Evolutionary Multitasking Across Multi and Single-Objective Formulations for Improved Problem Solving

Bingshui Da*‡, Abhishek Gupta*, Yew-Soon Ong*, Liang Feng† and Chen Wang‡
*School of Computer Engineering, Nanyang Technological University, Singapore 639798
†College of Computer Science, Chongqing University, China 400044
‡SAP Innovation Center, Singapore, 138602

***Abstract*—Traditionally, single-objective and multi-objective optimization have only considered a single problem in one run. However, the notion of evolutionary multitasking, which aims at solving multiple optimization problems simultaneously, has recently emerged in Evolutionary Computation (EC). It is inspired by the implicit parallelism of population-based search, which attempts to take advantage of implicit genetic transfer in a multitasking environment. According to optimization literature, transforming a single-objective optimization (SOO) problem into a multi-objective optimization (MOO) problem has often been found to remove local optima. Motivated by the aforementioned idea and the concept of multitasking, in this paper, we introduce a new strategy for tackling complex multi-modal problems. In particular, we solve the original (or target) SOO task together with an artificially formulated MOO task in a multitask setting. Therein, the MOO task is expected to provide a useful inductive bias to the search progress of the target SOO task by leveraging on the transferable knowledge shared between them, thereby helping overcome local optima and effectively guiding the population towards more promising regions of the search space.**

I. Introduction

Over the years, the algorithms of evolutionary computation (EC) have become popular tools for solving complex real-world optimization problems. Within EC, Evolutionary algorithms (EAs) are optimization meta-heuristics that are inspired by biological evolution and follow Darwinian principles of *Natural Selection*[1, 2]. Based on a collection of simple rules mimicking natural evolutionary processes, EAs have been demonstrated to be capable of solving nonlinear, multi-modal, and discrete NP-hard problems efficiently [3, 4].

Traditionally, the design of EAs has been focused on efficiently dealing with a single problem at a time. This includes a variety of applications in single-objective optimization (SOO) [1], multi-objective optimization (MOO) [5] and more recently in many-objective optimization [6]. Notably, there has seldom been any effort made towards evolutionary multitasking, *i.e.*, solving multiple optimization problems simultaneously using a single population of evolving individuals. It is only very recently that the notion of evolutionary multitasking has been formalized under the label of *multifactorial optimization* (MFO) [7–10]. In MFO, each task contributes a unique *factor* influencing the evolution of a single population of individuals. Accordingly, the main aim is to harness the underlying commonalities between the tasks to improve performance characteristics of the optimization process.

In [7], in order to efficiently realize the evolutionary multitasking paradigm, a multifactorial evolutionary algorithm (MFEA) has been proposed. The MFEA is inspired by bio-cultural models of *multifactorial inheritance* [11, 12] which suggest that the complex developmental traits among offspring are influenced by the interaction of genetic as well as cultural factors. With this in mind, it is contended that the multiple optimization tasks in a single multitask setting represent multiple blocks of cultural bias coexisting in the same environment. The evolution of a population of individuals in such a *multicultural* environment facilitates the autonomous transfer of knowledge (in the form of encoded genetic material) across the different tasks. As a result, if there happen to exist some form of underlying commonalities or relatedness between tasks, these get spontaneously harnessed, thereby improving convergence characteristics. From the context of machine learning, a similar case study in [13] involved a repertoire of related problems that were created to improve regression modeling using genetic programming. However, to the best of our knowledge, no equivalent concept in the field of *optimization* has yet been thoroughly investigated for the purpose of improved problem solving in complex real-world domains.

An interesting approach to solving complex optimization problems is the reformulation of an SOO problem as a related MOO problem, an idea that was proposed in [14]. According to Knowles *et. al.* [14], the possibility of reducing local optima by the aforementioned process of multiobjectivization can be illustrated using an abstract building block problem. As an experimental case study, the authors applied their algorithm for solving the traveling salesman problem (TSP), achieving some noteworthy results. However, it was also identified in [14] that while the multi-objective reformulation could often achieve performance enhancements, there continued to be several examples in which the SOO approach was still superior.

Inspired by the previous study, in this paper, we aim to further improve optimization performance for complex NP-bard problems by combining an original SOO formulation together with its associated MOO reformulation in a single multitasking environment. The artificially generated MOO task, with hopefully reduced local optima, is expected to act as a *helper task* which aids the search process of the original (or target) problem via the process of implicit genetic transfer. To elaborate, we first decompose the target SOO problem into an associated MOO problem, and then employ the evolutionary multitasking engine to solve both problems in conjunction using the same population of evolving individuals. As a consequence, it is expected that we may successfully leverage upon the unique advantages provided by both approaches (i.e., SOO and MOO), effectively overcoming local optima to converge towards globally optimal solutions more consistently. As a realization of the proposed multitasking paradigm, we continue to use the TSP for our experimental studies in this paper.

