To appear in *Optimization Methods & Software*
Vol. 00, No. 00, Month 20XX, 1–14

Evolutionary Algorithm with a Directional Local Search for Multiobjective Optimization in Combinatorial Problems

Krzysztof Michalak[a*]

[a] *Department of Information Technologies, Institute of Business Informatics, Wroclaw University of Economics, Wroclaw, Poland*

(*Received 00 Month 20XX; final version received 00 Month 20XX*)

Evolutionary algorithms are often employed to multiobjective optimization, because they process an entire population of solutions which can be used as an approximation of the Pareto front of the tackled problem. It is a common practice to couple local search with evolutionary algorithms, especially in the context of combinatorial optimization. In this paper a new local search method is proposed that utilizes the knowledge concerning promising search directions. The proposed method can be used as a general framework and combined with many methods of iterating over a neighbourhood of an initial solution as well as various decomposition approaches. In the experiments the proposed local search method was used with an evolutionary algorithm and tested on 2-, 3- and 4-objective versions of two well-known combinatorial optimization problems: the Travelling Salesman Problem (TSP) and the Quadratic Assignment Problem (QAP). For comparison two well-known local search methods, one based on Pareto dominance and the other based on decomposition, were used with the same evolutionary algorithm. The results show that the evolutionary algorithm coupled with the directional local search yields better results than the same evolutionary algorithm coupled with any of the two reference methods on both the TSP and QAP problems.

Keywords: multiobjective optimization; evolutionary computing; local search; decomposition; Travelling Salesman Problem; Quadratic Assignment Problem

1. Introduction

Multiobjective optimization has been, for many years, a field of active study in which researchers tackle theoretical as well as practical challenges. Without a loss of generality, a multiobjective optimization problem can be defined as follows:

$$\begin{array}{l} \text{minimize } F(x) = [f_1(x), \ldots, f_m(x)] \\ \text{subject to } x \in \Omega, \end{array} \tag{1}$$

where:
Ω - the decision space,
m - the number of objectives.

Because there exist m different objectives it is usually not possible to choose one, the best solution which minimizes all $f_j(x)$, $j = 1, \ldots, m$ simultaneously. Similarly, it is not always possible to determine which of any given two solutions is better than the other one. Instead, a concept of Pareto domination [7, 27] is used to define a relation over a set

*Corresponding author. Tel.: +48 71 36 80 377 Email: krzysztof.michalak@ue.wroc.pl

of all possible solutions to the problem (1). For any two points $x_1, x_2 \in \Omega$ we say that x_1 *dominates* x_2 ($x_1 \succ x_2$) if:

$$\begin{array}{l} \forall j \in \{1, \ldots, m\} : f_j(x_1) \leq f_j(x_2) \\ \exists j \in \{1, \ldots, m\} : f_j(x_1) < f_j(x_2) \end{array} \tag{2}$$

A solution $x \in \Omega$ is *nondominated* (also called *Pareto optimal*) if:

$$\neg \exists x' \in \Omega : x' \succ x. \tag{3}$$

The set of all nondominated solutions in the decision space Ω is called the *Pareto set* and its image in objective space R^m is called the *Pareto front*.

Multiobjective optimization can be performed using a wide variety of optimization methods. Among others, well-known methods such as the steepest descent [11] and tabu search [14] have been adapted to multiobjective optimization. One of the proposed approaches to multiobjective optimization is also to use the simulated annealing method in which the probability of making a transition from the current state to a new state depends on the energies assigned to these states and on a global time-varying parameter representing the temperature. In the paper [33] several criteria for the probability of accepting the new solution are discussed. The paper [35] proposed a simulated annealing algorithm PDMOSA using a Pareto-based fitness.

