# Vector Field Benchmark for Collective Search in Unknown Dynamic Environments

Palina Bartashevich[000−0002−5908−8196], Welf Knors, and Sanaz Mostaghim

Faculty of Computer Science, University of Magdeburg, Magdeburg, Germany
palina.bartashevich@ovgu.de, welf.knors@st.ovgu.de,
sanaz.mostaghim@ovgu.de

**Abstract.** This paper presents a Vector Field Benchmark (VFB) generator to study and evaluate the performance of collective search algorithms under the influence of unknown external dynamic environments. The VFB generator is inspired by nature (simulating wind or flow) and constructs artificially dynamic environments based on time-dependent vector fields with moving singularities (vortices). Some experiments using the Particle Swarm Optimization (PSO) algorithm, along with two specially developed updating mechanisms for the global knowledge about the external environment, are conducted to investigate the performance of the proposed benchmarks.

## 1 Introduction

Swarm Intelligence algorithms such as Particle Swarm Optimization (PSO) [13] and Ant Colony Optimization [6] are shown to be very effective in solving optimization problems. Due to their distributed nature, they can be easily used in swarm robotic search scenarios [8]. In the past years, PSO has been successfully used in this context [5, 1, 16, 11]. However, there are only a few existing methods that have addressed the influence of the external environments on the collective search algorithms [2, 3, 12, 17, 9]. The main goal of this paper is to provide new benchmark problems, simulating the influence of the dynamic external environment (such as wind, flow, etc.), which will serve as a baseline testbed for the development of new collective search mechanisms, that are robust to the unknown perturbations and can be further employed in real-world applications. In [2], the authors have introduced vector fields to simulate the external dynamics in a PSO-based collective search scenario designed for a swarm of aerial micro-robots. This approach only considered static vector fields, which are rather rare in nature, as the external dynamics change over time (i.e. unsteady flows). In this paper, we consider time-dependent vector fields and propose a unified method, called Vector Field Benchmark (VFB), to construct such dynamic environments using singular points [4]. In order to test the proposed VFB system, experiments are made using VFM-PSO [2] under the composition of changing vector fields and moving singular points. The results show that the VFB system can give different properties by simply setting the environmental types. The paper is organized as follows. We define VFB generator in Section 2. In Section 3, we describe the generalized “VFB-Map Exploration Framework” in the context of which we test the proposed benchmark. Section 4 contains several experiments to test the performance of VFB. The paper is concluded in Section 5.

## 2 Vector Fields Benchmark (VFB)

The time-varying dynamics of the environment are modeled by the unsteady vector fields with or without vortices, which are described below. In general, most vortex definitions are characterized by means of differential properties of the observed vector fields. For simplicity, our VFB functions are limited to a two-dimensional space (as horizontal wind).

**Definition 1**. *A* **Vector Field** $\boldsymbol{VF}$ *on a planar domain* $D \subset \mathbb{R}^2$ *is a function assigning to each point* $(x, y) \in D$ *a 2-dimensional vector* $\boldsymbol{VF}(x, y) = (u(x, y), v(x, y))$.

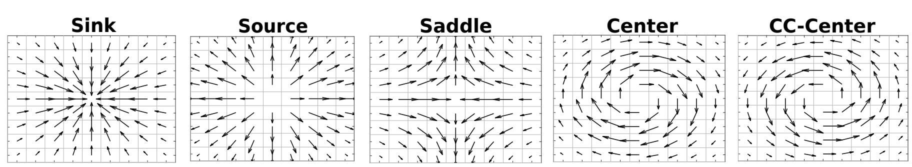
**Definition 2**. *A point* $(x_0, y_0) \in D$ *is singular for* $\boldsymbol{VF}$ *if* $\boldsymbol{VF}(x_0, y_0) = (0, 0)$. The values at any point $\boldsymbol{p} \in D$ of local vector field $\boldsymbol{SP}$ defined by corresponding singular point can be calculated as follows [14]:

$$\boldsymbol{SP}(\boldsymbol{p}) = e^{-d||\boldsymbol{p}-\boldsymbol{p_0}||^2} JV(\boldsymbol{p} - \boldsymbol{p_0}), \tag{1}$$

where $JV$ is the Jacobian matrix of the desired Singular Point, $p_0 = (x_0, y_0)$ is the center of the Singular Point and $d$ is a decay constant limiting the intensity of the Singular Point influence with increasing distance to its center $\boldsymbol{p_0}$.

