Parallel predator-prey interaction for evolutionary multi-objective optimization

Grimme Christian, Joachim Lepping, Papaspyrou Alexander

Nat Comput
DOI 10.1007/s11047-011-9266-9

Parallel predator–prey interaction for evolutionary multi-objective optimization

Christian Grimme · Joachim Lepping ·
Alexander Papaspyrou

Abstract Over the last decade, the predator–prey model (PPM) has emerged as an alternative algorithmic approach to multi-objective evolutionary optimization, featuring a very simple abstraction from natural species interplay and extensive parallelization potential. While substantial research has been done on the former, we for the first time review the PPM in the light of parallelization: We analyze the architecture and classify its components with respect to a recent taxonomy for parallel multi-objective evolutionary algorithms. Further, we theoretically examine benefits of simultaneous predator collaboration on a spatial population structure and give insights into solution emergence. On the prey level, we integrate a gradient-based local search mechanism to exploit problem independent parallelization and hybridize the model in order to achieve faster convergence and solution stability. This way, we achieve a good approximation and unfold further parallelization potential for the model.

Keywords Multi-objective optimization · Predator prey model · Parallelization · Classification

1 Introduction

For most real-life problems, decision makers face the challenge of finding good solutions in an environment of contradicting objectives. Optimal solutions, however, are always trade-offs: The improvement of one objective will inevitably lead to the deterioration of another. This problem type is well-known as *multi-objective optimization problems (MOPs)* and being tackled by mathematical and computer science research for more than a century.

Formally, a MOP can be defined as a real-valued problem with m objective functions $f : \mathbb{R}^n \rightarrow \mathbb{R}$ regarding a vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T \in \mathbb{R}^n$ in decision space. Without loss of generality, we assume that all objectives have to be minimized simultaneously, leading to a new definition of dominance for contradicting objectives. In the decision space $\mathbb{R}^n$, we formulate $\mathcal{F} = (f_1, \ldots, f_m) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ as a mapping into the objectives space. Now, for elements $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, we say that $\mathbf{x}$ dominates $\mathbf{y}$ if $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$ for all $i \in \{1, \ldots, m\}$ and denote this relation by $\mathbf{x} \preceq \mathbf{y}$. If an $i \in \{1, \ldots, m\}$ with $f_i(\mathbf{x}) < f_i(\mathbf{y})$ exists, we say $\mathbf{x}$ truly dominates $\mathbf{y}$ and denote it by $\mathbf{x} \prec \mathbf{y}$. An element of the optimal solution set is called Pareto-optimal if no other element truly dominates it in the decision space.

Obviously, the elements in the optimal solution set are only partially ordered and, as such, incomparable. Therefore, such problems are often approached using a-posteriori optimization: First, all optimal trade-off solutions are found in order to draw a picture of the achievable range of possibilities. Then, one out of these is selected as solution to be applied to the original problem. That selection or decision making step is usually done by humans who incorporate further experiences, problem knowledge, previously unexpressed preferences, or other aspects which are too

C. Grimme (✉) · J. Lepping · A. Papaspyrou
Robotics Research Institute, Section Information Technology,
TU Dortmund University, Otto-Hahn-Strasse 8,
44227 Dortmund, Germany
e-mail: Christian.Grimme@tu-dortmund.de

J. Lepping
e-mail: Joachim.Lepping@tu-dortmund.de

A. Papaspyrou
e-mail: Alexander.Papaspyrou@tu-dortmund.de

Published online: 17 August 2011

fuzzy to be tackled through computation. In contrast to that, the first step of trade-off generation is in most cases supported by algorithms which ideally provide the whole set of possible results. Unfortunately, for most MOPs it is algorithmically very difficult to find such a solution set.

During the last two decades, nature-inspired methods have therefore emerged as alternatives for the approximation of such solution sets. They are supposed to be efficient if the problem space (fitness landscape) is very large or even unknown so that established methods are conceptually not applicable. Multi-Objective Evolutionary Algorithms (MOEAs) follow the selection and variation paradigm from natural evolution. They represent problem solutions as a population of individuals and foster the survival of the fittest following a generational approach: The best-adapted solutions are allowed to produce offspring by carefully incorporating random variation. In this way, an ancestry-based search space exploration is realized that allows for parallelization at the level of evaluating the individuals regarding their respective fitness[1].

Because of the aforementioned partially-ordered dominance relation, the selection of ancestors for the next generation's offspring is a non-trivial problem. Modern MOEAs introduce complex mechanisms (Deb et al. 2000; Emmerich et al. 2005) for ranking the quality of individuals. Although—for many problems—not exceeding the effort of fitness evaluation, the computations which pave the path for performing sensible selection are getting increasingly expensive. This cost mainly results from conserving the basic evolutionary loop also in MOEAs. While the evaluation of a population and subsequent selection is relatively easy in single-objective evolutionary algorithms due to a single dominance criterion, computation of multiple objectives and of a substitute indicator for dominance ranking are usually bottlenecks in the algorithmic scheme. Ironically, little effort has been put into tackling this problem on the structural level: Instead, extensive research has been conducted on the parallelization of selection metrics computation (Alba 2005). This is all the more remarkable since such an approach will somewhat ameliorate the increased computational cost, but leave the structural problems largely untouched and thus impair scalability.

The predator–prey model (PPM), originally proposed by Schwefel, takes an alternative approach for handling multi-objective optimization problems under parallelization aspects (Laumanns et al. 1998). In contrast to established MOEAs, this method relies on an agent-based single-objective selection scheme. The basic idea is motivated from predator and prey interplay in nature, where prey individuals evolutionarily adapt to threatening predator individuals. Within the algorithm's abstraction, prey represent the solutions of a multi-objective problem while predators reflect the pursued objectives. The simultaneous influence from all predators forces the prey individuals to resist different threats, thereby approximating optimal trade-off solutions. Until now, various modifications (Grimme and Schmitt 2006; Grimme and Lepping 2007; Grimme et al. 2007) have been applied to the original PPM towards a building block-oriented model. The original intent of those works was to allow the flexible integration of problem specific knowledge leading to problem-specific algorithms. Nevertheless, both the original and the building block-oriented approach expose inherent parallelism which should be contemplated as well.

In this paper, we discuss the PPM with respect to identifying components that benefit from parallelism, evaluating emergent behavior, and proposing ways of hybridization. To this end, we fit the PPM into accepted taxonomies for parallelization of multi-objective evolutionary algorithms. Moreover, we show how the PPM inherently realizes different parallelization aspects within a single framework and exemplarily demonstrate the interaction of the different components for mutation and a special kind of recombination. Finally, we motivate this model as an environment for further research in parallel multi-objective optimization as it leverages decoupled interaction of actors to multiple parallelization layers. We explicitly do not compare the PPM to dominance- or indicator-based MOEAs, but rather evaluate the model in full depth to spotlight its ease of use with respect to parallelism.

The rest of the paper is organized as follows: In Sect. 2, we comprehensively review popular algorithms in multi-objective optimization and give some exemplary insight on parallelization approaches. Then, we describe three approaches for classification of algorithmic parallelization in multi-objective optimization and subsequently build the basis for our perspective on parallelization paradigms. In Sect. 3, we give a general definition of the reviewed predator–prey model and explain its classification into a common three-layer taxonomy for parallelization in multi-objective optimization. Next, in Sect. 4, we analyze the dynamics of interplay in the predator–prey model. There, we exemplarily investigate the behavior of mutation and recombination on the parallel cooperation level. Further, we discuss on the independent parallelization level how local search is to be integrated. Finally, in Sect. 5, we conclude our work with a perspective on future research and development.

[1] Considering parallelization, this approach comes with an inherent barrier at the moment when parents are selected for the next generation. For example in case of time consuming evaluation of only a single individual, the generational selection mechanism is blocked until completion of that specific individual.

2 Background

Over the last two decades, various heuristics have been proposed for applying evolutionary and natural principles to multi-objective optimization problems; see Deb (2001; Coello et al. 2007) for a detailed review. In the following, we introduce the most popular and review them with respect to parallelization. Eventually, we discuss three classification schemes for such algorithms.

2.1 Multi-objective optimization

Most successful approaches use the Pareto dominance relation to select efficient solutions during evolution: the Non-dominated Sorting Genetic Algorithm (NSGA-II) applies non-dominated ranking and sorting to determine reproduction candidates. Other approaches like SPEA2 (Zitzler et al. 2001) and PAES (Knowles and Corne 2000) utilize an archive as a central component for efficient solution conservation and offspring generation. Both rely on crowding-based measures (Deb et al. 2000) as a global component for diversity preservation: the former applies this mechanism on the population itself, and the latter uses it for archive reorganization.