Evolutionary methods are often used to tackle multiobjective optimization problems, because their population-based nature allows them to process in one run an entire set of solutions, which approximate the true Pareto front. Evolutionary algorithms used for multiobjective optimization often fall into one of the two categories: based on Pareto dominance or based on decomposition. Algorithms from the former class use the relation (3) either directly for selection of specimens (e.g. NSGA [34] and NSGA-II [8]) or for fitness assignment (e.g. SPEA [39] and SPEA2 [38]). Algorithms based on decomposition transform the initial multiobjective problem into a set of scalar problems using a decomposition procedure. For example, this approach is used in the Multiobjective Evolutionary Algorithm Based on Decomposition (MOEA/D) algorithm [25, 37]. In one of the simplest decomposition procedures scalar objectives are obtained by calculating weighted sums of the initial objectives with various weight vectors.

Local Search in Evolutionary Optimization

It is a commonly used approach to augment an evolutionary algorithm (EA) with a local search method, which is used to improve the solutions found by the EA. This practice is very often used in the case of combinatorial optimization problems. Early attempts at combining the local search with evolutionary optimization include the Genetic Local Search method developed, among others, by Ishibuchi [16] and Jaszkiewicz [18] for multiobjective optimization in such problems as the TSP and flowshop scheduling. Further developments in the area of memetic algorithms include the Memetic Pareto Archived Evolution Strategy (M-PAES) [24], which has been applied successfully to combinatorial problems, for example to the multiobjective knapsack problem [23]. Memetic algorithms have been applied to many optimization problems including the flowshop scheduling problem [17], pick-up and delivery problems [28], the orienteering problem [12] and the vehicle routing problem [32]. A survey by Knowles and Corne [22] apart from presenting even more optimization problems to which memetic algorithms can be applied, discusses various mechanisms used in these algorithms (such as dominance ranking, scalarization, niching, etc.). It also describes performance measures used for assessing the quality of results produced by optimization algorithms and discusses conditions under which various mechanisms can improve the working of memetic algorithms.

In the case of multiobjective optimization a local search based on Pareto dominance (PLS) is often used [2–4, 26, 29]. Pareto dominance can be used for replacing solutions during the local search, but also for determining the areas in which an improvement of solutions can be expected [15]. In this paper the Pareto Local Search is used as a reference method to which the new local search method is compared. The PLS used in this paper is a best-improvement search based on the neighborhood generated using the 2-opt operator [5]. The neighbourhood generation using the 2-opt operator works as follows. For a given permutation π the neighbourhood $N(\pi)$ is generated by considering all possible pairs of two different indices $\langle i, j\rangle$. For each given pair a new permutation is generated by reversing the order of the elements with indices between i and j (inclusive). This operation may "wrap around" the last element in the permutation when $j < i$. Thus, applying the 2-opt operator to a permutation $\pi = (\pi_1, \ldots, \pi_k)$ for $i < j$ produces $(\pi_1, \ldots, \pi_{i-1}, \pi_j, \ldots, \pi_i, \pi_{j+1}, \ldots, \pi_k)$. For $i > j$ the 2-opt operator produces $(\ldots, \pi_i, \pi_{j+1}, \ldots, \pi_{i-1}, \pi_j, \ldots)$. For example if $\pi = (1, 2, 3, 4, 5, 6, 7, 8)$ the new permutation produced for $i = 2$ and $j = 6$ is $\pi = (1, 6, 5, 4, 3, 2, 7, 8)$ and the new permutation produced for $i = 6$ and $j = 2$ is $\pi = (7, 6, 3, 4, 5, 2, 1, 8)$.

Another common approach to local search is to use objective aggregation similar to that performed in the decomposition-based optimization algorithms such as the MOEA/D [25, 37]. The aim of decomposition is to convert a multiobjective optimization problem (1) into a set of scalar problems. One of the commonly used decomposition methods is the weighted sum decomposition. This decomposition method uses a set of H weight vectors $\{\lambda^{(i)}\}_{i\in\{1,\ldots,H\}}$. Each of the vectors $\lambda^{(i)}$ satisfies the following conditions: $\lambda^{(i)}[j] \geq 0$ for all $j \in \{1, \ldots, m\}$, $\sum_{j=1}^{m} \lambda^{(i)}[j] = 1$. The initial problem is decomposed into H scalar subproblems associated with $\lambda^{(i)}$ vectors. In the i-th subproblem the scalar objective is calculated as a weighted sum of the initial objectives $f_j(x)$, $j = 1, \ldots, m$ with weights equal to the elements of the corresponding weight vector $\lambda^{(i)}$. Other decomposition methods include the Tchebycheff decomposition and boundary-intersection decomposition [6].

