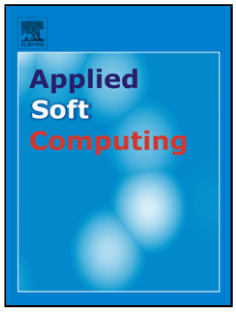


Title: Meta-Lamarckian Learning in Multi-Objective Optimization for Mobile Social Network Search

Author: Andreas Konstantinidis Savvas Pericleous Christoforos Charalambous

PII: S1568-4946(18)30085-1
DOI: https://doi.org/doi:10.1016/j.asoc.2018.02.026
Reference: ASOC 4718

To appear in: *Applied Soft Computing*

Received date: 2-3-2016
Revised date: 28-11-2017
Accepted date: 16-2-2018

Please cite this article as: Andreas Konstantinidis, Savvas Pericleous, Christoforos Charalambous, Meta-Lamarckian Learning in Multi-Objective Optimization for Mobile Social Network Search, *<![CDATA[Applied Soft Computing Journal]]>* (2018), https://doi.org/10.1016/j.asoc.2018.02.026

Research Highlights

- A realistic Multi-Objective Mobile Social Network Search (MO-MSNS) optimization problem is investigated.
- A decompositional MOEA hybridized with a Meta-Lamarckian approach, coined MOEA/D-ML, which learns from the problem's properties and objective functions, is proposed.
- MOEA/D-ML is evaluated on mobility and social behaviour patterns derived from the real data of GeoLife and DBLP datasets and a trace-driven experimental methodology.
- The generalizability of MOEA/D-ML is also evaluated on the well-known multi-objective combinatorial optimization problem Permutation Flowshop Scheduling Problem.
- The proposed MOEA/D-ML approach successfully learns the behaviour of individual local search heuristics during the evolution and adaptively follows the pattern of the best performing heuristics at different areas of the objective space of different benchmark test instances and for different problems.

Meta-Lamarckian Learning in Multi-Objective Optimization for Mobile Social Network Search

Andreas Konstantinidis, Savvas Pericleous and Christoforos Charalambous

Department of Computer Science and Engineering, Frederick University, Nicosia, Cyprus

Abstract

Mobile Social Networks (MSNs) have recently brought a revolution in socially-oriented applications and services for mobile phones. In this paper, we consider the search problem in a MSN that aims at simultaneously maximizing the user's search outcome (recall) and mobile phone performance (battery usage). Because of the conflicting nature of these two objectives, the problem is dealt within the context of *Multi-Objective Optimization (MOO)*. Our proposed approach hybridizes a Multi-objective Evolutionary Algorithm based on Decomposition (*MOEA/D*) with a *Meta-Lamarckian (ML) learning* strategy that learns from the problem's properties and objective functions. The ML strategy is devised for adaptively select the best performing local search heuristic for each case, from a pool of general-purpose heuristics, so as to locally optimize the solutions during the evolution. We evaluated our propositions on a realistic multi-objective MSN search problem using trace-driven experiments with real mobility and social patterns. Extensive experimental studies reveal that the proposed method successfully learns the behaviour of individual local search heuristics during the evolution, adaptively follows the pattern of the best performing heuristics at different areas of the objective space and offers better performance in terms of both convergence and diversity than its competitors.

The proposed Meta-Lamarckian based MOEA does not utilize any problem-specific heuristics, as most cases in the literature do, facilitating its applicability to other combinatorial MOO problems. To test its generalizability the proposed method is also evaluated on various test instances of the well-studied multi-objective Permutation Flow Shop Scheduling Problem.

Keywords: multi-objective optimization, evolutionary algorithms, local search, decomposition, meta-lamarckian learning, smartphones, social networks

1. Introduction

The widespread deployment of mobile smartphone devices and the advent of social networks have brought a revolution in social-oriented applications and services for smartphones [1] and the emergence of the so-called *Mobile Social Networks (MSNs)*, mainly composed of mobile users carrying smartphones that are used for sharing and collaboration [2, 3]. For example, Google Latitude and Facebook Places enable users to share their location, rank places and events, check-in to favorite places, provide their location history and query for *real-time* data (e.g., content, interests, comments, ideas and places). Currently, the bulk of social networking services designed for smartphone communities, rely on centralized or cloud-like architectures. Smartphone clients upload their captured objects (e.g., images uploaded to Instagram, video traces uploaded to Youtube, etc.,) to a central entity that subsequently takes care of the content organization and dissemination. Smartphones have asymmetric communication mediums with a slow up-link, thus continuously transferring massive amounts of data to a central authority through WiFi/3G/4G connections, can drain smartphone battery faster, increase query response times and quickly degrade the network health. A major goal while developing such a mobile-social network service is often (1) to maximize the outcome (i.e., the query response or *recall*) without (2) deteriorating the resources of the smartphone devices (i.e., *minimize energy consumption*) satisfying, at the same time, several constraints. These two objectives are conflicting and the respective problem is treated within the context of *Multi-Objective Optimization(MOO)*.

A *Multi-objective Optimization Problem (MOP)* [4, 5] can be mathematically formulated as

$$\text{minimize } \; F(X) = (f_1(X), \ldots, f_k(X)), \quad \text{subject to } \; X \in \Omega, \tag{1}$$

where Ω is the decision space and $X \in \Omega$ is a decision vector. $F(X)$ consists of k objective functions $f_i : \Omega \rightarrow \Re, i = 1, \ldots, k$, where $\Re^k$ is the objective space. The objectives often conflict with each other and improving on one objective may lead to deterioration of another. Thus, no single solution exists that can optimize all objectives simultaneously. In that case, the best trade-off solutions, called the set of Pareto optimal (or non-dominated) solutions, is often required by a decision maker. The Pareto optimality concept is formally defined as

Definition 1. A vector $u = (u_1, \ldots, u_k)$ is said to dominate another vector $v = (v_1, \ldots, v_k)$, denoted as $u \prec v$, iff $\forall i \in \{1, \ldots, k\}$, $u_i \leq v_i$ and $u \neq v$.

Definition 2. A feasible solution $X^* \in \Omega$ of problem (1) is called *Pareto optimal*

solution, iff $\nexists Y \in \Omega$ such that $F(Y) \prec F(X^*)$. The set of all Pareto optimal solutions is called the Pareto Set (PS), denoted as,

$$PS = \{X \in \Omega | \ \nexists Y \in \Omega, F(Y) \prec F(X)\}.$$

The image of the PS in the objective space is called the Pareto Front (PF)

$$PF = \{F(X) | X \in PS\}.$$

Multi-objective Evolutionary Algorithms (MOEAs) can obtain an approximate PF in a single run by accommodating different forms of operators to iteratively generate a population of such solutions. A major goal of MOEAs when dealing with a MOP is to produce (i) a diverse set of non-dominated solutions that is (ii) as close as possible to the real PF. Several techniques were proposed for improving the performance of MOEAs along those directions. For example, MOEAs are combined with niching mechanisms such as crowding distance estimation [6] to improve diversity, and/or local search methods [7] to improve convergence. The hybridization of MOEAs with local search heuristics, also known as Memetic Algorithms (MAs) [8], has been proven efficient in the past, giving rise to new challenges such as how to select the appropriate local search method within a pool of local search methods in order to identify, in an effective manner, the best local solution within a neighbourhood.

In Single Objective Optimization (SOO), an Evolutionary Algorithm (EA) is often hybridized with a Local Search (LS) method either randomly or deterministically. In the former case, either a LS method is randomly selected a priori and used for the whole evolution or a LS method is randomly selected from a pool of LS methods at each generation. In the latter case, problem-specific LS methods are designed and deterministically combined with an EA based on the characteristics of the objective function. However, the choice of the correct LS method and/or the design of a problem-specific LS method for each single objective that needs to be optimized can be a difficult and tedious process [9]. This is mainly due to the fact that different search algorithms, other than uniform random search, might introduce some kind of different bias into its search that makes a method good for some classes of problems but not for others. Ong and Keane in [10] propose a Meta-Lamarckian learning (or adaptive search) strategy that intelligently selects the most suitable LS method from a pool of LSs during the evolution.

In MOO, however, things are even more complicated for the following reasons: (i) in most cases there is no (or limited) knowledge of the problem domain

and MOEAs are often used as a black box [11], and (ii) in addition to the issues that are commonly considered when selecting a LS for SOO (e.g., different LSs having different bias on different classes of problems as well as on different instances of the same problem domain), in MOO, due to the multiple often conflicting objectives involved, there are also different biases on the same instance of the same problem domain but for different objectives.

In this paper, we combine *Meta-Lamarckian (ML) learning* with MOEA to study a realistic combinatorial Multi-objective Optimization Mobile Social Network Search (*MO-MSNS*) problem initially defined in [12]. We propose a new algorithm named MOEA/D-ML, which follows the general framework of MOEA based on Decomposition (MOEA/D) [13], combined with a pool of generalized Local Search (LS) methods and a Meta-Lamarckian Learning strategy to adaptively learn the effectiveness of each LS, and select the best performing LS for each objective function of each problem instance of each class of problems, online during the evolution. Here it is important to notice that no problem-specific heuristics are used in the design of the proposed approach, in contrast to other research studies that incorporate Meta-Lamarckian learning strategies (e.g., [14]), so as to facilitate its generalizability to other multi-objective combinatorial optimization problems. MOEA/D-ML is first evaluated using mobility and social behaviour patterns derived from the real data of GeoLife [15] and DBLP [16] datasets and a trace-driven experimental methodology. It is then also applied to various benchmark test instances of the well-known multi-objective Permutation Flow Shop Scheduling Problem (PFSSP) [7, 14, 17].

The rest of the paper is organized as follows: Section 2 provides related work on Mobile Social Networks, Multi-Objective Optimization and MOEAs, as well as Meta-Lamarckian learning and adaptive search. Section 3 introduces the system model as well as the problem definition and formulation. Section 4 presents our MOEA/D-ML approach, its internal structures and procedures. MOEA/D-ML is evaluated in Section 5 on realistic scenarios (by combining real datasets) and experimental parameters on the proposed MO-MSNS problem, as well as on various instances of the PFSSP. Finally, Section 6 concludes this paper.

2. Related Work

In this section, we provide related research work that lie at the foundation of our propositions.

2.1. Mobile Social Networks

Searching a smartphone social network to share objects of interest (e.g., photos, videos, text etc.,) can be roughly classified into: (i) *Blind* Search [18, 19, 20], where smartphone users propagate the query using an unsophisticated (e.g., random, TTL property) approach to as many nodes in the network as possible, and (ii) *Informed* Search [21, 22], where smartphone users utilize semantic or location information to forward queries to specific nodes in the network.

In this paper, we adopt the search approach presented in [12], that belongs class (ii) with the difference that a centralized approach is utilized where smartphone devices subscribe to a centralized registry. Similar to [22], a content summary mechanism (i.e., profile) is used for discovering mobile users that will participate in a query $\mathcal{Q}$ by the centralized node. In our adopted setting, the content summary of each mobile user is stored at the centralized node upon its registration and it features continuous sharing of data that can be utilized to create a number of collaborative scenarios (e.g., BikeNet [23]).

A central component to realize such scenarios is the availability of some high-level communication structure, such as *Query Routing Trees (QRTs)*. In [24], the authors present a technique that profiles the activities of the user in order to minimize the number of communication packets transmitted in the smartphone network. In [25], the authors form QRTs using flooding in order to continuously track mobile events and relay data to the query user. However, this approach suffers from significant energy waste as all nodes continuously and actively participate in the smartphone network.

Moreover, trying to optimize only a single objective (e.g., minimize energy) individually by ignoring and/or constraining the others (e.g., minimize network resources consumption, maximize recall etc.), often results in losing "better" solutions, since in a smartphone social network, minimizing the energy and maximizing the recall (i.e., quantity and quality of related objects of interest retrieved) are conflicting objectives and a set of trade-off candidate solutions is required.

2.2. Multi-Objective Evolutionary Algorithms

Multi-Objective Evolutionary Algorithms (MOEAs) are proven efficient and effective in dealing with MOPs. This is due to their population-based nature that allows them to approximate the whole PS (PF) in a single run. MOEAs general frameworks are often classified into three main categories: (i) the MOEAs based on Pareto Dominance [4] such as MOGA [26], NSGA-II [6], SPEA-II [27], which are mainly characterized by a selection operator based on Pareto-domination and a reproduction operator used iteratively; (ii) the decompositional MOEAs [28]

such as MOEA/D [13], MOGLS [29, 30], which are based on conventional aggregation approaches and usually decompose a MOP into a number of scalar SOO sub-problems, which are weighted aggregations of the individual objectives; and (iii) the indicator-based MOEAs such as the Basic Indicator-Based Evolutionary Algorithm (B-IBEA) and the Adaptive IBEA (A-IBEA) [31, 32], which allow to adapt the search according to arbitrary performance measures. For recent surveys on the state of the art of MOEAs please refer to [33, 34].

The combination of a MOEA with *Local Search (LS)* is known as *Hybrid (Memetic) MOEAs*. The first hybrid MOEA was implemented by Ishibuchi et al. [29] as a Multi-Objective Genetic Local Search (MOGLS) approach, in which the multiple objectives were aggregated into a scalar fitness function using random weights for parent selection and LS. Jaszkiewicz [30] has further improved MOGLS performance by improving the parent selection. Following this direction, several researchers have designed hybrid MOEAs by applying LS to all individuals [35] either at the end of each generation [29, 30], or just at the last generation [36]. In particular, the general MOEA/D framework proposed by Zhang and Li in [13] considers the hybridization of MOEA/D with LS as an optional step.

Even though there is a variety of hybrid MOEAs available in the literature, their applications on Multi-Objective Mobile Social Network Search optimization problems are still rare. For example, Liu et al. in [37] and Ma et al. in [38] focus on topology-related MOPs utilizing MOEAs for optimizing the community structure of social networks without considering any user-related objectives such as user satisfaction or the performance of users' smartphone devices. Our work is more related to the work in [12] in which the authors have applied the conventional MOEA/D on a tri-objective search optimization problem in a social community of smartphone users providing better results than the state-of-the-art Pareto-dominance based approach NSGA-II. Their major aim was the design of a principled framework composed of an optimizer (MOEA/D), a posterior decision maker and a Peer-to-Peer search approach. A real prototype system was developed for the ubiquitous Android Operating System and was utilized in real conditions.

The work presented in this paper focuses on the algorithmic aspects of solving a real-life bi-objective MSN search optimization problem and in general, on the application of adaptive learning strategies on a hybrid decompositional MOEA for addressing multi-objective combinatorial optimization problems. We propose MOEA/D-ML, a hybrid MOEA/D combined with Meta-Lamarckian learning to adaptively select the most appropriate LS method from a pool of generalized LS heuristics. To the best of our knowledge, there is no similar hybridization of a decompositional MOEA and Meta-Lamarckian learning in the literature.

2.3. Meta-Lamarckian Learning and Adaptive Search

Even though the hybridization of EAs with LS methods, individualistically and/or randomly, is easy and is proven efficient to achieve both good exploration and exploitation simultaneously, it still raises several issues. From a SOO point of view, Reeves has observed in [39] that different LS operators provide different number of local optima and induce considerably different landscapes [40]. Furthermore, Davis [9] has effectually argued that hybridizing GAs with the most successful LS method for a particular problem should work no worse than either GA or LS alone. On the other hand, if one does not know which LS method best suits a problem in hand, there is a great chance for the hybrid EA to perform worse than GA alone. The major impact of LS methods on the performance of MAs can be found in [41, 42].

Moreover in [10], Ong and Keane further expand the aforemention statements by arguing that with so many LS methods available in the literature, it is almost impossible to know which is the most relevant to the problem, especially in the absence of any knowledge on its cost surface a priori. They go on to propose an adaptive search approach, coined Meta-Lamarckian learning MA, which injects some intelligent means on the correct selection of a LS approach from a pool of LSs for a particular problem while the search is progressing.

Meta-Lamarckian (or adaptive) strategies can be characterized as cooperative and/or competitive or individualistic. Competition is when LS methods with higher fitness improvements are rewarded with higher chance to be selected for subsequent optimizations. Cooperation is when LSs and their improvement rewards act together for the selection of a LS for a subsequent optimization, and individualistic is when a single LS is used on the problem. A common reward measure η of the improvements contributed by a LS to a solution that has been searched is defined in [10] as

$$\eta = \beta \frac{|pf - cf|}{\mu}, \tag{2}$$

where pf is the initial function fitness of a parent solution[1] before local search, cf is the final function fitness of the child solution after applying local search and μ is the number of LS function evaluation calls made to reach from pf to cf. The term $|pf - cf|/\mu$ is the absolute reward measure and β signifies the relative

[1] in EAs terminology a solution, a chromosome and a decision vector terms are used interchangeably

reward that scales the absolute reward in proportion to the ability to produce high quality solutions compared to the best known solution obtained so far during the evolution. Often β is set as σ/cf for minimization and cf/σ for maximization problems, where σ is the fitness of the best solution encountered so far.

The approach of Ong and Keane in [10], cannot be directly used in MOO and cannot be combined with a MOEA based on Pareto dominance. The reason is that MOEAs based on Pareto dominance tackle a MOP as a whole and the improvement towards the direction of one objective often results in the deterioration of others. The MOEA/D approach, on the other hand, alleviates this difficulty by decomposing the MOP into a set of SOO subproblems, which are solved using SOO techniques and neighbourhood information. To do that, each sub-problem is usually associated with a weight vector, which can be used as a measure of its solution's objective preference and its position in the objective space. Meta-Lamarckian learning can therefore be used to learn the effect of each LS during the evolution and choose the most appropriate LS to locally optimize a solution along the direction of the preferred objective of each subproblem.

The authors in [43] tackled a continuous MOP by using a multi-objective evolutionary algorithm hybridized with a Lamarckian learning strategy, coined Multi-Objective Lamarckian Immune Algorithm (MLIA) for improving the Non-dominated Neighbor Immune Algorithm (NNIA). The Lamarckian learning performs a greedy search so as to generate improved decision vectors around non-dominated individuals in less crowded regions of the current Pareto Front (PF). The Powell's conjugate direction method is then adopted for locally searching the continuous objective space of the considered MOPs. Here it is important to note that although the focus of this study is on discrete MOPs, the proposed approach could also be applied on continuous MOPs [44], such as the well-known ZDT and DLZT, by adopting a pool of LS heuristics suitable for searching a continuous objective space. This, however, is out of the scope of this paper.

