MO-MFCGA: Multiobjective Multifactorial Cellular Genetic Algorithm for Evolutionary Multitasking

Eneko Osaba[†*], Javier Del Ser[†‡], Aritz D. Martinez[†], Jesus L. Lobo[†], Antonio J. Nebro[§] and Xin-She Yang[¶]
[†]TECNALIA, Basque Research and Technology Alliance (BRTA), 48160 Derio, Bizkaia, Spain
[‡]University of the Basque Country (UPV/EHU), 48013 Bilbao, Bizkaia, Spain
[§]Dept. de Lenguajes y Ciencias de la Computacion, ITIS Software, Universidad de Malaga, Spain
[¶]Middlesex University London, The Burroughs, London NW4 4BT, United Kingdom
[*]Corresponding author. Email: eneko.osaba@tecnalia.com

***Abstract*—Multiobjetive optimization has gained a considerable momentum in the evolutionary computation scientific community. Methods coming from evolutionary computation have shown a remarkable performance for solving this kind of optimization problems thanks to their implicit parallelism and the simultaneous convergence towards the Pareto front. In any case, the resolution of multiobjective optimization problems (MOPs) from the perspective of multitasking optimization remains almost unexplored. Multitasking is an incipient research stream which explores how multiple optimization problems can be simultaneously addressed by performing a single search process. The main motivation behind this solving paradigm is to exploit the synergies between the different problems (or tasks) being optimized. Going deeper, we resort in this paper to the also recent paradigm Evolutionary Multitasking (EM). We introduce the adaptation of the recently proposed Multifactorial Cellular Genetic Algorithm (MFCGA) for solving MOPs, giving rise to the Multiobjective MFCGA (MO-MFCGA). An extensive performance analysis is conducted using the Multiobjective Multifactorial Evolutionary Algorithm as comparison baseline. The experimentation is conducted over 10 multitasking setups, using the Multiobjective Euclidean Traveling Salesman Problem as benchmarking problem. We also perform a deep analysis on the genetic transferability among the problem instances employed, using the synergies among tasks aroused along the MO-MFCGA search procedure.**

***Index Terms*—Multiobjective Optimization, Transfer Optimization, Evolutionary Multitasking, Cellular Genetic Algorithm, Traveling Salesman Problem.**

I. Introduction

Efficiently solving optimization problems driven by multiple conflicting objectives has been a recurrent concern that has been under active study by the research community. The main rationale for this continued activity is the prevalence of multiple objectives in real-world decision making processes, each corresponding to e.g., different evaluation criteria for the decision variables or interests/preferences imposed by different stakeholders of the considered scenario. Addressing these multiobjective optimization problems (MOPs) implies the discovery of a set of solutions which provides a balance among the considered objectives [1]. In this strand of optimization problems, one solution can be declared to be *Pareto optimal* when there is no other solution better than it in all considered objectives. This is the main reason why the presence of objectives, when mutually competitive, do not lead to a single solution, but rather to a set of solutions differently trading among the objectives of the problem.

Over decades a myriad of solvers has been proposed for efficiently dealing with MOPs. The adoption of meta-heuristic algorithms relying on either Evolutionary Computation or Swarm Intelligence has gained a remarkable popularity in recent years [2], being nowadays one of the most preferred alternatives for solving multiobjective problems. Among them, the Nondominated Sorting Genetic Algorithm II (NSGA-II [3]), the multiobjective evolutionary algorithm based on decomposition (MOEA/D [4]) or the Speed-constrained Multiobjective Particle Swarm Optimization (SMPSO [5]) stand out from the numerous algorithmic proposals in the field.

Although MOPs have been so far tackled via different solvers, it has not been until recently when Transfer Optimization [6] has emerged as a promising paradigm for this kind of optimization problems. Transfer Optimization is an incipient research field with a growing momentum in the current research panorama [7]. The main inspiration underneath this paradigm is the exploitation of the knowledge generated during the optimization of one problem for efficiently solving another related or unrelated problem. Within Transfer Optimization three different branches can be distinguished: sequential transfer, multiform optimization, and multitasking. Among them, we focus our attention on multitasking, which aims to simultaneous address different optimization problems of equal priority by dynamically exploiting synergies existing among them. Multitasking is often approached through the lenses of evolutionary computation, forging what is nowadays referred to as Evolutionary Multitasking (EM [8]). EM deals with multitasking environments by adopting concepts, operators and search strategies drawn from evolutionary computation, leveraging their design flexibility and adaptability to cope with problems that potentially feature different search spaces.

To date, several studies have demonstrated that multitasking

The authors would like to thank the Basque Government through EMAITEK and ELKARTEK funding grants. J. Del Ser also acknowledges funding support from the Department of Education of the Basque Government (Consolidated Research Group MATHMODE, IT1294-19). Antonio J. Nebro is supported by the Spanish Ministry of Science and Innovation via Grant PID2020-112540RB-C41 (AEI/FEDER, UE) and Andalusian PAIDI program with grant P18-RT-2799.

optimization can efficiently solve multiple problems at once, including MOPs [9]. Despite the efforts conducted to adapt EM for solving MOPs, efficient multitasking solvers to deal with problems comprising several objectives is still scarce when compared to those designed for single-objective problems. We assume this noted lack of contributions as our principal motivation to adapt the Multifactorial Cellular Genetic Algorithm (MFCGA) recently proposed in [10] to the case where tasks are MOPs, giving rise to the Multiobjetive MFCGA (MO-MFCGA). Briefly explained, MFCGA is an EM metaheuristic algorithm inspired by the foundations of cellular genetic algorithms [11] and multifactorial optimization [12], the latter being a specific materialization of the EM paradigm that has shown great efficiency when tackling different practical multitasking setups.

We assess the performance of the proposed MO-MFCGA approach over multiobjective multitasking environments by designing an extensive experimentation using the well-known Traveling Salesman Problem (TSP [13]) as the baseline problem for benchmarking purposes. Specifically, we have used 10 multiobjective Euclidean TSP (MO-ETSP) instances for building 10 different multitasking use cases, whose results are presented and discussed towards ascertaining the competitiveness of MO-MFCGA with respect to the Multiobjective Multifactorial Evolutionary Algorithm (MO-MFEA [9]), which is arguably the standard algorithmic bearer of this specific research area. Finally, another remarkable contribution of this work is the analysis of the genetic knowledge transfer emerged among the MO-ETSP instances in use. According to the insights given recurrently in the related literature (e.g. [14]), analyses of this kind are required to fully understand the behavior of new algorithmic proposals in regards to the exchange of knowledge among tasks, posing a valuable addition to the state of the art and providing insights for future follow-up studies.

The rest of this manuscript is structured as follows: Section II reviews background related to EM, MO-MFEA and MO-TSP. Next, Section III describes the main adaptations made to the seminal MFCGA to tackle MOPs, leading to the definition of the proposed MO-MFCGA. The experimental setup and the discussion of the results are given and discussed in Section IV. Section V finishes this paper by drawing conclusions and outlining future work rooted on our findings.

