Evolutionary Algorithms and Multi-Objectivization for the Travelling Salesman Problem

Martin Jähne
Fraunhofer Institute for Autonomous Intelligent Systems
Sankt Augustin, Germany
martin.jaehne@iais.fraunhofer.de

Xiaodong Li
School of Computer Science and Information Technology
RMIT University
Melbourne, Australia
xiaodong.li@rmit.edu.au

Jürgen Branke
Institute AIFB
Universität Karlsruhe (TH)
Karlsruhe, Germany
branke@aifb.uni-karlsruhe.de

ABSTRACT

This paper studies the multi-objectivization of single-objective optimization problems (SOOP) using evolutionary multi-objective algorithms (EMOAs). In contrast to the single-objective case, diversity can be introduced by the multi-objective view of the algorithm and the dynamic use of objectives. Using the travelling salesman problem as an example we illustrate that two basic approaches, a) the addition of new objectives to the existing problem and b) the decomposition of the primary objective into sub-objectives, can improve performance compared to a single-objective genetic algorithm when objectives are used dynamically. Based on decomposition we propose the concept "Multi-Objectivization via Segmentation" (MOS), at which the original problem is reassembled. Experiments reveal that this new strategy clearly outperforms both the traditional genetic algorithm (GA) and the algorithms based on existing multi-objective approaches even without changing objectives.

Categories and Subject Descriptors

G.1.6 [**Global Optimization**]: Multi-objective otpimization – multi-objectivization

General Terms

Algorithms, Performance

Keywords

Multi-Objective Optimization, Multi-Objectivization, Travelling Salesman Problem, Genetic Algorithms

1. INTRODUCTION

Over the last decade there has been a great deal of research that focused on the application of evolutionary algorithms (EAs) for multi-objective optimization (EMO)[5, 2]. Multi-objective EAs are algorithms capable of optimising several different objectives at the same time, and can identify a set of solutions representing non-dominated trade-offs between conflicting objectives. There has been intense research activity on developing algorithms that can find good approximations of the Pareto-optimal set and on their application to real-world multi-objective problems (MOOP).

However, there have been very few works dealing with the application of multi-objective optimization to single-objective problems (SOOP) [11, 10, 1, 12, 14, 9, 8]. Recent research has demonstrated that such a multi-objectivization of the original problem, as introduced by Knowles et al. [11], can lead to accelerated search performance. It has been shown both theoretically and practically how difficult problems in single-objective optimization can be overcome and the effect of getting trapped in local optima can be minimised [11, 10, 1, 12, 14, 9, 8].

Previous studies concentrated on two basic approaches of multi-objectivization, namely the decomposition of the original objective and the addition of new objectives. The effectiveness of algorithms based on decomposition to facilitate improved search was demonstrated in [11, 14, 9]. Adding new objectives, on the other hand, can make search harder [5, 2, 1]. Brockhoff et al. [1] recently presented general theoretical results for this approach, showing that it may have beneficial or detrimental effects on the runtime of a given problem. They cited Jensen [10] and Knowles et al. [11] as empirical examples that multi-objectivization can reduce the computational cost of the optimization process. The work of Knowles et al. [11] is based on the approach of decomposing the original problem. Jensen on the other hand realizes the concept of additional objectives. In order to demonstrate that multi-objectivization is beneficial they compared the performance of a single-objective algorithm to its multi-objective versions on single-objective problems. So far, a comparison between the two concepts using the same algorithm and problem has not been done. Especially possible benefits and drawbacks provided by the application of evolutionary multi-objective algorithms to single-objective real-world problems have not been explored. This paper addresses this issue using the travelling salesman problem (TSP) as an example. As well as comparing related approaches of Jensen [10] and Knowles et al. [11] to a traditional GA, a comparative evaluation and analysis of the different multi-objective algorithms will be provided.

The Jensen and Knowles' approaches were already criti-

GECCO'09, July 8–12, 2009, Montréal, Canada

cised due to the problem of city pair choices used in decomposition and an unacceptable shift away of the search focus when additional objectives are used, respectively. To overcome these issues we propose a new method of decomposition. In this new approach the helper-objectives introduced by Jensen [10] are used for decomposition. That way, objectives are nested and consist of overlapping segments. This hybridized approach "Multi-Objectivization via Segmentation" (MOS), provides the opportunity to use different segmentation schemes. Three such schemes are proposed in this paper.

In experiments on several problem instances of the TSP, the influences of different ways of multi-objectivization on search and the resulting performances will be analyzed.

The remainder of the paper is structured as follows: Section 2 discusses related approaches of multi-objectivization in TSP, and introduces the new concept MOS. Section 3 describes the algorithms that were used in our experiments, and Section 4 presents the corresponding results. Section 5 provides the concluding remarks.