**Definition 3**. *The* **spatial Jacobian** $JV$ *is an* $n \times n$ *matrix that contains a first-order description of how the flow* $\boldsymbol{VF}$ *behaves locally around a given location.* Following Hartman-Grobman theorem [10], singular points can be partly classified by looking to the eigenvalues of the Jacobian matrix at that point (see Fig. 1, where $k$ denotes the spread of $\boldsymbol{SP}$).

However, in measured data, the vector field $\boldsymbol{VF}$ is not given as a differentiable function. Following that, we discretize a domain $D \subset \mathbb{R}^2$ of the vector field and assume that we have the values of $\boldsymbol{VF}$ at the points $(x_i, y_j)$ of a regular grid of size $M \times N$ cells. We will denote the unit cell $c_{i,j}$ of the grid by sample point $(x_i, y_j)$ as follows $c_{i,j} = (u_{i,j}, v_{i,j}) = \boldsymbol{VF}(x_i, y_j)$.




$\begin{pmatrix} -k & 0 \\ 0 & -k \end{pmatrix}$ $\begin{pmatrix} k & 0 \\ 0 & k \end{pmatrix}$ $\begin{pmatrix} -k & 0 \\ 0 & k \end{pmatrix}$ $\begin{pmatrix} 0 & -k \\ k & 0 \end{pmatrix}$ $\begin{pmatrix} 0 & k \\ -k & 0 \end{pmatrix}$

Fig. 1: Five types of Singular Points shown on grids with $9 \times 9$ cells with $k = 5$ and corresponding values of $JV$ matrix below.

**Definition 4**. *An* **Unsteady Vector Field** *varies over time and is given as a time-dependent map* $\boldsymbol{VF}(\boldsymbol{x}, t) = \boldsymbol{VF}(x, y, t) : D \times T \to D$. *It can be also written as an* $(2+1)$*-dimensional steady field* [7]: $\boldsymbol{VF}(x, y, t) = (u(x, y, t), v(x, y, t), t)$.

**VFB Generator** When there are 2 or more Singular Points in a Vector Field, the velocities are calculated by simply adding the vector fields $\boldsymbol{SP}$ of each Singular Point. The overall VFB is defined as a sum of the underlying VF and the Singular Points influences:

$$VFB := \boldsymbol{VF}(\boldsymbol{p}, t) + \sum \boldsymbol{SP}(\boldsymbol{p}; JV(k), center : \boldsymbol{p_0}, decay : d, MoveType), \quad (2)$$

where singular points $\boldsymbol{p_0}$ are characterized by their type and strength $k$ performed with Jacobian $JV(k)$, and movement types $MoveType$. The dynamics of SPs are organized mainly by the moving center point $\boldsymbol{p_0}$ of the defined SP according to some law of movement (denoted by $MoveType$). Additionally, SPs movements are also affected by the underlying VF (if any), which means that the velocity vectors of the the underlying $\boldsymbol{VF}$ at the SPs current centers positions, i.e. $\boldsymbol{VF}(\boldsymbol{p_0})$, will be added to the $MoveType$ movement. The dynamics of underlying VFs, as well as of SPs, can also be complicated through multiplication on the Rotation Matrix $Rot(\alpha, t)$, thereby making their vectors rotating by some angle $\alpha$ at each time step $t$.

## 3 VFB-Map Exploration Framework

In order to test the proposed VFB functions, we take the same concept of Information Map (IM) approach for steady vector fields which was proposed in [2]. We adapt the concept for time-dependent flows with new ways of saving information in IM, which are supposed to catch the main features of unknown unsteady external dynamics. To estimate the external dynamics, the explorer population is used with simple movements based on the value of the VFB at their own positions. The explorers are coming from one and the same initial positions every $\Delta t$ time steps and save the information about VFBs magnitude and direction at their positions in the IM, which is a global and central archive accessible by all individuals. The optimizers follow the rules of PSO and retrieve the information about VFB from the IM to organize a better collective search process by full compensation of negative factors at already explored regions. The type of the optimizers decision, based on the IM, is not limited to full compensation (see for example, [3]), however, in this paper we consider only this type of action. As the VFB (i.e. $\boldsymbol{VF}(\boldsymbol{x}, t)$) changes over time, its values at the same positions might be different at different time steps $t$. So it is the question of how to store the measured values in IM, in order to take the relation between the past and the present into account. We present two update mechanisms of the IM for storing collected data:

(1) **Recent (Rec)** saves only the most recent measured values of the cells and left them unchangeable in the IM until their next visit. When a cell is visited a second time, all information from the first visit is replaced.

(2) **Evaporating Mean (EM)** computes the mean of measured values, if the cell $c_{i,j}$ is visited several times, and applies an evaporation operator $\rho_t$ to it, which linearly decreases the saved value for the $c_{i,j}$ by subtracting from its initially saved value the evaporation rate $\rho_t^0$ constantly at each time step $t$. The evaporation continues until the value in $c_{i,j}$ reaches the minimum limit of $\rho^{min}$, after which it is stopped, in order to preserve the information in already information-starved environment. The value of evaporation rate itself is constant $\rho_t^0 = \rho^0$ unless the cell is visited only once, otherwise its value is decreased by dividing on the number of cell visits $N_{vis}$ for each cell $c_{i,j}$ individually, as a means of 'confidence' in the saved value. In other words, the more times the cell is visited the less it is evaporated, i.e. $\rho_t^0(c_{i,j})$ is a function, which takes a value $(\sum N_{vis}(c_{i,j}))^{-1}\rho_{t-1}^0(c_{i,j})$ if $N_{vis}(c_{i,j}) > 1$ and $\rho_{t-1}^0(c_{i,j})$ otherwise.

Both of the above approaches consider the cells separately, therefore we refer to them as discrete methods denoted by Rec-D and EM-D. We also consider their continuous variants (denoted by Rec-C and EM-C) by using interpolation and extrapolation for the rest of the cells inside and outside the convex hull, defined by the cells with already saved information in IM. The *Nearest* interpolation [15] is used, as the fastest interpolation method among the others known in the literature (what is sufficient for time-dependent changes).

## 4 Experimental Study

The goal of the experiments is to demonstrate the usage of the introduced VFBs and to estimate the performance of the proposed updating mechanisms.

**Parameter Settings** Similar to [2], we use a VFM-PSO algorithm with 20 optimizers initialized randomly over the search space $S : [-15, 15] \times [-15, 15]$. The velocity limit $v_{max}$ is set to 2, inertia weight $w$ is selected to be 0.6 along with acceleration coefficients $C_1, C_2 = 1$. The number of explorers is set to 10 with frequency of update each $\Delta t = 10$ iterations. The total number of iterations is 150. The algorithm has been run on Sphere, Ackley and Rosenbrock over proposed further VFBs. We compare the proposed update mechanisms for the IM both for discrete (*Rec-D* and *EM-D*) and continuous (*Rec-C* and *EM-C*) variants. In the experiments, "*None*" indicates the approach without IM (i.e., without explorers). According to the preliminary experiments, the evaporation rate $\rho^0$ is set to 0.3 and $\rho^{min}$ is 0.5. Each experiment is repeated 30 times with different random initializations for both optimizers and explorers. Table 1 provides the function description of considered VFBs without SPs, i.e. VFB1-VFB3. Table 2 describes the VFBs containing moving SPs, i.e. VFB4-VFB7. For each VFB4-VFB7 one considers 9 singular points of at most two types with given coordinates $(x_0, y_0)$. Each scenario can have an underlying vector field, indicated as $VF$ in the last row of Table 2 and equations for which can be taken from Table 1. For all used in VFB4-VFB7 SPs, spread $k$ is set to 15 and decay $d$ is 0.4. $MoveType$ is defined by sinus law in horizontal direction. The only exception is VFB7, where SPs move according to the velocities of the underlying VF, i.e. Waves.