II. Background

This section provides a brief background on three specific aspects connected to this work: EM and Multifactorial Optimization (Section II-A), MFEA (Section II-B) and MO-TSP (Section II-C).

A. Evolutionary Multitasking and Multifactorial Optimization

As mentioned in the introduction, multitask optimization focuses on solving several optimization problems or tasks in a simultaneous manner. However, a fundamental premise to yield a more efficient search than solving each problem in isolation from each other is that knowledge transfer is held among related problems, so that the search is expedited when such relationships are synergistic [6]. In the multitasking mainstream, EM is often resorted for designing search algorithms that implement such knowledge transfer. The principal motivation for the adoption of concepts from evolutionary computation in this area is twofold: i) the inherent parallelism granted by a population of solutions, providing an suitable framework for dealing with concurrent tasks, and ii) the maintenance of a set of solutions over the search, which eases the exploration of synergies among the tasks and the exchange of genetic material among individuals [12]. Diverse perspectives have been given over the years for formalizing the EM concept, reviewed recently in comprehensive surveys on the matter [7]. However, there is a clear consensus in the literature around the capital role of multifactorial optimization as a leader paradigm to realize EM [15], with MFEA [12] as one of the most popular algorithms resulting from this paradigm.

Multifactorial optimization can be described as an EM scenario in which K concurrent problems or tasks (each characterized by its search space) must be optimized in a simultaneous fashion. The objective function of task T_k is represented as $f_k : \Omega_k \rightarrow \mathbb{R}$, where Ω_k denotes the search space of T_k. Assuming that all tasks are *minimization* problems, the main goal of multifactorial optimization is to reach a set of solutions $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K\}$ such that $\mathbf{x}_k = \arg\min_{\mathbf{x}\in\Omega_k} f_k(\mathbf{x})$. One of the differential features of multifactorial optimization is that it pursues this goal (i.e. finding $\{\mathbf{x}_k\}_{k=1}^{K}$) by exploring a single and unified search space Ω', so that solutions defined in this unified search space can be decoded to yield a task-specific solution for any of the K optimization problems at hand. To this end, algorithms embracing multifactorial optimization usually maintain a P-sized population of individuals $\{\mathbf{x}'_i\}_{i=1}^{P}$ (with $\mathbf{x}'_i \in \Omega'$), based on which several definitions are made:

Definition 1 (Factorial Rank): the factorial rank r_i^k of individual $\mathbf{x}'_i$ for task T_k is the index of that individual within the population, sorted in ascending order of the fitness value $f_k(\cdot)$ of its individuals over T_k.

Definition 2 (Skill Factor): the skill factor τ_i of individual $\mathbf{x}'_i$ is the task T_k for which it is assigned.

Definition 3 (Scalar Fitness): the scalar fitness φ_i of $\mathbf{x}'_i$ is given by $\varphi_i = 1/r_i^k$.

Many interesting research works have been published in the last few years around multifactorial optimization and its forefront method (MFEA). Among them, it is interesting the work in [16], where MFEA is firstly adapted for tackling discrete optimization problems. This work is of particular relevance for this work, since it introduces the unified discrete encoding strategy that lies at the core of our proposal. It is also worth highlighting the work conducted in [17], in which the adaptive variant of MFEA is proposed. In [18], MFEA is applied for dealing with multitasking environments under interval uncertainties. Also valuable is the study in [19], in which MFEA is used to solve several mobile agents path planning problems at once. Further related contributions can be found in the literature review offered in [7].

Focused on the specific context covered in this manuscript (EM for MOPs), the multiobjective variant of the canonical MFEA (MO-MFEA) was introduced in [9], posing an inflection point in the literature that had gravitated on single-objective problem formulations until then. Since its inception, MO-MFEA has been employed in a heterogeneous range of applications, such as electric power dispatch problems [20], operational indices optimization problems [21] or multiobjective pollution-routing problems [22]. Lastly, in [23] an adaptive version of the MO-MFEA has been introduced by the same authors that developed MFEA-II, an extension of the seminal MFEA that allows for the estimation of the intensity of knowledge exchange among tasks over the search.

B. Multiobjective Multifactorial Evolutionary Algorithm

MO-MFEA is a multifactorial optimization algorithm designed for multiobjective EM environments that is largely inspired by NSGA-II [3], which is the most popular multiobjective metaheuristic algorithm. In fact, in the concrete case of $K = 1$, MO-MFEA reduces to NSGA-II. Algorithm 1 describes algorithmically the main steps of MO-MFEA. We refer the audience to [9] for further details. However, it is worth mentioning that MO-MFEA has four main features that are crucial for guiding the multitasking search process:

Algorithm 1: Pseudocode of MO-MFEA [9]

Randomly generate a population $\mathbf{X}$ of P individuals
Assign a skill factor τ_i for every $\mathbf{x}'_i$ in $\mathbf{X}$
Decode and evaluate each $\mathbf{x}'_i$ only for task τ^i
Calculate the scalar fitness φ_i of each $\mathbf{x}'_i$
repeat
    Apply genetic operators on $\mathbf{X}$ to get offspring $\mathbf{X}^*$
    Determine the τ_i for each individual in $\mathbf{X}^*$
    Evaluate each individual in $\mathbf{X}^*$ in its skill factor τ_i
    Combine $\mathbf{X}$ and $\mathbf{X}^*$ to yield $\mathbf{Q} = [\mathbf{X}; \mathbf{X}^*]$
    Update scalar fitness φ_i for each individual in $\mathbf{Q}$
    Build the next population $\mathbf{X}$ by selecting the best P individuals in $\mathbf{Q}$ in terms of scalar fitness
until *termination criterion reached*
Return all individuals for each task T_k

- *Unified solution representation*: the selection of a suitable encoding strategy for $\mathbf{x}'_i$ is one of the most important aspects to consider when dealing with multifactorial optimization. A proper strategy is crucial for building a unified search space Ω' capable of fully representing the search spaces of all the K tasks under consideration. In this work we deal with MOPs whose solutions are permutations of a range of integers. Consequently, we resort to the well-known *permutation encoding* as the unified representation for $\mathbf{x}'_i$ [24]. Assuming K MO-TSP problems are simultaneously tackled, each MO-TSP instance T_k has its own dimensionality D_k equal to the number of cities to be visited. In this way, an individual $\mathbf{x}'_i$ is represented as a permutation of the integer set $\{1, 2, \ldots, D_{max}\}$, where $D_{max} = \max_{k\in\{1,\ldots,K\}} D_k$, that is, the maximum problem dimension among all the K tasks. Thus, if a solution $\mathbf{x}'_i$ has to be evaluated for a task T_k whose $D_k < D_{max}$, only the values lower than D_k are considered for building the solution $\mathbf{x}_k$ that can be evaluated as per $f_k(\cdot)$. These chosen values follow the same order as in $\mathbf{x}'_i$.