A hybrid Multi-Objective Particle Swarm Optimization (MOPSO) with Simulated Annealing is proposed in [14] for tackling a multi-objective permutation flow shop scheduling problem. A ranked-order value (ROV) rule based on a random key technique is employed to convert the continuous position values of particles to job permutations. A problem-specific LS based on the NEH-heuristic is first applied to good solutions with a specified probability $p_{ls} = 0.1$ to enhance the exploitation ability. To enrich the searching behavior and to avoid premature convergence, a LS based on Simulated Annealing, with multiple different possible neighborhoods (SWAP, INSERT, and INVERSE) is then applied with a specified probability $p_{SA} = 0.05$. An adaptive Meta-Lamarckian learning strategy is em-

ployed in order to decide which neighborhood will be used each time. MOPSO is evaluated on various instances of the permutation flow shop scheduling problem and compares favourably against MOGLS [7, 30].

We remark that the main differences between our research work and the work in [14] are: (i) for the construction of the MOPSO, problem-specific knowledge is considered that may affect its generalizability in other multi-objective combinatorial optimization problems; (ii) the actual effect of the Meta-Lamarckian learning strategy is not shown in any of their experimental series (e.g., comparison of the proposed algorithm with and without Meta-Lamarckian learning and/or replacing the ML strategy with random selection strategy) and (iii) our decompositional-based MOEA is able to adaptively learn the effectiveness of each LS and select the best performing LS at different areas of the objective space, during the evolution.

3. Problem Definition

In this section, we outline the adopted system model from [12] and then formulate the MO-MSNS problem. A table of respective symbols is shown in Table 1.

3.1. System Model

Overview

Let $\mathcal{C}$, denote a social networking service that maintains centrally a list of its M subscribed users $\mathcal{U}$ and their corresponding profiles $\mathcal{P}$ that record basic user details, authentication credentials, interests and friendship relations which can be used to define the conceptual social network graph $\mathcal{G}$ among the M users. In our setting, a user u_a uses a smartphone device to capture objects o_{ak} of interest at arbitrary moments.

Energy and Data Rate Model

We assume that when u_i is connected to $\mathcal{C}$, then $\mathcal{C}$ is aware of u_i's absolute and relative location. Each u_i features different Internet connection modalities that provide intermittent connectivity to $\mathcal{C}$ (e.g., WiFi, 2G/3G/4G), as well as peer-to-peer connection modalities that provide connectivity to nodes in spatial proximity (e.g., Bluetooth, Portable WiFi or NFC) [45]. Note that, each of the connection modalities comes at different energy and data transfer rate characteristics. For example, uploading or downloading large data items using Bluetooth can be more energy-efficient than using a radio network, but Bluetooth may not always be available and it is often slower. For more details, please refer to Section 2.1 in [12].

Social Network Search Techniques

Let an arbitrary user u_i, be interested in answering a query[2] $\mathcal{Q}$ over its social neighbourhood $\mathcal{G}'$ (i.e., nodes in $\mathcal{G}$ connected to u_i either directly or through intermediate nodes). For instance, let $\mathcal{Q}$ be a depth-bounded Breadth First Search query over u_i's neighbours in the graph $\mathcal{G}' \subseteq \mathcal{G}$. This kind of conceptual query can be realized either in (i) central, or (ii) distributed manner.

In (i), the multimedia objects o_{ak} and tags are all uploaded to $\mathcal{C}$ prior to query execution. Once $\mathcal{Q}$ is posted, $\mathcal{C}$ can locally derive the answers (using its local tag database) and return the answers to u_i. This model, currently utilized by all social networking sites (such as Twitter, Youtube, etc.), performs well in terms of query response time but performs poorly both in terms of data disclosure (i.e., objects o_{ak} and tags need to be continuously disclosed to $\mathcal{C}$) and performance (i.e., data transmission of large objects over radio links is both energy demanding and time consuming.)

In (ii), the objects and tags are all stored in-situ (on their owner's smartphone). In order to realize the search task, a querying node u_i downloads from $\mathcal{C}$ the addresses (e.g., IP:PORT address) of its first line neighbouring nodes in $\mathcal{G}'$. User u_i then contacts these nodes in order to conduct a depth-bounded Breadth First-Search in a P2P fashion (i.e., using a pre-specified Query Time-To-Live $\mathcal{Q}_{TTL} > 0$). Once some arbitrary node $u_j \in \mathcal{G}'$ receives $\mathcal{Q}$, it both looks at its local tags, in order to identify an answer and also forwards the request further until $\mathcal{Q}_{TTL} = 0$. The distributed approach improves the data-disclosure drawback of the central approach, but it is quite inefficient during search because $\mathcal{Q}$ has to go over a random neighbourhood rather than a neighbourhood that is contextually related to the query.

The search method adopted in this paper, aims at utilizing the advantages of both search approaches (i) and (ii) outlined above. The multimedia objects are kept in-situ for preserving privacy and facilitate location-awareness and the users only upload their profiles to $\mathcal{C}$, which will be responsible to derive and forward to the query user u_i, a Query Routing Tree X with the addresses of the contextually related users of the network along with the connection modality that each user should be contacted. Then the query user will conduct a depth-bounded Breadth First-Search in a P2P fashion to retrieve the data.

[2]Without loss of generality we assume simple Boolean keyword queries over tags.

Table 1: Table of Symbols

Symbol	Description
$\mathcal{C}$	Centralized Social Networking Service
$\mathcal{U}$	Users $\{u_1, u_2, ..., u_M\}$ of the Social Mobile Network
$\mathcal{P}$	Profiles $\{p_1, p_2, ..., p_M\}$ of users in $\mathcal{U}$ stored by $\mathcal{C}$.
o_{ak}	Object k (images, videos, etc.) recorded by user a.
$\mathcal{G}$	Conceptual Social Network Graph connecting users in $\mathcal{U}$.
$\mathcal{G}'$	Social neighbourhood of some arbitrary user ($\mathcal{G}' \subseteq \mathcal{G}$).
$\mathcal{Q}$	Query conducted in $\mathcal{G}'$.
X	Query Routing Tree (solution or individual) constructed to answer query $\mathcal{Q}$ ($X \subseteq \mathcal{G}'$).
$\mathcal{U}'$	Active users (nodes of $\mathcal{G}'$) connected to $\mathcal{C}$ during execution of $\mathcal{Q}$.

3.2. Problem Formulation

The *Multi-Objective Mobile Social Network Search* (**MO-MSNS**) problem focuses on improving the search operation of a smartphone user by optimizing the neighbour selection process. More precisely, given query $\mathcal{Q}$, a node aims to download from $\mathcal{C}$ an optimized *Query Routing Tree (QRT)* X, which minimizes the total Energy consumption and maximizes the Recall rate, according to the following formulation:

Given a social network of users, a query $\mathcal{Q}$ posted by an arbitrary user, a list of active users $\mathcal{U}'$, their coordinates and their profiles $\mathcal{P}$, define:

- The total **Energy** consumption of X

$$Energy(X) = \sum_{(u_a, u_b) \in X} e(u_a, u_b), \tag{3}$$

  where $e(u_a, u_b)$ denotes the energy consumption for transmitting one byte of data over the respective edge (WiFi, Bluetooth and 3G) using the energy profiling of the devices according to the energy model.

- **Recall** rate of X

$$Recall(X, \mathcal{Q}) = \frac{Relevant(\mathcal{Q}) \cap Retrieved(X, \mathcal{Q})}{Relevant(\mathcal{Q})} \tag{4}$$

where

$$\textit{Relevant}(\mathcal{Q}) = \bigcup_{u_a \in \mathcal{U}'} \bigcup_{k} o_{ak},$$

denotes the set of all objects o_{ak} from active users u_a that are relevant to $\mathcal{Q}$, that is, the profile p_a of user u_a contains terms found in $\mathcal{Q}$ and

$$\textit{Retrieved}(X, \mathcal{Q}) = \bigcup_{u_a \in X} \bigcup_{k} o_{ak},$$

denotes the actual set of objects o_{ak} from active users u_a that have been retrieved in response to $\mathcal{Q}$ over X, and p_a contains terms found in $\mathcal{Q}$.

The query processor then aims to:

$$\textit{minimize} \quad F(X) = (f_1(X), f_2(X)), \quad \text{subject to } X \subseteq \mathcal{G}', \tag{5}$$

with **objective functions**

$$f_1 = Energy(X), \qquad f_2 = -Recall(X, Q), \tag{6}$$

defined above in Equations 3 and 4, respectively.

4. Proposed Approach

This section details the proposed MOEA/D with Meta-Lamarckian Learning approach, named MOEA/D-ML.

4.1. MOEA/D framework

The MOEA/D-ML builds upon the decompositional generic MOEA/D framework proposed by Zhang and Li in [13], which requires the following pre-processing steps:

Decomposition: Initially, the MO-MSNS is decomposed into a number of scalar subproblems using the Tchebycheff approach with a set of uniformly distributed weight vectors as follows. Given the objective vector $F(X) = (f_1(X), f_2(X))$, a weight vector λ^i, which remains fixed for each subproblem for the whole evolution [46] and a reference point $z^* = (z_1, z_2)$, which is a vector with all the best values z_k found so far for each objective f_k, the objective function of a subproblem i is stated as:

$$g(X|\lambda^i, z^*) = \sum_{k=1}^{2} \{\lambda_k^i | f_k(X) - z_k|\}. \tag{7}$$

Assuming that there are N weight vectors $\lambda^1, \ldots, \lambda^N$ then the original MOP is decomposed to N scalar subproblems.

Representation: Each scalar subproblem with an objective function $g(X|\lambda^i, z^*)$, $i = 1, \ldots, N$ has a representative solution X that is the best solution found so far for that particular subproblem during the evolution. The set of all representative solutions at each generation is called the Internal Population (IP), where the IP is equal to $|IP| = N$. In MO-MSNS problem, a solution[3] X is a QRT of fixed-size $|\mathcal{G}'|$, i.e., the active smartphone users that can participate in the resolution of $\mathcal{Q}$. Therefore without loss of generality, let X be a fixed-size vector in which each index a corresponds to a user u_a and the value of that index corresponds to u_a's parent. The root of the tree is the query user (for simplicity noted as u_1). A negative value -1 in any position indicates that the given user is not currently selected in the query routing tree X.

Subpopulations/Neighbourhoods: Moreover, N subpopulations (or neighbourhoods) are generated for each subproblem. The neighbourhood B^i of a subproblem associated with a weight vector λ^i is composed of the indexes of the subproblems whose associated weight vectors are the $T << N$ closest (in terms of Euclidean distance) to λ^i including itself. This is due to the argument of Zhang and Li in [13] that the optimal solutions of the i^{th} and j^{th} subproblems are close to each other in the search space iff the λ^i and λ^j are close to each other in the weight space. Therefore, the genetic information of the i^{th} subproblem should be helpful for solving the j^{th} subproblem and vice-versa.

At each generation gen, the population IP is evolved by generating a new solution for each subproblem. For the i^{th} subproblem with $g(X|\lambda^i, z^*)$ a new solution Y, known as offspring, is generated using the genetic operators (i.e., crossover and/or mutation). Then a solution Z is generated by locally optimizing solution Y using a local search heuristic. The local search heuristic is adaptively selected from a pool of local search heuristics of size L using a reward vector $R^i = (r_1, \ldots, r_L)$, assigned to each subproblem i, which is constructed and updated at each generation by a Meta-Lamarckian (ML) learning approach. Finally, solution Z is used for the following updates. (1) Z is selected as the new representative of the i^{th} subproblem and therefore replaces the current best solution

[3]The terms *"solution"*, *"individual"*, *"vector"* and *"QRT"* are used interchangeably.

X, iff $g(Z|\lambda^i, z^*) < g(X|\lambda^i, z^*)$[4]. (2) For each $j = 1, \ldots, n$, update the reference point, if $z_j < f_j(Z)$ then set $z_j = f_j(Z)$. (3) Update the set of non-dominated solutions (i.e., Pareto Front - PF) found so far during the evolution as follows, $PF = PF \cup \{Z\}$ if Z is not dominated by any solution $X \in PF$, and $PF = PF \setminus \{X\}$ if $Z \prec X$, for all $X \in PF$. (4) Update the neighbourhood B^i of the i^{th} subproblem: the new solution Z of the i^{th} subproblem is compared with all representative solutions X in the neighbourhood $B^i = \{i_1, \ldots, i_T\}$. Z updates X iff $g(Z|\lambda^j, z^*) < g(X|\lambda^j, z^*)$, for all $j \in \{i_1, \ldots, i_T\}$. The same process is followed for all N subproblems. The evolution stops after a termination criterion is satisfied, such as a maximum number of generations gen^m is reached, or the PF has not converged after a fixed number of consecutive generations gen^c.

4.2. MOEA/D-ML main steps

The detailed steps of **MOEA/D-ML** are presented below:

Input:

- a MO-MSNS problem instance (see Subsection 3.2);
- a termination criterion (gen^m);
- the number of decomposed subproblems N and thus weight vectors $\{\lambda^1, \ldots, \lambda^N\}$.
- the pool of local search heuristics of size L.
- the size of the neighbourhood T of each subproblem.

Output: a set of non-dominated solutions PF.

Step 1: Initialization

1.1 Set $PF = \emptyset$;

1.2 Decompose the MO-MSNS problem into N scalar subproblems;

1.3 Initialize $IP = \{X^1, \ldots, X^N\}$ corresponding to subproblem with weight vectors $\lambda^1, \ldots, \lambda^N$ respectively, and evaluate it using Eq. (7);

1.4 Compute the Euclidean distance between each pair of the weight vectors to construct the neighbourhood $B^i = \{i_1, \ldots, i_T\}$ for each subproblem i, so that $\lambda^i_1, \ldots, \lambda^i_T$ are the T closest weight vectors to λ^i (including λ^i itself);

[4] $g(Z|\lambda^i, z^*) > g(X|\lambda^i, z^*)$ for maximization subproblems

1.5 Set $R^i = (r_1^i, \ldots, r_L^i) = \mathbf{0}$, where $i = 1, \ldots, N$;

Step 2: Main Loop

2.1 Set $gen = 1$;

2.2 **Genetic Operation:** For the i^{th} subproblem generate a new solution Y^i using conventional genetic operators (i.e., Selection, Crossover and Mutation as in [13]). In particular, two parent solutions are randomly selected from the neighbourhood B^i of the i^{th} subproblem. The two parent solutions are recombined using a two-point crossover to produce a new solution - the offspring - with a probability r_c. The offspring is then modified with a random mutation operator with a probability r_m. Finally, evaluate the new solution Y^i using Eq. (7).

2.3 **Meta-Lamarckian Learning:** Then select a local search heuristic from the pool of local search heuristics using a Meta-Lamarckian learning approach and the reward vector R^i. Apply the selected local search to Y^i to generate Z^i and evaluate it using Eq. (7). Calculate the reward r_j of the selected local search heuristic $j \in \{1, \ldots, L\}$ using a reward function that measures the improvement contributed by j to generate Z^i with respect to X^i and Y^i. Update the reward vector $R^i = (r_1, \ldots, r_L)$, accordingly.

Step 3: Update: Use solution Z^i to update z^*, IP, PF and B^i. If $i < N$ then $i = i + 1$ and goto Step 2.2;

Step 4: Termination: If the termination criterion $gen = gen^m$ is satisfied then terminate the algorithm and output the PF, otherwise goto Step 2.1;

4.3. Definition of Reward Function

During the Meta-Lamarckian learning in Step 2.3 a *reward* (ranging from 0 to 1) is calculated to measure the improvement contributed by a local search heuristic when applied to Y to generate a new solution Z, recalling that solution X is the best solution found so far during the evolution for subproblem i, using the

following rules:

$$r = \begin{cases} 1 & \text{(a) if } g(Z) < g(X) < g(Y) \\ \frac{g(Y)-g(Z)}{g(X)-g(Z)} & \text{(b) if } g(Z) < g(Y) \leq g(X) \\ \frac{g(Y)-g(Z)}{g(Y)-g(X)} & \text{(c) if } g(X) \leq g(Z) < g(Y) \\ 0 & \text{(d) otherwise} \end{cases} \quad (8)$$

where $g(X)$, $g(Y)$ and $g(Z)$ correspond to $g(X|\lambda^i, z*)$, $g(Y|\lambda^i, z*)$ and $g(Z|\lambda^i, z*)$, respectively. The proposed reward function and rules reflect to the actual contribution of the local search approach in the scalar objective function space by taking into consideration the actual replacement of $g(Z)$ towards the optimal solution with respect to $g(X)$ and $g(Y)$ as follows:

- Rule (a) rewards the local search heuristic with the maximum possible value (i.e., 1), since it generates a solution Z that is better than the current best solution X even if the genetic operation generated a poorest offspring Y than X.
- Rule (b) rewards the local search heuristic with a possibly moderate value, since it generates a solution Z that is the same or better than the current best X but the improvement builds upon the already improved solution Y that is generated by the genetic operation.
- Rule (c) rewards the local search heuristic with a possibly moderate value, as it generates a solution Z that is better than Y, but at most as good as X.
- In Rule (d) the local search has not been rewarded since no improvement is observed. Note that $g(Z|\lambda^i, z*) = g(Y|\lambda^i, z*)$ in the worst case.

4.4. Pool of general Local Search (LS) heuristics

In order to maintain the robustness and generalizability of our proposition as well as to promote the adaptiveness and contribution of our learning strategy that follows, we decided to use six general local search heuristics, which have been frequently used on permutation or sequencing problems [47] in the past:

- **Swap Heuristic** (Sw): randomly selects and swaps the parent nodes of two users $\{u_i, u_j\} \in Y$.