Table 1: Function descriptions for VFBs without singularities: VFB1-VFB3.

| | | |
|---|---|---|
| VBF1 | CrossRot | $VF(x_1, x_2) = Rot(5, t) * (x_2, x_1)$ |
| VBF2 | Waves | $VF(x_1, x_2) = (10, cos(x_1 - 0.5 * t) * 3)$ |
| VBF3 | UniformRot | $VF(x_1, x_2) = Rot(5, t) * (3, 3)$ |
| | Sheared | $VF(x_1, x_2) = (x_1 + x_2, x_2)$ |

Table 2: Parameters descriptions for VFBs with singularities: VFB4-VFB7.

| | VBF4 | VBF5 | VBF6 | VBF7 | Coordinates $(x_0, y_0)$ |
|---|---|---|---|---|---|
| type | Source | Source | Center | Center | (-10,-8) (-10,10) (0,-1) (0,-6) (10,12) |
| | Saddle | Saddle | | | (-10,1) (0,-10) (0,8) (10,3) |
| VF | Cross | Sheared | None | Waves | |

**Results** Figure 2 shows a comparison of median fitness values obtained using *None*, *Rec-D*, *EM-D*, *Rec-C* and *EM-C* (from left to right) within considered VFB (i.e. VFB1-VFB7 indicated by columns) on the corresponding objective function (indicated by rows). Since the main objective of the experiments is to demonstrate the usage of the introduced VFBs and to estimate the performance of the proposed exploration techniques, we have made multiple pairwise statistical comparison tests to identify which of the approaches are specifically different. Pairwise Mood's median tests were performed for VFBs, which have indicated statistical differences in at least one of the medians, i.e. VFB2-VFB5 and VFB7. For visual representation of the statistical differences between approaches the reader is referred to Figure 2. The boxes, which do not share any letter in common within one and the same VFB over certain objective function, indicate statistical differences between compared types of updating mechanisms. From this we can see that the obtained results reveal our hypothesis as on the most of the considered VFBs, regardless of the objective function, discrete mechanisms (*Rec-D*, *EM-D*) are not statistically different from each other and *None*. While almost in all of the cases, continuous updating mechanisms (*Rec-C*, *EM-C*) are statistically different from *None* and their discrete analogies (i.e. *Rec-D*, *EM-D*). *Rec-C* seems to be the most successful among the presented approaches, as its median fitness values are significantly lower than those using *EM-C* on VFB2, VFB3, VFB4 and VFB7. Although, *EM* approach was supposed to be a compromise between taking changes into account and compensating for the cases with partial covering by the VF at a time (as on VFB7), *EM-C* is statistically worse than *Rec-C* on all considered VFBs, including VFB7.

In the following we also report the convergence behavior of the proposed algorithms. Figure 3 illustrates Euclidean distances between the center of the swarm and the obtained global best over the iterations for VFB2, VFB7 and VFB8, which is a composition of VFB2 and VFB7. It can be observed that *None*, *Rec-D* and *EM-D* variants have similar behavior and do not change over iterations on VFB2 and VFB8, while *EM-C* and *Rec-C* reproduce an oscillation behavior, indicating that they have found the equilibrium point and it does not change with time anymore. The results differ for VFB7, as in this case the SPs only partially influence the movements of the optimizers, so we expect that the continuous