2. MULTI-OBJECTIVIZATION IN TSP

2.1 TSP

The travelling salesman problem (TSP) is a classical combinatorial optimisation problem. It consists of a set of N cities c_1,..., c_N and an associated N x N distance matrix M. The entries in M represent the distances between the cities, so $M(c_1, c_2)$ is the distance from c_1 to c_2, where $M(c_1, c_2) = M(c_2, c_1)$. The objective is to find a Hamiltonian path (a circular path visiting each city exactly once) with the smallest possible total distance. If $\pi = (\pi_1, \pi_2, ..., \pi_N)$ is a permutation of (1, 2,...,N) representing the tour of the cities, then the distance associated with the tour can be calculated as

$$D(\pi) = \sum_{i=1}^{N} M(c_{\pi[i]}, c_{\pi[i\oplus 1]})$$
$$where \quad i \oplus 1 = \begin{cases} i+1 & if \quad i < N, \\ 1 & if \quad i = N. \end{cases} \tag{1}$$

Figure 1 shows an exemplary fitness calculation of the primary objectitve (PO) using a $N = 5$ cities tour. For a good introduction to TSP, see [3]. The problem instances used in this work are all Euclidean instances in the plane. They are all available for download at the TSPLIB website [1] or the travelling salesman website at Georgia Tech [2].

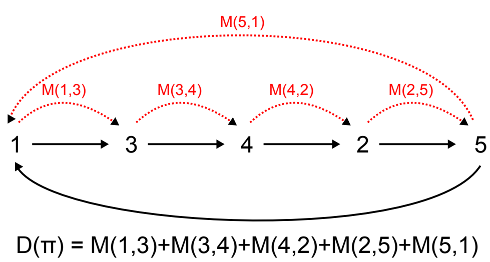

Figure 1: SO Case: fitness calculation

[1] http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/
[2] http://www.tsp.gatech.edu/

2.2 Related Approaches

2.2.1 *Multi-Objectivization Via Decomposition*

Knowles et al. [11] introduced a way to multi-objectivise the TSP using the approach of decomposition of the original objective. In order to use a method that is exemplary for the TSP class they have chosen to simply divide the problem into two distinct sub-tours, each to be minimized. The sub-tours are defined by two cities, and the two-objective formulation of the TSP becomes:

$$f_1(\pi, a, b) = \sum_{i=\pi^{-1}[a]}^{\pi^{-1}[b]-1} M(c_{\pi[i]}, c_{\pi[i\oplus 1]})$$
$$f_2(\pi, a, b) = \sum_{i=\pi^{-1}[b]}^{N} M(c_{\pi[i]}, c_{\pi[i\oplus 1]}) + \sum_{i=1}^{\pi^{-1}[a]-1} M(c_{\pi[i]}, c_{\pi[i\oplus 1]}) \tag{2}$$

where $\pi^{-1}[x]$ denotes the position of x in π and a and b are parameters (cities specified a priori) defining f_1 and f_2, and if $\pi^{-1}[b] < \pi^{-1}[a]$ they are swapped. It is intended that a and b are chosen arbitrarily. Notice that the sum of the two objectives is the same as the quantity to be minimized in the single-objective TSP. This ensures that each of the original optima becomes a Pareto-optimum under the new set of objectives, as required by their definition of multi-objectivization [11]. Figure 2 shows an example of how the two objectives are calculated.

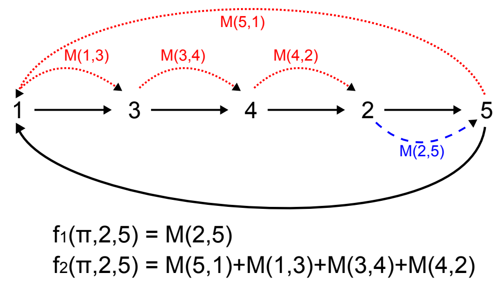

Figure 2: MO(Decomposition): fitness calculation

This decomposition method [11] was criticised by Jensen [10]. He found these new objectives had a weakness for symmetric problems. He argues that *"given two cities a and b defining f_1 and f_2, two different representations of the same tour can be Pareto-incomparable, with two different values for f_1 and f_2. Thus, despite representing the same tour, generally π_1 and π_2 will be given different objectives, and will be non-identical, and incomparable in the Pareto-sense" [10].* However, in [11], the two cities to split the given tour are chosen, where the cities are swapped if the position of city b is smaller than the position of city a in the tour. That way, different fitness values for identical tours are prevented. Furthermore, when duplicates are avoided, it does not matter at all.

The choice of city pairs to be used in the multi-objective algorithms was partially investigated by Knowles et al. [11]. Their results reveal the choice of city pairs influences the performance of the algorithm. Depending on the cities chosen, differently sized objectives are the consequence. If the two cities are direct neighbours in the optimal tour, we even obtain the single-objective case. Furthermore, depending on specific problem characteristics such a choice can be very destructive due to interdependency of the problem components.

