK, 2009.
[2] P. Misra, P. Enge, Global Positioning System: Signals, Measurements, and Performance, Ganga–Jamuna Press, MA, US, 2010.
[3] J. Bosak, T. Bray, Xml and the second-generation web, Sci. Am. 280 (1999) 89–93.
[4] G. Beni, J. Wang, Swarm intelligence in cellular robotic systems, in: Proceedings of NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, pp. 26–30.
[5] C. Blum, X. Li, Swarm intelligence in optimization, in: C. Blum, D. Merkle (Eds.), Swarm Intelligence, Springer Verlag, Berlin, 2008, pp. 43–86.
[6] W. Banzhaf, G. Beslon, S. Christensen, J.A. Foster, F. Képès, V. Lefort, J.F. Miller, M. Radman, J.J. Ramsden, Guidelines: from artificial evolution to computational evolution: a research agenda, Nat. Rev. Genet. 7 (2006) 729–735.
[7] X.S. Yang, A new metaheuristic bat-inspired algorithm, in: C. Cruz, J. González, G.T.N. Krasnogor, D.A. Pelta (Eds.), Nature Inspired Cooperative Strategies for Optimization (NISCO 2010), Studies in Computational Intelligence, vol. 284, Springer Verlag, Berlin, 2010, pp. 65–74.
[8] I.J. Fister, D. Fister, X.-S. Yang, A hybrid bat algorithm, Electrotech. Rev. 80 (2013) 1–7.
[9] I.J. Fister, D. Fister, I. Fister, Differential evolution strategies with random forest regression in the bat algorithm, in: Proceeding of the Fifteenth Annual Conference Companion on Genetic and Evolutionary Computation Conference Companion, GECCO '13 Companion, ACM, New York, NY, USA, 2013, pp. 1703–1706.
[10] Friel. http://home.trainingpeaks.com/blog/article/estimating-training-stress-score-tss, 2009.
[11] S. Das, P. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (2011) 4–31.
[12] X.-S. Yang, X. He, Bat algorithm: literature review and applications, Int. J. Bio-Inspir. Comput. 5 (2013) 141–149.
[13] A. Ušaj, A Short Review of the Basis of Sport Training, Faculty of Sport, Ljubljana, Slovenia, 2003.
[14] E. Banister, Modeling elite athletic performance, Physiol. Test. Elite Athl. (1991) 403–424.
[15] M. Garey, D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman & Co., New York, NY, USA, 1979.
[16] A. Eiben, J. Smith, Introduction to Evolutionary Computing, Springer-Verlag, Berlin, 2003.
[17] C. Darwin, The Origin of Species, P.F. Collier, New York, 1859.
[18] J. Kennedy, R. Eberhart, The particle swarm optimization: social adaptation in information processing, in: D. Corne, M. Dorigo, F. Glover (Eds.), New Ideas in Optimization, McGraw Hill, London, UK, 1999, pp. 379–387.
[19] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (2007) 459–471.
[20] D. Karaboga, B. Akay, A survey: algorithms simulating bee swarm intelligence, Artif. Intell. Rev. 31 (2009) 61–85.
[21] S. Das, S. Biswas, S. Kundu, Synergizing fitness learning with proximity-based food source selection in artificial bee colony algorithm for numerical optimization, Appl. Soft Comput. 13 (2013) 4676–4694.
[22] X.-S. Yang, Firefly algorithm, lévy flights and global optimization, in: M. Bramer, R. Ellis, M. Petridis (Eds.), Research and Development in Intelligent Systems XXVI, Springer, London, 2010, pp. 209–218.
[23] I. Fister, I.-J. Fister, X.-S. Yang, J. Brest, A comprehensive review of firefly algorithms, Swarm Evol. Comput. 13 (2013) 34–46.