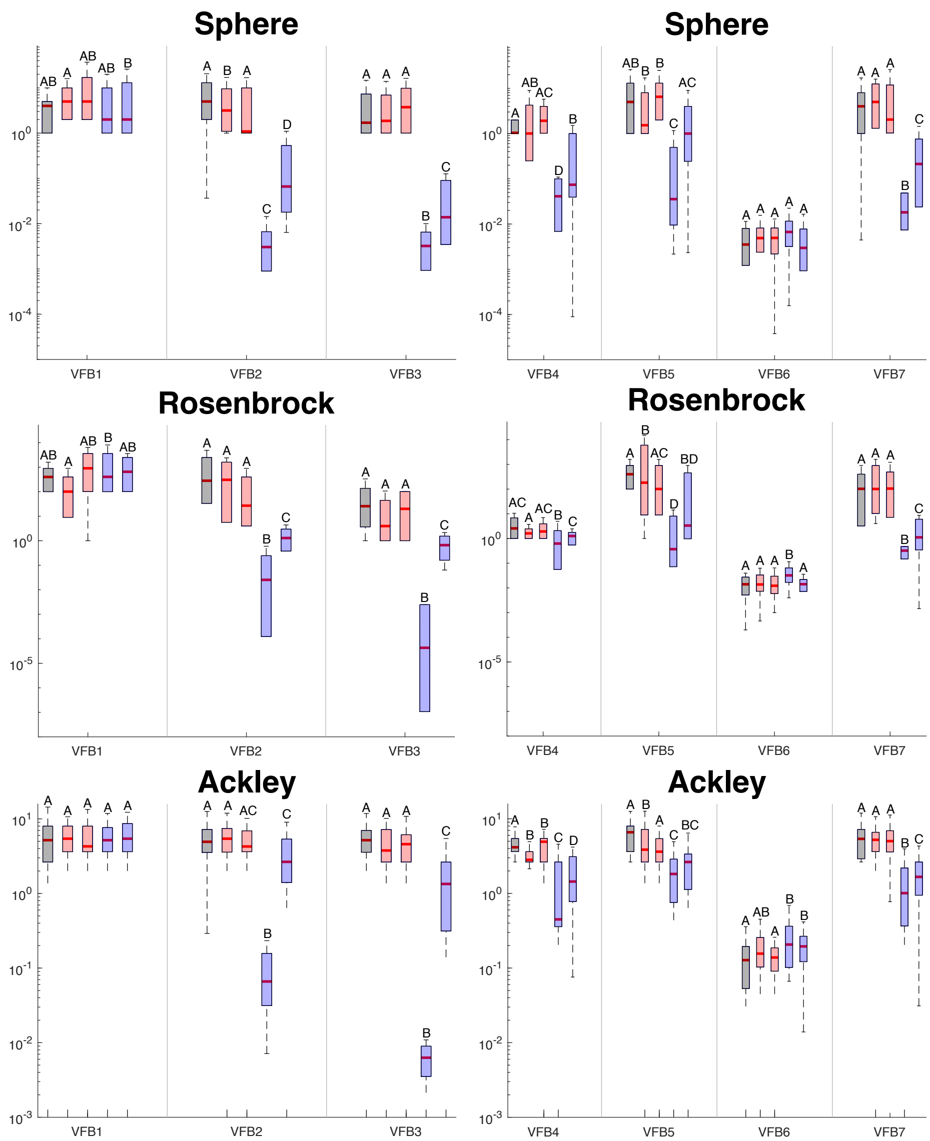



Fig. 2: Boxplots of the median fitness values obtained by *None*, *Rec-D*, *EM-D*, *Rec-C* and *EM-C*(from left to right) within VFB1 to VFB7. The central mark on each boxplot indicates the median. Boxplots which share at least one common letter within one and the same VFB indicate not statistical difference in median fitness values with significance level $\alpha = 0.05$ according to Pairwise Mood's Test.

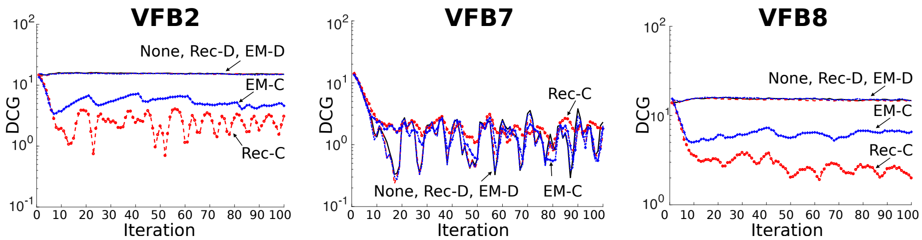



Fig. 3: Distances between the center of the swarm and the global best over the iterations obtained on Ackley function over VFB2 (full covering VF without SPs), VFB7 (consists only of SPs) and VFB8 (composition of VFB2 and VFB7).

update mechanisms do not help as they disturb the movements themselves at the positions where there is no VF influence. This can be observed on the performance of *EM-C* and *Rec-C* in VFB7, while the other update approaches help to converge until a change in the environment occurs. The changes in the environment can be depicted by the oscillating behaviors in the convergence plots. The results reported in Fig. 3 are obtained on the Ackley search landscape. However, our experiments show that the observed movement patterns (i.e. oscillations) are the same for all the other considered objective functions on the same VFBs. The only difference is in the value of the drift (i.e. vertical shift on the plots in Fig. 3), as it is defined by the found global best solution. In comparison to standard PSO problems (i.e. without VFBs), where distance between the swarm center and the global best is constantly decreasing as the particles converge to the best, acting under VFBs the swarm can not really converge. Therefore, in order to improve its performance, we observe oscillations in certain limited area around the best so-far obtained solution.