Algorithm 2: Assortative mating in MO-MFEA

Consider candidate parents $\mathbf{x}'_1$ and $\mathbf{x}'_2$
Generate a random real number $rand$ between 0 and 1
if $\tau^{(1)} = \tau^{(2)}$ *or* $rand < rmp$
    $(\mathbf{x}'^{,*}_1, \mathbf{x}'^{,*}_2) =$ crossover+mutation$(\mathbf{x}'_1, \mathbf{x}'_2)$
else
    $\mathbf{x}'^{,*}_1 =$ mutation$(\mathbf{x}'_1)$, $\mathbf{x}'^{,*}_2 =$ mutation$(\mathbf{x}'_2)$
end

- *Assortative mating*: this mechanism is built under the premise that individuals are more prone to interact with mates coming from similar cultural background [12]. Following this assumption, the search process of MO-MFEA prioritizes mating solutions with the same skill factor τ_i. We depict in Algorithm 2 the specific assortative mating operator used by MO-MFEA. It should be noted that the random mating probability $rmp \in \mathbb{R}[0, 1]$ is a control parameter that must be set before the search is run.

Algorithm 3: Cultural transmission in MO-MFEA

Consider an generated offspring individual $\mathbf{x}'^{,*}_1$
if $\mathbf{x}'^{,*}_1 =$ crossover+mutation$(\mathbf{x}'_1, \mathbf{x}'_2)$ *and* $rand < 0.5$
    $\mathbf{x}'^{,*}_1$ inherits τ_1 from $\mathbf{x}'_1$
else if $\mathbf{x}'^{,*}_1 =$ crossover+mutation$(\mathbf{x}'_1, \mathbf{x}'_2)$ *and* $rand \leq 0.5$
    $\mathbf{x}'^{,*}_1$ inherits τ_2 from $\mathbf{x}'_2$
else if $\mathbf{x}'^{,*}_1 =$ mutation$(\mathbf{x}'_i)$ for any $i \in \{1, 2\}$
    $\mathbf{x}'^{,*}_1$ inherits τ_i from $\mathbf{x}'_i$
end

- *Offspring evaluation*: this strategy guarantees the computational scalability of the solver by imposing that each produced offspring is evaluated only on one task. More concretely, a generated individual is evaluated in the task corresponding to the skill factor inherited from its parents. The inheritance of the skill factor is driven by the mechanisms known as *Cultural Transmission via Selective Imitation*, which is described in Algorithm 3.
- *Scalar fitness-based selection*: this is the survival criterion of MO-MFEA, which follows an elitist strategy. As such, the best P individuals in terms of scalar fitness σ_i are the ones retained in the population for the next generation. To this end, individuals in the population must be sorted in terms of their scalar fitness σ_i, which must reflect the quality of the solutions in terms of Pareto optimality. To this end, in MO-MFEA the computation of the scalar fitness relies on

the crowding distance and the first non-dominated front. We recommend reading [9] for further details.

C. Multiobjective Traveling Salesman Problem

As mentioned previously, this study utilizes MO-ETSP instances as the tasks evolved in a EM setting for measuring the performance of our proposed MO-MFCGA. Formally, the single-objective version of the TSP can be defined as a complete graph $G = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ is the set of $N = |\mathcal{V}|$ nodes of the graph (*cities*), and $\mathcal{A} = \{(v_i, v_j) : (v_i, v_j) \in \mathcal{V} \times \mathcal{V} \text{ for } i \neq j\}$ the group of edges interconnecting these nodes. Furthermore, each edge $(v_i, v_j) \in \mathcal{V}$ has an associated cost c_{ij}, which in the symmetric TSP is the same in both directions, i.e., $c_{ij} = c_{ji}$. The main objective of the canonical TSP is to find a complete path that, starting and finishing at the same node, visits each node exactly once and minimizes the total cost of the route. The TSP is one of the most intensively studied optimization problems in the history of the optimization field, attracting an interest of the research community that lasts to the present day [25].

In this paper we deal with the multiobjective MO-ETSP variant of the TSP, in which the minimization of the tour cost must be done over M planes (networks) simultaneously [26]. Each of such planes is characterized by different values of the edge costs c_{ij}, resulting from the Euclidean distance between every node of the network, each located in a different position in every plane $m = 1, \ldots, M$. The fact that such locations vary gives rise to a Pareto trade-off between the paths minimizing the total cost of each of the M planes under consideration. Several other formulations of the MO-TSP can be found in the literature, but MO-ETSP is the most frequently used one in the related studies. To cite a few, in [27] a multiobjective Artificial Bee Colony is proposed for dealing with the MO-ETSP. The work in [28] proposed a novel method coined as MOEA/NSM, which integrates features from NSGA-II, SPEA2 and MOEA/D. Chen et al. introduced in [29] a multiobjective evolutionary method based on the Physarum-inspired computational model. In [30], a genetic algorithm was utilized to tackle the MO-ETSP. Multiobjective ant colonies were also explored for this same problem in [31].

III. Proposed Multiobjective Multifactorial Cellular Genetic Algorithm

As mentioned in Section II-B, MO-MFEA hinges on four different concepts: *unified solution representation*, *assortative mating*, *offspring evaluation* though the application of *cultural transmission via selective imitation*, and *scalar fitness-based selection*. We have considered and reformulated these four concepts when designing our proposed MO-MFCGA. Algorithm 4 shows its pseudocode, inspired by both cellular genetic algorithms, MFEA and the optimization of MOPs.

Regarding the unified representation employed, the same approach described in Section II-B for MO-MFEA has been employed. Furthermore, classical mutation and crossover mechanisms have been used as genetic search operators. At each generation, every individual $\mathbf{x}'_i$ of the population undergoes these two operators (without mutation nor crossover probabilities), producing two new solutions: $\mathbf{x}_i^{\prime,mut}$ and $\mathbf{x}_i^{\prime,cross}$. On the one hand, $\mathbf{x}_i^{\prime,mut}$ is produced by applying the mutation operator to $\mathbf{x}'_i$. On the other hand, $\mathbf{x}_i^{\prime,crossover}$ is the result of crossing $\mathbf{x}'_i$ with a random neighbor $\mathbf{x}'_j$ from the cellular neighborhood $\mathbf{X}_i^{\circledast} \subset \mathbf{X}$ of $\mathbf{x}'_i$. The geometry and neighborhood structure of the cellular grid is one of the parameters of MO-MFCGA, which can be set to any of the available options usually considered in the literature related to cellular automata (e.g., Moore or von Neumann).