[24] M. Dorigo, G. Di Caro, The ant colony optimization meta-heuristic, in: D. Corne, M. Dorigo, F. Glover (Eds.), New Ideas in Optimization, McGraw Hill, London, UK, 1999, pp. 11–32.
[25] I. Fister, A Comprehensive Review of Bat Algorithms and Their Hybridization, Master Thesis, University of Maribor, Slovenia, 2013.
[26] S. Rao, Engineering Optimization: Theory and Practice, John Wiley & Sons, Inc., New York, NY, USA, 2009.
[27] S. Kirkpatrick, C. Gelatt, M. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671–680.
[28] X.-S. Yang, Bat algorithm for multi-objective optimisation, Int. J. BioInspir. Comput. 3 (2011) 267–274.
[29] A.H. Gandomi, X.-S. Yang, A.H. Alavi, S. Talatahari, Bat algorithm for constrained optimization tasks, Neural Comput. Appl. 22 (2013) 1239–1255.
[30] P.W. Tsai, J.S. Pan, B.Y. Liao, M.J. Tsai, V. Istanda, Bat algorithm inspired algorithm for solving numerical optimization problems, Appl. Mech. Mater. 148 (2012) 134–137.
[31] L. Guo, A novel hybrid bat algorithm with harmony search for global numerical optimization, J. Appl. Math. 2013 (2013).
[32] G. Wang, L. Guo, H. Duan, L. Liu, H. Wang, A bat algorithm with mutation for UCAV path planning, Sci. World J. 2012 (2012).
[33] P. Musikapun, P. Pongcharoen, Solving multi-stage multi-machine multi-product scheduling problem using bat algorithm, Int. Proc. Econ. Develop. Res. 35 (2012).
[34] J. Xie, Y. Zhou, Z. Tang, Differential lévy-flights bat algorithm for minimization makespan in permutation flow shops, in: Intelligent Computing Theories and Technology, Springer, 2013, pp. 179–188.
[35] D. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Massachusetts, 1996.
[36] J. Koza, Genetic Programming 2 – Automatic Discovery of Reusable Programs, MIT Press, Cambridge, USA, 1994.
[37] T. Bäck, Evolutionary Algorithms in Theory and Practice – Evolution Strategies, Evolutionary Programming, Genetic Algorithms, Oxford University Press, Oxford, UK, 1996.
[38] L. Fogel, A. Owens, M. Walsh, Artificial Intelligence through Simulated Evolution, John Wiley, New York, US, 1966.
[39] J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems, IEEE Trans. Evol. Comput. 10 (2006) 646–657.
[40] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer Verlag, Berlin, 1996.
[41] I. Fister, D. Fister, S. Fong, I. Fister, Widespread mobile devices in applications for real-time drafting detection in triathlons, J. Emerg. Technol. Web Intell. 5 (2013) 310–321.
[42] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, J. Am. Stat. Assoc. 32 (1937) 675–701.
[43] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, Ann. Math. Stat. 11 (1940) 86–92.
[44] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[45] D. Wolpert, W. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1997) 67–82.
[46] S. Das, S. Biswas, B. Panigrahi, S. Kundu, D. Basu, A spatially informative optic flow model of bee colony with saccadic flight strategy for global optimization, IEEE Trans. Cybern. PP (2014) 1.

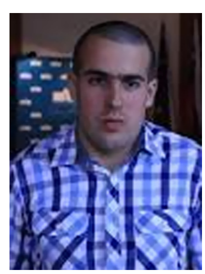
Iztok Fister graduated in computer science from the University of Ljubljana in 1983. In 2007, he received his Ph.D. degree from the Faculty of Electrical Engineering and Computer Science, University of Maribor. Since 2010, he has worked as a Teaching Assistant in the Computer Architecture and Languages Laboratory at the same faculty. His research interests include computer architectures, programming languages, operational researches, artificial intelligence, and evolutionary algorithms. He is a member of IEEE.



Samo Rauter received Ph.D. degree in kinesiology at Faculty of sport at Univeristy of Ljubljana, Slovenia in 2012. He is currently working at the Institute of sport with top level athletes and is also an Assistant Professor at the Faculty of sport at University of Ljubljana, Slovenia. His research interests include several aspects of sport science.

Xin-She Yang received his D.Phil. in Applied Mathematics from University of Oxford. Now he is a Reader in Modelling and Simulation at Middlesex University, UK, an Adjunct Professor at Reykjavik University, Iceland, and a Distuiguished Guest Professor at Xi'an Polytechnic University, China. He is also the IEEE CIS Chair for the Task Force on Business Intelligene and Knowledge Management, and the Editor-in-Chief of International Journal of Mathematical Modelling and Numerical Optimisation (IJMMNO).

Karin Ljubič is a student of medicine at Faculty of Medicine at University of Maribor. Her research acitivities include pulmonary medicine, internal medicine and sport science.



Iztok Fister Jr. received his B.Sc. and M.Sc. from Computer Science in 2011 and 2013, respectively. Currently, he is working towards his Ph.D degree. His research activities encompasses swarm intelligence, pervasive computing and programming languages.