The decomposition-based local search (DLS) maintains its own set of weight vectors and thus it does not depend on whether the main evolutionary algorithm is decomposition based. After each generation of the evolutionary algorithm the DLS is performed for each of the specimens in the population as presented in Algorithm 1.

Note: in Algorithm 1 the $\cdot$ symbol denotes the dot product.

The N(x) denotes a problem-specific neighbourhood of a solution x. For example, in this paper the neighbourhood search for the TSP and the QAP problems iterates over all the solutions that can be generated from a given solution x using the 2-opt operator. The Directional Local Search (DirLS) method proposed in this paper uses the DLS search procedure to perform the search for a given specimen using a weight vector assigned by the DirLS algorithm described later in the paper. Also, the DLS is used as a second reference method to which the DirLS method is compared. In the reference DLS method the weights are calculated based on the nadir point $n^* \in R^m$ that consists of the worst values of the objective found during the search and the current position of the solutions in the population.

For a specimen $x^{(i)}$ with the objectives vector $\{f_1(x^{(i)}), \ldots, f_m(x^{(i)})\}$ the corresponding weight vector $\lambda^{(i)}$ is calculated based on a vector of absolute differences between the objective values and nadir point coordinates:

$$\delta^{(i)} = [|f_1(x^{(i)}) - n_1^*|, \ldots, |f_m(x^{(i)}) - n_m^*|] \quad . \tag{4}$$

Algorithm 1 The main loop of the decomposition-based local search.

IN:
x_0 - a solution found by the evolutionary algorithm
$F_0 = F(x_0)$ - a vector of the values of objectives for the solution x_0
$\lambda = (\lambda_1, \ldots, \lambda_m)$ - the weight vector associated with the solution x_0

OUT:
x - an improved solution

```
x = x_0
v = F_0 · λ
do
  v_old = v
  N = N(x)
  foreach x' ∈ N
    F' = [f_1(x'), ..., f_m(x')]
    v' = F' · λ
    if v' < v
      x = x'
      v = v'
    endif
  end foreach
while v < v_old
return x
```

The weights are calculated by normalizing vectors $\delta^{(i)}$:

$$\lambda^{(i)} = \delta^{(i)} / \sum_{j=1}^{m} \delta^{(i)}[j] \quad . \tag{5}$$

Improved solutions found by a local search procedure are usually incorporated to the population that undergoes evolution. In the case of Pareto-based evolutionary algorithms the improved solutions replace specimens which they dominate. When a decomposition-based evolutionary algorithm is used, the easiest way to perform the local search is to search along the aggregated objectives using the same weight vectors as the original algorithm. For example in a recent paper Ke et al. [20] use a single-objective local search using the same weight vectors as the original algorithm and, additionally, a subpopulation which undergoes the Pareto Local Search. The latter of these search procedures is, therefore, a well-known Pareto Local Search (PLS), while the former is a decomposition-based local search (DLS) which does not adapt the weight vectors as the DirLS does.

The rest of this paper is structured as follows. In section 2 the directional local search algorithm is presented. Section 3 describes the combinatorial problems used for tests. In section 4 the experimental setup is outlined and the results are discussed. Section 5 concludes the paper.