- **Double Swap** (DSw): performs the random swap heuristic twice.
- **Copy Heuristic** (CH): randomly selects the parent node of a user u_i and copies it to another user u_j.
- **Shift Heuristic** (Sh): randomly chooses to perform either a backward shift or a forward shift. A backward shift randomly selects a parent node from its current position of user u_i and inserts it at a position before a user u_j, where $i > j$. A forward shift is similar to backward shift, but it selects a parent node from its current position of user u_i and inserts it at a position after a customer u_j, where $i < j$.
- **Double Shift** (DSh): is the combination of Backward and Forward Shift.
- **Inverse** (IH): is the selection of a portion of Y^i and reversing its order.

In all cases, the local search approaches are used greedily. That is, the local search starts from solution Y^i and continues for a pre-defined number I of iterations. Each time an improvement is achieved and an improved solution Z^i is obtained, the search continues from the improved solution Z^i for the remaining moves, otherwise the search continues from the original Y^i solution. Note that moves that lead to infeasibility such as disconnecting the tree or forming cycles are not allowed during local search.

In [47], Reeves has mentioned some interesting relationships on local search operators for SOO. For example, one can say that Swap is the weakest from all these operators and an optimal solution with respect to any other operator is also optimal for Swap heuristic. In that sense, Swap is subsumed by all other operators in the pool of local search approaches. Similarly, Forward and Backward Shifts are subsumed by Double Shift, which should theoretically produce better solutions given that its search space is twice as large. One can say that this is also true for the Inverse heuristic, in which the Copy heuristic is subsumed. However, in this article the type of questions that are considered interesting and challenging for investigation are: (i) whether one operator of the same size outperforms another in all test instances and all objectives; considering the fact that different operators induce different landscapes and that the combination of these operators in different times of the evolution may perform better and faster; (ii) whether there is a bias of different operators in different areas of the objective space, for dissimilar classes of MO-MSNS instances and if this bias can be learned during the evolution.

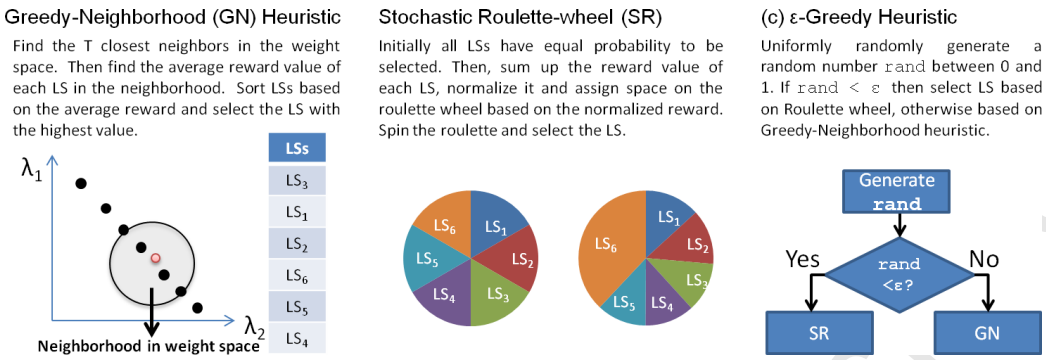


Figure 1: The adaptive search heuristics.

4.5. Proposed Learning Strategy

The idea behind the adaptive strategy for Meta-Lamarckian learning in the proposed decompositional MOEA aims at providing insights to those questions by learning the effectiveness of each local search heuristic in dealing with the current objectives, problem instances and problem area as the search progresses.

At the beginning of the evolution and for a predefined number of generations g^t each single local search heuristic is given the opportunity to hybridize with the MOEA for locally optimizing the solution Y^i of each subproblem i. The reward of each local search is calculated using Equation 8. Those initial reward values will be used later to guide future LS choices and will change dynamically as the overall search progresses. This is commonly known as the training stage, after which the learning phase takes over, using the proposed ϵ-greedy strategy, which probabilistically alternates between the following two learning strategies.

The **Greedy Neighbourhood-based** (GN) strategy works as follows:

Step 1: Locate the neighbourhood B^i of size T of a subproblem i.

Step 2: Find the average reward value $\overline{R^i} = (\overline{r_1}, \ldots, \overline{r_L})$ in the neighbourhood B^i.

Step 3: Select local search j with the maximum average reward value $\overline{r_j}$.

For each parent solution X in the population to be searched, the GN strategy locates its T closest neighbours from the archived database by using simple Euclidean measures. Note that each neighbour solution Y is associated with a pool

of LS strategies and their current rewards achieved so far. Then the average reward of each LS from the pool of LS approaches is measured. Finally, the LS with the highest average reward in the neighbourhood is used to improve the parent solution X. After LS, the X and the reward of the selected LS are updated in the database.

The **Stochastic Roulette-wheel** (SR) strategy works as follows:

Step 1: Sum the reward values $r_j \in R^i$ of all local search approaches for subproblem i.

Step 2: Determine normalized relative reward value of each member of R^i.

Step 3: Assign space on the roulette wheel for each local search based on the normalized value.

Step 4: Generate a random number between 0 and 1, select the local search corresponding to the portion of the wheel in which the chosen random number falls.

The SR strategy ensures that the probability that a LS approach is selected is biased from its own previous performance, which changes dynamically as the overall search progresses. In particular, each time a parent solution X is about to be locally searched, a biased roulette wheel is used to pick up the subsequent LS, based on the rewards taken and archived over all previous LSs. The highest a LS's reward for particular X, the highest the probability to be selected for future local searches. This strategy, ensures diversity in the choice of LS approaches since it restricts a LS from completely dominating the search.

Finally, the proposed ϵ-**greedy** strategy, which is based on a pre-defined $0 < \epsilon < 1$ parameter, generates a random number $rand$ between 0 and 1 and selects the Stochastic Roulette-wheel strategy if $rand < \epsilon$ or the Greedy Neighbourhood-based strategy otherwise. Each time a local search approach j is adaptively selected and used on Y^i to generate a solution Z^i, the reward value of $r_j \in R^i$ is updated using Equation 8.

The ϵ-greedy strategy promotes both cooperation and competition. It promotes competition by giving the opportunity of a LS that performs better along the direction of one objective and within a particular neighbourhood of a particular problem and/or a particular instance of the problem, to be rewarded with greater chances of being selected for subsequent optimizations during the evolution. It promotes cooperation by allowing the solutions of subproblems to exchange neighbourhood information, not only about the genotype and fitness of

the individuals as in the conventional MOEA/D, but also about the performance of each LS on a particular subproblem i and its neighbourhood B^i.

5. Experimental Studies

In this section, we introduce the experimental setup followed in our experimental studies. First we discuss the real data sets utilized to design our synthetic data sets and test instances on the MO-MSNS problem. We then describe the various methods that are compared with our proposed MOEA/D-ML and their parameter settings, as well as, the performance metrics utilized to evaluate the algorithms' performance. In order to evaluate the efficacy of incorporating the Meta-Lamarckian (ML) learning approach in MOEA/D seven experimental series are developed. We then compare the proposed approach with the state-of-the-art in MOEAs based on Pareto-Dominance, the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [6]. Finally, we test the generalizability of the proposed method on eight test instances of the well-known Multi-Objective Permutation Flow Shop Scheduling Problem (PFSSP). We introduce three experimental series in order to compare its performance against the Multi-Objective Genetic Local Search (MOGLS) approach and other MOEA/D variants. Statistical and sensitivity analyses are also provided in order to support the evaluation results.

5.1. Experimental Setup for MO-MSNS

In our experimental studies, we examined a mobile social scenario that is derived from the following two real datasets as in [12]:

GeoLife [15] (*mobility*): This real dataset by Microsoft Research Asia includes 1,100 trajectories of a human moving in the city of Beijing over a life span of two years (2007-2009). The average length of each trajectory is $190,110$ points, while the maximum trajectory length is 699,600 points. Notice that 95% of the GeoLife dataset refers to a granularity of 1 sample every 2-5 seconds or every 5-10 meters.

DBLP [16] (*social*): This real dataset by the DBLP Computer Science Bibliography website, includes over 1.4 million publications in XML format. In particular, the dataset records the paper titles, paper urls, co-authors, links between papers and authors and other useful semantics. In order to map this dataset to our problem, we assume that each object is an author's paper. We also assume that each object is "tagged" by the keywords found in the paper title.

In order to link the above datasets we have constructed a mobile social scenario that uses the DBLP social dataset and GeoLife mobility dataset. The DBLP

dataset is used to construct a social graph $\mathcal{G}$ of authors that are related based on their research interests (i.e., keywords of their articles' titles) as well as their co-authorships that are attributes of the DBLP dataset. Then we have mapped each DBLP author to a trajectory of the Geolife dataset. That is, we have extracted 1,100 authors from the DBLP dataset and we have mapped them to the 1,100 trajectories of the Geolife dataset using a 1-1 correspondence. This resulted in a social graph with 1,100 mobile DBLP authors moving in the city of Beijing.

In our experiments, we utilize the following two queries:

```
-- Query 1
SELECT S.title, S.url
FROM SmartphoneUsers S, Query Q
WHERE (distance(S.x,S.y,Q.x,Q.y) < 10 KM)
      AND S.Title LIKE '%optimization%';

-- Query 2
SELECT S.title, S.url
FROM SmartphoneUsers S, Query Q
WHERE (distance(S.x,S.y,Q.x,Q.y) < 10 KM)
      AND S.Title LIKE '%data%';
```

where "S.x,S.y" represent the (x, y) coordinates of a Smartphone user in S and "Q.x,Q.y" represent the (x, y) coordinates of the query user. The query search will be conducted within an area of radius 10 KM.

Table 2: Experimental Execution Scenarios and Test Instances for MO-MSNS.

Test Instance	Keyword Query $\mathcal{Q}$	Time	Active Users $\mathcal{U}'$	Total Objects	Relevant Objects
T1	Query1	1:48:4[0-9] am	95	8884	183
T2	Query2	1:48:4[0-9] am	95	8884	657
T3	Query1	11:34:3% am	121	10691	201
T4	Query2	11:34:3% am	121	10691	859
T5	Query1	17:34:[1-5]% pm	139	12316	231
T6	Query2	17:34:[1-5]% pm	139	12316	1004
T7	Query1	20:[2-3]4:3% pm	165	14630	254
T8	Query2	20:[2-3]4:3% pm	165	14630	1162

In our experimental studies, we have examined eight test instances as summarized in Table 2, denoted as T and designed using the popular Factorial design process [48], which represent mobile social scenarios of various time periods (e.g., 1:48:4[0-9] corresponds to a network snapshot of 10 seconds at 1:48 am), in order to capture different mobility patterns that are inherent in the GeoLife dataset

with different number of active users and for different queries in order to vary the number of relevant objects.

5.2. MO-MSNS Algorithms: MOEA/D-ML, NSGA-II and other MOEA/D-variants

The hybridization of the conventional MOEA/D with a single LS heuristic that is used throughout the search is coined Individualistic MOEA/D (note that this hybridization is an optional step for the conventional MOEA/D framework [13]). In this paper, six different Individualistic MOEA/D approaches are designed, i.e., MOEA/D-Sw, MOEA/D-CH, MOEA/D-DSw, MOEA/D-Sh, MOEA/D-DSh and MOEA/D-IH, by hybridizing MOEA/D with swap, copy, double swap, shift, double shift and inverse heuristics, respectively.

The idea of allowing the competition and cooperation among different LSs [49] has given rise to the so-called adaptive strategies with Meta-Lamarckian learning. The MOEA/D combined with the most basic Meta-Lamarckian learning scheme which selects LSs with a simple Random Walk (RW) over the available methods is denoted as MOEA/D-RW. This method does not adapt but at least it gives the opportunity to every LS approach to locally improve a solution. Finally in this paper, three decompositional MOEAs with different Meta-Lamarckian learning strategies are designed. That is, (i) MOEA/D-SR that uses the Stochastic Roulette-wheel approach as its learning strategy, (ii) the MOEA/D-GN that uses the Greedy Neighbourhood-based strategy throughout the evolution and (iii) the proposed MOEA/D-ML with the ϵ-greedy strategy. For ease of reference, the abbreviations of all MOEA/D variants used in the experimental series are summarized in Table 3.

Finally, the proposed MOEA/D-ML is compared with the state-of-the-art in MOEAs based on Pareto-dominance NSGA-II. NSGA-II maintains a population IP of size N at each generation gen, for gen^m generations. NSGA-II adopts the same evolutionary operators (i.e. selection, crossover and mutation) for offspring reproduction as MOEA/D. The key characteristic of NSGA-II is that it uses a fast non-dominated sorting and a crowded distance estimation for comparing the quality of different solutions during selection and to update the IP and the PF. We refer interested readers to [6] for details.

5.3. Performance Metrics

The performance of a MOEA is often evaluated from two perspectives. That is, the obtained non-dominated set should be (i) as close to the true Pareto Front as possible and mainly corresponds to the convergence (or quality) of the obtained

Table 3: Abbreviations of MOEA/D variants used in the experimental series

Notation	Description
MOEA/D	Conventional MOEA based on Decomposition [13].
MOEA/D-Sw	Individualistic MOEA/D hybridized with Swap Heuristic.
MOEA/D-DSw	Individualistic MOEA/D hybridized with Double-swap Heuristic.
MOEA/D-CH	Individualistic MOEA/D hybridized with Copy Heuristic.
MOEA/D-Sh	Individualistic MOEA/D hybridized with Shift Heuristic.
MOEA/D-DSh	Individualistic MOEA/D hybridized with Double-shift Heuristic.
MOEA/D-IH	Individualistic MOEA/D hybridized with Inverse Heuristic.
MOEA/D-RW	MOEA/D with the Random Walk learning strategy.
MOEA/D-SR	MOEA/D with the Stochastic Roulette-wheel learning strategy.
MOEA/D-GN	MOEA/D with the Greedy Neighbourhood-based learning strategy.
MOEA/D-ML	Proposed MOEA/D with the ϵ-greedy learning strategy.

solutions, and (ii) distributed as diversely and uniformly as possible. In the literature, there is no single metric that can reflect both of these aspects and thus a number of metrics are often used [50, 6, 11, 7, 51]. In this study, we have used the following four metrics:

- **Coverage (C):** commonly used for comparing two sets of non-dominated solutions A and B, originally proposed by Zitzler and L. Thiele in [52]. The $C(A, B)$ metric, which is often considered as a MOEA quality metric, calculates the ratio of the non-dominated solutions in B dominated by the non-dominated solutions in A, divided by the total number of non-dominated solutions in B. Hence,

$$C(A, B) = \frac{|\{x \in B | \exists y \in A : y \prec x\}|}{|B|}.$$

  Therefore, $C(A, B) = 1$ means that all non-dominated solutions in B are dominated by the non-dominated solutions in A. Note that $C(A, B) \neq 1 - C(B, A)$.

- **Distance from reference set (I_D):** defined by Czyzzak et al. in [53] as follows:

$$I_D(A) = \frac{\sum_{y \in R}\{min_{x \in A}\{d(x, y)\}\}}{|R|}.$$

This shows the average distance from a solution in the reference set R to the closest solution in A. The smaller the value of I_D is then the closer the set A is to R indicating better convergence. In the absence of the real reference set (i.e., PF) in MO-MSNS the average distance of each single point to the nadir point is used.

- **Hypervolume (I_H):** originally proposed by Zitzler et al. in [27] indicating the area dominated by at least one solution in the obtained non-dominated set A. Therefore high I_H indicates better diversity. The metric is formally defined as

$$I_H(A) = \int_{z \in \cup_{x \in A}} \dots \int_{HV(f(x), f^*)} 1.dz,$$

where $HV(f(x), f^*) = [f_1(x), f_1^*] \times \dots \times [f_m(x), f_m^*]$ is the Cartesian product of the closed intervals $[f_i(x), f_i^*], i = 1, \dots, m$. Since we consider minimization objectives the reference point $f^* = (f_1^*, \dots, f_m^*)$ is the ideal worst point, i.e., $f_i^* = max_{x \in \Omega} f_i(x), \forall i = 1, \dots, m$.

- **Number of Non-Dominated Solutions (NDS):** a straightforward metric proposed by Weicker et al. in [51] that is usually considered in cases of real-life discrete optimization problems such as the MO-MSNS showing the cardinality or the number of Non-Dominated Solutions in set A, i.e.

$$NDS(A) = |A|.$$

In these cases, it is more desirable to obtain a high number of $NDS(A)$ in order to provide an adequate number of Pareto optimal choices. In contrast, and usually in cases of continuous optimization [13], a high number of NDS is not desirable, since the decision making procedure becomes more complicated and more time consuming. However, the NDS should be considered in combination with other metrics (e.g. Δ and C metrics), since it is usually desirable to have a high number of NDS when the solutions is of high quality (i.e. low C-metric) and spread (i.e. low Δ-metric) in the objective space.

5.4. Experimental Layout and Algorithmic Settings for MO-MSNS

In the experimental studies on the MO-MSNS problem that follow, several decompositional MOEAs have been examined and compared with the proposed MOEA/D-ML approach:

- **Experimental series 1** examines the effect of each LS on the MOEA/D individualistically. The conventional MOEA/D approach as proposed by Zhang and Li in [13] is compared with the six individualistic MOEA/D variants defined above (i.e., MOEA/D-Sw, MOEA/D-DSw, MOEA/D-CH, MOEA/D-Sh, MOEA/D-DSh and MOEA/D-IH).

- **Experimental series 2** compares MOEA/D-ML and the two best performing Individualistic MOEA/Ds of Experimental series 1.

- **Experimental series 3** compares MOEA/D-ML and the MOEA/D-RW variant that uniformly randomly selects a local search heuristic from the pool of generalized local search heuristics.

- **Experimental series 4** compares three MOEA/Ds combined with different adaptive learning strategies: i) MOEA/D-SR with Stochastic Roulette-wheel , ii) MOEA/D-GN with Greedy Neighbourhood-based learning, and iii) MOEA/D-ML with ϵ-greedy learning strategy.

- **Experimental series 5** compares MOEA/D-ML and NSGA-II.

- **Experimental series 6** provides statistical analysis on the performance of both MOEA/D-ML and NSGA-II.

- **Experimental series 7** provides sensitivity analysis on the step size (number of iterations) of local search of the MOEA/D-ML.