2.2.2 *Using Helper-Objectives (HO)*

In contrast to Knowles et al. [11], decomposing the TSP objective function into two (or more) terms, Jensen [10] uses additional objectives to the existing primary objective and identifies so called helper-objectives for the TSP. Motivated by the weaknesses of the previous approach [11], Jensen introduced a helper-objective overcoming these problems:

$$h(\pi, p) = \sum_{i=1}^{|p|} M(c_{\pi[\pi^{-1}[p[i]]\ominus 1]}, c_{p[i]}) + M(c_{p[i]}, c_{\pi[\pi^{-1}[p[i]]\oplus 1]}) \quad (3)$$

where p is a subset of $\{1, 2, ..., N\}$ and $\ominus 1$ is the reverse of $\oplus 1$. The helper-objective $h(\pi, p)$ is the sum of distances in the path incident on the cities in p. This helper-objective has the property that all solutions representing the same tour will share the same objective value. Any number of helpers $\{h_1(\pi, p_1), h_2(\pi, p_2), ..., h_N(\pi, p_N)\}$ can be used. The helper-objectives are generated by creating a number of random sets $\{p_1, p_2, ..., p_N\}$, where each city had a 50% probability of being in a given set p. Note, distances are multiple-selected, if two cities are neighbours in the given tour.Figure 3 shows an example of how the two objective values are then calculated using a $N = 5$ cities tour and the random set $p = \{4, 5\}$.

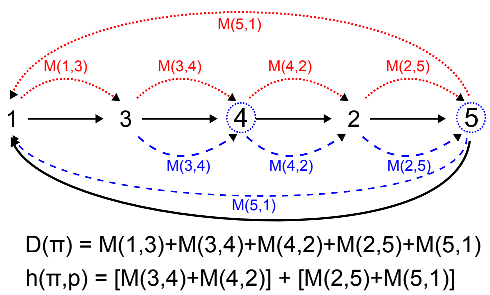

Figure 3: MO(Add. Objectives): fitness calculation

As Jensen [10] points out, when helper-objectives are used in a wrong way, the *"focus of the search will be shifted away from the primary objective to an unacceptable degree"*. Since HOs are generally in conflict with the primary goal, the search will concentrate more on solutions which have good fitness values in the second objective. In other words search will concentrate on the information contained in the used helper-objectives. In fact, the use of an additional objective could decrease the number of local optima, as Jensen argued. On the other hand experiments reveal, selection pressure increases and depending on the used helper-objective search is likely to be shifted away from global optimum. Thus, to use additional objectives beneficially one has to find suitable objectives, which guide search in the right direction.

In order to overcome the problem of an early convergence (cf. Section 2.2.2) Jensen [10] applies his additional objectives dynamically. Since he did not know how to schedule the different helper-objectives, he simply changes them deterministically depending on processing time. He found using a random ordering of 10 different helper-objectives each used for the same amount of time within one run to be the most promising approach. However, using such a scheme can be seen as a way to maintain diversity. Of course, this improves the performance particularly in comparison to a single-objective algorithm not using an additional diversity mechanism.

2.3 Multi-Objectivization via Segmentation

2.3.1 *The Concept of MOS*

To overcome the issues mentioned above, we propose a new method of decomposition. In this new approach the helper-objectives introduced by Jensen [10] are used for decomposition. That way, objectives are nested and consist of overlapping segments. This hybridized approach is named "Multi-Objectivization via Segmentation" (MOS). The formal description of the new MOS-approach is as follows:

$$\begin{aligned} s_1(\pi, p) &= \sum_{i=1}^{|p|} M(c_{\pi[\pi^{-1}[p[i]]\ominus 1]}, c_{p[i]}) + M(c_{p[i]}, c_{\pi[\pi^{-1}[p[i]]\oplus 1]}) \\ s_2(\pi, p^C) &= \sum_{i=1}^{|p^C|} M(c_{\pi[\pi^{-1}[p^C[i]]\ominus 1]}, c_{p^C[i]}) + M(c_{p^C[i]}, c_{\pi[\pi^{-1}[p^C[i]]\oplus 1]}) \end{aligned} \quad (4)$$

where p is again a subset of $\{1, 2, ..., N\}$ and p^C is the complementary set of p. The two new objectives $s_1(\pi, p)$ and $s_2(\pi, p)$ are the sum of distances in the path incident on the cities in p and p^C, respectively. The objectives were generated by creating a number of random sets $\{p_1, p_2, ..., p_N\}$ and $\{p_1^C, p_2^C, ..., p_N^C\}$, where each city is attributed to one of them with a 50% probability. Figure 4 shows an example of how the two objective values are calculated using a $N = 5$ cities tour and the two complementary sets $p = \{2, 3, 5\}$ and $p^C = p' = \{1, 4\}$. Notice that the sum of the two objectives is exactly twice the quantity to be minimized in equation (1). This again fulfils the requirement that each of the original optima becomes a Pareto-optimum under the new set of objectives. This approach is able to overcome all aspects that