In contrast, indicator-based methods aggregate solution quality in a single value to enable their comparison. Simple approaches perform this aggregation via weighting; however, choosing adequate weights is an intricate process, often leading to non-satisfactory results. More sophisticated approaches use elaborate aggregation methods, for example the $\mathcal{S}$-Metric Selection algorithm (SMS-EMOA). Based on the hypervolume measure of Zitzler et al. (2000), the algorithm selects solutions which contribute most to the overall hypervolume. Although it outperforms the dominance-based approaches for more than four objectives, the indicator calculation is rather expensive (Emmerich et al. 2005).

2.2 Algorithmic parallelization

Both approaches rely on global properties of the multi-objective domain: the calculation of dominance sets and quality indicators include the consideration of all available solutions; this also holds for diversity preservation and causes an additional parallelization barrier. To address this problem, various modifications of the algorithmic structure have been made to enable parallel processing.

A classical parallelization strategy is applied by Powell and Hollingsworth (2007) who assume a central NSGA-II algorithm instance that submits expensive function evaluations to Web.service-based clusters or Grid nodes for parallel execution. Xiong and Li (2003) present a parallel version of SPEA2 which comprises threads in order to handle individual as well as archive-related evaluations. They additionally introduce independently performing subpopulations that allow individual migration for genetic information exchange during execution. Okuda et al. (2002) use a similar approach, but allow the integration of arbitrary MOEAs for each subpopulation. Parsopoulos et al. (2004) also build upon subpopulations to enable particle swarms to start parallel searches that migrate intermediate results on a ring topology in order to gain an overall solution for multi-objective problems. Recently, Zhang et al. (2010) proposed a multi-objective distributed efficient global optimization algorithm (MOEA/D-EGO) that divides costly problem evaluations into subproblems. Subsequently, all subproblems are considered in parallel to generate and evaluate test points for later consideration in selection. In a model-based way, the search space is explored in parallel to enhance selection and to reduce costly evaluations.

Finally, there are many approaches that use agents or independently acting, spatially distributed individuals: Rowe et al. (1996) propose parallel diffusion GAs in multi-objective optimization. This approach assumes a population in which all individuals interact with their direct neighbors on a lattice structure. As a main advantage of this approach, the authors identify speedup through mere local dominance comparison. Besides that, they presume beneficial niching properties regarding different objectives and a diversity preservation effect of population distribution. Sadly, little evidence for this behavior is presented. Nebro et al. (2007) use a toroidal grid population structure and neighborhood interaction to propose a cellular GA for multi-objective optimization. They store non-dominated solutions in an external archive and apply a feedback mechanism on it to randomly seed individuals to a new generation. Siirola et al. (2004) introduce agents to induce external pressure on the distributed population. In this toolbox approach, several optimizers are acting on a population in shared memory. Although this approach is quite similar to the PPM, it still follows a generational paradigm and does not unfold its full potential of distribution, asynchronism, and parallelization. Furthermore, the influence of different agents is not investigated in detail.

2.3 Classification of parallel MOEAs

The exemplified discussion of parallelization approaches in multi-objective optimization shows different types of parallelism and distribution in algorithms. As in single-objective optimization (Crainic and Toulouse 2002; Alba and Tomassini 2002), several classification schemes for MOEAs have been published.

Coello et al. (2007) identify four parallelization models: The *master–slave* model follows the paradigm of a centralized master process that distributes function evaluations to slave processes for parallel computation. The *island* model assumes isolated evolving subpopulations. Most common representatives of this type either allow individual migration between populations or mimic species that search the same or different parts of decision space as a kind of multi-start approach. The third type is called *diffusion* model and is represented by a massively distributed single population. Herein, evolution takes place on the local level and slowly spreads through the population: Only one or few individuals of the spatially distributed population are managed by a single processor. A fourth class captures hierarchical combinations of the former without introducing an additional paradigm.

Nebro et al. (see in Chap. 16 of Alba 2005) differentiate single-walk and multiple-walk parallelization. The former aims to accelerate computation by parallelizing function evaluations or search operators. For the latter, an additional separation regarding the Pareto-set generation is done: the centralized Pareto-set approach relies on a central data structure allowing the direct construction of a global Pareto-optimal set, while the distributed approach acts independently and on a local level, presuming a later merging of results. Unlike Coello's classification, the authors consider the topology of the algorithm's distribution as mere consequence of the chosen parallelization strategy.

Talbi et al. (2008) identify three levels of hierarchical parallelization: The *self-contained parallel cooperation* includes cooperating subpopulations or multi-start variants (as in the island model). Here, several algorithms work together to achieve parallelization for both function evaluation and spatial distribution. On the second and third level, the authors consider parallelization approaches for speeding up algorithms while leaving their structures untouched. They distinguish between *problem independent parallelization* (level 2) and *problem dependent parallelization* (level 3). The former mainly comprises the master–slave approach of parallel fitness evaluation in a generation, while the latter focuses on subdividing single evaluations to speed them up. According to the authors, this leads to the application of several solvers with different objectives, the decomposition of fitness evaluation for a single objective, or even the parallelization of multiple evaluation runs for the same calculation under uncertainty.

3 A parallel predator prey model

For our studies, we base upon a tailored model loosely adopting Schwefel's original ideas using predator–prey interaction for multi-objective optimization (Laumanns et al. 1998). In this section, we give a brief history of the model's development, a definition of the model's components, and a description of the algorithm's working principle.

Predator–prey interaction was initially considered by Laumanns et al. (1998) as a simple approach to multi-objective optimization. They introduced predators as environmental influences that follow a single selection criterion on a spatially distributed prey population. Therein, trade-off solutions were expected to emerge from simultaneous action of different predator species selecting regarding different objectives. Although principally successful, the approach was unable to ensure convergence to the efficient solution set and adequate solution diversity at the same time. Several modifications have been proposed to address these problems. Deb (2001) introduces a different evaluation scheme by randomly aggregating objectives in a single predator while Li (2003) adds more dynamics by allowing prey movement. Schmitt et al. (2005), in turn, split up the spatial population into so called demes in which a (μ, κ, λ)-ES with self-adaptation is applied. Finally, we iteratively proposed to modularize the whole model in order to enable a building block approach (Grimme and Schmitt 2006; Grimme and Lepping 2007) for (1) easily combining arbitrary actors and (2) better exploitation of the parallel potential of the model. This allowed us to additionally introduce a local search mechanism to prey individuals (Grimme et al. 2009).

3.1 General model definition

The interaction environment for our system is represented by a graph, usually a toroidal grid, see Fig. 1, which is populated by both predator and prey individuals. The latter represent possible solutions of a multi-objective optimization problem and are immobile (each individual inhabits a single, fixed vertex). As such, there exist as many prey individuals as graph nodes, building the population. Furthermore, all prey are of equal kind and can therefore be classified by a single species. On the contrary, predator individuals can differ from each other regarding their properties, such as their consumption characteristics and reproduction model.

The first property is the selection criterion, which defines a relationship between two prey individuals with respect to a single objective. This is usually realized as extinctive selection of the worst prey. The second property describes a variation operator, which specifies the creation model for the replacement of consumed prey individuals. For these variation operators mutation as well as different recombination schemes may be considered. For further investigation we here consider Gaussian mutation and in some cases Simplex recombination, a special case of local intermediate recombination, as operators. Every predator is constructed from a single selection criterion and a variation

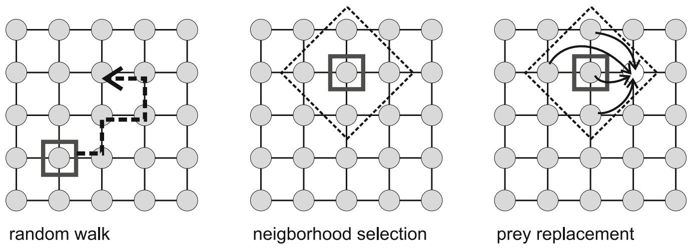


Fig. 1 Exemplary depiction of the three major algorithmic phases during predator action. At first, the predator performs a random walk. Then it spans a neighborhood evaluating all contained prey regarding its its single objective and removing the worst. Finally, the predator triggers a reproduction from the remaining prey to fill up the vacant place

operator, which both determine the predator species. That is, for a problem with two objectives $\mathcal{F} = (f_1, f_2)$, a predator species may be defined by f_1 as single selection criterion and Gaussian mutation as variation operator. Another predator species may comprise f_2 as objective and Simplex recombination as variation influence. According to Grimme and Lepping (2007), the possibility to arbitrarily combine selection and variation allows to flexibly induce specific influences on the population. In contrast to prey, predators are allowed to move within the interaction environment. This roaming behavior is usually implemented by a random walk. Figure 1 schematically shows the random predator's (squared box) movement throughout the spatial population.