The algorithmic parameters are set as follows: termination criteria gen^m=250 and $gen^c = 25$, population size and number of subproblems N=120, crossover rate r_c=0.9, mutation rate r_m=0.5, neighbourhood size T=10, the size of the pool of local search heuristics $L = 6$, the number of iterations for each local search is set to $I = 10$ and the training phase for the MOEA/D with Meta-Lamarckian Learning approaches is set to $g^t = 10$. For the MOEA/D-ML that utilizes the proposed ϵ-greedy strategy, the ϵ is set to 0.8. Moreover, in all simulations the recall and energy objectives are evaluated as in Subsection 3.2. All algorithms were coded in Java programming language and run on an Intel(R) Core(M) i5 CPU 2.4GHz Windows 7 server with 4 GB RAM. Note that in our experimental studies we have used the same number of function evaluations for all methods, for fairness, and each algorithm is executed 20 times in each study. Statistical analysis on the multiple runs is provided in the experimental studies below.

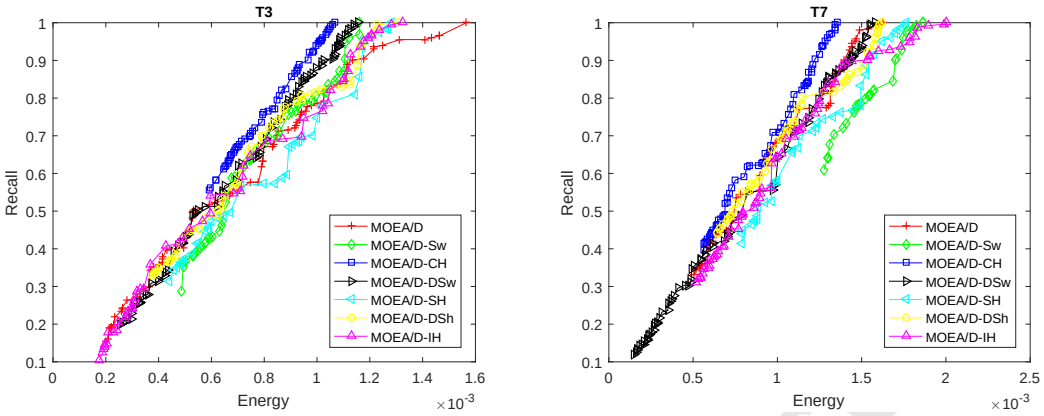


Figure 2: Experimental Series 1 (MO-MSNS) - Comparison between conventional MOEA/D with Individualistic MOEA/D variants on two representative test instances T3 and T7.

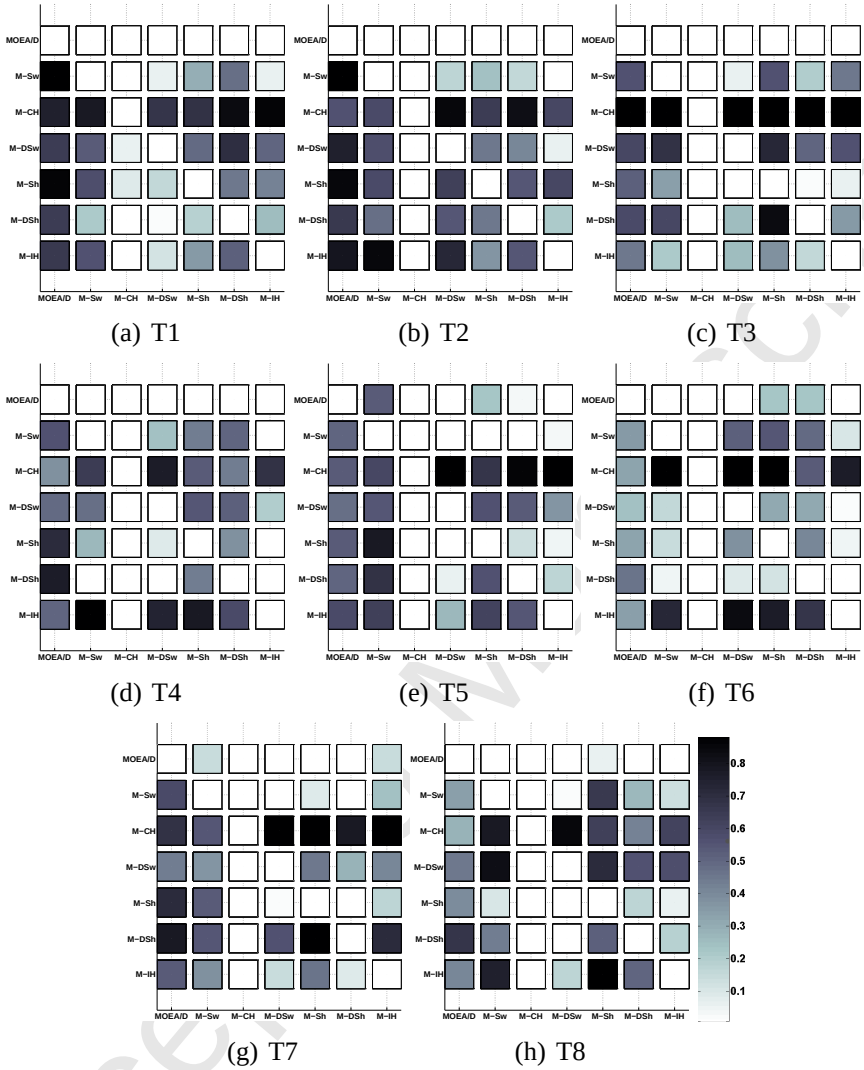
5.5. Experimental Results for the MO-MSNS problem

The results of the seven experimental series are presented below:

5.5.1. Experimental Series 1 (MO-MSNS) - Effect of LS on MOEA/D

In experimental series 1, we examine the effect of each LS on the MOEA/D individualistically. The main purpose of this experiment is not just to test if the hybridization of the MOEA/D with a single LS improves its performance, but also to examine the behaviour of each LS on various test instances of the MO-MSNS problem in general as well as during the evolution. Therefore, we examine and compare the conventional MOEA/D and its six individualistic variants, i.e., MOEA/D-Sw, MOEA/D-CH, MOEA/D-DSw, MOEA/D-Sh, MOEA/D-DSh and MOEA/D-IH. The algorithms are evaluated on all eight test instances of Table 2 using the performance metrics of Subsection 5.3.

Figure 2 shows that the hybridization of MOEA/D with any Local Search heuristic improves the performance of the conventional MOEA/D in terms of both convergence and diversity. The improvement, however, achieved by some local search heuristics such as the Copy (MOEA/D-CH), the Double Swap (MOEA/D-DSw) and the Inverse (MOEA/D-IH) is higher than the others. For example, it is visually possible to argue that the MOEA/D-CH provides a good convergence and diversity in both T3 and T7. However, one can say that it does not converge well towards the energy objective. MOEA/D-DSw and MOEA/D-IH, on the other hand, provide a better diversity and convergence with respect to the energy objective. These are the test instances with fewer relevant objects of interest and



(a) T1 (b) T2 (c) T3

(d) T4 (e) T5 (f) T6

(g) T7 (h) T8

Figure 3: Experimental Series 1 (MO-MSNS) - Comparison between the conventional MOEA/D and the six Individualistic MOEA/D variants in terms of the Coverage metric C(A,B). In this figure, the darker a square is, the closer the C(A,B) is to 1 (as illustrated by the colorbar in the bottom-right corner of the figure) where A and B are the MOEAs depicted on the horizontal and vertical axes, respectively. For example, the square in row 1 and column 1 of T1 (in subfigure (a)) represents the value C(M-IH,MOEA/D)=0.71 and shows that the hybrid MOEA/D with the Inverse Heuristic (IH) outperforms the conventional MOEA/D.

Table 4: Results of Experimental Series 1 (MO-MSNS) - Comparison between the conventional MOEA/D and the six Individualistic MOEA/D variants in terms of the performance metrics I_D and I_H. The best results of each test instance are denoted in bold.

Alg:	MOEA/D		M-Sw		M-CH		M-DSw		M-Sh		M-DSh		M-IH	
TI	I_D	I_H	I_D	I_H	I_D	I_H	I_D	I_H	I_D	I_H	I_D	I_H	I_D	I_H
1:	0.2602	0.2600	0.0912	0.4980	**0.0595**	0.7120	0.0647	**0.8450**	0.0856	0.5670	0.0759	0.7370	0.0640	0.6690
2:	0.2257	0.2590	0.0769	0.5390	**0.0445**	**0.9270**	0.0660	0.7710	0.0754	0.6760	0.0584	0.7500	0.0849	0.5920
3:	0.3416	0.0710	0.0925	0.5200	0.1030	0.2620	**0.0760**	0.7000	0.1044	0.5170	0.0920	0.5290	0.0815	**0.8160**
4:	0.1477	0.3190	0.0694	0.6580	**0.0421**	**0.8580**	0.0646	0.7780	0.0990	0.5600	0.1290	0.4310	0.0667	0.6880
5:	0.1508	0.2710	0.1472	0.3490	0.0827	0.5280	**0.0730**	**0.7590**	0.1112	0.3910	0.0955	0.4740	0.0945	0.5890
6:	0.2907	0.1460	0.0819	0.7050	**0.0528**	0.7330	0.0672	**0.7330**	0.0834	0.7090	0.1190	0.4130	0.0737	0.6310
7:	0.1653	0.2900	0.1533	0.1320	0.0873	0.4380	**0.0657**	**0.8220**	0.1195	0.3340	0.0960	0.3920	0.0854	0.5010
8:	0.2283	0.1090	0.0687	0.6560	**0.0536**	**0.8540**	0.0631	0.7440	0.0780	0.5410	0.1037	0.3420	0.0834	0.5750
mean:	0.23	0.22	0.10	0.51	**0.07**	0.66	**0.07**	**0.76**	0.09	0.53	0.09	0.50	0.07	0.63
std:	0.07	0.09	0.03	0.19	0.02	0.23	0.00	0.05	0.02	0.12	0.02	0.15	0.01	0.09

therefore require more effort in finding the "appropriate" users to participate in the Query Routing Tree (QRT). This is also shown by the statistical results summarized in Tables 4, 5 and Figure 3. Table 4 shows a comparison of the MOEA/D variants in terms of the distance between the obtained PF and the reference set (I_D) and the hypervolume (I_H). The results show that the MOEA/D-CH converges closer to the reference set in five out of eight test instances (lowest value of I_D metric along rows) and provide the best diversity in three test instances (highest value of I_H metric along rows). Moreover, the MOEA/D-DSw obtains a better I_D in the remaining three test instances and provide a better diversity in half of the test instances. Finally, the MOEA/D with the Inverse Heuristic (IH) provides the best diversity in test instance T3.

Figure 3, shows a comparison between the MOEA/D variants in terms of the C-metric. The colored squares of the subfigures represent the C(A,B) value, where A and B are the MOEAs depicted on the vertical and horizontal axes of the subfigure, respectively. For example, in T1 of subfigure 3(a), the fifth row represents all C(M-CH,B) values of the MOEA/D with the Copy Heuristic (CH) with respect to the remaining algorithms, substituting B with the respective algorithm shown on the x-axis from left to right. The darker a square is the closer the C-value is to 1. Therefore, examining the top row of each subfigure (a-h), the results show that the PF obtained by the conventional MOEA/D does not dominate any of the PFs obtained by the other MOEA/D hybrids.

In addition, the PF obtained by the MOEA/D-CH is of higher quality compared to the other MOEA/Ds even if it does not converge towards the energy objective in some test instances (see Figure 2) as it is discussed above. Therefore, one can say that there is a preference of different LS approaches towards differ-

Table 5: Results of Experimental Series 1 (MO-MSNS) - Comparison between the conventional MOEA/D and the six Individualistic MOEA/D variants in terms of NDS. The best results of each test instance are denoted in bold.

TI	MOEA/D	M-Sw	M-CH	M-DSw	M-Sh	M-DSh	M-IH
1:	9	58	**110**	95	57	68	82
2:	13	63	**163**	94	64	97	49
3:	5	52	60	**83**	39	53	54
4:	30	85	**156**	91	53	29	84
5:	23	30	72	**79**	40	52	59
6:	8	63	**138**	64	64	40	80
7:	18	30	67	**86**	37	57	64
8:	16	80	**110**	**110**	77	58	62
mean:	15.25	57.63	**109.5**	87.75	53.88	56.75	66.75
std:	8.35	20.23	40.53	13.44	14.39	20.13	13.49

ent objective functions for the same test instances of the same problem domain. This is the reason why the quality of the non-dominated solutions obtained by MOEA/D-DSw and MOEA/D-IH is also higher in some cases. Finally, Table 5 summarizes the number of NDS obtained by each MOEA/D variant for each test instance T1-T8 of Table 2. The results show that MOEA/D-CH provides more Pareto-Optimal choices to the Decision Maker in five out of eight test instances, with MOEA/D-DSw having the higher NDS in the remaining three.

The conclusions drawn from experimental series 1 are summarized below:

(i) the traditional MOEA/D is outperformed by almost all MOEA/D hybrids in almost all cases, showing that the hybridization with local search heuristics improves its performance and

(ii) there is preference in the choice of the local search heuristic in different test instances as well as for different objectives of the same test instance.

The next experimental series aims to further justify the last conclusion of experimental series 1.

5.5.2. Experimental Series 2 (MO-MSNS) - Effect of Meta-Lamarckian Learning: comparing MOEA/D-ML and two best performing individualistic MOEA/Ds

In this experimental series, we compare our proposed MOEA/D-ML method and the two individualistic variants MOEA/D-CH and MOEA/D-DSw that performed best in experimental series 1 (that is, MOEA/D hybridised with the Copy

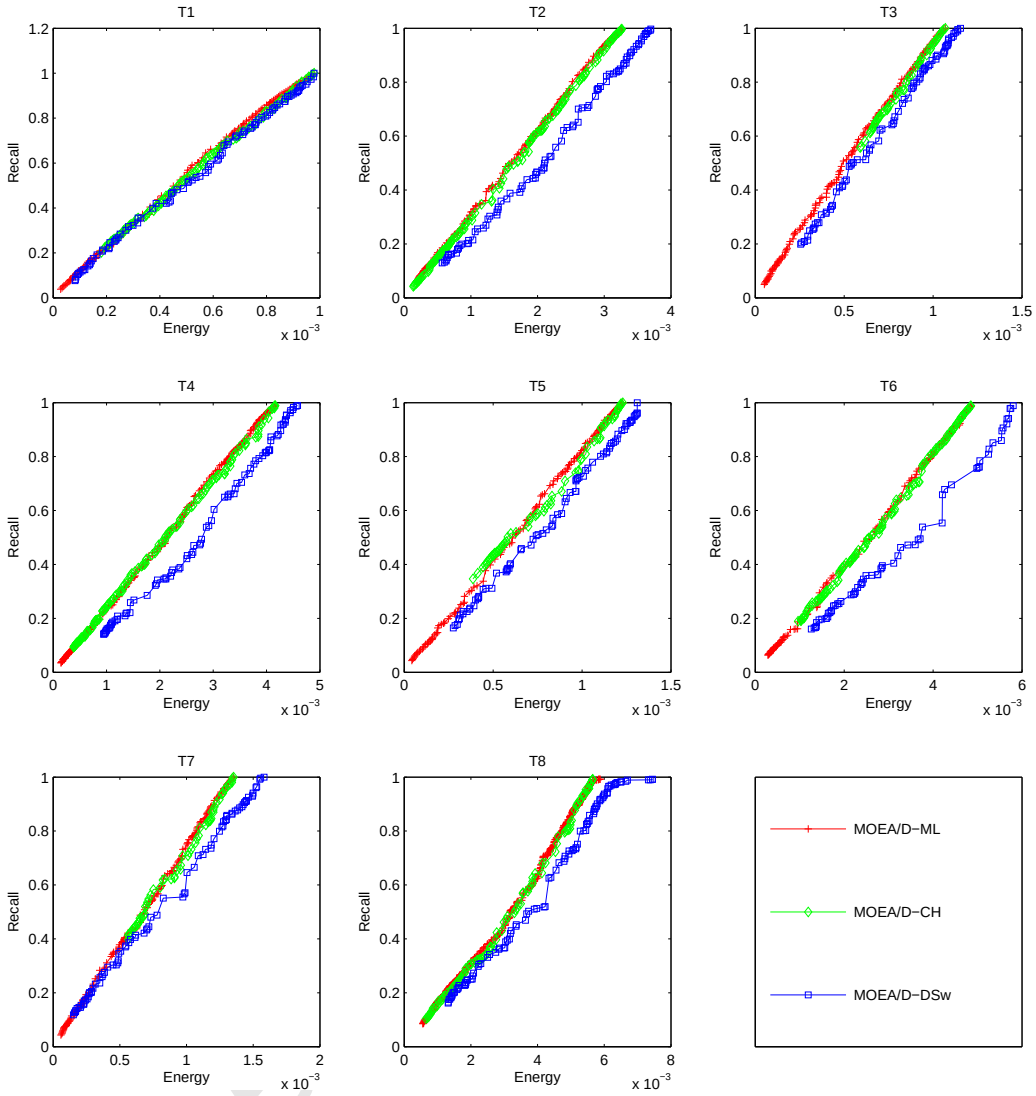


Figure 4: Experimental Series 2 (MO-MSNS) - The effect of Meta-Lamarckian Learning: The proposed MOEA/D-ML, i.e., MOEA/D with the ϵ-greedy Meta-Lamarckian learning approach is compared with MOEA/D-CH and MOEA/D-Sw, the best performing individualistic MOEAs (hybridized with the Copy and Double-Swap heuristics, respectively) of experimental series 1.

and Double Swap Heuristics, respectively). We adopt both statistical comparison in terms of the performance metrics introduced in Subsection 5.3, as well as visual comparison where necessary in all test instances of Table 2.

Table 6: Results of Experimental Series 2 (MO-MSNS) - MOEA/D-ML (i.e., the proposed MOEA/D with the ϵ-greedy learning strategy) is compared with individualistic variants MOEA/D-CH and MOEA/D-DSw (i.e., MOEA/D's hybridized with the Copy and Double Swap Heuristic, respectively), that performed well in Experimental Series 1, in terms of the performance metrics I_D, I_H and NDS (top) and coverage metric C (bottom). The best results of each test instance are denoted in bold.