The remainder of this paper is organized as follows. In Section 2 we introduce the preliminaries of this paper. This includes a brief overview of multi-objective optimization and a discussion on the basic concepts of evolutionary multitasking. In Section 3, the decomposition strategy employed for reformulating SOO to MOO, as was introduced in [14] in the context of TSPs, is described. In Section 4, we describe the multi and single-objective optimization multifactorial evolutionary algorithm (M&S-MFEA) for the TSP. Thereafter, in Section 5, numerical experiments are carried out showcasing the efficacy of the proposed algorithm. Section 6 concludes the paper with some directions for future work.

II. Preliminaries

A. Multi-objective Optimization

The general mathematical formulation of a multi-objective minimization problem can be stated as follows

$$\min \ (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \dots, f_k(\boldsymbol{x})), \quad (1a)$$
$$s.t. \ \ \boldsymbol{x} \in \boldsymbol{X}. \quad (1b)$$

Here, $\boldsymbol{X}$ represents the set of all feasible solutions, and k is the dimensionality of the objective space. Considering any two solutions $\boldsymbol{x}_1$ and $\boldsymbol{x}_2 \in \boldsymbol{X}$, $\boldsymbol{x}_1$ is said to dominate $\boldsymbol{x}_2$ iff $\forall i \in \{1, 2, \dots, k\}$ $f_i(\boldsymbol{x}_1) \le f_i(\boldsymbol{x}_2)$ and $\exists j \in \{1, 2, \dots, k\}$ such that $f_j(\boldsymbol{x}_1) < f_j(\boldsymbol{x}_2)$. Moreover, a solution $\boldsymbol{x}^*$ is said to be optimal in multi-objective sense (or Pareto optimal) if $\boldsymbol{x}^* \in \boldsymbol{X}$ and there exists no feasible $\boldsymbol{x}$ such that $f(\boldsymbol{x})$ dominates $f(\boldsymbol{x}^*)$.

It can be seen from this description that for all $k > 1$ there will generally exist a set of Pareto optimal solutions, instead of a single optimal solution (as is the case for single objective optimization problems). The image of these solutions in the objective space is said to constitute the *Pareto front* or the *Pareto surface*.

MOO is a research area within multiple criteria decision making [15] where the target is to find the optimum trade-off between a set of objective functions, *i.e.*, $f_1, f_2, \dots, f_k$ in Fig 1. In other words, by MOO one attempts to deduce a favorable balance between two or more conflicting objectives of *the same underlying problem*. Thus, although multiple aspects or objectives are taken into consideration, MOO only deals with a single optimization task during each run.

B. Multifactorial Optimization: Formalizing Evolutionary Multitasking

It is noted that since MOO instances comprise multiple objectives at a time, there may (inaccurately) appear to exist some form of conceptual overlap between them and evolutionary multitasking. Thus, in the discussion that follows, we highlight the distinction between the two paradigms.

Basically, while MOO accounts for multiple objectives of the same optimization task, in contrast, evolutionary multitasking tackles multiple distinct optimization tasks at once. As shown in Fig 1, $F_1, F_2, \dots, F_k$ represent the objective functions of each task separately. The aim of evolutionary multitasking is to get the *best solution* for each of the tasks, rather than finding a desirable trade-off between their objectives. Further, note that one or more tasks in evolutionary multitasking could themselves be an MOO (i.e., any one $F_1, F_2, \dots, F_k$ could be vector-valued), thereby highlighting the greater generality of the paradigm.

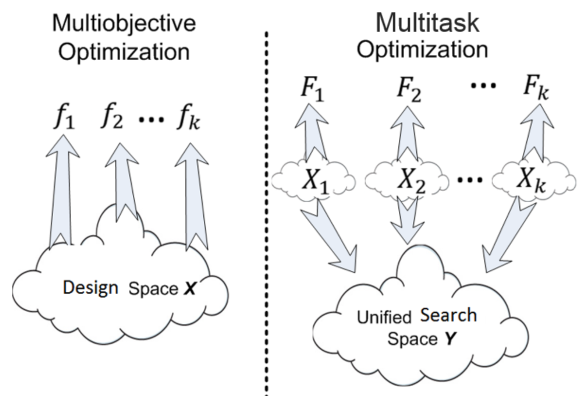


Fig. 1: Distinguishing between evolutionary multitasking and MOO. In MOO, all objective functions typically depend on a common search space. In contrast, each task in evolutionary multitasking has its own search space, thereby making search space unification an important requirement. Furthermore, note that each task in multitasking could itself have multiple objectives.