Algorithm 4: Pseudocode of MO-MFCGA

Generate a population $\mathbf{X}$ of P random individuals
Assign skill factor τ_i for every $\mathbf{x}'_i \in \mathbf{X}$
Evaluate each $\mathbf{x}'_i$ only for task τ_i
Let $\mathbf{X}_i^{\circledast}$ denote the set of neighbors of $\mathbf{x}'_i$
repeat
    for $i = 1, \ldots, P$ **do**
        Randomly choose a neighbor $\mathbf{x}'_j \in \mathbf{X}_i^{\circledast}$
        $\mathbf{x}_i^{\prime,cross} = \text{crossover}(\mathbf{x}'_i, \mathbf{x}'_j)$
        $\mathbf{x}_i^{\prime,mut} = \text{mutation}(\mathbf{x}'_i)$
        Evaluate $\mathbf{x}_i^{\prime,mut}$ and $\mathbf{x}_i^{\prime,cross}$ only for task τ_i
        Add $[\mathbf{x}'_i, \mathbf{x}_i^{\prime,mut}, \mathbf{x}_i^{\prime,cross}]$ to $\mathbf{Q}_{\tau_i}$
    end
    for $k = 1, \ldots, K$ **do**
        Rank Q_k by *Fast Non-Dominated Sorting*
    end
    for $i = 1, \ldots, P$ **do**
        $\mathbf{x}_i = \text{bestRank}(\mathbf{x}'_i, \mathbf{x}_i^{\prime,mut}, \mathbf{x}_i^{\prime,cross})$ within Q_{τ_i}
    end
until *termination criterion reached*
Return all individuals for each task T_k

Once $\mathbf{x}_i^{mutation}$ and $\mathbf{x}_i^{crossover}$ are created, they are evaluated by following the same procedure described for MO-MFEA (Subsection II-B), which guarantees the computational efficiency of the method. This way, $\mathbf{x}_i^{\prime,mut}$ and $\mathbf{x}_i^{\prime,cross}$ are only evaluated for task T_{τ_i}, where τ_i is the skill factor of $\mathbf{x}'_i$. This is an essential difference to the scalar factor inheritance mechanism of MO-MFEA (Algorithm 3), since each individual in the cellular grid is devoted to the optimization of the same MOP during the entire search of MO-MFCGA. Furthermore, as in MO-MFEA, the initial assignment of skill factors ensures an equal proportion of solutions allocated to each task.

Once $\mathbf{x}_i^{\prime,mut}$ and $\mathbf{x}_i^{\prime,cross}$ have been created and evaluated, they are appended to a temporary population $\mathbf{Q}_{\tau_i}$, together with the original *parent* individual $\mathbf{x}'_i$. As a result, the K subpopulations $\{\mathbf{Q}_k\}_{k=1}^K$ store the previous individuals in $\mathbf{X}$ and all the offspring generated in a single generation of the search process. In order to select which individuals survive in the cellular grid for subsequent generations, we adopt a *local improvement selection* mechanism. Through this procedure, for each position i of the population $\mathbf{X}$, the best solution among $\mathbf{x}'_i$, $\mathbf{x}_i^{\prime,mut}$ and $\mathbf{x}_i^{\prime,cross}$ survives to the next iteration, whereas the other two individuals are discarded. In this case, in order to decide the best of these three candidates, the temporary population Q_{τ_i} is ranked by using the *Fast Non-Dominated Sorting* criterion from [3] over the search space spanned by the objectives of task T_{τ_i}. Thus, the best individual is the one positioned in the best ranking position as per this criterion. Ties in this ranking are resolved by prioritizing $\mathbf{x}_i'$ (the solution is substituted only if it is improved), followed by $\mathbf{x}_i^{\prime,cross}$ (sharing of knowledge) and $\mathbf{x}_i^{\prime,mut}$ (local improvement).

IV. Experimental Setup, Results and Discussion

As mentioned in previous sections, the experimentation described on this work comprises several *test cases* comprising several two-objective MO-ETSP instances. Specifically, 10 different MO-ETSP instances have been designed, whose composition is detailed in Table I. It should be noted that these instances are the combination of 5 single-objective Euclidean TSP instances retrieved from the Krolak/Felts/Nelson dataset, which is part of the well-known TSPLIB repository. KroBC100, for example, is the combination of KroB100 and KroC100 instances, involving $N = 100$ nodes (cities) each. The combination of these Krolak/Felts/Nelson instances for synthesizing multiobjective problems is common practice in the MO-ETSP related literature. Our experiments using these designed test cases is to compare the performance of MO-MFCGA and MO-MFEA, but also to examine the exchange of genetic material of MO-MFCGA over the search, which is realized through its grid neighborhood structure and crossover operator (lines 7 and 8 of Algorithm 4).

TABLE I
Composition of the MO-ETSP instances under consideration

Instance	kroA100	kroB100	kroC100	kroD100	kroE100
kroAB100	✓	✓			
kroAC100	✓		✓		
kroAD100	✓			✓	
kroAE100	✓				✓
kroBC100		✓	✓		
kroBD100		✓		✓	
kroBE100		✓			✓
kroCD100			✓	✓	
kroCE100			✓		✓
kroDE100				✓	✓

Each of the 10 generated multitasking test cases requires that MO-MFCGA and MO-MFEA should solve all its assigned MO-ETSP instances. We summarize all these test cases in Table II. As it can be seen, 5 different test cases are composed of 4 MO-ETSP instances, 4 comprise 6 MO-ETSP cases, while the last test case covers all the 10 MO-ETSP instances. The main motivation for generating these test cases is two-fold. On one hand, these 10 configurations ensures the heterogeneity and variety of the test cases is assured, that is, each MO-ETSP instance is considered in a similar number of test cases. On the other hand, with these configurations we can explore the capability of MO-MFCGA to exploit known synergies between MO-ETSP instances that share a single-objective Euclidean TSP problem. In other words, MO-ETSP instances such as kroAB100 and kroAC100 can be placed under the focus of our knowledge transfer study, as both instances have in common the kroA100 TSP instance in their composition.

TABLE II
Summary of the 10 designed multitasking test cases

Test Case	MO-ETSP tasks involved
TC_4_1	kroAB100, kroAC100, kroAD100, kroAE100
TC_4_2	kroAB100, kroBC100, kroBD100, kroBE100
TC_4_3	kroAC100, kroBC100, kroCD100, kroCE100
TC_4_4	kroAD200, kroBD100, kroCD100, kroDE100
TC_4_5	kroAE100, kroBE100, kroCE100, kroDE100
TC_6_1	kroAC100, kroAD100, kroAE100, kroBC100, kroBD100, kroBE100
TC_6_2	kroAB100, kroAC100, kroBD100, kroBE100, kroCD100, kroCE100
TC_6_3	kroAC100, kroAD100, kroBD100, kroBE100, kroCD100, kroCE100
TC_6_4	kroAB100, kroAC100, kroAD100, kroBE100, kroCE100, kroDE100
TC_10	kroAB100, kroAC100, kroAD100, kroAE100, kroBC100, kroBD100, kroBE100, kroCD100, kroCE100, kroDE100

In what refers to the configuration of the algorithms under comparison, similar operators and parameters have been considered for both MO-MFCGA and MO-MFEA for the sake of a fair and rigorous comparison. Table III shows the specific configurations used in the experiments. Parameter values have been established based on those reported in previous work related to cellular genetic algorithms, MO-MFEA and discrete MFEAs [9], [16]. A further refinement phase was performed based on a grid of values around typical configurations of these solvers, reporting the results of the best configurations found during this phase. Furthermore, we embrace methodological guidelines for benchmarking metaheuristic algorithms [32] and perform 20 independent runs for each multitasking test case and algorithm involved in the experimentation. As stopping criterion, both algorithms stop their execution after $750 \cdot 10^3$ objective function evaluations. Furthermore, all experiments have been performed on an Intel Xeon E52650 v3 2.30 GHz processor with 32 GB RAM. To ensure reproducibility of these results and support future studies using MO-MFCGA, a Java implementation of our proposed algorithm is available in https://git.code.tecnalia.com/aritz.martinez/mo-mfcga.