Alg:	MOEA/D-ML		MOEA/D-CH		MOEA/D-DSw		MOEA/D-ML	MOEA/D-CH	MOEA/D-DSw
TI	I_D	I_H	I_D	I_H	I_D	I_H	NDS	NDS	NDS
1:	**0.0506**	**0.9210**	0.0595	0.6640	0.0647	0.8520	**150**	110	95
2:	**0.0436**	0.8980	0.0445	**0.9160**	0.0660	0.7330	**219**	163	94
3:	**0.0491**	**0.9000**	0.1030	0.2320	0.0760	0.6280	**160**	60	83
4:	**0.0373**	**0.9160**	0.0421	0.8210	0.0646	0.6730	**255**	156	91
5:	**0.0537**	**0.9160**	0.0827	0.4630	0.0730	0.6810	**149**	72	79
6:	**0.0485**	**0.8880**	0.0528	0.6640	0.0672	0.6560	**168**	138	64
7:	**0.0455**	**0.9270**	0.0873	0.4090	0.0657	0.7840	**171**	67	86
8:	**0.0288**	**0.8240**	0.0536	0.8250	0.0631	0.7070	**452**	110	110
mean:	**0.04**	**0.90**	0.07	0.62	0.07	0.71	**215.5**	109.5	87.75
std:	0.01	0.03	0.02	0.24	0.00	0.07	102.49	40.53	13.44

Test Inst.	C(MOEA/D-ML,MOEA/D-SR)	C(MOEA/D-SR,MOEA/D-ML)	C(MOEA/D-ML,MOEA/D-GN)	C(MOEA/D-GN,MOEA/D-ML)
1:	**0.5533**	0.2000	**0.8133**	0.0526
2:	**0.8721**	0.0000	**0.8447**	0.0000
3:	**0.4375**	0.0000	**0.7937**	0.0000
4:	**0.4510**	0.4423	**0.7569**	0.0000
5:	**0.5235**	0.4028	**0.8054**	0.0000
6:	**0.3433**	0.3406	**0.7619**	0.0000
7:	**0.4152**	0.2388	**0.8070**	0.0116
8:	**0.5752**	0.5273	**0.8429**	0.0455
mean:	**0.52**	0.27	**0.80**	0.01
std:	0.16	0.20	0.03	0.02

MOEA/D-ML is more effective than the individualistic MOEA/D variants. The results of Figure 4 show that the proposed approach provide better diversity and convergence than the two best performing individualistic approaches (i.e., MOEA/D-CH and MOEA/D-DSw) of experimental series 1. Here it is more important to notice the adaptive behaviour of the proposed hybrid with respect to the other two approaches. For example, considering the experimental results of test instance $T7$ (left bottom corner of Figure 4), it is evident that on the one hand MOEA/D-CH provides high quality non-dominated solutions for the sub-problems that favor the recall objective (i.e., upper half of the PF) and stops converging when the weights start favoring the energy objective. On the other hand, the MOEA/D-DSw provides some low-quality (compared to those of MOEA/D-CH) non-dominated solutions in that part of the PF and converges well towards the energy objective. The proposed MOEA/D, however, adaptively follows the pattern of each local search heuristic where they perform well, absorbs the best non-dominated solutions and finally provides a diverse and high quality set of Pareto-optimal solutions. The improvement on the performance of the MOEA/D

hybridized with an adaptive local search and a Meta-Lamarckian learning strategy is also summarized in the statistical results of Table 6. The results show that the MOEA/D-ML clearly outperforms the individualistic MOEA/D variants in all test instances with respect to all metrics adopted in this article.

Table 7: Results of Experimental Series 3 (MO-MSNS) - The proposed MOEA/D-ML, i.e., MOEA/D with the proposed Meta-Lamarckian learning approach (ϵ-greedy) is compared with the MOEA/D-RW Heuristic in terms of the performance metrics I_D, I_H, NDS and C. The best results of each test instance are denoted in bold.

Alg:	MOEA/D-ML		MOEA/D-RW		MOEA/D-ML	MOEA/D-RW	C(MOEA/D-ML,	C(MOEA/D-ML,
TI	I_D	I_H	I_D	I_H	NDS	NDS	MOEA/D-RW)	MOEA/D-RW)
1:	**0.0479**	**0.913**	0.0557	0.9	**150**	113	**0.74**	0.09
2:	**0.04**	**0.89**	0.049	0.794	**219**	98	**0.78**	0.06
3:	**0.045**	**0.91**	0.059	0.74	**160**	88	**0.8**	0.02
4:	**0.035**	**0.94**	0.06	0.69	**255**	86	**0.7**	0.00
5:	**0.05**	**0.93**	0.06	0.797	**149**	85	**0.71**	0.12
6:	**0.044**	**0.88**	0.047	0.80	**168**	118	**0.8**	0.05
7:	**0.042**	**0.92**	0.064	0.84	**171**	86	**0.7**	0.08
8:	**0.02**	**0.86**	0.063	0.737	**452**	81	**0.6**	0.23
mean:	**0.0418**	**0.8986**	0.0578	0.7913	**215.12**	94.37	**0.7210**	0.0812
std:	0.0081	0.0061	0.0344	0.0617	102.6846	13.9687	0.060	0.0721

5.5.3. Experimental Series 3 (MO-MSNS) - Effect of Meta-Lamarckian Learning: comparing MOEA/D-ML and MOEA/D-RW

In this experimental series, we compare the proposed MOEA/D with Meta-Lamarckian learning (ϵ-greedy strategy) and the MOEA/D with Random Walk (MOEA/D-RW) that uniformly randomly selects a local search heuristic from the pool of local search heuristics. We adopt both statistical comparison in terms of the performance metrics introduced in Subsection 5.3 as well as visual comparison where necessary in all test instances of Table 2.

The results of Table 7 show that the proposed MOEA/D-ML (ϵ-greedy) technique provides a more diverse (i.e., high I_H) Pareto Front, that is closer to the reference set (i.e., low I_D) and with more Pareto optimal solutions (i.e., high NDS) of higher quality (i.e., high C) compared to the MOEA/D-RW variant, in all eight test instances. In particular, MOEA/D-ML provides about 20% better I_D than MOEA/D-RW on average, it approximates the reference set by around 90% compared to around 80% of MOEA/D-RW and provides significantly more Pareto Optimal solutions. Finally, the Pareto Front solutions obtained by the proposed approach dominate around 72% of the Pareto Front solutions obtained by MOEA/D-RW while only around 8% are dominated, on average.

Furthermore, Figure 5 visually shows the dominance of the proposed MOEA/D-ML approach with respect to MOEA/D-RW in terms of both diversity and conver-

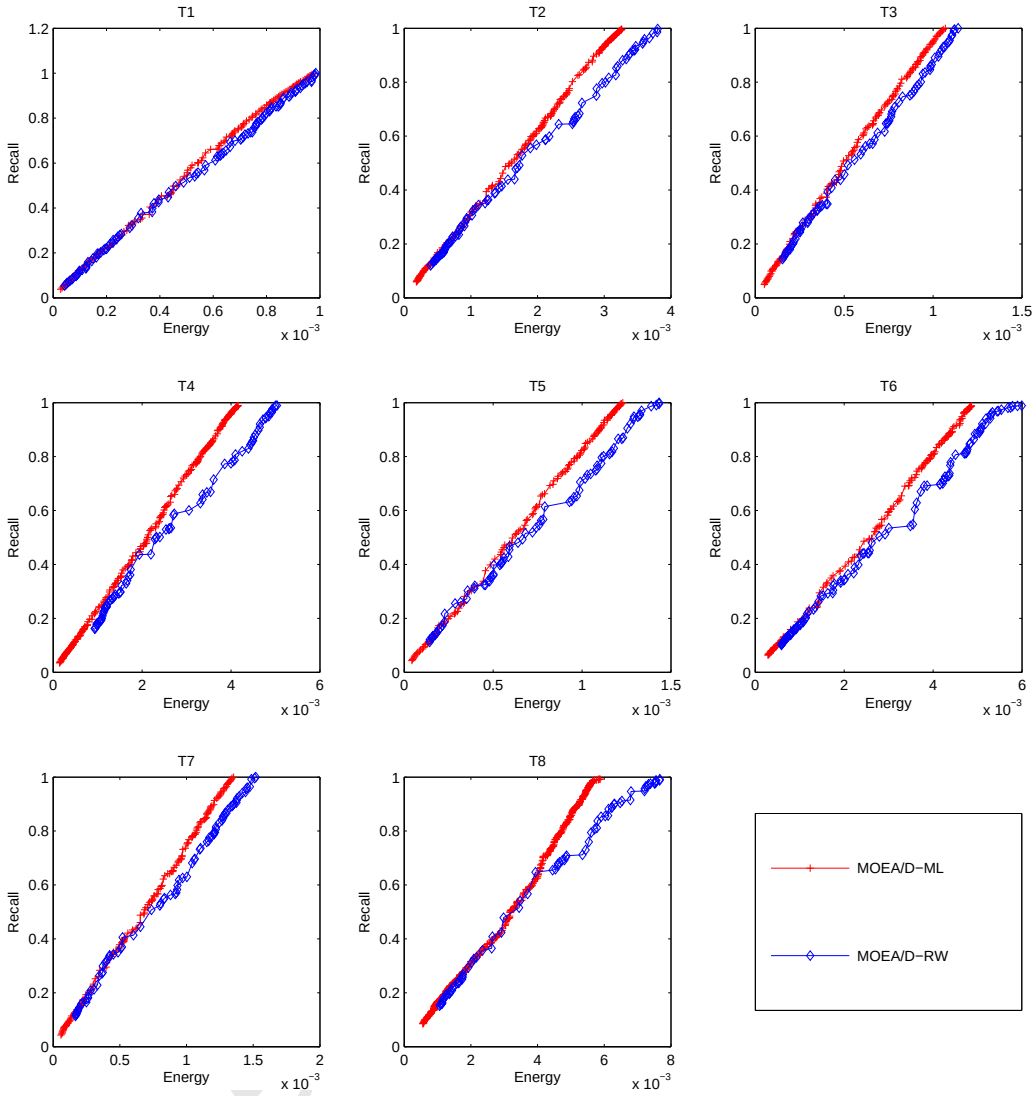


Figure 5: Experimental Series 3 (MO-MSNS) - Comparison of MOEA/D-ML (with the ϵ-greedy learning strategy) with MOEA/D-RW (that uniformly randomly selects at each hybridization step a LS heuristic from the pool of generalized LS heuristics)

gence, in all eight test instances. Here it is important to notice that MOEA/D-RW does not obtain Pareto-optimal solutions of low energy consumption when recall is low (i.e., bottom-left of plots) and obtains solutions of poor energy-consumption

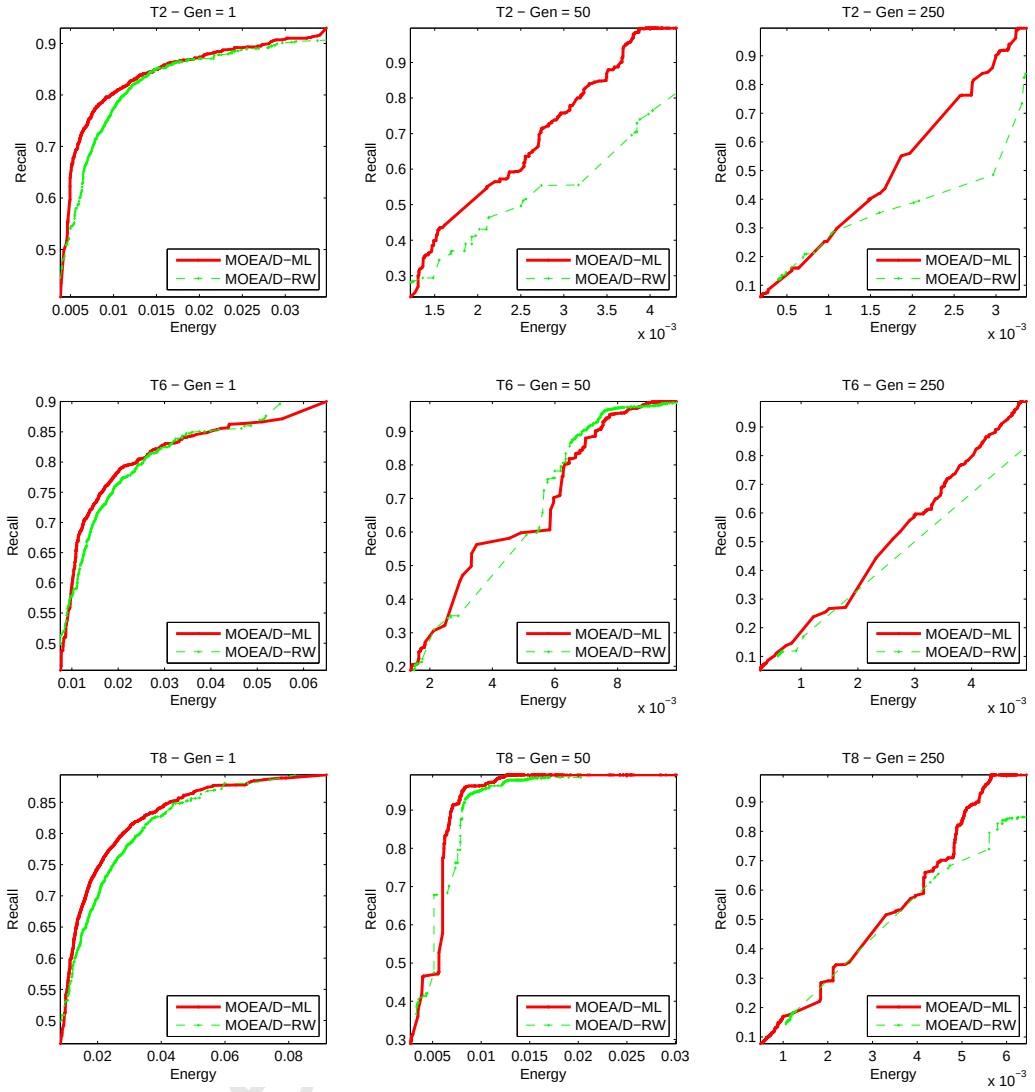


Figure 6: Experimental Series 3 (MO-MSNS) - The effect of Meta-Lamarckian Learning: Comparison of the proposed MOEA/D-ML (with the ϵ-greedy learning strategy) with MOEA/D-RW (that uniformly randomly selects at each hybridization step a LS heuristic from the pool of generalized LS heuristics) in terms of convergence of the internal population during the evolution.

when the recall is high (top-right of plots). This is due to the fact that random walk does not search the objective space efficiently and therefore cannot obtain solutions close to the extremes.

Finally, Figure 6 demonstrates the convergence of the internal populations (i.e., IP) of the two approaches compared in this section during the evolution. Note that for better visualization we randomly selected and presented the results of three test instances (i.e., T2, T6 and T8) for generations $gen = 1, 50, 250$. The results show that at the beginning, i.e., $gen = 1$, Figure 6 (left), both approaches start from a similar low-quality, low-diversity population. The proposed MOEA/D-ML approach, however, after few generations, i.e., $gen = 50$, Figure 6 (center), starts improving both the diversity and quality of the obtained solutions. This is due to the fact that Meta-Lamarckian learning approach requires some iterations to start learning and adapting to the needs of each test instance with respect to the local search selection. At the end, i.e., $gen = 250$, Figure 6 (right), MOEA/D-ML provides a better approximation towards the reference set compared to the MOEA/D-RW. Here it is important to notice that the better performance of the proposed approach is both in terms of convergence and diversity. The latter, for example for test instance T2 (top-row of Figure 6), improves approximately 10% in the first 50 generations and another 25% for the last 200 generations, where the MOEA/D-RW improves 10% in the first 50 generations and just 12% in the last 200 generations.

5.5.4. Experimental Series 4 (MO-MSNS) - Comparison between different Meta-Lamarckian Learning strategies

In this experimental series, we present a comparison between the decompositional MOEA with the three Meta-Lamarckian learning strategies introduced in Subsection 5.2, namely, MOEA/D-SR (Stochastic Roulette-wheel), MOEA/D-GN (Greedy Neighbourhood-based) and MOEA/D-ML (ϵ-greedy), in terms of the performance metrics introduced in Subsection 5.3.

The results of Table 8 show that the proposed MOEA/D-ML approach outperforms both MOEA/D with Meta-Lamarckian learning variants MOEA/D-SR and MOEA/D-GN in most cases. In particular, MOEA/D-ML provides a more diverse (i.e., high I_H) Pareto Front, that is closer to the reference set (i.e., low I_D) and with more Pareto-optimal solutions (i.e., high NDS) in six out of eight test instances. Moreover, the PF obtained by the proposed approach is of higher quality (i.e., high C) compared to the PF obtained by the MOEA/D-SR in six cases and the PF obtained by the MOEA/D-GN in all eight cases. Therefore, it is reasonable to argue that the ϵ-greedy collaboration between the two Meta-Lamarckian approaches in more effective than using them individually.

Table 8: Results of Experimental Series 4 (MO-MSNS) - The proposed MOEA/D-ML, i.e., MOEA/D with the proposed Meta-Lamarckian learning approach (ϵ-greedy) is compared with the MOEA/D-SR (Stochastic Roulette-wheel) and the MOEA/D-GN (Greedy Neighbourhood-based) approaches in terms of the performance metrics I_D, I_H and NDS (top) and coverage metric C (bottom). The best results of each test instance are denoted in bold.