In a multitask setting with k tasks, the j^{th} task, denoted T_j, has a search space $\boldsymbol{X}_j$ with an objective function $F_j : \boldsymbol{X}_j \to \mathbb{R}$. The goal of such a multitasking instance can be stated mathematically as shown below,

$$deduce \ \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_k\} \quad (2a)$$
$$= \arg\min\{F_1(\boldsymbol{x}), F_2(\boldsymbol{x}), \dots, F_K(\boldsymbol{x})\} \quad (2b)$$
$$s.t. \ \ \boldsymbol{x}_j \in \boldsymbol{X}_j, j = 1, 2, \dots, k. \quad (2c)$$

Next, we will introduce some basic concepts as were proposed in [7]. Consider k distinct optimization tasks are presented to a single evolutionary solver at the same time. Without loss of generality, all optimization tasks are assumed to be minimization problems. In order to develop a suitable algorithm for evolutionary multitasking, it is necessary to conceive a valid measurement to evaluate the fitness of individuals in a multitasking environment, based on the Darwinian principle of natural selection. Note that all the individuals in a population P are encoded in a $unified\ search\ space$ $\boldsymbol{Y}$ encompassing the search spaces of all constitutive tasks, *i.e.*, $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_k$ are all subsumed in $\boldsymbol{Y}$ as shown in Fig 1. Importantly, each individual can be decoded into a task-specific solution representation with respect to all the optimization tasks. Accordingly, the decoded form of p_i can be written as $\boldsymbol{x}_1^i, \boldsymbol{x}_2^i, \ldots, \boldsymbol{x}_k^i,$, where $\boldsymbol{x}_1^i \in \boldsymbol{X}_1$, $\boldsymbol{x}_2^i \in \boldsymbol{X}_2$, $\ldots$, $\boldsymbol{x}_k^i \in \boldsymbol{X}_k$. The main motivation behind the unified search space is to merge the building blocks corresponding to different tasks into a unified pool of genetic material [4], thereby allowing the EA to process them in parallel and facilitate the implicit genetic transfer to kick in.

The following pair of metrics are defined for every individual p_i, where $i \in 1, 2, \ldots, |P|$, in a multitasking environment comprising scalar-valued optimization tasks only, i.e., it is assumed *temporarily* that $F_1, F_2, \ldots, F_k$ are all scalar.

- *Factorial Rank*: The *factorial rank* r_{ij} of p_i on task T_j is simply the index of p_i in the list of population members sorted in ascending order with respect to F_j.
- *Scalar Fitness*: The list of *factorial ranks* $\{r_{i1}, r_{i2}, \cdots, r_{iK}\}$ of an individual p_i is reduced to a *scalar fitness* φ_i based on its best rank over all tasks; *i.e.* $\varphi_i = 1/\min\{r_{i1}, r_{i2}, \cdots, r_{iK}\}$.

The comparison between individuals are carried out simply based on the scalar fitness. For example, individual p_a is considered to dominate individual p_b in *multifactorial sense* simply if $\phi_a > \phi_b$. Another important property of the aforementioned definitions and comparison procedure is that if individual p^* maps to the global optimum of any one task, then $\varphi^* \geq \varphi_i$, for all $i \in 1, 2, \ldots, |P|$. In other words, *multifactorial optimality* is guaranteed if convergence to a global optimum is achieved.

- *Multifactorial Optimality*: An individual p^*, with a list of objective values$\{f_1^*, f_2^*, \ldots, f_k^*\}$, is considered *optimum* in multifactorial sense iff $\exists j \in \{1, 2, \ldots, K\}$ such that $f_j^* \leq F_j(\boldsymbol{x}_j)$, for all feasible $\boldsymbol{x}_j \in \boldsymbol{X}_j$.

III. Creating MOO Helper Tasks for Evolutionary Multitasking: A Traveling Salesman Problem Exemplar

In this part, we briefly introduce the classical TSP and the algorithm proposed in [14] for reformulating it as a multi-objective optimization problem. The artificially generated MOO task brings to the table some unique properties which when combined with the target SOO task in a multitask setting can be expected to provide improved convergence characteristics towards globally optimum solutions.

The TSP is a classical combinatorial optimization problem in the field of computer science. It consists of a set of N cities $c_1, c_2, \ldots, c_N$ and an associated $N \times N$ distance matrix M representing the distances between arbitrary two cities, *i.e.*, $M(c_1, c_2)$ is the distance from c_1 to c_2. In this paper, we only focus on solving symmetric TSPs, i.e., in which the distance matrix M is a symmetric matrix. The objective, quite straightforwardly, is to find a Hamiltonian path (a circular path visiting every city exactly once) with the smallest possible distance [16, 17]. TSP can be formulated as an integer linear program, but it has been shown to be NP-hard. For further descriptions, please refer to [18]. Using $\pi = (\pi_1, \pi_2, \ldots, \pi_N)$ as a permutation of $(1, 2, \ldots, N)$ denoting the sequence in which the salesman must visit cities in the round trip, the cost, i.e., the total distance associated with the tour, can be calculated as

$$D(\pi) = \sum_{i=1}^{N} M(c_{\pi[i]}, c_{\pi[i \oplus 1]}) \tag{3a}$$

$$where \quad i \oplus 1 = \begin{cases} i+1 & if\ i < N \\ 1 & if\ i = N \end{cases} \tag{3b}$$