TABLE III
Parametrization of MO-MFCGA and MO-MFEA

Parameter	MO-MFCGA	MO-MFEA
Population size P	200	
crossover($\cdot$)	Order crossover	
mutation($\cdot$)	2-opt	
Crossover probability		0.9
Mutation probability		0.1
Grid, neighborhood	2D, Moore	

A. Results and Discussion

Our discussion departs from Table IV, which depicts the average and standard deviation of two multi-objective quality indicators (hypervolume, HV, and inverted generational distance plus, IGD+) reached by MO-MFCGA and MO-MFEA in all the 10 multitasking test cases detailed above. Results are reported for each MO-ETSP instance, and

TABLE IV

MEAN AND STANDARD DEVIATION OF THE HV AND IGD+ MULTIOBJECTIVE QUALITY INDICATORS ACHIEVED BY MO-MFEA AND THE PROPOSED MO-MFCGA OVER EACH OF THE MO-ETSP INSTANCES OF THE TEST CASES UNDER CONSIDERATION

Test case	kroAB100		kroAC100		kroAD100		kroAE100		kroBC100		kroBD100		kroBE100		kroCD100		kroCE100		kroDE100	
	MFEA	MFCGA	MFEA	MFCGA	MFEA	MFCGA	MFEA	MFCGA	MFEA	MFCGA	MFEA	MFCGA	MFEA	MFCGA	MFEA	MFCGA	MFEA	MFCGA	MFEA	MFCGA
TC_4_1	$0.58_{0.02}$	$0.75_{0.01}$	$0.58_{0.01}$	$0.74_{0.01}$	$0.6_{0.02}$	$0.76_{0.01}$	$0.57_{0.02}$	$0.75_{0.01}$	–	–	–	–	–	–	–	–	–	–	–	–
	$0.92_{0.02}$	$0.03_{0.01}$	$0.95_{0.03}$	$0.03_{0.01}$	$0.89_{0.02}$	$0.04_{0.01}$	$0.94_{0.02}$	$0.04_{0.01}$	–	–	–	–	–	–	–	–	–	–	–	–
TC_4_2	$0.59_{0.02}$	$0.75_{0.01}$	–	–	–	–	–	–	$0.56_{0.02}$	$0.73_{0.01}$	$0.59_{0.02}$	$0.74_{0.01}$	$0.6_{0.02}$	$0.74_{0.01}$	–	–	–	–	–	–
	$0.96_{0.02}$	$0.03_{0.01}$	–	–	–	–	–	–	$0.97_{0.02}$	$0.03_{0.01}$	$0.96_{0.02}$	$0.03_{0.01}$	$0.92_{0.02}$	$0.03_{0.01}$	–	–	–	–	–	–
TC_4_3	–	–	$0.61_{0.02}$	$0.75_{0.01}$	–	–	–	–	$0.62_{0.02}$	$0.76_{0.01}$	–	–	–	–	$0.57_{0.02}$	$0.75_{0.01}$	$0.58_{0.02}$	$0.74_{0.01}$	–	–
	–	–	$0.9_{0.03}$	$0.04_{0.01}$	–	–	–	–	$0.87_{0.03}$	$0.03_{0.01}$	–	–	–	–	$0.94_{0.04}$	$0.04_{0.01}$	$0.93_{0.03}$	$0.03_{0.01}$	–	–
TC_4_4	–	–	–	–	–	–	–	–	–	–	$0.61_{0.02}$	$0.76_{0.01}$	–	–	$0.57_{0.02}$	$0.73_{0.01}$	–	–	$0.61_{0.02}$	$0.76_{0.01}$
	–	–	–	–	–	–	–	–	–	–	$0.96_{0.01}$	$0.03_{0.01}$	–	–	$0.95_{0.02}$	$0.04_{0.01}$	–	–	$0.92_{0.02}$	$0.03_{0.01}$
TC_4_5	–	–	–	–	–	–	$0.59_{0.02}$	$0.76_{0.01}$	–	–	–	–	$0.59_{0.03}$	$0.73_{0.01}$	–	–	$0.63_{0.02}$	$0.75_{0.01}$	$0.59_{0.03}$	$0.73_{0.01}$
	–	–	–	–	–	–	$0.91_{0.02}$	$0.03_{0.01}$	–	–	–	–	$0.89_{0.03}$	$0.03_{0.01}$	–	–	$0.84_{0.02}$	$0.03_{0.01}$	$0.92_{0.03}$	$0.03_{0.01}$
TC_6_1	–	–	$0.64_{0.02}$	$0.82_{0.01}$	$0.62_{0.02}$	$0.81_{0.01}$	$0.66_{0.02}$	$0.85_{0.01}$	$0.65_{0.02}$	$0.84_{0.01}$	$0.64_{0.02}$	$0.81_{0.01}$	$0.66_{0.01}$	$0.83_{0.01}$	–	–	–	–	–	–
	–	–	$0.66_{0.02}$	$0.03_{0.0}$	$0.71_{0.02}$	$0.03_{0.01}$	$0.64_{0.03}$	$0.03_{0.01}$	$0.64_{0.02}$	$0.03_{0.01}$	$0.7_{0.02}$	$0.03_{0.01}$	$0.64_{0.02}$	$0.03_{0.01}$	–	–	–	–	–	–
TC_6_2	$0.68_{0.01}$	$0.85_{0.01}$	$0.62_{0.02}$	$0.79_{0.01}$	–	–	–	–	–	–	$0.63_{0.03}$	$0.81_{0.01}$	$0.65_{0.02}$	$0.82_{0.01}$	$0.6_{0.03}$	$0.8_{0.01}$	$0.64_{0.02}$	$0.83_{0.01}$	–	–
	$0.62_{0.02}$	$0.03_{0.01}$	$0.67_{0.02}$	$0.03_{0.01}$	–	–	–	–	–	–	$0.69_{0.03}$	$0.03_{0.0}$	$0.65_{0.01}$	$0.03_{0.01}$	$0.72_{0.04}$	$0.03_{0.01}$	$0.66_{0.02}$	$0.03_{0.0}$	–	–
TC_6_3	–	–	$0.62_{0.02}$	$0.81_{0.01}$	$0.61_{0.03}$	$0.82_{0.01}$	–	–	–	–	$0.63_{0.02}$	$0.8_{0.01}$	$0.66_{0.02}$	$0.82_{0.01}$	$0.64_{0.02}$	$0.84_{0.01}$	$0.61_{0.02}$	$0.79_{0.01}$	–	–
	–	–	$0.68_{0.02}$	$0.03_{0.01}$	$0.67_{0.02}$	$0.03_{0.01}$	–	–	–	–	$0.68_{0.02}$	$0.03_{0.0}$	$0.63_{0.02}$	$0.03_{0.0}$	$0.66_{0.02}$	$0.03_{0.01}$	$0.7_{0.02}$	$0.03_{0.01}$	–	–
TC_6_4	$0.62_{0.03}$	$0.81_{0.01}$	$0.64_{0.02}$	$0.82_{0.01}$	$0.6_{0.03}$	$0.79_{0.01}$	–	–	–	–	–	–	$0.66_{0.02}$	$0.82_{0.01}$	–	–	$0.64_{0.03}$	$0.82_{0.01}$	$0.64_{0.03}$	$0.8_{0.01}$
	$0.66_{0.02}$	$0.03_{0.01}$	$0.65_{0.01}$	$0.03_{0.0}$	$0.7_{0.02}$	$0.03_{0.01}$	–	–	–	–	–	–	$0.68_{0.02}$	$0.03_{0.01}$	–	–	$0.68_{0.03}$	$0.03_{0.01}$	$0.66_{0.02}$	$0.02_{0.0}$
TC_10	$0.57_{0.03}$	$0.79_{0.01}$	$0.55_{0.04}$	$0.78_{0.01}$	$0.53_{0.03}$	$0.77_{0.01}$	$0.53_{0.04}$	$0.8_{0.01}$	$0.55_{0.04}$	$0.79_{0.01}$	$0.56_{0.05}$	$0.79_{0.01}$	$0.61_{0.03}$	$0.81_{0.01}$	$0.54_{0.03}$	$0.8_{0.01}$	$0.57_{0.03}$	$0.8_{0.01}$	$0.57_{0.02}$	$0.79_{0.01}$
	$0.71_{0.03}$	$0.04_{0.01}$	$0.72_{0.03}$	$0.03_{0.01}$	$0.77_{0.03}$	$0.04_{0.01}$	$0.78_{0.05}$	$0.04_{0.01}$	$0.74_{0.04}$	$0.03_{0.0}$	$0.76_{0.06}$	$0.03_{0.01}$	$0.68_{0.03}$	$0.04_{0.01}$	$0.74_{0.05}$	$0.04_{0.01}$	$0.73_{0.03}$	$0.04_{0.01}$	$0.73_{0.02}$	$0.04_{0.01}$