Alg:	MOEA/D-ML		MOEA/D-SR		MOEA/D-GN		MOEA/D-ML	MOEA/D-SR	MOEA/D-GN
TI	I_D	I_H	I_D	I_H	I_D	I_H	NDS	NDS	NDS
1:	**0.0496**	**0.9430**	0.0498	0.8870	0.0656	0.7430	**150**	147	76
2:	0.0436	0.8940	**0.0404**	**0.9770**	0.0765	0.5620	219	**250**	75
3:	**0.0491**	0.9090	0.0519	**0.9280**	0.0734	0.7320	**160**	150	75
4:	**0.0373**	**0.9310**	0.0471	0.8500	0.0906	0.5120	**255**	213	54
5:	0.0537	**0.9260**	**0.0502**	0.9140	0.0574	0.7840	149	**167**	88
6:	**0.0485**	**0.9270**	0.0566	0.7820	0.0890	0.5810	**168**	101	63
7:	**0.0455**	**0.9260**	0.0565	0.8740	0.0951	0.4870	**171**	129	53
8:	**0.0288**	**0.8400**	0.0525	0.8170	0.1118	0.2630	**452**	139	35
mean:	**0.0445**	**0.912**	0.0506	0.8786	0.08242	0.583	**215.5**	162	64.875
std:	0.0079	0.03263	0.0052	0.06257	0.0175	0.1712	102.49	47.92	16.98

Test Inst.	C(MOEA/D-ML,MOEA/D-SR)	C(MOEA/D-SR,MOEA/D-ML)	C(MOEA/D-ML,MOEA/D-GN)	C(MOEA/D-GN,MOEA/D-ML)
1:	**0.4267**	0.3605	**0.7800**	0.0395
2:	0.2466	**0.5360**	**0.7626**	0.0000
3:	**0.6563**	0.1667	**0.8313**	0.0000
4:	**0.5569**	0.1268	**0.6745**	0.0000
5:	0.1879	**0.6707**	**0.7383**	0.1023
6:	**0.5774**	0.2178	**0.6786**	0.0000
7:	**0.3743**	0.3643	**0.6667**	0.0000
8:	**0.7458**	0.7410	**0.5398**	0.0000
mean:	**0.471**	0.397	**0.70**	0.01
std:	0.196	0.23	0.089	0.036

5.5.5. Experimental Series 5 (MO-MSNS) - Comparison between MOEA/D-ML and NSGA-II

In this experimental series, we compare the proposed MOEA/D with Meta-Lamarckian learning (ϵ-greedy strategy) and NSGA-II, the state-of-the-art on Pareto-dominance based approaches. We adopt both a statistical comparison in terms of the performance metrics introduced in Subsection 5.3 as well as a visual comparison in all test instances of Table 2.

Figure 7 visually shows the dominance of the proposed MOEA/D-ML approach with respect to NSGA-II in terms of both diversity and convergence, in all eight test instances. MOEA/D-ML provides a diverse set of non-dominated solutions that smoothly cover the objective space where at the same time its quality adequately approximates the extreme objective fitness values (i.e., energy = 0 and recall = 1). On the other hand, NSGA-II in the absence of Meta-Lamarckian learning and local search obtains a PF of both poor diversity and quality. In particular, NSGA-II finds difficulties in obtaining non-dominated solutions in the direction

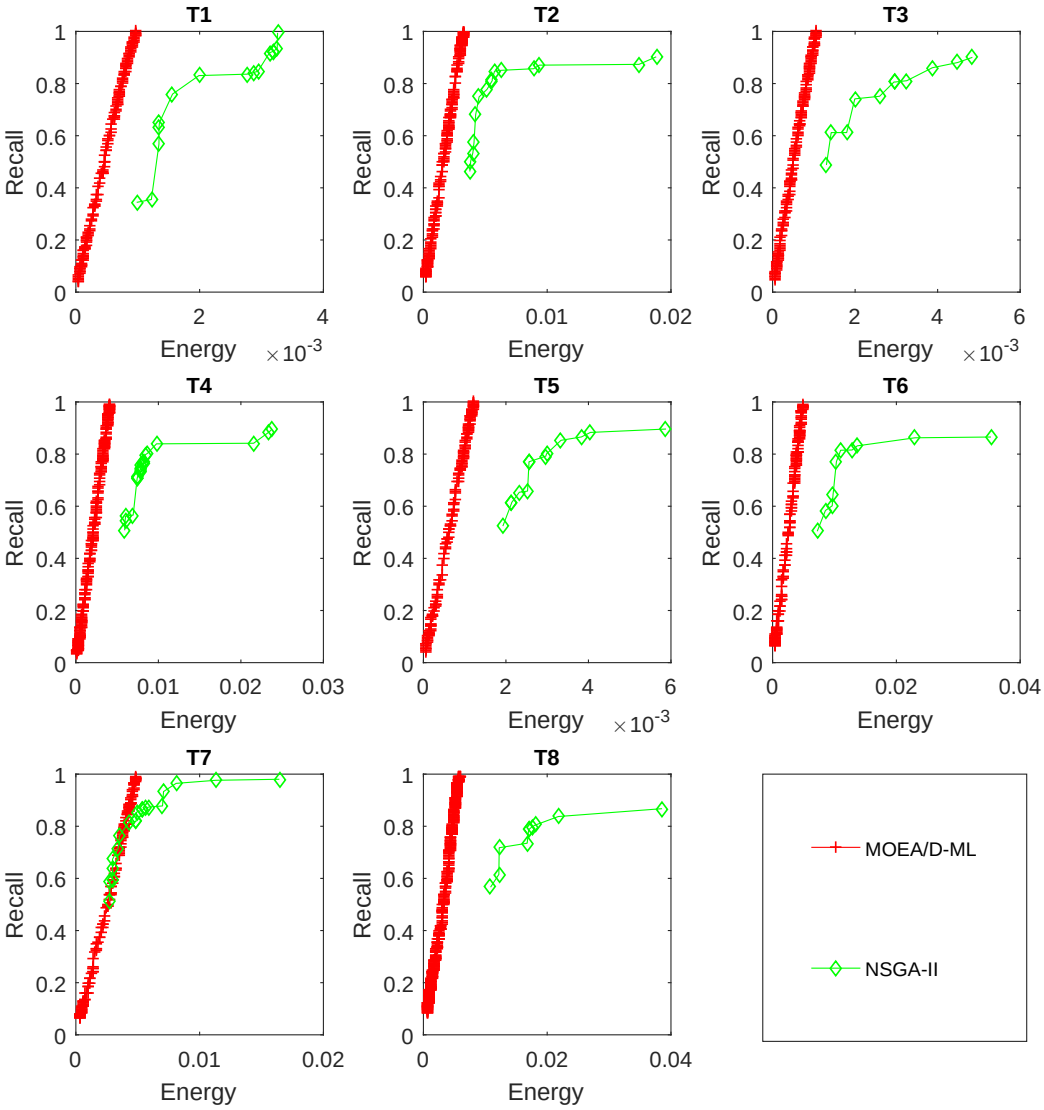


Figure 7: Experimental Series 5 (MO-MSNS) - Comparison[6]of MOEA/D-ML (with the ϵ-greedy learning strategy) with NSGA-II in all test instances T1-T8 of Table 2.

of the more demanding energy objective.

The results of Table 9 support the previous observations since MOEA/D-ML

[6]Note that a common legend for all sub-figures appears in the bottom-right box.

Table 9: Results of Experimental Series 5 (MO-MSNS) - The proposed MOEA/D-ML is compared with NSGA-II in terms of the performance metrics I_D, I_H, NDS and C. The best results of each test instance are denoted in bold.

Alg:	MOEA/D-ML		NSGA-II		MOEA/D-ML	NSGA-II	C(MOEA/D-ML,NSGA-II)	C(NSGA-II, MOEA/D-ML)
TI	I_D	I_H	I_D	I_H	NDS	NDS		
1:	**0.05**	**0.95**	0.19	0.45	**150**	15	**0.64**	0.00
2:	**0.04**	**0.94**	0.18	0.43	**219**	15	**0.61**	0.00
3:	**0.05**	**0.93**	0.21	0.40	**160**	11	**0.56**	0.00
4:	**0.04**	**0.96**	0.16	0.38	**255**	20	**0.48**	0.00
5:	**0.05**	**0.95**	0.20	0.32	**149**	13	**0.58**	0.00
6:	**0.04**	**0.93**	0.21	0.40	**168**	10	**0.56**	0.00
7:	**0.04**	**0.92**	0.17	0.41	**171**	20	**0.31**	0.30
8:	**0.03**	**0.91**	0.21	0.30	**452**	10	**0.46**	0.00
mean:	**0.05**	**0.98**	0.21	0.39	**215.5**	13.70	**0.54**	0.04
std:	0.0027	0.0135	0.03	0.10	102.49	4.06	0.1046	0.00

provides better results in all four performance metrics with respect to NSGA-II. In particular, MOEA/D-ML provides about four and three times better performance with respect to the I_D and I_H metrics, respectively. It also provides around 200 more non-dominated solutions, on average, and its non-dominated solutions dominate 54% of the non-dominated solutions obtained by NSGA-II. MOEA/D-ML also provides a low standard deviation in most cases indicating a consistency on its performance along different test instances in the same MOP.

Table 10: Results of Experimental Series 6 (MO-MSNS) - Statistical analysis on the best results obtained by MOEA/D-ML (M) and NSGA-II (N) in terms of mean and standard deviation on the performance metrics I_D, I_H, NDS and C in all eight test instances (T1-T8). Student t-test is performed to evaluate the significance of the final results. Note that h="+" indicates rejection of the null hypothesis that the two sets are not significantly different with a significance level $\alpha = 0.05$, where h= "-" indicates the alternative.

Metric:	$I_D(M)$	$I_D(N)$	$I_H(M)$	$I_H(N)$	$NDS(M)$	$NDS(N)$	$C(M,N)$	$C(N,M)$
mean:	0.05	0.21	0.98	0.39	215.5	13.70	0.54	0.00
std:	0.0027	0.03	0.0135	0.1	102.49	4.06	0.1046	0.04
t-test (h):	+		+		+		+	

5.5.6. Experimental Series 6 (MO-MSNS) - Statistical Analysis

In this experimental study, we perform statistical analysis on the results obtained by MOEA/D-ML and NSGA-II on 20 runs x eight (8) test instances. In particular, the two approaches are compared with respect to their mean and standard deviation values, as well as by using statistical hypothesis tests, i.e., the *Student's sample t-test* and the *one-way ANOVA*. The t-test is carried out for comparing the results between the two approaches and the one-way ANOVA is carried out when

the results of multiple runs of each algorithm is compared. Each test returns an h value on the null hypothesis that the average results are not significantly different against the alternative that the average results are significantly different. The h = "+" indicates a rejection on the null hypothesis and h = "-" indicates a failure to reject the null hypothesis with a given significance level α.

Table 11: Results of Experimental Series 6 (MO-MSNS) - Statistical analysis on the results obtained by MOEA/D-ML (M) and NSGA-II (N) in all 20 runs in terms of average and standard deviation on the performance metrics I_D, I_H, NDS and C. Moreover a one-way ANOVA is performed in order to evaluate the robustness of the two approaches. Note that h="+" indicates rejection of the null hypothesis that the sets obtained during the 20 runs for each approach are not significantly different with a significance level $\alpha = 0.05$, where h="-" indicates the alternative.

Metric:		$I_D(M)$	$I_D(N)$	$I_H(M)$	$I_H(N)$	$NDS(M)$	$NDS(N)$	$C(M,N)$	$C(M,N)$
T1	mean:	0.05	0.21	0.98	0.39	157.30	13.70	0.54	0.00
	std:	0.0027	0.03	0.0135	0.10	11.2354	4.06	0.1046	0.00
T2	mean:	0.05	0.18	0.98	0.44	159.20	16.70	0.61	0.00
	std:	0.0027	0.03	0.0135	0.10	11.2354	4.06	0.1046	0.00
T3	mean:	0.05	0.20	0.94	0.40	147.30	13.80	0.57	0.00
	std:	0.0027	0.03	0.0135	0.10	11.2354	4.06	0.1046	0.00
T4	mean:	0.05	0.18	0.95	0.36	157.90	16.20	0.54	0.00
	std:	0.0027	0.03	0.0135	0.10	11.2354	4.06	0.1046	0.00
T5	mean:	0.04	0.20	0.97	0.35	191.00	15.20	0.49	0.00
	std:	0.0027	0.03	0.0135	0.10	11.2354	4.06	0.1046	0.00
T6	mean:	0.04	0.20	0.97	0.37	184.50	15.90	0.52	0.00
	std:	0.0027	0.03	0.0135	0.10	11.2354	4.06	0.1046	0.00
T7	mean:	0.04	0.19	0.96	0.36	175.50	15.20	0.46	0.03
	std:	0.0027	0.03	0.0135	0.10	11.2354	4.06	0.1046	0.00
T8	mean:	0.04	0.19	0.96	0.34	206.30	17.50	0.46	0.00
	std:	0.0027	0.03	0.0135	0.10	11.2354	4.06	0.1046	0.00
T1-8	mean:	0.05	0.19	0.96	0.38	172.38	15.53	0.53	0.00
	std:	0.0018	0.01	0.01	0.02	25.90	1.54	0.03	0.03
t-test (h)		+		+		+		+	
Anova	p:	0.4240	0.8180	0.1597	0.7773	0.2836	0.3690	0.3782	0.4484
Anova	h:	-	-	-	-	-	-	-	-

Table 10 shows a t-test statistical analysis on the best results of each MOEA approach for each performance metric. This test indicates a rejection ($h = +$) of the null hypothesis that the results obtained between the two approaches are not significantly different. This supports our previous observations that MOEA/D-ML significantly outperforms NSGA-II.

Table 11 shows a statistical analysis of the two approaches over 20 independent runs on all eight test instances. The top part of the table indicates the mean and standard deviation results for each test instance (T_i) over 20 runs, the middle

part shows the mean and standard deviation of all 20x8=160 test instances (T1-T8) of each performance metric for both the MOEA/D-ML and NSGA-II. The results clearly demonstrate the superiority of MOEA/D-ML in all test instances individually as well as in all runs together with respect to all performance metrics, since it provides better mean values and better standard deviation in almost all cases. The bottom part of the table shows the results of the two statistical hypothesis tests with a significance level $\alpha = 0.05$. The t-test indicates a rejection ($h = +$) of the null hypothesis that the results obtained between the two approaches are not significantly different in all eight test instances over all 20 runs. The ANOVA test demonstrates the robustness of the proposed approach along the 20 independent runs, since the results show a failure to reject ($h = +$) the null hypothesis that the results are not significantly different ($p > \alpha$). This indicates that the proposed MOEA/D-ML approach, as well as NSGA-II, consistently provide similar performance over a number of independent runs.

5.5.7. *Experimental Series 7 (MO-MSNS) - Sensitivity Analysis on step size of Local Search*

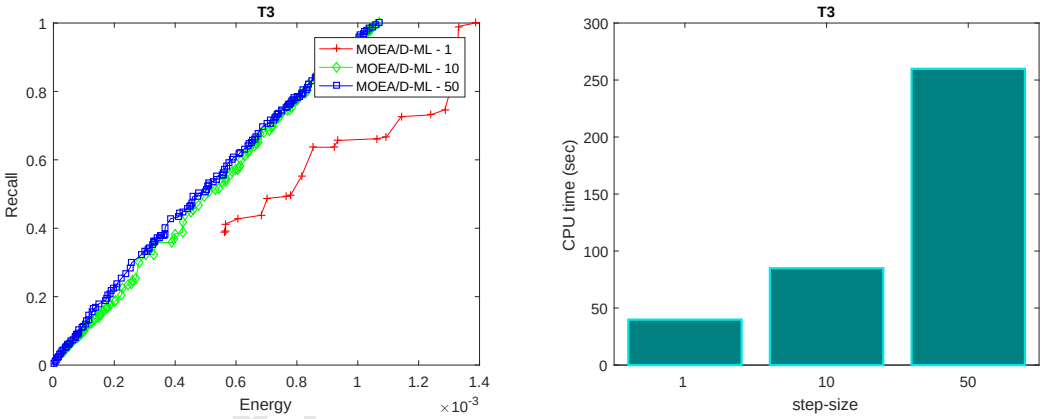


Figure 8: Experimental Series 7 (MO-MSNS) - Sensitivity Analysis: The effect of step size (iterations) of local search heuristics on MOEA/D-ML approach on test instance T3.

In this last experimental series, it is examined how the local search step-size (i.e., the number of iterations each time a local search heuristic is selected) affects the performance of the proposed MOEA/D-ML approach with respect to the quality and diversity of the obtained Pareto-Front as well as the required CPU time (in seconds) required to obtain that particular non-dominated set of solutions over a

fixed termination criterion (i.e., fixed maximum number of generations). Figure 8 clearly shows a trade-off between the quality and diversity of the obtained PF and the required CPU time. The increase of the local search step-size results in a non-linear improvement on the quality and diversity of the PF meaning that the more iterations allowed for the local search the better the results are, but the improvement decreases as the step-size increases. On the other hand, the increase on the step-size cause a significant increase of the CPU time. Therefore, an average step-size=10 is considered as a good choice providing both good quality and diversity in the PF and a reasonable CPU time, at the same time.

5.6. Generalizability: Permutation Flow Shop Scheduling Problem (PFSSP)

In this subsection, we evaluate the performance of the proposed MOEA/D-ML approach on the multi-objective Permutation Flow Shop Scheduling Problem (PPFSP) with respect to the popular MOGLS approach proposed in [7] in order to validate its generalizability over a well-known combinatorial MOP.

Given a permutation of n jobs and a series of m machines, and for $i = 1, ..., n$ and $k = 1, ..., m$, the processing times $P(i, k)$ of job i on machine k and the due dates d_i of job i, PFSSP can be formulated as follows: Each of these n jobs has to be processed sequentially from the first machine to the last, in the same order. In other words, sequence changes are not allowed, so once the sequence of jobs is scheduled on the first machine, this sequence remains unchanged on the other machines. After completion on one machine a job joins the queue at the next machine, all queues are assumed to operate under the FIFO discipline. Each machine can process at most one job at any given time, and it can not be interrupted. Machines never breakdown and are available throughout the scheduling period. Each job is available at time zero, and can be processed by at most one machine in any given time. The set-up times of the jobs on machines are sequence independent and are included in processing times.

The aim is to determine a permutation $\pi = (\pi_1, \pi_2, ..., \pi_n)$, i.e., a processing order of the jobs on each machine, which minimizes the following two objectives: i) makespan $C_{max}(\pi)$, and ii) maximum tardiness $T_{max}(\pi)$.