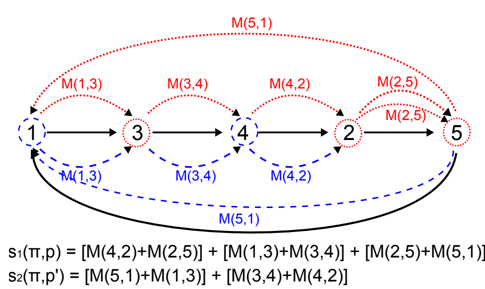

Figure 4: MO(Segmentation): fitness calculation

were criticised about the approaches of Jensen and Knowles et al., (see Section 2.2.1). First, these objectives have the property that all solutions representing the same tour will share the same objective value, as required by the definition of multi-objectivization of Knowles et al. [11]. Second, interdependencies of the problem components are minimized through the complex segmentation. This prevents choices of "bad objectives", ensuring a balanced trade-off between the two objectives. Third, thanks to the equally sized objectives, search focuses consequentially on the crucial area and will not be shifted away form the global optimum.

2.3.2 Enhanced MOS

The new approach MOS provides the opportunity to vary the segmentation of a tour. Knowing some problem characteristics, an adaptive construction of useful objectives is possible. At each generation, the population can be analysed using the current configuration to generate newly composed objectives. This way, search can be influenced and ideally guided out of local optima. In [11] it was already mentioned that *"the skill of the researcher...is to seperate out the conflicting aspects of the problem"*. In this Section we put forward three heuristic approaches of separating segments into two or more objectives.

Expected Value of distances.

One approach is to split the tour into two sets of cities that differ in average distance to their direct neighbours. That way cites with shorter links are separated from cities that are far away from next connected cities. Equation (5) shows how it is calculated, whether a city belongs to subset p or not.

$$\begin{aligned} p &= \{c_i \in N | \mu_i^s > \hat{\mu^s}\} \\ with\ \mu_i^s &= \frac{\sum_{s=1}^{S} d_i^s}{S}\ and\ \hat{\mu^s} = \frac{\sum_{i=1}^{N} \mu_i^s}{N} \\ where\ d_i^s &= M(c_{\pi[\pi^{-1}[p[i]]\ominus 1]}, c_{p[i]}) \\ &\quad + M(c_{p[i]}, c_{\pi[\pi^{-1}[p[i]]\oplus 1]}) \end{aligned} \tag{5}$$

where S is the number of individuals that were used for calculations and d_i^s is the sum of the distances between city i and its two neighbours in a given solution s. μ_i^s stands for the mean distances of city i to its neighbours over all individuals contained in S. $\hat{\mu^s}$ is the average of mean distances (μ_i^s) over all cities. Each city that has a mean above the average over all cities is chosen for the first subset. All remaining cities form the second subset. These subsets are then used in the original MOS approach as the two objectives. As a result of this approach, trade-off between objectives becomes more conflictive and search concentrates more on crucial parts of a given problem. It is mostly the longer distances that have to be changed to escape from local optima. This approach is able to trade-off these important and long distances with shorter distances.

Standard Deviation of distances.

Built on the first approach, standard deviation of distances is calculated using the parameters that were already defined in equation (5). Instead of just using mean of distances as information, in this second approach, we take into account the variation that one city has within the population. That way, cities that have a high bandwidth in distances to their neighbours will automatically become crucial for the search process. By dividing cities with high standard deviation and cities with a low standard deviation into two subsets, optimization concentrates on important parts of a given population. This approach is formally defined as follows:

$$\begin{aligned} p &= \{c_i \in N | \sigma_i^s > \hat{\sigma^s}\} \\ with\ \sigma_i^s &= \frac{\sum_{s=1}^{S} (\mu_i^s - d_i^s)}{S}\ and\ \hat{\sigma^s} = \frac{\sum_{i=1}^{N} \sigma_i^s}{N} \end{aligned} \tag{6}$$

where σ_i^s is the standard deviation of the distances between city i and its two neighbours in a given tour over all individuals. $\hat{\sigma^s}$ stands for the mean standard deviation over all cities and individuals. Each city that has a standard deviation above the average over all cities is chosen for the first subset. All remaining cities form the second subset. These subsets are then used in the original approach MOS as the two objectives.

Expected Value of different neighbours.

In the third approach the information about the distribution of cities in a population is used. Summing up the number of different neighbours that each city has in a given population it can be deduced which cities are crucial. Two sub-tours are created by calculating the average number of neighbours for each city and the corresponding mean. One city is chosen for the first sub-tour when the average number of neighbours is above the mean value as the following term illustrates:

$$\begin{aligned} p &= \{c_i \in N | \mu_i^s > \hat{\mu^s}\} \\ with\ \mu_i^s &= \frac{n_i^s}{S}\ and\ \hat{\mu^s} = \frac{\sum_{i=1}^{N} \mu_i^s}{N} \end{aligned} \tag{7}$$

where n_i^s is the number of different neighbours of city i in subset S. In doing so, search should concentrate on cities that differ most in terms of neighbours within the population.

Table 3 shows an overview of all presented approaches.