Note: The upper value $mean_{std}$ in the cell corresponding to every (algorithm, MO-ETSP, test case) combination indicates the HV statistics, whereas the bottom value shows IGD+ statistics.

are computed over the 20 independent runs performed for every test case. The HV is arguably the most widely used indicator in the multiobjective community, and it evaluates the solutions by simultaneously taking into account the convergence and diversity. Thus, the higher the HV, the better the algorithm can be declared to perform. The IDG+ measures both convergence and diversity of Pareto front approximations, and it is a weak Pareto-compliance variant of the basic IDG. In this regard, the lower the IGD+ is, the better the algorithms performance can be considered to be.

The results shown in Table IV are conclusive: we confirm that MO-MFCGA reaches better results when compared to MO-MFEA in all the instances and over every test case considered in this experimental benchmark. Gaps between both algorithms are wide in term of both quality indicators, which can be also stated to be statistically significant as per the results of a Wilcoxon rank-sum test performed over the results of every (MO-ETSP, test case) combination. The p-value obtained by this test is lower than $1e-3$ in all cases, determining that the differences noted between the indicator values of MO-MFEA and MO-MFCGA are relevant. To qualitatively inspect these results, we focus on the most complex multitasking test case TC_10 and depict in Figure 1 the overall non-dominated set of solutions (aggregated over the 20 runs) obtained by both MO-MFCGA and MO-MFEA for all the 10 instances that compose this demanding test case. The visual assessment of these Pareto fronts reveals that indeed, Pareto fronts approximations found by MO-MFCGA dominate those of MO-MFEA, in terms of both convergence and spread. This reinforces our conclusion that MO-MFCGA outperforms MO-MFEA over each test case of our experiments.

B. Analysis of the Genetic Transfer between Tasks

In this last section of our experimentation, we analyze the knowledge transferred between the 10 considered MO-ETSP instances in the most complex test case (TC_10). This study aims to i) analyze the positive genetic transfer among considered tasks; ii) discover which instances can be declared to be synergistically related and why; and iii) empirically quantify the inter-task interactions produced by MO-MFCGA along the 20 executions of the TC_10 test case.

As we have already discussed in [10], our single-objective MFCGA was proven to be especially appropriated for analyzing positive knowledge transfer held during the search. This characteristic is directly inherited by the MO-MFCGA presented in this paper, mainly by virtue of the *local improvement selection* criterion used to retain the best individuals in the cellular grid. As a result of this replacement mechanism, an individual $\mathbf{x}_i'$ of the population is replaced only if any of the generated $\mathbf{x}_i^{\prime,mut}$ or $\mathbf{x}_i^{\prime,cross}$ performs better as per its assigned task (i.e., its skill factor). Therefore, if $\mathbf{x}_i^{\prime,cross}$ replaces $\mathbf{x}_i'$, we can state that a positive transfer has occurred from $\mathbf{x}_j' \in \mathbf{X}_i^{\circledast}$ to $\mathbf{x}_i'$ (we refer to Algorithm 4 and Section III for notation details). In the specific case of the MO-ETSP, a positive genetic contribution of task τ_j towards task τ_i is materialized through the insertion of part of $\mathbf{x}_j'$ into $\mathbf{x}_i'$ by means of the crossover operator.