Let $C(\pi_i; k)$ denote the completion time of job π_i on machine k. Then the completion times for this permutation can be recursively calculated as follows:
$C(\pi_1, 1) = P(\pi_1, 1)$
$C(\pi_i, 1) = C(\pi_{i-1}, 1) + P(\pi_i, 1), \ \ (2 \leq i \leq n)$;
$C(\pi_1, k) = C(\pi_1, k-1) + P(\pi_1, k), \ \ (2 \leq k \leq m)$;
$C(\pi_i, k) = max\{C(\pi_{i-1}, k), C(\pi_i, k-1)\} + P(\pi_i, k), \ \ (2 \leq i \leq n, \ 2 \leq k \leq m)$.

The makespan (total completion time) is the time required to complete the last job of permutation π on the last machine m, defined as $C_{max}(\pi) = C(\pi_n, m)$.

The lateness of job π_i is defined as $L_{\pi_i} = C(\pi_i, m) - d_{\pi_i}$ and the tardiness of job π_i is defined as $T_{\pi_i} = max\{L_{\pi_i}, 0\}$. The tardiness of permutation π is defined as $T_{max}(\pi) = max_{1 \leq i \leq n}\{T_{\pi_i}\}$.

5.6.1. Experimental Setup and Algorithms for PFSSP

We evaluate the performance of MOEA/D-ML on eight benchmark test instances of m -machine, n-job permutation flow shop scheduling problems as summarized in Table 12.

Table 12: Test Instances (PFSSP) - T1-T4 refer to the test problems from Ishibuchi et. al. [7] and T5-T8 refer to the test problems from Bin-Bin Li et. al. [14].

Test Instance	# of jobs	# of machines	α	β
T1	20	20	-	-
T2	40	20	-	-
T3	60	20	-	-
T4	80	20	-	-
T5	20	10	0.2	0.6
T6	20	10	0.2	1.2
T7	20	10	0.4	0.6
T8	20	10	0.4	1.2

The first four test instances (i.e., T1-T4) were initially defined in [7] as follows: the processing time of each job on each machine was specified as a random integer in the interval $[1, 99]$. The due date of each job was specified by adding a random integer in the interval $[-100, 100]$ to its actual completion time in a randomly generated schedule.

The next four test instances (i.e., T5-T8) were initially defined in [14] as follows: the processing time of each job in every machine is uniformly distributed in interval $[1, 99]$, whereas the due date of each job is uniformly distributed in interval $[Q(1 - a - (b/2)), Q(1 - a + (b/2))]$, where a and b represent the tardiness factor of jobs and the dispersion range of due dates, respectively, and Q is a lower bound of makespan estimated as:

$$Q = max\left\{ \max_i \sum_{k=1}^{m} p_{i,k}, \max_{1\leq k\leq m} \left\{ \sum_{i=1}^{n} p_{i,k} + \min_i \sum_{l=1}^{k-1} p_{i,l} + \min_i \sum_{l=k+1}^{m} p_{i,l} \right\} \right\}.$$

Then the following four scenarios about due dates are considered as in [14], where each scenario is determined by a different combination of the values of a and b. In general, the due dates are more restrictive when a increases, whereas the due dates are more diversified when b increases.

- T5: Scenario 1) low tardiness factor ($a = 0.2$) and small due date range ($b = 0.6$);
- T6: Scenario 2) low tardiness factor ($a = 0.2$) and wide due date range ($b = 1.2$);
- T7: Scenario 3) high tardiness factor ($a = 0.4$) and small due date range ($b = 0.6$);
- T8: Scenario 4) high tardiness factor ($a = 0.4$) and wide due date range ($b = 1.2$);

We have compared the proposed MOEA/D-ML against (i) the popular MOGLS that also does not utilize any problem-specific heuristics (e.g., NEH [14]), and (ii) all MOEA/D variants summarized in Table 3, except the MOEA/D-CH, since the Copy Heuristic (CH) causes infeasible solutions in the PFSSP; instead MOEA/D-SwA (Swap Adjacent) is used. The algorithmic parameters in the following experimental studies are set as follows: termination criterion gen^m=1000, population size and number of subproblems N=500, crossover rate r_c=0.8, mutation rate r_m=0.2, neighbourhood size T=14, the size of the pool of local search heuristics $L = 6$, the number of iterations for each local search is set to $I = 10$ and the training phase for the MOEA/D with Meta-Lamarckian Learning approaches is set to $g^t = 100$. For the MOEA/D-ML that utilizes the proposed ϵ-greedy strategy, the ϵ is set to 0.8. Note that in our experimental studies we have used the same number of function evaluations for all methods, for fairness, and each algorithm is executed 20 times in each study.

5.6.2. Experimental Series 1 (PFSSP) - MOEA/D-ML vs. MOGLS

In experimental series 1, we compare the performance of the proposed MOEA/D-ML approach against the MOGLS in all eight test instances (i.e, T1-T4 from [7] and T5-T8 from [14]) of Table 12 in terms of all performance metrics introduced in Subsection 5.3.

Figure 9 shows the superiority of the proposed approach when also applied to this well-known multi-objective combinatorial problem. In particular, MOEA/D-ML provides better quality and diversity than MOGLS in all eight test distances. The superiority of MOEA/D-ML increases as the complexity of the test instances

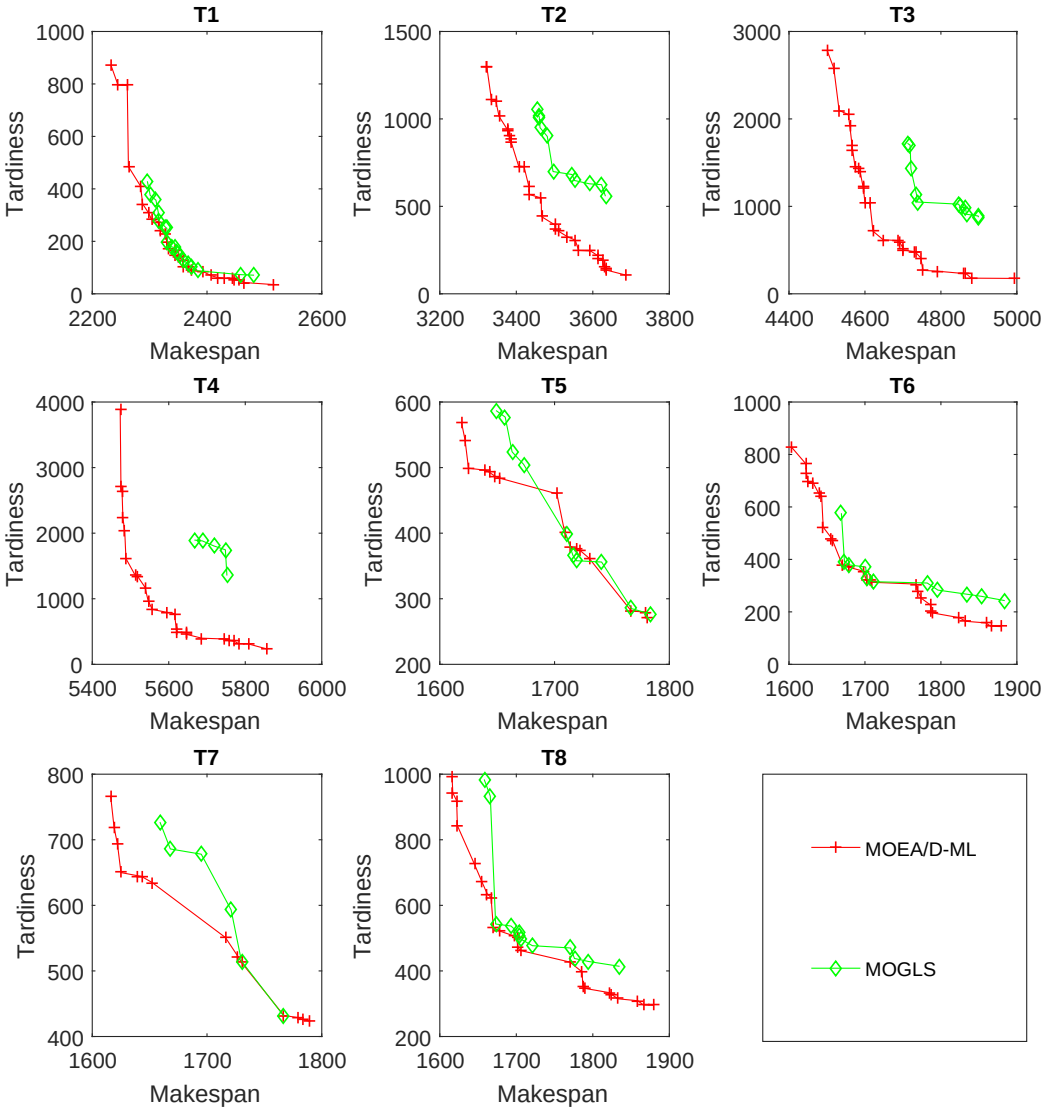


Figure 9: Experimental Series 1 (PFSSP) - Comparison of MOEA/D-ML (with the ϵ-greedy learning strategy) with MOGLS in both the test instances T1-T4 from [7] and T5-T8 from [14].

increases. That is the performance of MOGLS, for example, in test instances T1 (the first test instance of [7] with the lowest number of jobs) and T5 (the first test instance of [14] with the lowest values of parameters α and β) is relatively comparable to, but slightly worse than, MOEA/D-ML. This is more evident in the

Table 13: Results of Experimental Series 1 (PFSSP) - The proposed MOEA/D-ML is compared with MOGLS in terms of the performance metrics I_D, I_H, NDS and C. Note that T1-T4 refer to the test problems OF Ishibuchi et. al. [7] and T5-T8 refer to the test problems OF Bin-Bin Li et. al. [14]. The best results of each test instance are denoted in bold.

Alg:	MOEA/D-ML		MOGLS		MOEA/D-ML	MOGLS	C(MOEA/D-ML,MOGLS)	C(MOEA/D-ML,MOGLS)
TI	I_D	I_H	I_D	I_H	NDS	NDS		
1:	62.14	**0.09**	**56.15**	0.07	**29**	18	**0.72**	0.00
2:	**115.03**	**0.06**	224.65	0.03	**30**	11	**0.80**	0.00
3:	**213.18**	**0.08**	322.82	0.04	**29**	11	**0.79**	0.00
4:	**264.99**	**0.06**	687.43	0.02	**24**	5	**0.58**	0.00
5:	**49.54**	**0.03**	66.93	0.02	**16**	10	**0.56**	0.20
6:	**69.76**	**0.08**	82.01	0.05	**27**	11	**0.70**	0.00
7:	**57.74**	**0.03**	94.52	0.02	**14**	6	**0.57**	0.17
8:	**71.17**	**0.08**	99.65	0.06	**24**	12	**0.79**	0.00
mean:	**112.94**	**0.063**	204.27	0.038	**24.12**	10.5	**0.688**	0.046
std:	81.44	0.023	216.02	0.019	6.08	3.96	0.104	0.086

statistical results summarized in Table 13.

Table 13 shows that MOEA/D-ML outperforms MOGLS in almost all test instances with respect to all performance metrics. In particular, MOEA/D-ML provides two times better performance in both the I_D and I_H metrics. It also provides about two times more non-dominated solutions and its non-dominated solutions dominate 68% of the non-dominated solutions obtained by MOGLS, on average. MOEA/D-ML also provides a relatively low standard deviation in all cases indicating a consistency at its performance along different test instances in the well-known PFSSP.

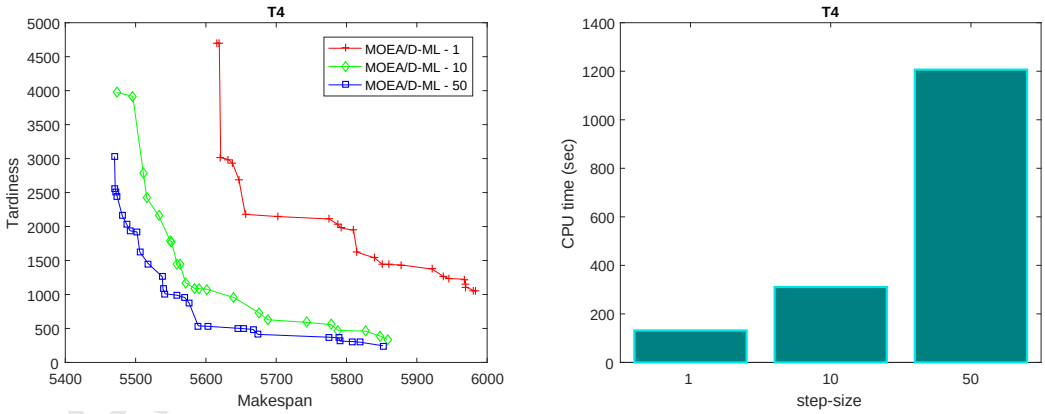


Figure 10: Experimental Series 2 (PFSSP) - Sensitivity Analysis: The effect of step size (iterations) of local search heuristics on MOEA/D-ML approach on test instance T4.

5.6.3. *Experimental Series 2 (PFSSP) - Sensitivity Analysis on step size of Local Search*

Similarly to experimental series 7 (Subsection 5.5.7) of the MO-MSNS problem, this experimental series examines how the local search step-size (i.e., the number of iterations each time a local search heuristic is selected) affects the performance of the proposed MOEA/D-ML approach with respect to the quality and diversity of the obtained Pareto-Front as well as the required CPU time (in seconds) required to obtain that particular non-dominated set of solutions over a fixed termination criterion (i.e., fixed maximum number of generations). Figure 10 clearly shows the trade-off between the quality and diversity of the obtained PF and the required CPU time. Once again, the increase of the local search step-size results in a non-linear improvement on the quality and diversity of the PF in the sake of a significant increase on the CPU time.

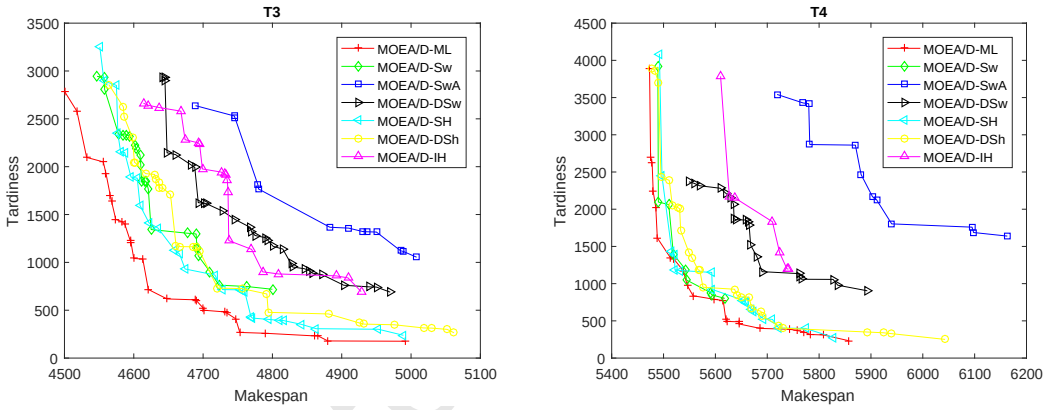


Figure 11: Experimental Series 3 (PFSSP) - Effect of Meta-Lamarckian Learning: Adaptiveness of MOEA/D-ML and comparison against all Individualistic MOEA/D variants in test instances T3 and T4.

5.6.4. *Experimental Series 3 (PFSSP) - Effect of Meta-Lamarckian Learning*

One of the major contributions of this research study is the demonstration of the adaptiveness of the proposed MOEA/D-ML in learning the effectiveness of each LS from a pool of generalized LS methods, online, and selecting the best performing LS for each objective function of each problem instance of each class of problems, during the evolution. Therefore, in this experimental series, we discuss the effect of the proposed Meta-Lamarckian learning approach (ϵ-greedy strategy) on the MOEA/D when applied to the PFSSP.

Table 14: Results of Experimental Series 3 (PFSSP) - Comparison between the MOEA/D-ML and the six Individualistic MOEA/D variants in terms of the performance metrics I_D and I_H. The best overall results of each test instance are denoted in bold and the best results among the individualistic approaches are underlined.

Alg:	M-ML		M-Sw		M-SwA		M-DSw		M-Sh		M-DSh		M-IH	
TI	I_D	I_H	I_D	I_H	I_D	I_H	I_D	I_H	I_D	I_H	I_D	I_H	I_D	I_H
1:	62.14	0.09	71.34	0.09	95.51	0.07	83.46	0.07	65.01	0.09	**61.93**	**0.10**	88.80	0.07
2:	115.03	0.06	171.28	0.09	210.14	0.05	170.15	0.07	140.06	**0.10**	**103.11**	0.07	219.28	0.08
3:	**213.18**	**0.08**	348.21	0.06	445.13	0.04	271.50	0.05	260.37	0.07	253.63	0.07	387.87	0.06
4:	**264.99**	0.06	549.95	0.07	691.76	0.03	338.00	0.07	292.09	**0.10**	306.25	0.08	731.31	0.06
5:	**49.54**	0.03	73.48	0.03	63.00	0.03	66.48	0.02	60.74	0.03	60.78	**0.04**	64.66	0.02
6:	**69.76**	0.08	70.11	**0.09**	105.37	0.08	78.82	0.07	69.19	0.08	61.18	**0.09**	78.00	0.06
7:	**57.74**	0.03	59.75	0.03	87.51	0.02	65.46	**0.04**	73.05	0.02	75.67	**0.04**	83.28	0.03
8:	**71.17**	0.08	65.25	0.07	109.70	0.06	69.32	0.06	71.29	**0.09**	67.28	**0.09**	73.22	0.07
mean:	**112.94**	0.063	176.17	0.0662	226.015	0.047	142.89	0.056	128.97	**0.072**	123.728	**0.072**	215.8	0.056
std:	81.44	0.023	180.46	0.02	225.53	0.02	107.06	0.018	94.69	0.03	98.41	0.02	236.40	0.02

Figure 11 shows a visual comparison between the proposed MOEA/D-ML with all individualistic MOEA/D variants in the most representative test instances T3 and T4. Statistical comparison between all approaches in all test instances follows. The results show that the best performing Individualistic approach is the MOEA/D-DSh, which obtained better quality and diversity than all the Individualistic approaches, overall. The makespan objective, however, shows a slight preference on the MOEA/D-Sw approach since it provides the best objective value in both test instances, where it provides a very poor performance on the other objective. The MOEA/D-Sh, on the other hand, provides comparable results to (but slightly worse than) the other two approaches in both objective functions and all test instances. The MOEA/D-ML approach adaptively learns the performance of each Individualistic LS for each objective function and selects the best for each case. Consequently, the proposed approach clearly outperforms all Individualistic MOEA/D variants in terms of both diversity and quality of the obtained PF.