According to Knowles *et. al.* [14], the transformed MOO version of the original TSP is formulated by decomposing the original problem. They simply segment the original TSP into two distinct sub-tours, each to be minimized, and the objectives of each sub-tour are defined using the following formula:

$$f_1(\pi, a, b) = \sum_{i=\pi^{-1}[a]}^{\pi^{-1}[b]} M(c_{\pi[i]}, c_{\pi[i]\oplus 1}) \tag{4a}$$

$$f_2(\pi, a, b) = \sum_{i=\pi^{-1}[b]}^{N} M(c_{\pi[i]}, c_{\pi[i]\oplus 1}) \tag{4b}$$

$$+ \sum_{i=1}^{\pi^{-1}[a]-1} M(c_{\pi[i]}, c_{\pi[i]\oplus 1}) \tag{4c}$$

where $\pi^{-1}[x]$ denotes the position of x in π, and the two cities a and b are specified in priori. If $\pi^{-1}[a] < \pi^{-1}[b]$, a and b are swapped. Noticing that the sum of the two objectives is the same as the objective of the original single-objective TSP, the original optima are ensured to become a Pareto-optimum under the new set of objectives. The choice of the city pair a and b are selected arbitrarily, which was partially investigated in [14].

The experiment results in [14] indicated that *in some cases* tackling the TSP as a MOO problem can often reduce local optima and facilitate improved optimization. However, it was also noted that in some other cases, SOO remained the superior approach. Thus, the overall results lead to the contention that by somehow enabling the search characteristics of SOO and MOO to act in concert, high quality solutions may be achieved more consistently. An elegant way of achieving the above is to allow evolutionary multitasking to take over and autonomously harness the complementarities between the two approaches.

IV. M&S-MFEA: The Multifactorial Evolutionary Algorithm for Multi and Single-Objective Optimization

The principal hypothesis of this paper is that by combining a complex optimization task together with a closely related, but often simpler, task in a single multitasking environment, it may be possible to achieve substantial performance improvements by harnessing the unique search benefits provided by both tasks. Since the MOO reformulation of the TSP is oftentimes successful at removing several local optima, it may act as a simpler *helper task* that aids the optimization search for the standard SOO formulation of the NP-hard TSP. However, it is noted that combining SOO and MOO during multitasking leads to some algorithmic challenges, which will be resolved hereafter.

When dealing with SOO and MOO tasks in conjunction during multitasking, a matter of concern is the prescription of a meaningful *factorial rank* of an individual with respect to a constitutive MOO task. Thus, we begin this section by first describing a simple approach for achieving the above. Thereafter, based on the prescribed ordering scheme, we present details of the new M&S-MFEA.

For the sake of brevity, the concepts of Non-dominated Front (NF) and Crowding Distance (CD) in constrained multi-objective optimization are directly adopted from the literature [19]. We do not elaborate on their interpretations in this paper since these concepts have been well-established over several years of multi-objective optimization research.

Notice that for ranking individuals in a population, it is sufficient to define a preference relationship between two individuals and show that the binary relationship satisfies the properties of irreflexivity, asymmetry, and transitivity. To this end, let us consider a pair of individuals p_1 and p_2 with non-dominated fronts NF_1 and NF_2 and crowding distances CD_1 and CD_2, respectively. With the aim of facilitating a diverse distribution of points along the PF, we prescribe individual p_2 to be preferred over p_1 (i.e., $p_2 \succ p_1$) if any one of the following conditions holds:

- $\text{NF}_2 < \text{NF}_1$
- $\text{NF}_2 = \text{NF}_1$ and $\text{CD}_2 > \text{CD}_1$

For the aforementioned preference relationship, the satisfaction of the necessary properties can be simply shown as below.

- Property 1 (*Irreflexivity*): $p_i \nsucc p_i$, for all $p_i \in P$

Proof: Suppose $p_i \succ p_i$. Then, either (a) $\text{NF}_i < \text{NF}_i$ or (b) $\text{NF}_i = \text{NF}_i$ and $\text{CD}_i > \text{CD}_i$. However, since $\text{NF}_i = \text{NF}_i$ and $\text{CD}_i = \text{CD}_i$, the supposition leads to a contradiction.

- Property 2 (*Asymmetry*): If two individuals p_1 and p_2 satisfy $p_2 \succ p_1$, then $p_1 \nsucc p_2$.

Proof: Suppose $p1 \succ p2$. Then, either (a) $\text{NF}_1 < \text{NF}_2$ or (b) $\text{NF}_1 = \text{NF}_2$ and $\text{CD}_1 > \text{CD}_2$. However, according to $p_2 \succ p_1$, we have either (a) $\text{NF}_2 < \text{NF}_1$ or (b) $\text{NF}_2 = \text{NF}_1$ and $\text{CD}_2 > \text{CD}_1$. Thus, the supposition leads to a contradiction.

- Property 3 (*Transitivity*): If $p_2 \succ p_1$ and $p_3 \succ p_2$, then it must also be the case that $p_3 \succ p_1$.