As predators roam throughout the population, their interaction with prey is locally restricted. Formally, such an area of interaction is called neighborhood and given by a neighborhood function. As most common function the Von-Neumann neighborhood is applied, comprising all nodes—and thus prey—which can be reached by a fixed amount of steps in the grid structure. Figure 1 depicts a Von-Neumann neighborhood of size 1. Contrary to Laumanns, we do not allow diverging neighborhoods for selection and reproduction, to enable reproduction from prior evaluated parents (Grimme and Schmitt 2006).

The algorithmic transition from one state into another is conducted by a predator movement together with the application of an evolutionary process in the predator's neighborhood, see Fig. 1. In this figure, the predator moves according to the walk scheme to a target vertex. Then, a Von-Neumann neighborhood is spanned around that vertex defining the set of prey which are considered for the evolutionary process. The worst prey (in case of extinctive selection) within the neighborhood is selected regarding the predators selection criterion and consumed from the subpopulation. Afterwards, the free place on the population structure is taken by a new individual. This is bred by applying the predator's operator (e.g mutation or recombination). This process is repeated for every predator in parallel.

3.2 Layered model view

In order to structure the different levels of parallelism in the PPM, we use the above mentioned classification of Talbi et al. (2008) and map them onto the PPM's parallelization layers, see Fig. 2.

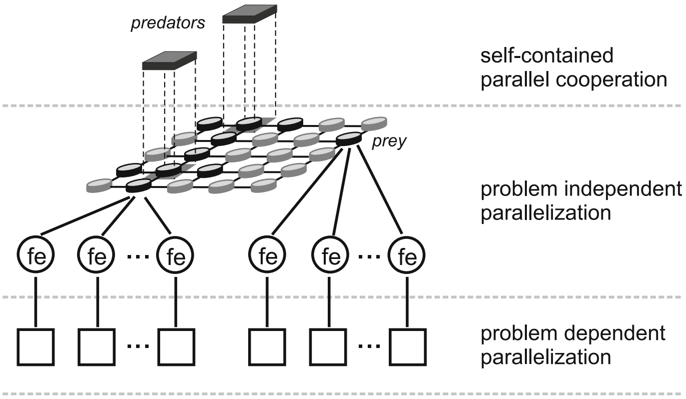


Fig. 2 Schematic classification of components in the predator-prey model in three levels of parallelization (fe denotes function evaluations). The first level shows the parallel acting of predators which is an inherent part of the model. Additionally, level two represents the general batch parallelization of expensive function evaluations. Finally, the third level denotes that even the function evaluations can be parallelized depending on their specific characteristics

3.2.1 Layer 1: Self-contained parallel cooperation

Predator movement and interaction on a spatially distributed population is an integral part of algorithmic structure in the PPM. This way, predators affect a subset of prey population locally regarding a single objective. Only the application of multiple predator species is expected to collaboratively yield efficient solutions for MOPs. Although independently acting and completely unaware of each other, predators work on the same global set of distributed prey using the underlying population structure as common environment. As such, changes to a local neighborhood performed by a single predator indirectly affect other predators' selection and variation behavior leading to trade-off solutions.

In terms of collaborative parallelization this emergent behavior of predators holds two important structural benefits: implicit distribution and parallelization of global dominance considerations as well as flexible adjustment of the algorithm. The distribution and parallelization due to predator action collapses dominance evaluation to single objective function computation and emergence through long-term predator interplay. Thus, the barrier of function evaluations from global dominance computation disappears.

The flexible adjustment of the algorithm is realized by predator behavior parameters like random walk radius, variation operators, or number of predators. For example, if predators are restricted to a specific spatial niche, either seldom or even no interaction can occur. In contrast, a completely coupled walk scheme is likely to result in cooperative influences on the prey.

3.2.2 Layer 2: Problem independent parallelization

On the problem independent parallelization level we capture two aspects of PPM modules in the general context of function evaluations. Although these processes are independent such that they can be executed in an *embarrassingly parallel*[2] fashion like it is possible for any MOEA, the PPM provides two specific properties motivated from the collaborative parallelization level.

First, in order to eliminate a prey within the predator's spanned neighborhood the whole subset of surrounded prey must be evaluated. For this purpose, a whole batch of evaluations is executed in parallel. A typical problem in embarrassingly parallel executions are barriers as they hamper the procedural progress. If one evaluation takes much longer than all others a barrier occurs which may lead to a low utilization of computing resources. This problem typically limits the embarrassingly parallel evaluation of a whole population in panmictic evolutionary models. The problem is also conceptually apparent in the PPM but due to rather small selection neighborhoods almost negligible. Further, the decomposition of MOPs to single objective evaluations allows independent computation of function values. In case of separable MOPs, each objective evaluation can be considered individually. Otherwise—for non-separable MOPs—all objectives have to be evaluated. However, additionally available information obtained from other predators can be cached and used later.

Second, the various prey individuals may perform an integrated local search mechanism, see Grimme et al. (2009) and Sect. 4.2. Any local search exclusively affects the decision space and is therefore completely independent of the population's spatial structure. The same holds for any autonomous optimization steps engaged by the prey. As the preys' immobility on the spatial structure is part of the PPM concept, actually no interaction occurs at that level. Thus, prey-triggered local search procedures can be asynchronously parallelized regardless of both the optimization problem and the higher level PPM configurations and predator interaction. For parallel evaluation of all objectives in the context of local search, the same barrier considerations as for local neighborhoods can be applied.

3.2.3 Layer 3: Problem dependent parallelization

At the lowest level, we consider the problem itself which might be parallelized by customized evaluation procedures. If the optimization problem has no mathematical formulation the evaluation is typically based on simulations performed by external software tools. In case of large scale simulations, it is often recommended to decompose the problem itself into sub-components which can be computed in parallel. Most real-world simulation softwares—such as CFD-ACE+[3], ANSYS CFX[4], or MATLAB[5]—support automatic problem decomposition and MPI-triggered parallel execution in order to benefit from an expected speedup. However, this concept is no special feature of the PPM and can be realized in any other optimization algorithm in the same way. Nevertheless, we show that the PPM does not involve any restrictions at the lowest parallelization level such that traditional approaches are directly applicable. However, in this paper we focus on levels 1 and 2 as methods for problem decompositions have been

[2] This is a technical term from high performance computing. It expresses a complete parallelization without any communication or synchronization between concurrent processes.

[3] http://www.esi-group.com/.

[4] http://www.ansys.com/.

[5] http://www.mathworks.com/.

exhaustively discussed in related literature, see for example Alba (2005).

4 Aspects of parallel cooperation

The previously described components of the model are used to realize different levels of parallel interaction. Each component of the model (e.g. walk, mutation, recombination) represents an atomic entity within the predator–prey optimization framework. In general, every component is independent and uncoupled such that they may be executed fully parallel.

In order to investigate parallel cooperation properties, effects, and benefits of the presented PPM, we have to understand interplay dynamics of acting entities on the spatially distributed population and the effects of evolutionary components in the algorithm. As such, we have to consider the population structure and simultaneously focus on very local processes that are involved. To this end, we discuss both the implicit predator collaboration as well as a hybridization approach with gradient-based local search and show the influence of the model's components on the different levels of the parallel architecture.

4.1 The predators' influence on the population and its visualization

In order to allow for deeper insights into the PPM's dynamics, a visualization concept is desired that is able to display the entire population in both positional mapping and objective value. Since the development of a general visualization technique is quite difficult for multi-objective optimization problems with multi-modal objective functions, we restrict ourselves to the easy test problem in Eq. 1, which suffices to show important effects.

$$\mathcal{F}_1(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} x_1^2 + x_2^2 \\ (x_1 - 2)^2 + x_2^2 \end{pmatrix} \text{ with } \mathbf{x} \in [-10, 10]^2 \qquad (1)$$

The Pareto optimal set of this problem is given by a line between (0,0) and (2,0) in decision space with the endpoints as optimal solutions for $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ respectively. Thus, a good solution for $f_1(\mathbf{x})$ is a bad solution for $f_2(\mathbf{x})$ simultaneously and vice versa. This yields the following construction rule for a two dimensional representation of the population.