The above observations are also supported by the statistical results summarized in Tables 14-16. The results show a comparison between the proposed MOEA/D-ML and all Individualistic MOEA/D variants in all eight test instances with respect to performance metrics I_D, I_H and NDS as well as a comparison between the MOEA/D-ML and the two best performing Individualistic approaches with respect to the C-metric. MOEA/D-ML performs better in six out of eight test instance with respect to the I_D metric, where the best performance in terms the I_H and NDS metrics varies between different MOEA/D variants. In terms of the C-metric MOEA/D-ML dominates 59% of the PF obtained by MOEA/D-Sh and 56% of the PF obtained by MOEA/D-DSh, on average in all test instances, where

Table 15: Results of Experimental Series 3 (PFSSP) - Comparison between the MOEA/D-ML and the six Individualistic MOEA/D variants in terms of NDS. The best overall results of each test instance are denoted in bold and the best results among the individualistic approaches are underlined.

TI	M-ML	M-Sw	M-SwA	M-DSw	M-Sh	M-DSh	M-IH
1:	29	29	16	21	**32**	30	24
2:	30	26	11	27	23	**33**	17
3:	29	25	13	29	28	**30**	20
4:	24	9	12	21	18	**27**	7
5:	**16**	8	13	10	12	11	9
6:	27	22	13	22	25	**29**	21
7:	**14**	12	7	10	8	11	7
8:	24	22	15	26	25	**27**	17
mean:	24.13	19.13	12.50	20.75	21.38	**24.75**	15.25
std:	6.08	8.22	2.73	7.25	8.14	8.70	6.69

only 1% of the MOEA/D-ML's PF is dominated by MOEA/D-Sh and only 8% of its PF is dominated by MOEA/D-DSh, on average.

Finally, Table 17 shows a comparison between the proposed MOEA/D with Meta-Lamarckian learning (ϵ-greedy strategy), the conventional MOEA/D [13] and the MOEA/D with Random Walk (MOEA/D-RW) that uniformly randomly selects a local search heuristic from the pool of local search heuristics, in terms of the performance metrics introduced in Subsection 5.3 in all test instances of Table 12. The results demonstrate the effect of the Meta-Lamarckian learning in improving the performance of the conventional MOEA/D and in adaptively

Table 16: Results of Experimental Series 3 (PFSSP) - Comparison between the MOEA/D-ML and the best performing Individualistic MOEA/D variants in terms of the C-metric. The best results of each test instance are denoted in bold.

Test Inst.	C(MOEA/D-ML,MOEA/D-Sh)	C(MOEA/D-Sh,MOEA/D-ML)	C(MOEA/D-ML,MOEA/D-DSh)	C(MOEA/D-DSh,MOEA/D-ML)
1:	**0.55**	0.00	**0.31**	0.10
2:	**0.93**	0.00	**0.93**	0.00
3:	**0.97**	0.00	**1.00**	0.00
4:	**0.79**	0.00	**0.83**	0.00
5:	**0.50**	0.00	**0.25**	0.09
6:	**0.37**	0.00	**0.44**	0.07
7:	**0.29**	0.00	**0.29**	0.27
8:	**0.33**	0.04	**0.46**	0.07
mean:	**0.59**	0.01	**0.56**	0.08
std:	0.27	0.01	0.31	0.09

Table 17: Results of Experimental Series 3 (PFSSP) - The proposed MOEA/D-ML is compared with the conventional MOEA/D and the MOEA/D-RW approaches in terms of the performance metrics I_D, I_H and NDS (top) and coverage metric C (bottom). The best results of each test instance are denoted in bold.

Alg:	MOEA/D-ML		MOEA/D		MOEA/D-RW		MOEA/D-ML	MOEA/D	MOEA/D-RW
TI	I_D	I_H	I_D	I_H	I_D	I_H	NDS	NDS	NDS
1:	**62.28**	0.07	79.66	0.07	63.04	**0.08**	**29**	19	**29**
2:	115.03	**0.10**	266.11	0.03	**104.09**	**0.10**	30	16	**34**
3:	213.18	**0.16**	432.96	0.04	**206.09**	0.13	29	17	**30**
4:	**265.21**	**0.12**	756.72	0.04	392.10	0.11	**24**	13	22
5:	**49.92**	**0.03**	78.40	0.01	57.49	**0.03**	**16**	8	14
6:	69.76	**0.10**	111.65	0.06	**67.83**	0.08	**27**	17	**27**
7:	**57.74**	**0.03**	71.90	0.03	66.28	**0.03**	**14**	13	10
8:	**71.17**	**0.08**	78.82	0.08	72.25	**0.08**	**24**	17	**24**
mean:	**113.04**	**0.09**	234.53	0.05	128.65	0.08	**24.13**	15	23.75
std:	81.44	0.04	247.04	0.02	117.20	0.04	6.08	3.51	8.19

Test Inst.	C(MOEA/D-ML,MOEA/D)	C(MOEA/D,MOEA/D-ML)	C(MOEA/D-ML,MOEA/D-RW)	C(MOEA/D-RW,MOEA/D-ML)
1:	**0.83**	0.00	0.14	**0.17**
2:	**1.00**	0.00	**0.60**	0.24
3:	**1.00**	0.00	**0.59**	0.13
4:	**1.00**	0.00	**0.67**	0.09
5:	**0.63**	0.00	**0.19**	0.14
6:	**0.85**	0.00	**0.48**	0.04
7:	**1.00**	0.00	**0.14**	0.10
8:	**0.88**	0.00	**0.42**	0.04
mean:	**0.90**	0.00	**0.40**	0.12
std:	0.13	0.00	0.22	0.07

selecting the best performing local search heuristic, rather than randomly selecting any local search heuristic from a pool of generalized local searches.

In particular, the MOEA/D-ML approach clearly outperforms the conventional MOEA/D approach in all test instances with respect to all performance metrics. Here it is important to notice that no non-dominated solution obtained by MOEA/D dominates any solution obtained by MOEA/D-ML, in any test instance. MOEA/D-ML provides better performance than MOEA/D-RW in five test instances with respect to I_D metric and in seven out of total eight test instances, with respect to I_H, NDS and C-metric.

6. Conclusions and Future Work

In this paper, we deal with a realistic Multi-objective Optimization Mobile Social Network Search (MO-MSNS) problem: given a query by a smartphone user in a MSN optimize the search operation by minimizing the total energy consumption and maximizing the recall rate.

Our proposed algorithm, namely MOEA/D-ML, follows the general frame-

work of MOEA/D, combined with a Meta-Lamarckian approach that learns from the problem's properties and objective functions. A strategy promoting both co-operation and competition was devised for adaptively selecting the best performing local search heuristic for each objective function of each problem instance of each class of problems, from a pool of general-purpose local search heuristics, so as to locally optimize the solutions during the evolution. To the best of our knowledge, this is the first such hybridization of a decompositional MOEA and a Meta-Lamarckian learning strategy, in the literature.

We evaluate our algorithm on mobility and social behaviour patterns derived from the real data of GeoLife and DBLP datasets and a trace-driven experimental methodology. Extensive experimental studies investigate the adaptiveness and performance of the proposed approach. The experimental results initially reveal that individualistic local search heuristics exhibit bias in different test instances of the MO-MSNS problem as well as in different areas of the objective space of the same test instance. It is then shown that the proposed MOEA/D-ML successfully learns from this behaviour during the evolution and adaptively follows the pattern of best performing local search heuristics at different areas of the objective space. As a result of this learning strategy, MOEA/D-ML selects the best non-dominated solutions and at the end provides a more diverse and high quality set of Pareto-optimal solutions compared to its competitors. It is also evident from the experimental results that the greedy collaboration between the Greedy Neighbourhood-based and Stochastic Roulette-wheel Meta-Lamarckian approaches is more effective than using them separately. The generalizability of the proposed MOEA/D-ML approach is finally evaluated on the well-known multi-objective Permutation Flow Shop Scheduling Problem (PFSSP) on various benchmark test instances and the results reinforce the aforementioned findings.

In the future, we aim at investigating our propositions in the context of many-objective optimization as well as applying our MOEA/D-ML approach in other real-life problems. Further research future directions may also include the hybridization of the proposed approach with problem-specific local search heuristics as well as heuristics suitable for searching continuous objective spaces for investigating the application of the MOEA/D-ML on continuous MOPs.

References

[1] G. H. Costa, F. Baldo, Generation of road maps from trajectories collected with smartphone–a method based on genetic algorithm, Applied Soft Computing 37 (2015) 799–808.

[2] M. Xiao, J. Wu, L. Huang, Community-aware opportunistic routing in mobile social networks, IEEE Trans. on Computers 63 (7) (2014) 1682–1695.

[3] P. Valencia, A. Haak, A. Cotillon, R. Jurdak, Genetic programming for smart phone personalisation, Applied Soft Computing 25 (2014) 86–96.

[4] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, Wiley and Sons, 2002.

[5] C. A. C. Coello, D. A. V. Veldhuizen, G. B. Lamont, Evolutionary Algorithms for Solving Multi-Objective Problems, Vol. 5, Kluwer Academic Publishers, 2002.

[6] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197.

[7] H. Ishibuchi, T. Yoshida, T. Murata, Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling, IEEE Transactions on Evolutionary Computation 7 (2) (2003) 204–223.

[8] P. Moscato, C. Cotta, A Gentle Introduction to Memetic Algorithms, Vol. 57, Springer, 2003, Ch. 5, pp. 105–144.

[9] L. Davis, Handbook of Genetic Algorithms, Van Nostrand Reinhold, 1991.

[10] Y. S. Ong, A. Keane, Meta-lamarckian learning in memetic algorithms, Evolutionary Computation, IEEE Transactions on 8 (2) (2004) 99–110.

[11] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, V. G. da Fonseca, Performance assessment of multiobjective optimizers: An analysis and review, IEEE Transactions on Evolutionary Computation 7 (2) (2003) 117–132.

[12] A. Konstantinidis, D. Zeinalipour-Yazti, P. Andreou, G. Samaras, P. K. Chrysanthis, Intelligent search in social communities of smartphone users, Distributed and Parallel Databases 31 (2) (2013) 115–149.

[13] Q. Zhang, H. Li, MOEA/D: A multi-objective evolutionary algorithm based on decomposition, IEEE Transactions on Evolutionary Computation 11 (6) (2007) 712–731.

[14] B.-B. Li, L. Wang, B. Liu, An effective pso-based hybrid algorithm for multiobjective permutation flow shop scheduling, IEEE Transactions on Systems, Man and Cybernetics 38 (2008) 818–831.

[15] Y. Zheng, L. Liu, L. Wang, X. Xie, Learning transportation mode from raw gps data for geographic applications on the web, in: WWW, 2008.

[16] DBLP, DBLP Computer Science Bibliography, http://dblp.uni-trier.de/xml/ (2010).

[17] M. Pinedo, Scheduling: Theory, Algorithms and Systems, 3rd Edition, Springer Science+Business Media, LLC, 2008.

[18] Gnutella, Gnutella peer-to-peer network, http://gnutella.wego.com (14 March 2000).

[19] Q. Lv, P. Cao, E. Cohen, K. Li, S. Shenker, Search and replication in unstructured peer-to-peer networks, in: 16th international conference on Supercomputing (ICS'02), New York, USA, 2002, pp. 84–95.

[20] B. Xu, O. Wolfson, C. Naiman, Machine learning in disruption-tolerant manets, ACM Transactions on Autonomous and Adaptive Systems (TAAS) 4 (4) (2009) 23.

[21] S.-K. Chen, P.-C. Wang, Design and implementation of an anycast services discovery in mobile ad hoc networks, ACM Transactions on Autonomous and Adaptive Systems (TAAS) 6 (1) (2011) 2.

[22] T. Repantis, V. Kalogeraki, Data dissemination in mobile peer-to-peer networks, in: 6th International Conference on Mobile Data Management (MDM'05), Ayia Napa, Cyprus, 2005, pp. 211–219.

[23] S. Eisenman, E. Miluzzo, N. Lane, R. Peterson, G. Seop-Ahn, A. Campbell, Bikenet: A mobile sensing system for cyclist experience mapping, ACM Transactions on Sensor Networks (TOSN'09) 6 (1).

[24] H. Tomiyasu, T. Maekawa, T. Hara, S. Nishio, Profile-based query routing in a mobile social network, in: Mobile Data Management, 2006. MDM 2006. 7th International Conference on, 2006, pp. 105 – 105. doi:10.1109/MDM.2006.127.

[25] A. Gahng-Seop, M. Musolesi, H. Lu, R. Olfati-Saber, A. Campbell, Metrotrack: Predictive tracking of mobile events using mobile phones, in: DCOSS, 2010, pp. 230–243.

[26] C. M. Fonseca, P. J. Fleming, Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization, in: S. Forrest (Ed.), Proceedings of the Fifth International Conference on Genetic Algorithms, University of Illinois at Urbana-Champaign, Morgan Kaufman, San Francisco, CA, , San Mateo, California, 1993, pp. 416–423.

[27] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization, in: Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems, Athens, Greece, 2002, pp. 95–100.

[28] A. Trivedi, D. Srinivasan, K. Sanyal, A. Ghosh, A survey of multiobjective evolutionary algorithms based on decomposition, IEEE Transactions on Evolutionary Computation 21 (3) (2017) 440–462.

[29] H. Ishibuchi, T. Murata, Multi-objective genetic local search algorithm and its application to flowshop scheduling, IEEE Transactions on Systems, Man and Cybernetics 28 (3) (1998) 392–403.

[30] A. Jaszkiewicz, On the performance of multiple-objective genetic local search on the 0/1 knapsack problem - a comparative experiment, IEEE Trans. Evolutionary Computation 6 (4) (2002) 402–412.

[31] E. Zitzler, S. Kunzli, Indicator-based selection in multiobjective search, in: in Proc. 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII, Springer, 2004, pp. 832–842.

[32] H. Ishibuchi, N. Tsukamoto, Y. Sakane, Y. Nojima, Indicator-based evolutionary algorithm with hypervolume approximation by achievement scalarizing functions, in: Proceedings of the 12th annual conference on Genetic and evolutionary computation, ACM, 2010, pp. 527–534.

[33] C. A. C. Coello, Evolutionary multi-objective optimization: Basic concepts and some applications in pattern recognition, in: MCPR, 2011, pp. 22–33.

[34] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, Q. Zhang, Multiobjective evolutionary algorithms: A survey of the state of the art, Swarm and Evolutionary Computation, Elsevier 1 (1) (2011) 32–49.

[35] A. Konstantinidis, K. Yang, Multi-objective energy-efficient dense deployment in wireless sensor networks using a hybrid problem-specific MOEA/D, Applied Soft Computing, Elsevier 11 (6) (2011) 4117–4134.

[36] K. Deb, T. Goel, A hybrid multi-objective evolutionary approach to engineering shape design., in: EMO'01, 2001, pp. 385–399.

[37] C. Liu, J. Liu, Z. Jiang, A multiobjective evolutionary algorithm based on similarity for community detection from signed social networks, Cybernetics, IEEE Transactions on 44 (12) (2014) 2274–2287.

[38] J. Ma, J. Liu, W. Ma, M. Gong, L. Jiao, Decomposition-based multiobjective evolutionary algorithm for community detection in dynamic social networks, The Scientific World Journal 2014.

[39] C. R. Reeves, Statistical properties of combinatorial landscapes: An application to scheduling problems, in: MIC2001: Proceedings of the 4th Metaheuristic International Conference, 2001, pp. 691–695.

[40] C. R. Reeves, A. V. Eremeev, Statistical analysis of local search landscapes, The Journal of the Operational Research Society 55 (7) (2004) pp. 687–693.

[41] W. E. Hart, Adaptive global optimization with local search, Tech. rep., PhD Dissertation, Univ. of California, San Diego, CA (1994).

[42] N. Krasnogor, Studies on the theory and design space of memetic algorithms, Ph.D. thesis, University of the West of England (2002).

[43] M. Gong, C. Liu, L. Jiao, G. Cheng, Hybrid immune algorithm with lamarckian local search for multi-objective optimization, Memetic Comp. 2 (2010) 47 – 67.

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

[45] G. Chatzimiloudis, A. Konstantinidis, C. Laoudias, D. Zeinalipour-Yazti, Crowdsourcing with smartphones, IEEE Internet Computing 16 (5) (2012) 36–44.

[46] Y. Jin, T. Okabe, B. Sendhoff, Adapting weighted aggregation for multiobjective evolution strategies, in: Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization, EMO '01, Springer-Verlag, London, UK, 2001, pp. 96–110.

[47] C. R. Reeves, Landscapes, operators and heuristic search, Annals of Operations Research 86 (1997) 473–490.

[48] D. C. Montgomery, Design and Analysis of Experiments, John Wiley & Sons, 2001.

[49] D. W. Johnson, R. T. Johnson, Cooperation and competition: Theory and research., Interaction Book Company, 1989.

[50] Y. Mei, K. Tang, X. Yao, Decomposition-based memetic algorithm for multiobjective capacitated arc routing problem, Evolutionary Computation, IEEE Transactions on 15 (2) (2011) 151–165.

[51] N. Weicker, G. Szabo, K. Weicker, P. Widmayer, Evolutionary multiobjective optimization for base station transmitter placement with frequency assignment, IEEE Trans. on Evolutionary Computation 7 (2) (2003) 189–203.

[52] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach., IEEE Trans. Evolutionary Computation (1999) 257–271.

[53] P. Czyzak, A. Jaszkiewicz, Pareto simulated annealing - a metaheuristic technique for multiple-objective combinatorial optimization, Journal of Multi-Criteria Decision Analysis 7 (1) (1998) 34–47.