## 5 Conclusions and Future Work

This paper presents new benchmark functions for simulating and modeling the external dynamics for swarm robotics applications. We propose to use time-dependent vector fields with moving singularities and analyze their influence on the existing PSO-based methods in the context of "VFB-Map Exploration Framework". The results illustrate the strong influence of the environment on the collective search. One feature, imposed by moving singularities, concerns the oscillating behavior in the convergence plots. We have tested two various schemes based on continuous and discrete updating mechanisms for storing the global information about the unknown environment. The results show the advantage of the continuous variant over discrete in unsteady environments without singularities, while this degrades for environments which contain ones. Our experiments illustrate that the VFB can be used as a good base for developing search algorithms and is not limited to the proposed exploration framework and PSO. In future, we aim to work on other swarm based collective search mechanisms on the presented VFB.

## References

1. Atyabi, A., Phon-Amnuaisuk, S., Ho, C.K.: Navigating a robotic swarm in an uncharted 2d landscape. Applied Soft Computing **10**(1), 149 – 169 (2010)
2. Bartashevich, P., Grimaldi, L., Mostaghim, S.: Pso-based search mechanism in dynamic environments: Swarms in vector fields. In: 2017 IEEE Congress on Evolutionary Computation. pp. 1263–1270 (2017)
3. Bartashevich, P., Koerte, D., Mostaghim, S.: Energy-saving decision making for aerial swarms: Pso-based navigation in vector fields. In: 2017 IEEE Symposium Series on Computational Intelligence. pp. 1–8 (2017)
4. Demazure, M.: Singular Points of Vector Fields, pp. 219–247. Springer Berlin Heidelberg (2000)
5. Doctor, S., Venayagamoorthy, G.K., Gudise, V.G.: Optimal pso for collective robotic search applications. In: Proceedings of the 2004 Congress on Evolutionary Computation. pp. 1390–1395, Vol.2 (2004)
6. Dorigo, M., Stützle, T.: Ant Colony Optimization. Bradford Company, MA, USA (2004)
7. Günther, T., Theisel, H.: The state of the art in vortex extraction. Computer Graphics Forum p. To appear (2018)
8. Hamann, H.: Scenarios of Swarm Robotics, pp. 65–93. Springer (2018)
9. Hao, M., Yang, W.X., Wang, Y., Zeng, M.: Collective odor source estimation and search in time-variant airflow environments using mobile robots **11**, 10415–43 (12 2011)
10. Helman, J., Hesselink, L.: Representation and display of vector field topology in fluid flow data sets. Computer **22**(8), 27–36 (1989)
11. Hereford, J.M., Siebold, M., Nichols, S.: Using the particle swarm optimization algorithm for robotic search applications. In: 2007 IEEE Swarm Intelligence Symposium. pp. 53–59 (2007)
12. Jatmiko, W., Sekiyama, K., Fukuda, T.: A mobile robots pso-based for odor source localization in dynamic advection-diffusion environment. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (2006)
13. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Neural Networks, 1995. Proceedings., IEEE International Conference on. vol. 4, pp. 1942–1948 vol.4 (1995)
14. Marin, R.D.C.: Vector field design notes (2008)
15. Rukundo, O., Hanqiang, C.: Nearest neighbor value interpolation. In: International Journal of Advanced Computer Science and Applications (2012)
16. Sheetal, Venayagamoorthy, G.K.: Unmanned vehicle navigation using swarm intelligence. In: In Proceedings of International Conference on Intelligent Sensing and Information Processing (2004)
17. Wang, X., Yi, P., Hong, Y.: Dynamic optimization for multi-agent systems with external disturbances. Control Theory and Technology **12**(2), 132–138 (2014)