3. THE ALGORITHMS

3.1 Operators

The operators used in this study are identical to the ones of Jensen [10]. The representation used was the permutation encoding often used in TSP [13]. A tour is represented by an N-gene permutation of the numbers 1-N with a starting point at city 1 and a fixed orientation of the tour, (cf. Section 2.1). In all algorithms tournament selection of size of two was used. The mutation operator used picks a random city in the tour, removes it and inserts it at a random position in the tour. The recombination used in the algorithms was the improved edged recombination operator (ER) [13]. The used algorithms employ a local search heuristic, called 2-opt [4]. For complexity reasons some modifications to this heuristic were made. The operator considers only the first 25 % out of $L_c = (c_a, c_b, ...)$ (List of other cities, sorted on the distance to c). The reproduction scheme used in all algorithms was the $(\mu + \lambda)$-reproduction [5], except for the original version of the single-objective GA employing an elite of one.

3.2 Traditional GA for TSP

To investigate the effects of multi-objectivization, we implemented a traditional GA for comparisons. It employs the operators described in [10]. Parameter settings were tested on several problem instances with various population sizes, crossover rates and mutation rates. Experiments revealed, settings already used in [10] showed the best performance. Population size is set to 140 with an offspring size of 140. Crossover rate is set to 0.4 and individuals not produced by ER were created using the mutation operator previously described. Local Search was applied to all solutions after crossover or mutation.

3.3 Multi-Objective GA for TSP

For multi-objective algorithms operators remain the same and experiments were conducted identically to the ones performed on the single-objective GA. Again, settings already used in [10] showed best performance. So, population size and offspring size are set to 100 and the crossover rate to 0.7. Settings are different compared to the traditional GA due to the different reproduction schemes. Mutation is applied for solutions not created by crossover and local search is applied to all newly created solutions. When using multi-objectivization at least two objectives are incorporated. Fitness values are calculated based on the different approaches, (cf. Section 2). Solutions are sorted using non-dominated sorting as in NSGA-II [6]. Additionally, in all multi-objective algorithms objectives can be changed dynamically as Jensen introduced in [10]. Whenever in this study it is referred to dynamic use of objectives we assume his approach.

4. RESULTS AND DISCUSSION

4.1 Basic Settings

All algorithms were run 500 times on each problem instance for the same number of function evaluations. The maximum number of evaluations for each instance was calculated by:

$$E(N, m) = \sqrt{N^3} \times m \qquad (8)$$

where N is the number of cities and m a parameter, here set to 15. This proportion between N and the maximum number of evaluations $E(N, m)$ allowed was taken from [10]. The local search operator was found to be the decisive factor with regard to processing time. For each combination of algorithm and problem instance, in Table 4 the average best tour length is given as a percentage above the optimal tour length. Note, if the optimum is found the corresponding value is 0.000. For brevity, the standard error is not included in the table, but for all of the experiments the standard error lies between 0% and 0.009% of the average best tour length and the average is 0.0044%. Statistical significance tests have been performed. The test statistic used was the Student's t-test [7] and the significance level was set to $\alpha = 0.05$. In each row numbers marked '+' are significantly better than the traditional GA, while numbers marked '–' are significantly worse than the traditional GA. For each problem instance, the smallest number has been highlighted bold. To investigate the influence of the dynamic use of objectives each multi-objective algorithm is tested both statically and dynamically. In the following we will only refer to them as 'static' and 'dynamic', respectively. To avoid premature convergence Jensen [10] introduces two new methods to maintain diversity, namely 'in-breeding control' and 'niche enforcement'. The former aims at controlling the similarity of parents and the latter at removing identical solutions. They both improved performance significantly, with niche enforcement being the most promising. A similar and even stricter method is introduced in this study, namely 'Duplicate Avoidance' (DA). In DA identical individuals are detected after local search and prevented, so that each solution can exist just once. DA can be seen as enhanced niche enforcement and a reasonable degree of diversity can be maintained by its implementation. All algorithms are tested 'original' as well as with using 'DA'.

4.2 Results of Related Approaches

4.2.1 "DA" and "dynamic objectives"

The results illustrated in Table 4, demonstrate quite clearly that including "Duplicate Avoidance" improves the performance of the traditional GA and the "Jensen Approach" significantly. There was no case, independently of static or dynamic use, where the performance was deteriorated seriously when using DA compared to the original setting. Especially the traditional GA benefits from this diversity maintaining method. The only approach not benefiting is the one of Knowles et al. (KWC Approach), where the additional diversity method seems to interfere with search. Another observation is that using objectives dynamically improves performance compared to a static use, in almost every case. Tables 1 and 2 show an overview of the improvement in average affected by the introduction of "DA" and the dynamic ues of objectives, respectively. Note, since the two settings are independent from each other there are four possible configurations that have to be considered. Each '+' means significantly improved, '–' significantly worse and approaches that could not be tested to be significantly different are marked '='.