Let $\mathcal{C} = (c_1, \ldots, c_l, c_{l+1})$ be a discrete color scale containing $l + 1$ colors. We apply the following mapping: l intervals of objective function $f_1(\mathbf{x})$, $\mathbf{x} = (x_1, x_2)^T$ with $0 \leq x_1 \leq 2$ are mapped to l colors. One color $l + 1$ is assigned to all remaining values of the objective function in order to mark them as being outside the considered range. Further, let $\mathcal{Q} = \{q_{ij} | i, j \in \{1, \ldots, k\}\}$ be a square divided in k^2 subsquares. Here, k is equal to the side length of the toroidal grid. Any subsquare $q_{ij} \in \mathcal{Q}$ denotes a prey's position on the torus. Finally, coloring subsquares from $\mathcal{Q}$ with colors from $\mathcal{C}$ results in a depiction of the population concerning both objectives.

The color of an optimal solution regarding objective $f_1(\mathbf{x})$ is given by c_1 while the optimum of objective $f_2(\mathbf{x})$ is represented by c_l. All other trade-off solutions are displayed by intermediate colors. An example of the introduced visualization method is shown in Fig. 3 along with its corresponding color scale.

Note that this method only holds for a small set of multi-objective optimization problems which show similar characteristics as the assessed problem. Furthermore, this visualization does not express anything about the convergence behavior to the Pareto-set and only yields a restricted statement on diversity due to the quadratic nature of both objectives.

4.1.1 Mutation analysis

First, we focus on predator influence with Gaussian mutation as an exclusive operator and apply the previously introduced simple test problem $\mathcal{F}_1(\mathbf{x})$ for our analysis. We observe the interaction of two predators and evaluate 100,000 predator walk steps each within a common toroidal structured population which is set up randomly in the beginning.

The results of optimization are shown in Fig. 4. Obviously, the whole population crowds at the extremal points of both objectives within the Pareto-set, see Fig. 4a, while no intermediary solutions are found. The corresponding visualized population structure is depicted in Fig. 4b where it comes apparent that both extreme solutions are represented by more or less large contiguous zones on the torus.

In the case of applying mutation only, this observation leads us to the assumption that diversity is lost due to the generation of similar offspring: on the long run only an interchange of individuals takes place and it is not possible

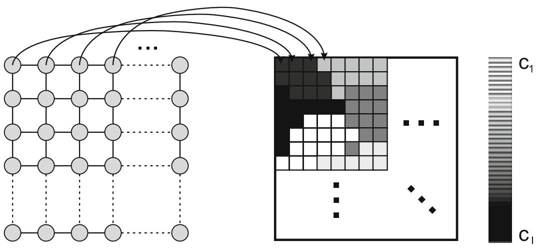
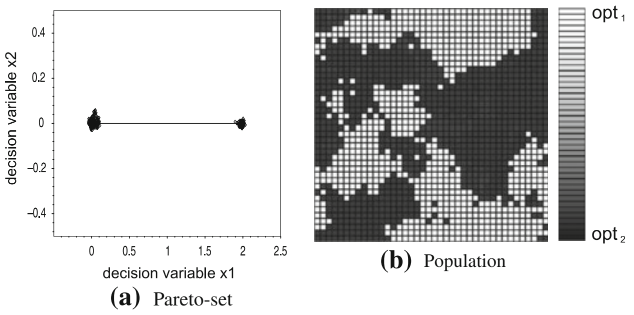


Fig. 3 Schematic depiction of the population visualization. The *color* of each subsquare denotes the proximity of the represented individual to the respective objective. The mapping of each prey's position to a subsquare is indicated by the arrows



(a) Pareto-set

(b) Population

Fig. 4 Population and Pareto-set after 100,000 evaluations of $\mathcal{F}_1$ with exclusive mutation. The *left-hand figure* shows the Pareto-set with the population crowding at the extremal solutions. The *right-hand figure* shows the corresponding colored representation of the search space, where the color in each subsquare denotes the quality of the residing prey

to induce diversity. In all contiguous areas the evolutionary process behaves similar to the single-objective case.

Moreover, we can observe squared edges between different extreme solution representations while no intermediary solutions are found at these borders. Although contrary to the assumption of emergence, intermediate solutions do not survive under exclusive influence of mutation. In the following, we try to find explanations for those observations.

Remember that in the predator–prey model, while the predator is moving randomly, the worst prey is replaced by a potentially better one. Thus, whenever a predator visits the same selection neighborhood again it becomes more likely that a prey is bred which is superior regarding the predator's objective. Thus, there is a strong tendency to move all prey within the selection neighborhood toward the predator's objective. This, however, comes along with a loss of diversity: once a prey has reached the near optimum for the objective it will no longer be consumed by the predator anymore and, respectively, the replacement of this individual becomes very unlikely.

In order to explain this phenomenon, we use a Markov chain that describes the transition probabilities of possible neighborhood states, see Fig. 5. We assume a single predator that resides at a static position within the torus graph and is not affected by any other predator. Furthermore, we consider a neighborhood of size 1 which involves five prey. Now, we assume a representation where one extreme solution is displayed as black prey and the other as white prey. For simplicity reasons we assume that the whole torus is already occupied by white prey (state s_0) and our predator selects regarding the black objective.

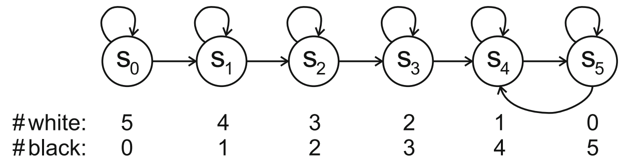


Fig. 5 Depiction of the Markov chain describing states and transitions of simple neighborhood mutation. Additionally, we denote the configuration in each state, starting with five white and ending with five black prey. The arrows represent state transitions

If the neighborhood consists of five white prey, we can always expect a white new mutated individual on the free position if the mutation step size is small. In this case, the probability of starting from state s_0 and ending in s_0 is approximately equal to 1. The same holds for the opposite configuration: a "white predator", black prey, and s_5 as starting state. This means that once the whole neighborhood is covered with kindred individuals it is almost impossible to leave this state anymore if the mutation step size is small. To be more precise, there is a probability of ϵ that an offspring prey keeps the color of its parent prey. We compute ϵ as the Cumulative Distribution Function (CDF) of a Gaussian distributed probability to perform a jump of at most 2 within the search space, depending on the mutation step size σ. Any larger jump in direction of the contradictory extremal solution would necessarily result in a direct change of a black prey into a white prey and vice versa, see Eq. 2. Consequently, the probability of the inverse event—that is the change in color of an offspring—is denoted by $\bar{\epsilon} = (1 - \epsilon)$.

$$\epsilon = \frac{1}{\sigma\sqrt{2\pi}} \int\limits_{2}^{\infty} \exp\left\{-\frac{x^2}{2\sigma^2}\right\} dx \qquad (2)$$

If σ is very small, ϵ will be approximately 1 denoting a rather stationary replication behavior. Contrary, a large σ will result in a small probability ϵ and result in a very transient behavior. This is also reflected in the state transitions in Fig. 5. Starting in state s_0 (where all prey are white), a stationary mutation will lead to s_0 again. A transient mutation leads to state s_1 with a neighborhood consisting of four white and one black prey. Due to the elitist selection mechanism, a transition back to s_0 is not possible for state s_1.