Proof: We are given that p_2 is preferred over p_1 according to the conditions stated earlier. Since $p_3 \succ p_2$ we also have either (a) $\text{NF}_3 < \text{NF}_2$ or (b) $\text{NF}_3 = \text{NF}_2$ and $\text{CD}_3 > \text{CD}_2$. If condition (a) is true then it implies $\text{NF}_3 < \text{NF}_1$. If condition (b) is true then it implies either (b.1) $\text{NF}_3 < \text{NF}_1$ or (b.2) $\text{NF}_3 = \text{NF}_1$ and $\text{CD}_3 > \text{CD}_1$. Therefore, if p_2 is preferred over p_1, and p_3 is preferred over p_2, then p_3 must also be preferred over p_1.

Accordingly, it is clear that the *factorial ranks* of individuals for any MOO task can easily be assigned by ranking them according to the two aforesaid conditions. The workflow of the M&S-MFEA, which utilizes this scheme in the multitasking approach to solving TSPs, is reported in Algorithm 1. Note that since we only deal with a single kind of optimization task, we can utilize a standard domain specific unification scheme (which implies that no added steps for decoding individuals from the common genotype space to the phenotype space are necessary here). To be precise, we use a *permutation based* description of $\boldsymbol{Y}$ in Fig 1 to represent candidate TSP solutions.

Algorithm 1 Workflow of the proposed M&S-MFEA for TSPs

Randomly select a city pair a and b for MOO reformulation of TSP instance.
Generate an initial population of TSP solutions and store it in $current\text{-}pop$.
Evaluate every solution with respect to the target SOO and the helper MOO tasks.
while stopping conditions are not satisfied **do**
  Apply genetic operators (crossover + mutation) on $current\text{-}pop$ to generate an $offspring\text{-}pop$.
  Evaluate the individuals in $offspring\text{-}pop$ with respect to the target SOO and the helper MOO tasks.[1]
  Concatenate $current\text{-}pop$ and $offspring\text{-}pop$ to form an $intermediate\text{-}pop$.
  Compute *factorial ranks* for all individuals in $intermediate\text{-}pop$. Standard sorting is used for SOO and (NF, CD) is used for MOO.
  Compute *scalar fitness* for all individuals in $intermediate\text{-}pop$.
  Select $fittest$ individuals from $intermediate\text{-}pop$ to survive into the next generation and form the new $current\text{-}pop$.
end while

The fundamental outcome of the mechanisms of the M&S-MFEA is that the population of evolving individuals is divided into two parts, each experiencing a distinctive *selection pressure*. While one part caters to the target SOO task, the other part caters to the artificially generated MOO task. As a result of the combined effect of the two unique selection pressures, the population is expected to simultaneously harness the search characteristics of SOO and MOO, thereby potentially providing solutions that are at least as good as those obtained by either approach independently.

V. Computational Studies

In this section, we will present our results for a selection of problems from the TSPlib [20].

[1]Note that, in the case of TSP, evaluation for one task directly provides objective values for the other task. Thus, significant added function evaluations are not entailed.

To demonstrate the competence of our proposed M&S-MFEA, we compare its performance against a standard single-objective evolutionary algorithm (SOEA) and a standard multi-objective evolutionary algorithm (MOEA). Note that the MOEA follows the basic steps of the popular NSGA-II procedure [19]. For fairness of comparison, we use the same permutation-based encoding scheme, crossover operator, mutation operator, and local search operator in the M&S-MFEA, SOEA and the MOEA. In particular, the variation operators are *order crossover* (OX) [21] and random swap for mutation. In order to facilitate convergence to high quality solutions, a local solution refinement step is applied to every individual. In our current implementation, we use 2-opt [22] as the preferred local search algorithm (with a maximum of 50 local search moves per individual). Finally, the parameter settings are also kept consistent across all algorithms, i.e., a population size of 100 individuals is employed, the probability of mutation is 0.2, and the termination condition is set to the completion of 150 generations.