Table 1: Improvement from static to dynamic

Approach	original	DA
Jensen Approach	+	+
KWC Approach	+	+
MOS Approach	=	=

Table 2: Improvement from original to DA

Approach	static	dynamic
Traditional GA	+	
Jensen Approach	+	+
KWC Approach	=	–
MOS Approach	=	+
Enh.MOS App. 1		=
Enh.MOS App. 2		+
Enh.MOS App. 3		+

4.2.2 Additional Objectives (Jensen Approach)

Compared to the single-objective algorithm it can be seen in Table 4 that the "Jensen Approach" outperforms the traditional GA in case of not-using DA. With an average over all problems of 0.0275% above the optimal solution, the multi-objective algorithm is much better than the traditional GA, achieving an average of 0.0418%. However, when using DA this is no longer true, i.e., we cannot ascertain a statistically significant difference between the two averages of 0.0136% and 0.0154%.

Table 3: Overview Approaches

Name	Type	Approach	Objectives	Variant	Algortihm	sta/dyn	adaptive
Traditional GA	single	–	primary obj.	–	GA	static	no
Jensen Approach	multi	additional obj.	primary+HO	attaching parts	NSGAII	sta/dyn	no
KWC Approach	multi	decomposition	sub-obj.	parting	NSGAII	sta/dyn	no
MOS Approach	multi	decomposition	sub-obj.	segmentation	NSGAII	sta/dyn	no
Enh. MOS Approaches	multi	decomposition	sub-obj.	segmentation	NSGAII	sta/dyn	yes

Table 4: Average best tour length in percent above the optimal tour length

problem	variant	traditional GA	Jensen (sta)	Jensen (dyn)	KWC (sta)	KWC (dyn)	MOS (sta)	MOS (dyn)	Enh. MOS 1	Enh. MOS 2	Enh. MOS 3
pr107	original	0.0234	0.0517–	0.0056+	0.0886–	0.0426–	0.0499–	0.0103+	0.0035+	**0.0030+**	0.0128+
	DA	0.0060	0.0421–	0.0063	0.1034–	0.0647–	0.0454–	**0.0018+**	0.0174–	0.0046	0.0116–
pr124	original	**0.0000**	0.0062–	**0.0000**	0.0006–	0.0002	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
	DA	**0.0000**	**0.0000**	**0.0000**	0.0003	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
pr136	original	0.0829	0.0964–	0.0257+	0.3170–	0.2448–	0.0325+	0.0184+	**0.0157+**	0.0224+	0.0205+
	DA	0.0338	0.0879–	0.0300	0.3236–	0.2789–	0.0341	0.0213+	**0.0209+**	0.0270+	0.0229+
pr152	original	0.0218	0.0378–	0.0026+	**0.0000+**	**0.0000+**	**0.0000+**	**0.0000+**	**0.0000+**	**0.0000+**	**0.0000+**
	DA	**0.0000**	0.0325–	0.0007	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
bier127	original	0.0029	0.0105–	0.0011	0.0315–	0.0249–	0.0006	**0.0000+**	0.0016	**0.0000+**	**0.0000+**
	DA	**0.0000**	0.0038	**0.0000**	0.0292–	0.0198–	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
eil101	original	0.1008	0.0827	0.0531+	0.1027	0.0817	**0.0251+**	0.0464+	0.0302+	0.0366+	0.0401+
	DA	**0.0076**	0.0455–	0.0219–	0.1107–	0.1005–	0.0108	0.0102	0.0102	0.0114	0.0114
xqf131	original	0.0553	0.0872–	0.0489	0.0379+	0.0089+	0.0174+	0.0248+	**0.0035+**	0.0223+	0.0291+
	DA	0.0124	0.0628–	0.0287–	0.0351–	0.0028+	0.0142	0.0181	**0.0014+**	0.0082	0.0099
kroA150	original	0.0047	0.0207–	0.0017	0.0270–	0.0063	0.0010+	0.0013	0.0012	**0.0009+**	0.0012
	DA	0.0009	0.0073	0.0025	0.0315–	0.0092–	0.0009	0.0010	0.0011	**0.0003+**	0.0006+
kroB150	original	0.0249	0.0377–	0.0211	0.0780–	0.0484–	0.0104+	**0.0093+**	0.0110+	0.0133+	0.0150+
	DA	**0.0044**	0.0257–	0.0073–	0.0729–	0.0576–	0.0105–	0.0070–	0.0082–	0.0086–	0.0062–
kroD100	original	0.0469	0.0511	0.0267+	0.0483	0.0200+	0.0033+	0.0049+	**0.0008+**	0.0090+	0.0195+
	DA	0.0064	0.0283–	0.0054	0.0431–	0.0315–	0.0028+	0.0010+	**0.0001+**	0.0017	0.0009
kroE100	original	0.1018	0.1051	0.0899	0.0873+	**0.0296+**	0.0480+	0.0550+	0.0459+	0.0430+	0.0582+
	DA	0.0612	0.0814–	0.0586	0.0844–	0.0570	0.0435+	0.0354+	0.0517	**0.0301+**	0.0382+
eil51	original	0.0366	0.0620–	0.0535–	0.0277	**0.0160+**	0.0333	0.0315	0.0230+	0.0329	0.0376
	DA	0.0300	0.0333	0.0230	0.0319	0.0390	**0.0164+**	0.0310	0.0202+	0.0263	0.0300
Average	original	0.0418	0.0541–	0.0275+	0.0706	0.0436	0.0185+	0.0168+	**0.0144+**	0.0153+	0.0195+
	DA	0.0136	0.0376–	0.0154	0.0722–	0.0551	0.0149	0.0103	0.0109+	**0.0099**	0.0110