With the above insight and both basic probability definitions, we can compute all possible transition probabilities between the states and formulate the transition matrix $\mathbf{T}$ of the Markov chain.

$$\mathbf{T} = \begin{bmatrix} \epsilon & \bar{\epsilon} & 0 & 0 & 0 & 0 \\ 0 & 0.75 - 0.5\bar{\epsilon} & 0.25 + 0.5\bar{\epsilon} & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0.25 + 0.5\bar{\epsilon} & 0.75 - 0.5\bar{\epsilon} & 0 \\ 0 & 0 & 0 & 0 & \bar{\epsilon} & \epsilon \\ 0 & 0 & 0 & 0 & \bar{\epsilon} & \epsilon \end{bmatrix} \qquad (3)$$

Within this matrix, the transition probabilities are arranged in ascending order of states for rows and columns respectively. Further, the transition probabilities directly result from our initial definitions and the current state. We exemplarily show the computation of those transition probabilities leading from s_1 to s_1 and s_2 respectively. In state s_1, we consider a neighborhood of size 1 which contains 3 remaining white parental prey, one black parental prey, and an empty place to be filled. Considering a uniformly random parent selection, we have a probability of $p_w = 0.75$ to select a white parent and a probability of $p_b = 0.25$ to select a black parent. From that we can easily calculate the probability P_{s_1} for stationary behavior as shown in Eq. 4.

$$\begin{aligned} P_{s_1} &= 0.75\epsilon + 0.25\bar{\epsilon} \\ &= 0.75(1 - \bar{\epsilon}) + 0.25\bar{\epsilon} \\ &= 0.75 - 0.5\bar{\epsilon} \end{aligned} \tag{4}$$

The probability to change to state s_2 is consequently given by the counter probability of P_{s_1}:

$$\begin{aligned} P_{s_2} &= 1 - P_{s_1} \\ &= 1 - (0.75 - 0.5\bar{\epsilon}) \\ &= 0.25 + 0.5\bar{\epsilon} \end{aligned}$$

With this model we compute the final transition probabilities after many steps of selections and mutations by computing the resulting transition matrix $\mathbf{T}^\infty = \lim_{t\to\infty} \mathbf{T}^t$ for a fixed mutation step size. Note that we refer to problem $\mathcal{F}_1(\mathbf{x})$ where the maximum distance of the Pareto-set has length 2, see Eq. 1 and Fig. 4a, and assume an immobile predator. The resulting matrices $\mathbf{T}^\infty_\sigma$ for $\sigma_1 = 0.5$ and $\sigma_2 = 1$ are shown in Eq. 5.

$$\mathbf{T}^\infty_{\sigma_1} = \begin{bmatrix} 0 & \cdots & 1 \\ \vdots & \ddots & 1 \\ & & 1 \\ & & 1 \\ & & 1 \\ & & 1 \end{bmatrix}, \mathbf{T}^\infty_{\sigma_2} = \begin{bmatrix} 0 & \cdots & 0.023 & 0.977 \\ \vdots & \ddots & 0.023 & 0.977 \\ & & 0.023 & 0.977 \\ & & 0.023 & 0.977 \\ & & 0.023 & 0.977 \\ & & 0.023 & 0.977 \end{bmatrix} \tag{5}$$

The convergence of $\mathbf{T}$ is pretty fast as for $t > 5$ the matrix $\mathbf{T}$ only slightly differs from the final matrix $\mathbf{T}^\infty$. After few mutations within the same neighborhood all individuals will probably be black regardless of the starting state. If we exemplarily set the mutation step size to $\sigma_1 = 0.5$, we can expect that—on the long run and independent of the initial representation—the neighborhood will consist of only similar prey. However, if the mutation is strong enough, it becomes more likely that a white prey is created from only black prey in states s_4 and s_5, see $\mathbf{T}^\infty_{\sigma_2}$.

If we also consider the walk of the predator, the observed structures in the population can be explained as follows: As the random walk steps are notably small and mutation strength is weak in this experiment, it is likely that the same area is considered for mutation several times. Therefore, once the neighborhood only consists of nearly optimal prey, the predator tends to conserve or even expand this area. If both predators act in this fashion, nothing but an interchange of solutions takes place at the boundaries. As shown in the theoretical analysis of mutation, an increment of mutation strength is expected to only delay—but never prevent—the crowding of the population at extremal points, see Eq. 5. Note that this process can already be observed within the first steps of predator movement where it contributes to the formation of stable areas. Such areas do not yet represent a single objective's extremal solution but an agglomerate around the neighborhood's best individual. In the long run, the areas touch and the one containing the worse individuals with respect to the predator's objective is made extinct. This again is due to the aforementioned principle.

4.1.2 *Recombination analysis and evaluation*

This section takes a closer look at the population's dynamics under the influence of predators that apply a special kind of intermediate or panmictic recombination, so-called Simplex recombination (Grimme and Schmitt 2006): it provides a rotation-independent variation of an offspring $\mathbf{x}' \in \mathbb{R}^n$ based on $n+1$ parent individuals $\mathbf{x}_i$, with $1 \leq i \leq n+1$. The calculation of the new individual $\mathbf{x}'$ in the solution space is done using

$$\mathbf{x}' = \sum_{i=1}^{n+1} \left((1 - \lambda_i) \prod_{j=0}^{i-1} \lambda_j \right) \mathbf{x}_i, \text{ with } \lambda_0 := 1 \text{ and } \lambda_{n+1} := 0 \tag{6}$$

The vectors $\mathbf{x}_i$ span the n-Simplex, while the λ_j values determine the position of $\mathbf{x}'$. In order to ensure a uniform distribution of selections of $\mathbf{x}'$, random numbers $z_j \sim \mathcal{U}(0,1)$ are generated and applied in

$$\lambda_j = \sqrt[k]{z_j} \text{ with } k = (n+1) - j \tag{7}$$

Further, we use the afore introduced methodology of representing a population, see Sect. 4.1, and exclude mutation from the evolutionary process. In previous studies, the simplex recombination was considered to support convergence to the Pareto-set as well as diversity preservation. These interpretations are based on observing global characteristics of the mechanism. In the following both global and local properties of simplex recombination on a spatial population will be reviewed.

Global properties: An obvious property of Simplex recombination is to collapse an initial population in its convex hull. If there is no selection pressure at all, the individuals urge to the center of gravity. Otherwise, the center of gravity is dislocated depending on the objectives. As a consequence, the final position of a collapsed population strongly correlates to the area the initial population covers and the objectives obeyed. Hence, an initial population with a convex hull surrounding the set of Pareto-optimal solutions will collapse near by this set while a population initially located beyond the Pareto-set will collapse close to the border of the initial convex hull. Because of the lack of mutation and the resulting lack of innovation, these bounds cannot be crossed.

Local properties: Apparently, predators with different objectives collaborate as long as prey solutions are outside the *feasible area*. For our test problem $\mathcal{F}_1$ we consider this area to be given by a subset of the decision space above and below the Pareto-set, bound in the first objective by the extremal points. Outside the feasible area, an advancement toward one objective is most probably also an advancement toward the other objective if the angle between the respective gradients is small.

Inside the feasible area, predators that favor objective $f_1(\mathbf{x})$ judge prey individuals which are good concerning objective $f_2(\mathbf{x})$ as being bad. This contradictory behavior, however, results in a significant slowdown of population convergence into the direction of the objectives' connection line. This effect is displayed in Fig. 6 which shows the contraction of population regarding the coordinate directions of decision space over time. The contraction process can obviously be divided in four phases of different behavior. First, the global effect resulting from the collaboration of both predator species yields to an equal contraction in x_1 and x_2 direction (phase 1). In phase 2, the population is inside the feasible area. Here, the population keeps on collapsing regarding x_2, as both predators favor this direction the same, while the predators' conflicting objectives seem to lead to a stagnation concerning the collapse of x_1. Thereafter, phase 3 is characterized by total stagnation in both directions. However, in this phase the objectives become less important for the selection process as the individuals begin to collapse to the center of the population. This effect becomes apparent with the beginning of phase 4, where individuals in the barycenter of the population are more frequently produced than those at the boundaries of the convex hull. Thus, they increasingly often participate in the reproduction process which leads to more intermediary solutions in the evolving population. Finally, this increased participation rate results in a contraction of equal strength for both coordinate directions.

Regarding phases 2 and 3, we focus on different processes inside the population when predators work against each other with respect to objective $f_1(\mathbf{x})$. Generally, there are two cases of neighborhood constellations in which the predators may reside. This is shown in Fig. 7, where the black colored prey is preserved by predator r_1 while r_2 spares white prey. Following, these cases are detailed:

1. All prey in the spanned neighborhood are black for r_1 or all white for r_2 (Fig. 7, cases 1 and 4). In this case the recombination has rather no effect as the predator can only create an offspring in the convex hull of quite similar prey. This effect is for some time similar to the replacement during mutation, see Sect. 4.1.1.
2. The predator resides on the border between areas with white and black prey individuals (Fig. 7, cases 2 and 3). If more than one prey is bad concerning the predators' objective, the recombination mixes the parental prey to an intermediary solution. This leads to an increasing number of intermediary solutions over time, while two boundary regions emerge (Fig. 7, cases 5 and 6). In one region black prey exist next to gray ones and in the analogue region white prey border gray ones. If the gray population has an appropriate size, it becomes more stable against both predators than the remaining black and white prey. This finally leads to a faster contraction to the barycenter of the population.