Bearing the above reasoning in mind, Figure 2 illustrates the amount of positive knowledge transfer occurred between each pair of MO-ETSP tasks along the 20 executions of TC_10 test case. The radius of each blue circle in this figure is proportional to the average amount of times that an individual featuring the skill factor shown in the column has shared some knowledge with a solution whose skill factor is represented in the row. For this reason, the wider the circle is, the more

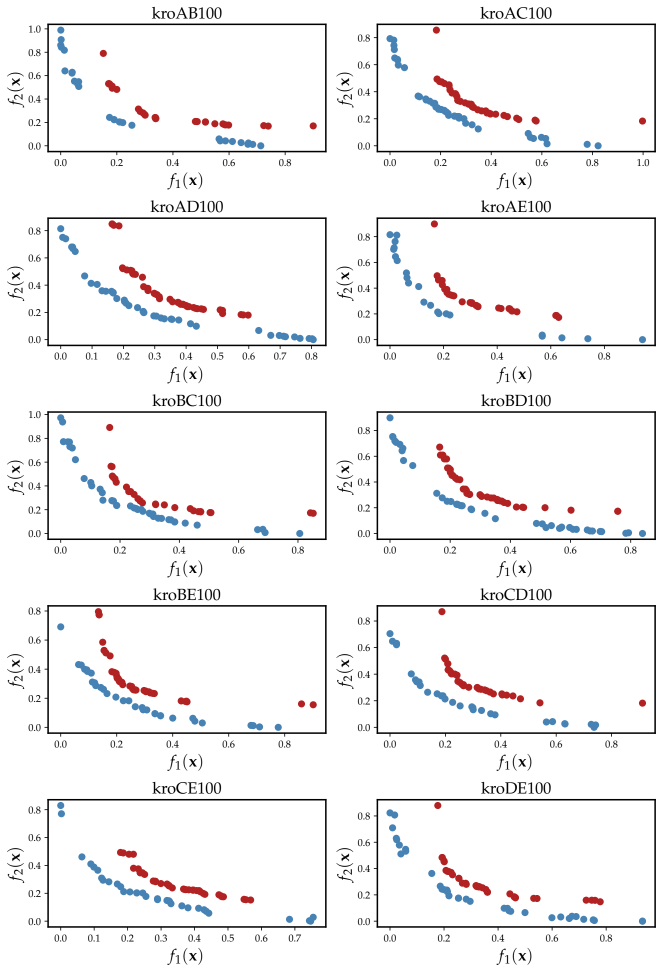

Fig. 1. Pareto Fronts obtained by MO-MFCGA and MO-MFEA in TC_10 test case. ●: MO-MFEA; ●: MO-MFCGA. Objective function values have been scaled to the range $[0, 1]$.

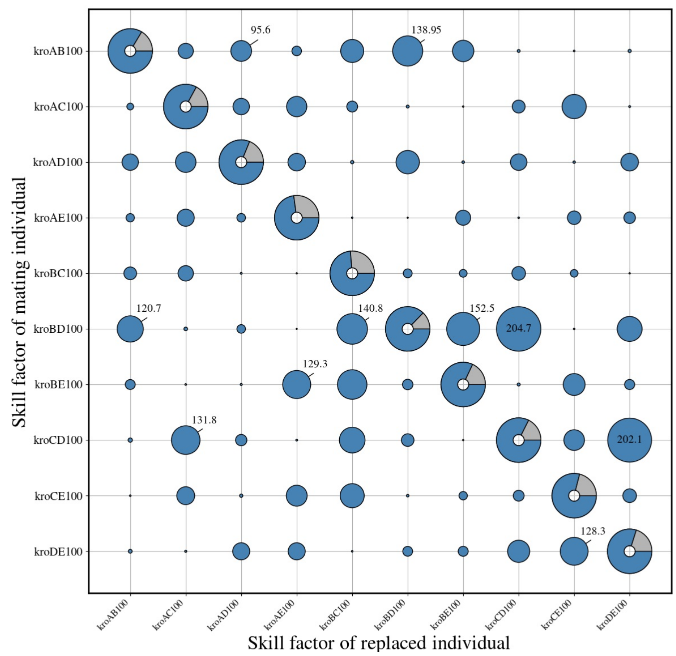

Fig. 2. Average intensities of genetic transfer between MO-ETSP instances registered when tackling the TC_10 test case.

intense the relation between the pair of instances can be thought to be. Circles placed in the diagonal of this matrix plot depict the sum of all inter-task (blue portion) and intra-task exchanges (gray portion), wherein intra-task exchanges stand for those cases in which the genetic transfer has occurred among individuals with the same skill factor.

Several interesting conclusions can be drawn by analyzing this figure. First, we observe that intense synergies exist among MO-ETSP cases that share one of their objectives. In other words, the exchange of knowledge among the pair (kroBD100, kroCD100), for example, is intense, mainly because they share one objective: the minimization of kroD100. Similar statements can be made – to a greater or lesser extent – in other cases with shared MO-ETSP instances, such as (kroAB100,kroBD100), (kroCD100,kroDE100) or (kroBE100,kroBC100). The second conclusion is the fact that, for pairs that do not share any of their compounding MO-ETSP instances, the exchange of knowledge is almost nonexistent. For this reason, the communication between these tasks can be thought to be negative. These two conclusions fully buttress the findings provided by some previously published studies, such as [12] or [6], which state that some kind of *partial domain overlap* should exist among different tasks in order to efficiently harness the transfer of genetic material. Finally, the amount of inter-task exchanges is found to be significantly greater than that of intra-task exchanges, proving the importance of the knowledge transfer for this specific case of study.

V. Conclusions and Future Work

This work has elaborated on the design and performance assessment of a novel approach coined as Multiobjetive Multifactorial Cellular Genetic Algorithm (MO-MFCGA) for dealing with EM scenarios whose tasks are MOPs. This method finds its main inspiration in the multiobjective variant of the well-known MFEA, as well as in core concepts of cellular genetic algorithms. Specifically, MO-MFCGA arranges its population as a grid structure, where different neighborhood relationships can be defined among individuals. These defined neighborhoods impose a degree of locality in the application of the evolutionary crossover operator, allowing for a controlled dispersion – as per the size and structure of the neighborhoods – of good solutions over the grid.

We have empirically measured the performance of our proposed MO-MFCGA by comparing it to that of MO-MFCGA over 10 different multitasking environments (test cases), each consisting of a subset of 10 multiobjective Euclidean traveling salesman problem (MO-ETSP) instances. The experimental outcomes verify that MO-MFCGA largely and consistently outperforms MO-MFEA in terms of the convergence and diversity of the Pareto front approximations discovered for every compounding task of the test cases. Furthermore, an analysis of the inter-task genetic transfer held during the search has revealed that MO-MFCGA effectively promotes knowledge exchange between tasks that are a priori known to be synergistically related as per the construction of the MO-ETSP instances.

In light of the promising results reported in this manuscript, further research lines are planned to extend the conclusions drawn in this preliminary research. First, we intend to adapt the inner search mechanisms featured by MO-MFCGA to other paradigms and real-world problems in the optimization field, with an emphasis on combinatorial and many-objective optimization problems [33], [34]. Furthermore, we will deeply analyze additional search procedures for the MO-MFCGA, as alternative survival strategies, local search methods, and the dynamic adaptation of the parameters of the cellular grid during the search.

References

[1] K. Deb, "Multi-objective optimization," in *Search methodologies*. Springer, 2014, pp. 403–449.

[2] J. Del Ser, E. Osaba, D. Molina, X.-S. Yang, S. Salcedo-Sanz, D. Camacho, S. Das, P. N. Suganthan, C. A. C. Coello, and F. Herrera, "Bio-inspired computation: Where we stand and what's next," *Swarm and Evolutionary Computation*, vol. 48, pp. 220–250, 2019.