On the other hand, the approach of Jensen using static objectives performs worse compared to traditional GA, independent of the use of DA, as we can see in 3rd and 4th columns of Table 4. There was no case where performance could be improved, but most of the problems were solved significantly worse. Thus one can say, the performance improvement presented in [10] comes from the iterative use of objectives rather than multi-objectivization itself. Of course, it is multi-objectivization providing the opportunity of using objectives dynamically, but even disregarding these additional effects, positive influences on performance should be realisable. Table 5 shows how many problems could be solved statistically significant better and worse by the proposed algorithms compared to the single-objective algorithm. Columns headed "+" contain the number of problems solved better and headed "–" contain the number at which performance is worse than the traditional GA.

Table 5: Performance compared to traditional GA

Approach	original				DA			
	static		dynamic		static		dynamic	
	+	−	+	−	+	−	+	−
Jensen Approach	0	9	5	1	0	9	3	0
KWC Approach	3	6	5	4	0	9	7	1
MOS Approach	8	1	9	0	3	2	4	1
Enh.MOS App. 1	-	-	9	0	-	-	4	2
Enh.MOS App. 2	-	-	10	0	-	-	4	1
Enh.MOS App. 3	-	-	9	0	-	-	4	2

4.2.3 Decomposition (KWC Approach)

The results for the approach of Knowles et al. [11] using sub-objectives are quite varied compared to the other approaches. In general, this approach performs a good deal better or a lot worse than the other algorithms (cf. Table 4). Considering the problem of city pair choices (discussed in Section 2.2.1), it might always depend on the specific problem instance, whether or not this approach is advantageous. Considering the averages over all tested problems this approach performs worse than all of the others. In contrast to general observations, the diversity methods do not seem to have any beneficial influence on this algorithm. Especially including DA seems not to have a big impact on performance.

4.3 Results for MOS-Approach

As shown in Table 4, the new "MOS Approach" is able to outperform the other approaches on many problem instances. In contrast to the approach of Jensen even the version using static objectives performs better compared to the original GA. Considering the average over all instances, the new approach is not tested to be significantly better than the traditional GA using DA, but with values of 0.0136% and 0.0149% they are at least as good. On the other hand, the dynamic use of MOS achieves an average of 0.0103% and thus performs better than the traditional GA, the approaches of Jensen [10] and Knowles et al. [11].

4.3.1 Results for Enhanced MOS-Approaches

Table 4 shows the "Enhanced MOS" using adaptive objectives, are able to improve the average performance even further, especially when the original version not including DA is used. The first of the enhanced approaches, which splits the tour based on the expected value of distances in the population seems to be the best. Using DA additionally, the second approach (standard deviation of distances) shows best results. However, none of the enhanced approaches are statistically better than the original "MOS Approach", except for the problem "kroB150" using DA. Overall, at least one of the MOS approaches is able to outperform the traditional GA on each instance.

4.4 Summary

To compare performance of the different algorithms, the average performance over all problems was tested using the Student's t-test [7] to determine whether two different results, each characterized by its mean, standard deviation and number of data points, are significantly different. In Tables 6, 7, 8 and 9 comparisons between all used algorithms are illustrated as for the four respective configurations. Approaches (row) that outperform another one (column) are marked "+". Approaches (row) that are outperformed significantly by the other (column) are marked "–". Approaches that are not significantly different are marked "=". Main observations are:

- Using static objectives, the "Traditional GA" outperforms the "Jensen Approach".
- Using static objectives, the new "MOS Approach" outperforms "Traditional GA", "KWC Approach" and "Jensen Approach".
- Using dynamic objectives, all approaches are able to outperform the "Traditional GA" except for the "KWC Approach"
- Using "DA", none of the other algorithm outperforms the "Traditional GA".
- The new "MOS Approach" performs at least as good as any of the other algorithms.

Table 6: Overview: static/original

Approach	Trad. GA	Jensen App.	KWC App.
Traditional GA			
Jensen Approach	−		
KWC Approach	=	=	
MOS Approach	+	+	+

5. CONCLUSIONS

In this paper we studied how the multi-objectivization of the TSP can facilitate improved search for evolutionary algorithms. We have adapted the approaches of Jensen [10] and Knowles et al. [11], which represent the two basic approaches of multi-objectivization, namely the addition of new objectives to the existing problem and the decomposition of the primary objective to sub-objectives. Built on the latter we have proposed the new concept "Multi-Objectivization via Segmentation" (MOS), which is based on reassembling the original problem. Three enhancements using adaptive objectives have been proposed.