The earlier described behavior results in the slowdown of contraction in the feasible area. This is rooted in the fact

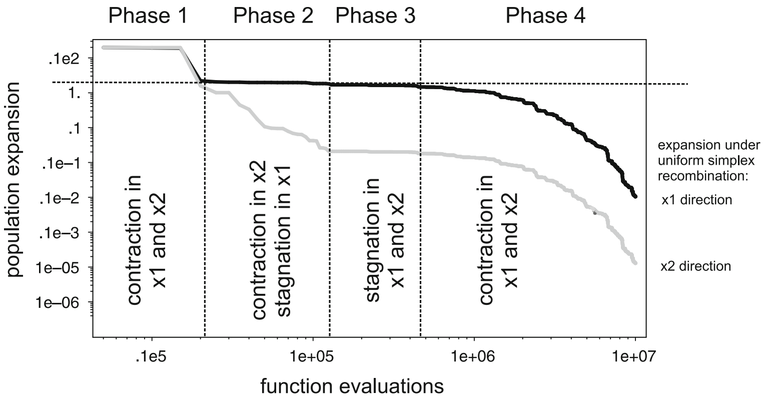


Fig. 6 Maximum expansion of the population's convex hull considering decision space components for simplex recombination. The figure shows that the recombinations influence can be divided into four phases of convergence

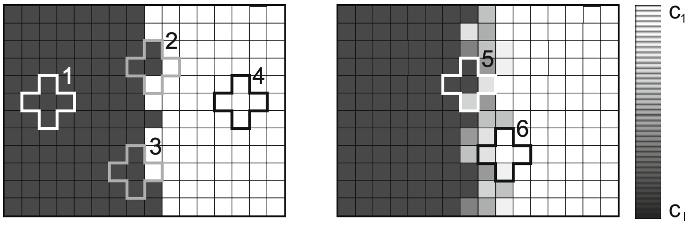


Fig. 7 Simplified neighborhood constellation on the spatial structure and its development under recombination. The *left-hand side* shows the situation with only mutation, where only solution interchange occurs. The *right-hand side* shows the same for activated recombination. In that case, trade-off solutions emerge on the boundaries between two contradictory areas

that individuals being a mixture of extremal solutions regarding both objectives are generated only on the border of two conflicting regions. As a consequence, possible trade-offs arise just in a fraction of the reproduction processes. Nevertheless, as Fig. 8 shows, even this initially slow process leads to contraction after many steps. The pictures of the population development where taken after 100,000 function evaluations (phase 2), 500,000 function evaluation (phase 3), and 2×10^6 function evaluations (phase 4).

4.1.3 *Combined application of reviewed operators*

By analyzing predator interplay with respect to associated variation operators independently, we were able to identify specific effects on the spatial population. For predators that bring in Gaussian mutation, we identified a tendency for spatial populations to develop contiguous areas of similar prey with respect to a single objective. Accompanying to this, there is a strong convergence towards extremal solutions, while no intermediary solutions are being created. Thus, the mutation operator's behavior degenerates to a plain replacement strategy. This holds independently from the variation of mutation step size, since an increase of this parameter only delays the crowding at extremal solutions as the final status. However, the random walk step size directly influences the aforementioned behavior, as this parameter softens the spatial structure. With longer random walks, this degenerates to an aggregated selection mechanism on a panmictic population.

Inclusion of predators bound to Simplex recombination supports global convergence and has a strong notion towards an emergence of intermediary solutions on the long run. The former, however, highly depends on the initial solution distribution in decision space as the collapsing behavior is only convergent to the true Pareto-set if it is enclosed by the convex hull of the population.

With these results at hand, we can demonstrate the potential of parallelity in the PPM by combining influences of both predator species in order to achieve better results for our simple test problem $\mathcal{F}_1$. Here, we set random walk step size to 1, mutation step size to 0.1, and perform 10^7 function evaluations.

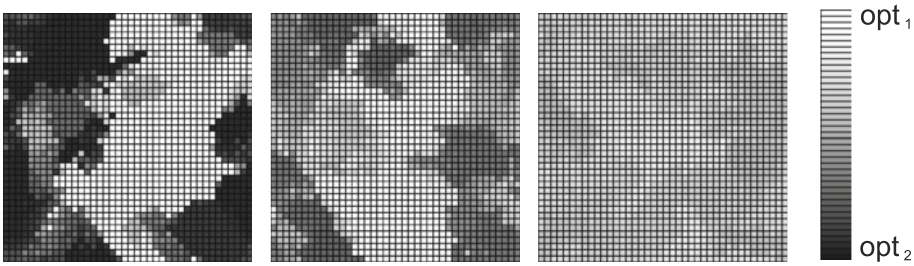
As depicted in Fig. 9, the simultaneous application of predator species reveals several characteristics that have been already identified for the non-combined case. At the beginning of the evolutionary process, the collapsing of the initial population towards the Pareto-set visualizes the strong influence of the recombination. Thereafter, the mutation operator dominates further development by preserving the extent of the population at both extremal solutions: apparently, the population keeps maximum expansion for the x_1 component while simultaneously oscillating on the x_2 component within the scale of the constant mutation step size.

Figure 10 shows the solution sets in decision space for both predator species individually as well as for the cooperating variant. This impressively demonstrates that the proposed use of collaborative parallelism yields better overall results.

4.2 Hybridization via prey individuals

Apart from collaboratively acting predator individuals on the cooperative parallelization level, we integrate

Fig. 8 Population development under simplex recombination, taken after 100,000, 500,000, and 2×10^6 function evaluations (from the left to the right). The figure shows the intermixing development during application of only recombination



independently executing local search procedures into the prey individuals. This contributes to stabilizing the search for solutions and also accelerates convergence of search towards required solutions. Since this hybridization works on a single decision variable without direct algorithmic cooperation, it happens on the problem independent layer of parallelization. Nevertheless, we will show on the basis of Grimme et al.'s work that the integration of local search indirectly contributes to the predator–prey model's approximation performance (Grimme et al. 2009).

4.2.1 Gradient-based local search

The inclusion of gradient-based, deterministic local search mechanisms into multi-objective optimization tasks has a long tradition which is rooted in an important inherent property of multi-objective problems: near the Pareto-set the gradients of a solution are almost contradictory.

Ester (1987) and colleagues (e.g. Peschel 1980) used this property for their early approaches in multi-objective optimization, while Brown and Smith (2005) review the basic theoretical principles of gradient-directed multi-objective search in EMO research. Harada et al. (2006) propose a Pareto-descent method, and Shukla (2007) focuses on the efficiency of such methods for unconstrained test problems and their effect on the computational complexity of the host algorithm. Recently, Schuetze et al. (2008) proposed and applied a 'hill-climber with sidestep' that also relies on gradient information and successfully integrated it into NSGA-II.

We integrate gradient information via the prey individuals into the predator–prey model. That is, we strive for finding the local descent direction for all objectives simultaneously and move a prey's solution along this direction.

Determining the Pareto-descent direction: Assuming that we are provided with gradient information for each objective, three cone types can be constructed:

1. *Descent cones*, which promise a benefit for all objectives and thus are preferable in order to reach the Pareto set,
2. *Contradictory cones*, which favor some objectives over others, and
3. *Ascent cones*, which lead to a deterioration of all objectives.