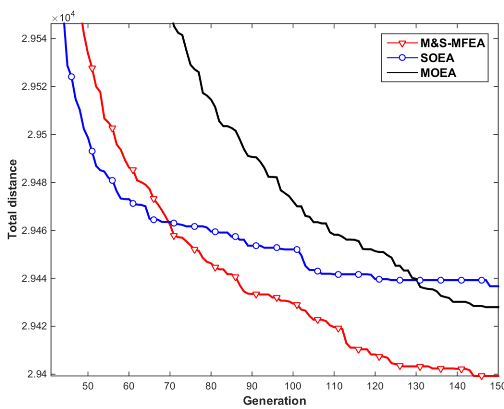


Fig. 2: Convergence trends for M&S-MFEA, SOEA and MOEA on TSP kroA200

For the purpose of qualitatively understanding the outcomes of evolutionary multitasking, we first consider the kroA200 instance from TSPlib [20]. Note that the results presented are averages across 30 independents runs of the solvers. As per [20], the best known result for the TSP instance kroA200 is 29368. To this end, the averaged outputs obtained by the M&S-MFEA, SOEA and MOEA are 29399.3, 29436.7, 29428.0, respectively. Interestingly, the M&S-MFEA can achieve the best known result on 7 out of the 30 runs, whereas the SOEA shows the tendency of getting continuously trapped at a local optimum. From the convergence trends in Fig 2, we observe that the SOEA converged as rapidly as the M&S-MFEA during the initial stages of the optimization process (i.e., during the first 50 generations), due to focused search on the original task. However, during the later stages of evolution, while the M&S-MFEA continues to successfully explore promising regions of the search space, SOEA gets stuck. This is because the MOO helper task aids in diversifying the population in the evolutionary multitasking setup by acting as an additional source of good quality genetic material. Comparing the trends of M&S-MFEA and MOEA, the former is found to converge faster because it autonomously combines the salient features of SOO and MOO (by the process of implicit genetic transfer), simultaneously benefiting from the focused search as well as the increased diversity.

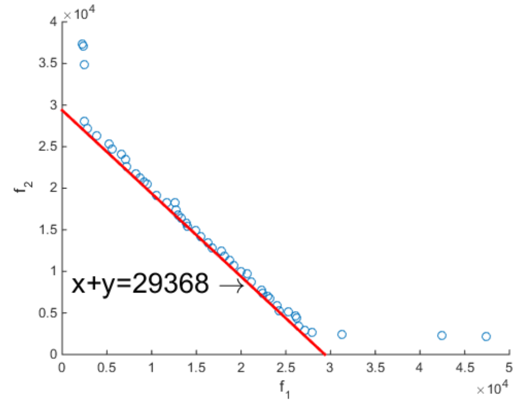


Fig. 3: Approximate Pareto front of the MOO helper task in instance kroA200, as obtained using M&S-MFEA

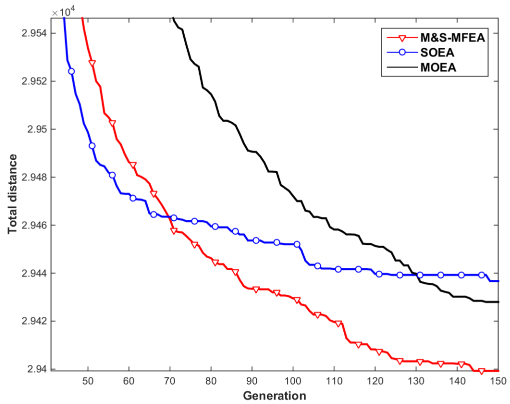
Fig 3 shows the approximate Pareto optimal front of the helper task (the MOO reformulation) obtained in the M&S-MFEA for instance kroA200. Therein, we find that the individuals (mapped to the objective space) lie more-or-less along a straight line given by the equation $f_1 + f_2 \approx 29368$. This condition demonstrates that the MOO helper task is indeed helpful in maintaining a diverse set of individuals in a promising region of the search space, where the overall objective value is at least close to the best known for kroA200. The fact that the M&S-MFEA effectively exploits this phenomenon has already been seen in Fig 2.

We show similar convergence trends in Figs 4a, 5 and 6 for instances kroA150, kroB200 and pr226, as well as the approximate Pareto optimal front of the helper task obtained in the M&S-MFEA for instance kroA150 in Fig 4b. Notice that in the majority of cases the M&S-MFEA shows notably superior performance to the SOEA and the MOEA.

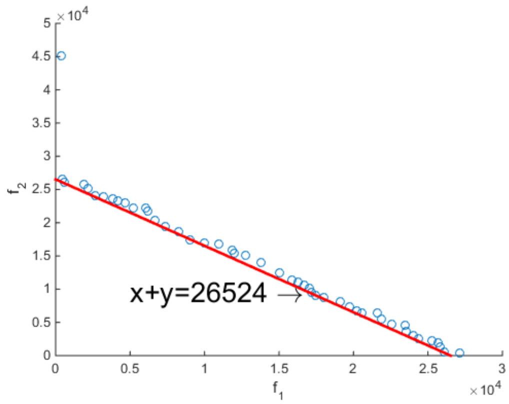
In Table I we present the final results (mean travel time and standard deviation) obtained for 21 different TSP instances averaged over 30 runs. Among these, in 18 out of the 21 cases, M&S-MFEA performs strictly better than SOEA. Further, in comparison to MOEA, M&S-MFEA performs at least as good on 16 out of the 21 instances, with the performance being strictly better in 14 cases. In summary, the numerical results highlight the global search efficacy and robustness achievable by the proposed evolutionary multitasking approach.