Experiments were conducted using dynamic objectives and an additional diversity method avoiding duplicates (DA).

Table 7: Overview: static/DA

Approach	Trad. GA	Jensen App.	KWC App.
Traditional GA			
Jensen Approach	–		
KWC Approach	–	=	
MOS Approach	=	+	+

Table 8: Overview: dynamic/original

Approach	Trad. GA	Jen.	Know.	MOS	EMOS1	EMOS2
Traditional GA						
Jensen Approach	+					
KWC Approach	=	=				
MOS Approach	+	+	=			
Enh.MOS App. 1	+	+	=	+		
Enh.MOS App. 2	+	+	=	=	–	
Enh.MOS App. 3	+	+	=	=	–	–

Table 9: Overview: dynamic/DA

Approach	Trad.GA	Jen.	Know.	MOS	EMOS1	EMOS2
Traditional GA						
Jensen Approach	=					
KWC Approach	=	=				
MOS Approach	=	+	+			
Enh.MOS App. 1	=	=	+	+		
Enh.MOS App. 2	=	=	+	=	–	
Enh.MOS App. 3	=	=	+	=	=	=

We demonstrated that all of the new algorithms were able to outperform the single-objective case when dynamic objectives are used. However, our results showed that this does not always hold true, especially when using objectives statically or including “DA”. Thus, one may argue the improvement in performance is attributed rather to diversity than to the multi-objectivization and that any other diversity method might achieve similar results (cp. [14]). Nevertheless, independent of which specific effects cause the improvement in performance we can state that multi-objectivization of the travelling salesman problem using evolutionary optimization techniques can be beneficial compared to single-objective optimization.

Furthermore, we have shown that the new algorithms based on the concept of “MOS” clearly outperform related approaches of Jensen and Knowles et al. Especially when objectives are used statically, the new approaches perform significantly better in average, regardless if DA is used or not. The mediocre performance of the approaches of Jensens and Knowles et al. is mainly caused by the misdirection of search using additional objectives and the problems of city pair choices, respectively. In contrast, the “MOS Approach” has the ability to maintain diversity and to form good building blocks, avoiding premature convergence.

Besides being an excellent diversity maintaining strategy our results suggest multi-objectivization results in a reduction of local optima. Since the new “MOS Approach” outperforms the traditional GA in the case where objectives are used statically (without “DA”), we may conclude that the decomposition of the original problem is effective.

6. REFERENCES

[1] D. Brockhoff, T. Friedrich, N. Hebbinghaus, C. Klein, F. Neumann, and E. Zitzler. Do additional objectives make a problem harder? In *Genetic and Evolutionary Computation*, pages 765–772, New York, NY, USA, 2007. ACM.

[2] C. A. Coello Coello and G. Lamont. *Applications of Multi-Objective Evolutionary Algorithms*. World Scientific Publishing, Singapore, 2004.

[3] David L. Applegate, Robert E. Bixby, Vasek Chvatal and William J. Cook. *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, 2007.

[4] David S. Johnson and Lyle A. Mcgeoch. The traveling salesman problem: A case study in local optimization. In *Local Search in Combinatorial Optimization*. Wiley and Sons. Chapter 8, E. Aarts and J.K. Lenstra (Eds.).

[5] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley, Chichester, UK, 2001.

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

[7] W. S. Gosset. The probable error of a mean. *Biometrika*, 6(1):1–25, March 1908. Originally published under the pseudonym “Student”.

[8] J. Handl, S. C. Lovell, and J. Knowles. Investigations into the effect of multiobjectivization in protein structure prediction. In *Parallel Problem Solving from Nature*, pages 702–711, Berlin, Heidelberg, 2008. Springer-Verlag.

[9] J. Handl, S. C. Lovell, and J. Knowles. Multiobjectivization by decomposition of scalar cost functions. In *Parallel Problem Solving from Nature*, pages 31–40, Berlin, Heidelberg, 2008. Springer-Verlag.

[10] Jensen, M.T. Helper-objectives: Using multi-objective evolutionary algorithms for single-objective optimisation. *Journal of Mathematical Modelling and Algorithms 3(4)*, pages 323–347, 2004.

[11] J. D. Knowles, R. A. Watson, and D. W. Corne. Reducing local optima in single-objective problems by multi-objectivization. In *Evolutionary Multi-Criterion Optimization*, pages 269–283. Springer, 2001.

[12] S. Watanabe, K. Sakakibara. Multiobjective approaches in single objective optimization environment. In *The 2005 IEEE Congress on Evolutionary Computation*, volume 2, pages 1714–1721, 2005.

[13] T. Starkweather, S. Mcdaniel, C. Whitley, K. Mathias, D. Whitley, and M. E. Dept. A comparison of genetic sequencing operators. In *Genetic Algorithms*, pages 69–76. Morgan Kaufmann, 1991.

[14] Z. Yan, L. Zhang, L. Kang, and G. Lin. A new moea for multi-objective tsp and its convergence property analysis. In *EMO*, pages 342–354, 2003.