Depending on the position of the point x in decision space relative to the Pareto set, the cones are of different size, see Fig. 11: if x is far from the Pareto set, the descent cone is large. Otherwise, the descent cone is rather small while the contradictory cone is large. Altogether with the

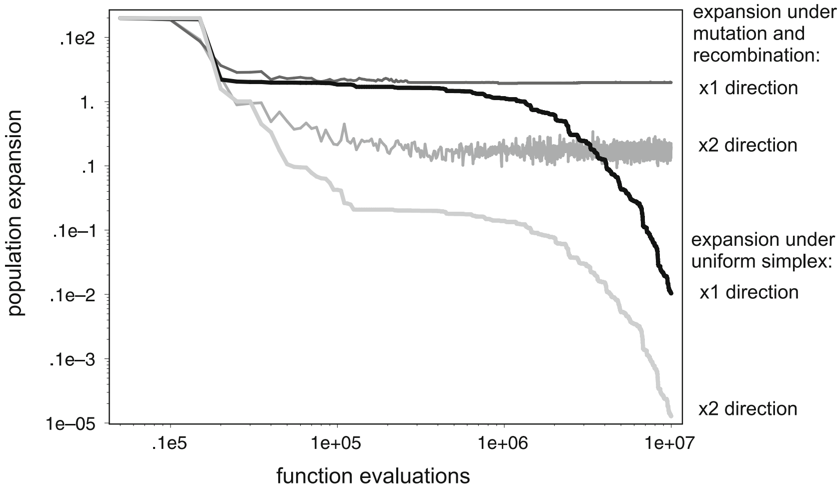


Fig. 9 Maximum expansion of the population's convex hull considering decision space components for both predator species. For comparison, the development under predators with only Simplex recombination is also depicted

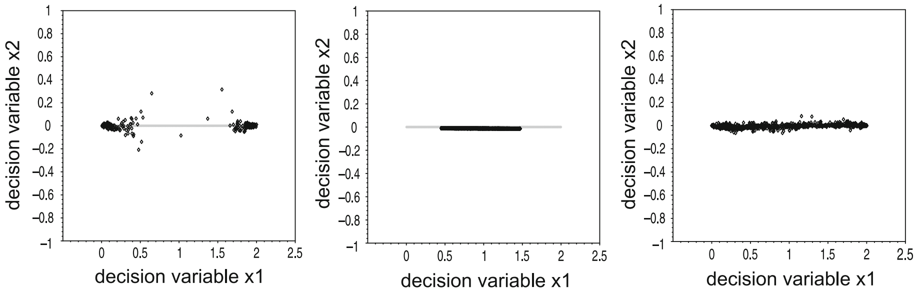


Fig. 10 Approximated Pareto-sets for the two-dimensional test problem $\mathcal{F}_1$, see Eq. 1. From *left* to *right* we applied predators only featuring mutation, simplex recombination and both predator species

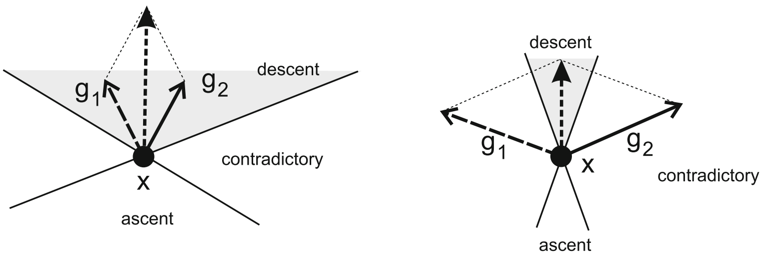


Fig. 11 Depiction of the applied local search based on approximated gradient directions for all objectives (here with two dimensions): if descent gradients point approximately to the same direction, the solution is far from the (local) Pareto set and the area of overall beneficial descent is large. Otherwise, the solution is close to the (local) Pareto-set and the beneficial area is small

gradient information, these properties are used to perform a local search step.

Formally, the search step length and direction is determined by combining the normalized gradient vectors as shown in Eq. 8.

$$\mathbf{x}' = \mathbf{x} + \mathbf{z}, \;\text{where}\; \mathbf{z} = \frac{1}{\sqrt{M}} \sum_{i=1}^{M} \omega_i \cdot \tilde{\mathbf{g}}_i \qquad (8)$$
$$\text{with}\; \mathbf{z}, \tilde{\mathbf{g}}_i \in \mathbb{R}^n \;\text{and}\; \omega_i = \mathcal{U}(1, M), \sum_{i=1}^{M} \omega_i = M$$

After combining the gradient vectors, the resulting step is normalized with respect to the dimensions to avoid large steps for $M > 2$ objectives. Simultaneously, each gradient vector is weighted with a uniformly distributed value to increase or reduce its influence in the heading direction. If point $\mathbf{x}$ is far from the Pareto set, the resulting direction $\mathbf{z}$ will not leave the descent cone. Close to the Pareto set, however, the weights will more frequently result in a step direction towards the contradictory cone. This is beneficial in order to favor diversity for solutions close to the Pareto set. Obviously, a parameter for switching between descent and diversity steps is not necessary: The local search mechanism automatically adjusts to the situation using gradient information inherent to the population.

Approximation of gradients: Usually, the finite difference method yields approximate information on the slope directions for a given point $\mathbf{x}$ within its nearest environment. However, this approach is computationally costly, as the number of function evaluations per objective grows with the dimension of decision space. Spall (1998), however, proposes a simultaneous perturbation method that works with two function evaluations per objective independent of decision space dimension. Although this approach is less exact in determining a gradient approximation, it determines an exact descent direction in the long run.

$$\tilde{\mathbf{g}}_i(\mathbf{x}) = \frac{f_i(\mathbf{x} + \sigma\mathbf{e}_i) - f_i(\mathbf{x} - \sigma\mathbf{e}_i)}{2\sigma} \qquad (9)$$

Equation 9 gives the very basic method to determine an approximated gradient $\tilde{\mathbf{g}}_\mathbf{i}$ for the i-th objective. The scalar value of σ denotes the perturbation strength, while $\mathbf{e}_i$ is a normalized random direction. In Spall's work, this direction is assumed to be Bernoulli ± 1 distributed.

4.2.2 *Evaluation of the hybridization approach*

We exemplarily focused on a more complex test problem than $\mathcal{F}_1$, which was used for visualization of predator dynamics. We consider Kursawe's test problem given in Eq. 10 which has non-convex parts in the Pareto-front and is disconnected in its Pareto-set (Kursawe 1991).

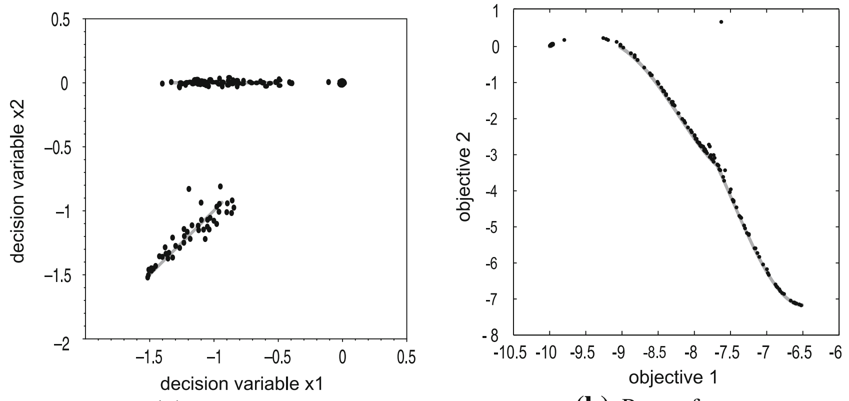
$$\mathcal{F}_2(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n-1} -10 \cdot \exp(-0.2 \cdot \sqrt{x_i^2 + x_{i+1}^2}) \\ \sum_{i=1}^{n} |x_i|^{0.8} + 5 \cdot \sin^3(x_i) \end{pmatrix} \qquad (10)$$
$$\text{with}\; \mathbf{x} \in [2, 4]^2$$

The main difficulty for previous predator–prey approaches and configurations was to conserve all parts of the Pareto-set. Because of the non-convex Pareto-set and the convexity properties of Simplex recombination (as shown in Sect. 4.1.2), we omit recombination from this analysis. Creating a point inside the convex hull of two solutions from different subsets of the disconnected Pareto-set will often lead to an infeasible trade-off

The combination of predators with only Gaussian mutation operators and the allowance of gradient-based local search in each individual yields a stable approximation of all Pareto-set parts, see Fig. 12.

In contrast to exclusively acting predators, this approach obviously comprises additional and simultaneous optimization of all available prey: Since the predator-induced evaluations are reduced and the number of function evaluations stay the same[6], the integration of local search into prey not only improves approximation

[6] With a maximum number of 200,000, which is a standard value for the predator–prey model.



(a) Pareto-set **(b)** Pareto-front

Fig. 12 Exemplary plot of $\mathcal{F}_2$ with integrated gradient search. In this setup, we used 225 individuals on a 15 × 15 toroidal grid

results but also increases parallelization and speed-up potential.