VI. Conclusion

In this paper, we have introduced a generic new strategy for tackling complex optimization problems. The efficacy of the proposal is demonstrated on the NP-hard TSP as a test case study. To elaborate, our approach involves solving a



(a) Convergence trends for M&S-MFEA, SOEA and MOEA on TSP kroA200



(b) Approximate Pareto front of the MOO helper task in instance kroA150, as obtained using M&S-MFEA

Fig. 4: Performances on TSP kroA200

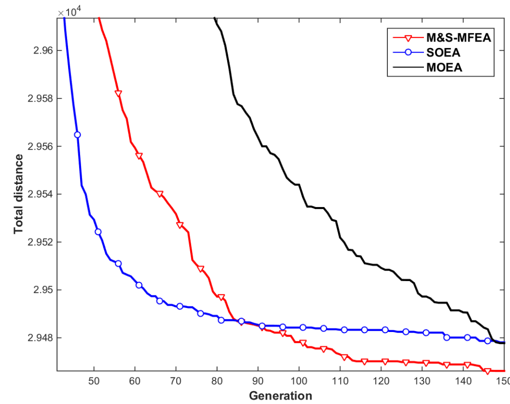


Fig. 5: Convergence trends for M&S-MFEA, SOEA and MOEA on TSP kroB200

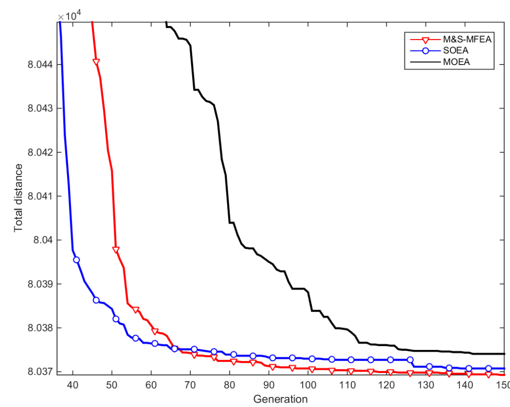


Fig. 6: Convergence trends for M&S-MFEA, SOEA and MOEA on TSP pr226

target single-objective optimization (SOO) task in conjunction with a closely related (but artificially generated) multi-objective optimization (MOO) task in the form of evolutionary multitasking. The motivation behind the proposal is that the associated MOO formulation can often act as a *helper task* that aids the optimization performance by leveraging upon the phenomenon of implicit genetic transfer. In particular, the number of local optima is known to be reduced in the MOO reformulation of the TSP. Thus, by allowing evolutionary multitasking to autonomously harness the complementarities between the SOO and MOO tasks, significant performance improvements are likely to be achieved. To this end, we have carried out computational experiments on a variety of TSP instances from the TSPlib [20]. The results verify our claims by depicting consistently superior solutions achieved by the multitasking approach.

There are several directions for future research extension of this work. For one, the present paper focuses on a specific domain of application, namely, the TSP. However, there are a variety of complex optimization problem domains of practical interest, in discrete as well as continuous optimization, where similar methods can be of significant use. A noteworthy application is in complex engineering design exercises where related design (optimization) tasks occur commonly. While the general practice is to invoke manual adaptation of knowledge from one task to another, evolutionary multitasking opens up possibilities for completely autonomous knowledge transfer (in the form of encoded genetic material) across tasks, thereby enabling significantly accelerated design stages.

References

[1] T. Bäck, U. Hammel, and H.-P. Schwefel, "Evolutionary computation: Comments on the history and current state," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 3–17, 1997.

[2] R. Dawkins, *The selfish gene*. Oxford university press, 2006, no. 199.

[3] Y.-S. Ong, M. H. Lim, and X. Chen, "Research frontier-memetic computationpast, present & future," *IEEE Comput. Intell. Mag*, vol. 5, no. 2, p. 24, 2010.

[4] L. Feng, Y.-S. Ong, M.-H. Lim, and I. Tsang, "Memetic search with interdomain learning: A realization between

TABLE I: Numerical experiment results on problems from TSPlib. Superior performance is highlighted in bold.