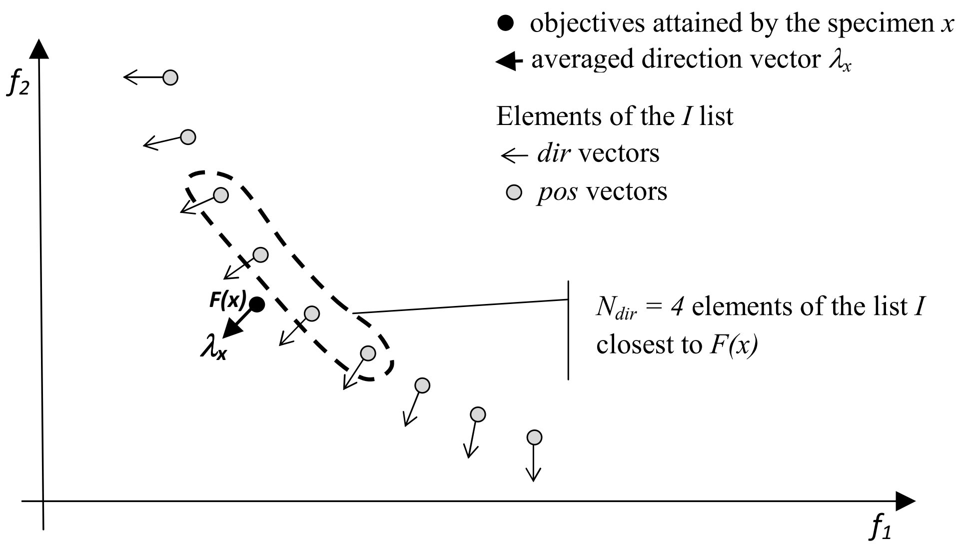

Figure 1. The concepts involved in the working of the Directional Local Search algorithm ($N_{dir} = 4$).

2. Directional Local Search

The Directional Local Search (DirLS) stores a list I of pairs $\langle pos, dir \rangle$, in which each of the elements contains a direction (dir) in which the last improvement was obtained during the local search phase at a given position (pos) on the Pareto front. This list is updated during the local search phase after each generation of the evolutionary algorithm. Information stored in the I list is used as follows during the local search phase. For each specimen x, N_{dir} elements are found for which the *pos* value is the closest to the current objectives $F(x)$ of the specimen x. The dir vectors of these elements are averaged and the resulting vector λ_x is used as the weights vector for a DLS procedure. The concepts involved in the working of the Directional Local Search algorithm are presented in Figure 1.

The working of the Directional Local Search algorithm is presented in Algorithm 2. In this algorithm the following procedures and functions are used:

InitPopulation - initializes a new population for the EA.

Evolve - performs an evolution of one generation of the EA. The implementation of this procedure depends of the selected evolutionary algorithm.

UpdateNadirPoint - updates the nadir point based on the solutions found so far. In the implementation presented in this paper the nadir point was calculated based on the archive of all nondominated solutions found by the algorithm.

Rand - Returns a number drawn from a given probability distribution. In the algorithm this function is used to draw a number from an uniform random distribution $U_{[0,1]}$.

GetNearestVectors - Returns N_{dir} vectors taken from dir elements of those elements of the list I that have *pos* vectors closest to the given value $F(x)$.

DLS - the Decomposition Local Search procedure presented in Algorithm 1.

In this paper the weighted sum decomposition and a neighbourhood search based on the 2-opt operator were used. The proposed framework, however, can be used with many other decomposition approaches and local search operators. Also, the working of the Directional Local Search does not rely heavily on the main evolutionary algorithm. It can be used both with EAs based on Pareto dominance as well as EAs based on decomposition. Other metaheuristic optimization methods can be coupled with the proposed local search method as well.

Algorithm 2 An evolutionary algorithm combined with the Directional Local Search algorithm.

N_{gen} - the number of generations of the EA
N_{pop} - the population size
N_{dir} - the number of direction vectors used to calculate each weight vector $\lambda^{(i)}$