5 Conclusion

In the presented work, we theoretically as well as conceptually examined the predator–prey model as an alternative concept for multi-objective optimization. In contrast to established MOEAs, this approach applies species interaction from nature to produce a Pareto-optimal solution set. Predators represent single objectives collectively threatening spatially distributed prey individuals, which denote solutions to the MOP. Prey are expected to adapt to all predators producing feasible trade-off solutions.

Due to the decoupled architecture, the model supports parallelism in multiple ways, which we (1) classified according to a recent taxonomy in multi-objective evolutionary research and (2) discussed for the top two levels in detail. For the level of collaborative parallelization, we focused on investigating predator interaction on the distributed population. In detail, we considered the effects of Gaussian mutation and Simplex recombination on the prey population theoretically and showed the algorithmic benefit of simultaneous action and collaboration for convergence as well as for diversity preservation.

Additionally, we addressed the level of problem independent parallelization via hybridization and outsourcing of function evaluations. While the latter is a standard application of parallelism in evolutionary computation, hybridization enables the activation of additional potential in the PPM. By allowing local search in prey individuals, convergence is supported. Simultaneously and especially important for approximating complex problems, this enables a stabilization of solutions in disconnected parts of the Pareto-set.

The exploration and exploitation of inherent parallelism in the PPM can offer a perspective for further research: Instead of optimizing selection mechanisms and operators to speed-up MOEAs, we gave an example for a simple and successful approach to multi-objective optimization. Although the PPM needs significantly more function evaluations than standard MOEAs like NSGA-II or SPEA, we have demonstrated that this drawback can be compensated by its enormous parallelization potential on multiple levels of the PPM models architecture. Certainly, further research in specific aspects of model architecture as well as integration of sophisticated local search methods can contribute to the model's applicability.

References

Alba E (2005) Parallel metaheuristics: a new class of algorithms. Wiley-Interscience, New York

Alba E, Tomassini M (2002) Parallelism and evolutionary algorithms. IEEE Trans Evol Comput 6(5):443–462

Brown M, Smith RE (2005) Directed multi-objective optimization. Int J Comput Syst Signal 6:3–17

Coello CAC, Lamont GB, Veldhuizen DAV (2007) Evolutionary algorithms for solving multi-objective problems In: Genetic and evolutionary computation, 2nd edn. Springer-Verlag, New York

Crainic TG, Toulouse M (2002) Parallel strategies for metaheuristics. In: Glover F, Kochenberger G (eds) Handbook of metaheuristics. pp 475–513 Kluwer Academic Publishers, Boston

Deb K (2001) Multi-objective optimization using evolutionary algorithms. Wiley, Chichester

Deb K, Agrawal S, Pratab A, Meyarivan T et al (2000) A fast elitist non- dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: Schoenauer M (eds) Parallel problem solving from nature VI. pp 849–858 Springer, Paris

Emmerich M, Beume N, Naujoks B (2005) An EMO algorithm using the hypervolume measure as selection criterion. In: Proceedings of the third international conference on evolutionary multi-criterion optimization (EMO), pp 62–76

Ester J (1987) Systemanalyse und mehrkriterielle Entscheidung. VEB Verlag Technik, Berlin

Grimme C, Lepping J (2007) Designing multi-objective variation operators using a predator–prey approach. In: Obayashi S (eds) Proceeding of the fourth conference on evolutionary multi-criterion optimization (EMO). Springer, Japan, pp 21–35

Grimme C, Schmitt K et al (2006) Inside a predator–prey model for multi-objective optimization: a second study. In: Beyer H-G (eds) Proceedings of the genetic and evolutionary computation conference (GECCO). ACM Press, New York, pp 707–714

Grimme C, Lepping J, Papaspyrou A et al (2007) Exploring the behavior of building blocks for multi-objective variation operator design using predator–prey dynamics. In: Thierens D (eds) Proceedings of the genetic and evolutionary computation conference (GECCO). ACM Press, London, pp 805–812

Grimme C, Lepping J, Papaspyrou A (2009) Adapting to the habitat: on the integration of local search into the predator–preymodel. In: Ehrgott M, Fonseca CM, Gandibleux X, Hao J-K, Sevaux M (eds) Proceedings of the fifth international conference on evolutionary multi-criterion optimization (EMO), pp 510–524

Harada K, Sakuma J, Kobayashi S (2006) Local search for multiobjective function optimization: pareto descent method. In: Proceedings of the conference on genetic and evolutionary computation (GECCO). ACM, New York, NY, USA, pp 659–666

Knowles J, Corne D (2000) Approximating the nondominated front using the pareto archived evolution strategies. Evol Comput 8(2):149–172

Kursawe F (1991) A variant of evolution strategies for vector optimization. In: Schwefel H-P, Männer R (eds) Parallel problem solving from nature I. Springer, Paris, pp 193–197

Laumanns M, Rudolph G, Schwefel H-P et al (1998) A spatial predator–prey approach to multi-objective optimization: a preliminary study. In: Bäck T (eds) Parallel problem solving from nature V. Springer, Paris, pp 241–249

Li X (2003) A real-coded predator–prey genetic algorithm for multiobjective optimization. In: Fonseca C et al (eds) Proceedings of the second evolutionary multi-criterion optimization conference (EMO), pp 207–221

Nebro HJ, Durillo JJ, Luna F, Dorronsoro B, Alba E et al (2007) Design issues in a multiobjective cellular genetic algorithm. In: Obayashi S (eds) Proceeding of the conference on evolutionary multi-criterion optimization. Springer, Berlin, pp 126–140

Okuda T, Hiroyasu T, Miki M, Watanabe S (2002) DCMOGA: Distributed cooperation model of multi-objective genetic algorithm. In: PPSN/SAB workshop on multiobjective problem solving from nature II (MPSN-II). Grenada, Spain

Parsopoulos KE, Tasoulis DE, Vrahatis MN (2004) Multi-objective optimization using parallel vector evaluated particle swarm optimization. In: Proceedings of the international conference on artificial intelligence and applications (IASTED). Innsbruck, Austria

Peschel M (1980) Ingenieurtechnische Entscheidungen. Modellbildung und Steuerung mit Hilfe der Polyoptimierung. VEB Verlag Technik, Berlin

Powell D, Hollingsworth J (2007) A NSGA-II, web-enabled, parallel optimization framework for NLP and MINLP. In: Proceedings of the conference on genetic and evolutionary computation (GECCO). ACM, New York, NY, USA, pp 2145–2150

Rowe J, Vinsen K, Marvin N (1996) Parallel GAs for multiobjective functions. In: Alander JT (eds) Second Nordic workshop on genetic algorithms and their applications (2NWGA). University of Vaasa, Vaasa, Finland, pp 61–70

Schmitt K, Mehnen J, Michelitsch T et al (2005) Using predators and preys in evolution strategies. In: Beyer H-G (eds) Genetic and evolutionary computation conference. ACM Press, New York, pp 827–828

Schuetze O, Sanchez G, Coello CAC (2008) A new memetic strategy for the numerical treatment of multi-objective optimization problems. In: Proceedings of the 10th annual conference on genetic and evolutionary computation. ACM, Atlanta, GA, USA, pp 705–712

Shukla PK (2007) On gradient based local search methods in unconstrained evolutionary multi-objective optimization. In: Obayashi S (eds) Proceeding of the fourth conference on evolutionary multi-criterion optimization (EMO). Springer, Heidelberg, pp 96–110

Siirola JD, Hauan S, Westerberg AW (2004) Computing pareto fronts using distributed agents. Comput Chem Eng 29(1):113–126. (PSE 2003)

Spall JC (1998) Overview of the simultaneous perturbation method for efficient optimization. Johns Hopkins APL Tech Dig 19(4):482–492

Talbi E-G, Mostaghim S, Okabe T, Ishibuchi H, Rudolph G, Coello Coella CA (2008) Parallel approaches for multiobjective optimization. Springer-Verlag, Berlin, Heidelberg, pp 349–372

Xiong S, Li F (2003) Parallel strength Pareto multi-objective evolutionary algorithm for optimization problems. In: Congress on evolutionary computation (CEC), vol 4. pp 2712–2718

Zhang Q, Liu W, Tsang E, Virginas B (2010) Expensive multiobjective optimization by MOEA/D with gaussian process model. IEEE Trans Evol Comput 14(3):456–474

Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: empirical results. Evol Comput 8(2):173–195

Zitzler E, Laumanns M, Thiele L (2001) SPEA2: improving the strength pareto evolutionary algorithm. Technical Report No. 103. Computer Engineering and Communication Networks Lab (TIK), ETH Zürich, Switzerland