#	Problem	Best known	M&S-MFEA	SOEA	MOEA
1	a280	2579	$\mathbf{2583.0 \pm 4.9}$	2583.4 ± 7.5	2586.7 ± 6.2
2	bier127	118282	$\mathbf{118322.9 \pm 125.8}$	118433.9 ± 196.1	118390.3 ± 126.9
3	ch130	6110	6118.2 ± 9.0	6128.4 ± 16.3	$\mathbf{6115.6 \pm 7.7}$
4	ch150	6528	6530.2 ± 6.1	6536.4 ± 11.3	$\mathbf{6528.9 \pm 2.6}$
5	d198	15780	$\mathbf{15790.2 \pm 9.8}$	15794.7 ± 12.8	15812.5 ± 23.8
6	gil262	2378	$\mathbf{2385.1} \pm 6.1$	2389.2 ± 10.2	$2386 \pm \mathbf{4.3}$
7	kroA150	26524	$\mathbf{26524.4 \pm 0.8}$	26538.5 ± 45.0	26527.3 ± 12.2
8	kroA200	29368	$\mathbf{29399.3 \pm 35.7}$	29436.7 ± 60.2	29428 ± 75.5
9	kroB150	26130	$\mathbf{26132.3 \pm 3.4}$	26146.8 ± 29.4	26137.4 ± 12.5
10	kroB200	29437	$\mathbf{29466.2} \pm 45.0$	29478.1 ± 71.0	$29477.8 \pm \mathbf{41.7}$
11	pr124	59030	$\mathbf{59030 \pm 0}$	59033.1 ± 11.7	$\mathbf{59030 \pm 0}$
12	pr136	96772	$\mathbf{96828.7 \pm 81.8}$	96895.3 ± 127.5	96859.1 ± 91.1
13	pr144	58537	$\mathbf{58537 \pm 0}$	$\mathbf{58537 \pm 0}$	$\mathbf{58537 \pm 0}$
14	pr152	73682	73695.6 ± 41.5	73736.4 ± 67.8	$\mathbf{73686.7 \pm 24.8}$
15	pr226	80369	$\mathbf{80369.3 \pm 1.0}$	80370.7 ± 4.5	80374.1 ± 7.8
16	pr264	49135	49167.6 ± 70.7	49186.7 ± 91.8	$\mathbf{49166.8 \pm 34.4}$
17	pr299	48191	$48403.3 \pm \mathbf{117.1}$	$\mathbf{48400.3} \pm 133.6$	48535.6 ± 123.1
18	rat195	2323	2334.8 ± 5.9	2336.9 ± 9.2	$\mathbf{2334.7 \pm 5.5}$
19	ts225	126643	$\mathbf{126651.3 \pm 33.6}$	126673.6 ± 63.8	126763.9 ± 91.0
20	tsp225	3916	3948.3 ± 16.6	$\mathbf{3946.4 \pm 14.2}$	3950.7 ± 14.7
21	u159	42080	$\mathbf{42082.9 \pm 16.1}$	42143.2 ± 128.6	42090.5 ± 57.7

CVRP and CARP," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 644–658, Oct 2015.

[5] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evol. Comput.*, vol. 3, no. 1, pp. 1–16, 1995.

[6] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review." in *IEEE C EVOL COMPUTAT*. Citeseer, 2008, pp. 2419–2426.

[7] A. Gupta, Y.-S. Ong, and L. Feng, "Multifactorial Evolution: Towards Evolutionary Multitasking," *IEEE Trans. Evol. Comput.*, 2015.

[8] A. Gupta, Y.-S. Ong, L. Feng, and K. C. Tan, "Multiobjective multifactorial optimization in evolutionary multitasking," *IEEE Trans. Cybern., accepted*.

[9] Y.-S. Ong and A. Gupta, "Evolutionary multitasking: A computer science view of cognitive multitasking," *Cognit. Comput.*, pp. 1–18.

[10] A. Gupta, J. Mańdziuk, and Y.-S. Ong, "Evolutionary multitasking in bi-level optimization," *Complex & Intelligent Systems*, vol. 1, no. 1-4, pp. 83–95, 2015.

[11] J. Rice, C. Cloninger, and T. Reich, "Multifactorial inheritance with cultural transmission and assortative mating. I. Description and basic properties of the unitary models." *Am. J. Hum. Genet.*, vol. 30, no. 6, p. 618, 1978.

[12] C. R. Cloninger, J. Rice, and T. Reich, "Multifactorial inheritance with cultural transmission and assortative mating. II. a general model of combined polygenic and cultural inheritance." *Am. J. Hum. Genet.*, vol. 31, no. 2, p. 176, 1979.

[13] K. Krawiec and B. Wieloch, "Automatic generation and exploitation of related problems in genetic programming," in *IEEE C EVOL COMPUTAT*. IEEE, 2010, pp. 1–8.

[14] J. D. Knowles, R. A. Watson, and D. W. Corne, "Reducing local optima in single-objective problems by multi-objectivization," in *Evol. multi-criterion optimization*. Springer, 2001, pp. 269–283.

[15] M. Zeleny and J. L. Cochrane, *Multiple criteria decision making*. University of South Carolina Press, 1973.

[16] M. T. Jensen, "Helper-objectives: Using multi-objective evolutionary algorithms for single-objective optimisation," *J. Math. Model. Algorithms*, vol. 3, no. 4, pp. 323–347, 2005.

[17] M. Jähne, X. Li, and J. Branke, "Evolutionary algorithms and multi-objectivization for the travelling salesman problem," in *Genet. and Evol. Comput. Conf.* ACM, 2009, pp. 595–602.

[18] D. L. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook, "The traveling salesman problem: A computational study," *AMC*, vol. 10, p. 12.

[19] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[20] G. Reinelt, "TSPLIB-A traveling salesman problem library," *ORSA J COMPUT*, vol. 3, no. 4, pp. 376–384, 1991.

[21] B. Fox and M. McMahon, "Genetic operators for sequencing problems," *FOGA*, vol. 1, pp. 284–009, 1990.

[22] G. A. Croes, "A method for solving traveling-salesman problems," *Oper. Res.*, vol. 6, no. 6, pp. 791–812, 1958.