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[4] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[5] A. J. Nebro, J. J. Durillo, J. Garcia-Nieto, C. C. Coello, F. Luna, and E. Alba, "SMPSO: A new PSO-based metaheuristic for multi-objective optimization," in *IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM)*. IEEE, 2009, pp. 66–73.

[6] A. Gupta, Y.-S. Ong, and L. Feng, "Insights on transfer optimization: Because experience is the best teacher," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 51–64, 2017.

[7] E. Osaba, A. D. Martinez, and J. Del Ser, "Evolutionary multitask optimization: a methodological overview, challenges and future research directions," *arXiv preprint arXiv:2102.02558*, 2021.

[8] Y.-S. Ong, "Towards evolutionary multitasking: a new paradigm in evolutionary computation," in *Computational Intelligence, Cyber Security and Computational Models*. Springer, 2016, pp. 25–26.

[9] A. Gupta, Y.-S. Ong, L. Feng, and K. C. Tan, "Multiobjective multifactorial optimization in evolutionary multitasking," *IEEE Transactions on Cybernetics*, vol. 47, no. 7, pp. 1652–1665, 2016.

[10] E. Osaba, J. Del Ser, A. D. Martinez, J. L. Lobo, and F. Herrera, "AT-MFCGA: An adaptive transfer-guided multifactorial cellular genetic algorithm for evolutionary multitasking," *Information Sciences*, vol. 570, pp. 577–598, 2021.

[11] B. Manderick, "Fine-grained parallel genetic algorithms," in *3rd International Conference on Genetic Algorithms*, 1989, pp. 428–433.

[12] A. Gupta, Y.-S. Ong, and L. Feng, "Multifactorial evolution: toward evolutionary multitasking," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 343–357, 2015.

[13] E. L. Lawler, J. K. Lenstra, A. R. Kan, and D. B. Shmoys, *The traveling salesman problem: a guided tour of combinatorial optimization*. Wiley New York, 1985, vol. 3.

[14] A. Gupta, Y.-S. Ong, B. Da, L. Feng, and S. D. Handoko, "Landscape synergy in evolutionary multitasking," in *IEEE Congress on Evolutionary Computation (CEC)*, 2016, pp. 3076–3083.

[15] B. Da, Y.-S. Ong, L. Feng, A. K. Qin, A. Gupta, Z. Zhu, C.-K. Ting, K. Tang, and X. Yao, "Evolutionary multitasking for single-objective continuous optimization: Benchmark problems, performance metric, and baseline results," 2017, arXiv:1706.03470.

[16] Y. Yuan, Y.-S. Ong, A. Gupta, P. S. Tan, and H. Xu, "Evolutionary multitasking in permutation-based combinatorial optimization problems: Realization with TSP, QAP, LOP, and JSP," in *IEEE Region 10 Conference (TENCON)*, 2016, pp. 3157–3164.

[17] K. K. Bali, Y.-S. Ong, A. Gupta, and P. S. Tan, "Multifactorial evolutionary algorithm with online transfer parameter estimation: MFEA-II," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 1, pp. 69–83, 2019.

[18] J. Yi, J. Bai, H. He, W. Zhou, and L. Yao, "A multifactorial evolutionary algorithm for multitasking under interval uncertainties," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 5, pp. 908–922, 2020.

[19] Y. Zhou, T. Wang, and X. Peng, "MFEA-IG: A multi-task algorithm for mobile agents path planning," in *IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2020, pp. 1–7.

[20] J. Liu, P. Li, G. Wang, Y. Zha, J. Peng, and G. Xu, "A multitasking electric power dispatch approach with multi-objective multifactorial optimization algorithm," *IEEE Access*, vol. 8, pp. 155 902–155 911, 2020.

[21] C. Yang, J. Ding, Y. Jin, C. Wang, and T. Chai, "Multitasking multiobjective evolutionary operational indices optimization of beneficiation processes," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 3, pp. 1046–1057, 2018.

[22] A. Rauniyar, R. Nath, and P. K. Muhuri, "Multi-factorial evolutionary algorithm based novel solution approach for multi-objective pollution-routing problem," *Computers & Industrial Engineering*, vol. 130, pp. 757–771, 2019.

[23] K. K. Bali, A. Gupta, Y.-S. Ong, and P. S. Tan, "Cognizant multitasking in multiobjective multifactorial evolution: MO-MFEA-II," *IEEE Transactions on Cybernetics*, vol. 51, no. 4, pp. 1784–1796, 2020.

[24] C. Bierwirth, D. C. Mattfeld, and H. Kopfer, "On permutation representations for scheduling problems," in *International Conference on Parallel Problem Solving from Nature*. Springer, 1996, pp. 310–318.

[25] E. Osaba, X.-S. Yang, and J. Del Ser, "Traveling salesman problem: a perspective review of recent research and new results with bio-inspired metaheuristics," *Nature-Inspired Computation and Swarm Intelligence*, pp. 135–164, 2020.

[26] A. Agrawal, N. Ghune, S. Prakash, and M. Ramteke, "Evolutionary algorithm hybridized with local search and intelligent seeding for solving multi-objective Euclidian TSP," *Expert Systems with Applications*, p. 115192, 2021.

[27] I. Khan, M. K. Maiti, and K. Basuli, "Multi-objective traveling salesman problem: an ABC approach," *Applied Intelligence*, vol. 50, no. 11, pp. 3942–3960, 2020.

[28] D. H. Moraes, D. S. Sanches, J. da Silva Rocha, J. M. C. Garbelini, and M. F. Castoldi, "A novel multi-objective evolutionary algorithm based on subpopulations for the bi-objective traveling salesman problem," *Soft Computing*, vol. 23, no. 15, pp. 6157–6168, 2019.

[29] X. Chen, Y. Liu, X. Li, Z. Wang, S. Wang, and C. Gao, "A new evolutionary multiobjective model for traveling salesman problem," *IEEE Access*, vol. 7, pp. 66 964–66 979, 2019.

[30] T. George and T. Amudha, "Genetic algorithm based multi-objective optimization framework to solve traveling salesman problem," in *Advances in Computing and Intelligent Systems*. Springer, 2020, pp. 141–151.

[31] I. Ariyasingha and T. Fernando, "Performance analysis of the multi-objective ant colony optimization algorithms for the traveling salesman problem," *Swarm and Evolutionary Computation*, vol. 23, pp. 11–26, 2015.

[32] A. LaTorre, D. Molina, E. Osaba, J. Del Ser, and F. Herrera, "Fairness in bio-inspired optimization research: A prescription of methodological guidelines for comparing meta-heuristics," *arXiv preprint arXiv:2004.09969*, 2020.

[33] R.-E. Precup, E.-I. Voisan, R.-C. David, E.-L. Hedrea, E. M. Petriu, R.-C. Roman, and A.-I. Szedlak-Stinean, "Nature-inspired optimization algorithms for path planning and fuzzy tracking control of mobile robots," in *Applied Optimization and Swarm Intelligence*. Springer, 2021, pp. 129–148.

[34] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence)*. IEEE, 2008, pp. 2419–2426.