$I = \emptyset$
$P_1 = \text{InitPopulation}()$
for $g = 1 \to N_{gen}$ **do**
  $P_{g+1} = \text{Evolve}(P_g)$
  $n^* = \text{UpdateNadirPoint}(P_{g+1}, n^*)$
  for $i = 1 \to N_{pop}$ **do**
    $max_i = \max\{x_i : [x_1, \ldots, x_m] \in P_{g+1}\}$
  end for
  for $i = 1 \to N_{pop}$ **do**
    $x = P_{g+1}[i]$
    if $\exists j : x_j = max_j$ **then**
      $\lambda^{(i)} = [0, \ldots, 0]$
      $\lambda^{(i)}_j = 1$
    else
      $\Theta = \frac{|I|}{N_{pop}}$
      if $(|I| \geq N_{dir})$ **and** $(\text{Rand}(U_{[0,1]}) < \Theta)$ **then**
        $\{v_i\}_{i=1,\ldots,N_{dir}} = \text{GetNearestVectors}(I, F(x), N_{dir})$
        $\{\lambda^{(i)}\} = \frac{v_1,\ldots,v_{N_{dir}}}{N_{dir}}$
      else
        $\delta^{(i)} = [|f_1(x) - n^*_1|, \ldots, |f_m(x) - n^*_m|]$
        $\lambda^{(i)} = \delta^{(i)} / \sum_{i=1}^{m} \delta^{(i)}[j]$
      end if
    end if
  end for
  $P'_{g+1} = \emptyset$
  for $i = 1 \to N_{pop}$ **do**
    $x = P_{g+1}[i]$
    $x' = DLS(x, F(x), \lambda^{(i)})$
    $P'_{g+1} = P'_{g+1} \cup \{x'\}$
    if $\forall j \in \{1, \ldots, m\} : x'[j] < x[j]$ **then**
      $I' = I' \cup \{\langle F(x'), F(x') - F(x)\rangle\}$
    end if
  end for
  $P_{g+1} = P'_{g+1}$
end for

3. Test Problems

The experiments were conducted on 2-, 3- and 4- objective versions of two well-known combinatorial optimization problems: the Travelling Salesman Problem (TSP) and the Quadratic Assignment Problem (QAP).

The Travelling Salesman Problem

The single objective Travelling Salesman Problem (TSP) is defined as follows [13]. For a given $n \times n$ cost matrix $C = [c_{ij}], i, j \in \{1, \ldots, n\}$:

$$\begin{array}{l} \text{minimize } f(p) = c_{p(n)p(1)} + \sum_{i=1}^{n-1} c_{p(i)p(i+1)} \\ \text{subject to } p \in \mathrm{P}_n, \end{array} \tag{6}$$

where:

P_n - the set of all permutations of numbers $1, \ldots, n$.

An m-objective TSP problem can be generated by calculating m objective functions $f_j(p),\ j = 1, \ldots, m$ each using a different cost matrix C_j of size $n \times n$. Biobjective TSP instances were obtained by combining single objective TSP instances kroAnnn and kroBnnn (where $nnn = 100, \ldots, 400$). Because for $nnn > 100$ only two instances kroAnnn and kroBnnn are available, the three- and four-dimensional test instances were generated using the following technique. For every instance size n three permutations π_2, π_3 and π_4 of n elements were generated using an online random permutation generator [19]. The first objective function $f_1(p)$ was calculated, using the equation (6), without any additional permutation with the cost matrix taken from the instance kroAnnn. The other objectives $f_j(p)$, $j \geq 2$ were calculated by permuting the argument p with one of the generated permutations, so $f_j(p) = f(\pi_j(p))$ for $j \geq 2$, where $f(p)$ was calculated using the equation (6). Objective functions generated using this approach reach their minima at different points in the search space (namely $p_0 = \mathrm{argmin}(f(p))$, $\pi_2^{-1}(p_0)$, $\pi_3^{-1}(p_0)$, $\pi_4^{-1}(p_0)$), thus creating a Pareto optimization problem. The random generator seeds used for calculating the permutations are presented in Table 1.

Table 1. Random generator seeds used for calculating the permutations used for calculating the objectives in a 3- and 4-objective TSP

Instance size	Seeds
100	9063, 22071, 27073
150	5101, 2173, 4181
200	1191, 22211, 12221
300	15231, 25241, 5251
400	14271, 29273, 7052

The Quadratic Assignment Problem

The single objective Quadratic Assignment Problem (QAP) is defined as follows [31]. For a given $n \times n$ distance matrix $D = [d_{ij}], i, j \in \{1, \ldots, n\}$ and a flow matrix $F = [f_{ij}], i, j \in \{1, \ldots, n\}$:

$$\begin{array}{l} \text{minimize } f(p) = \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} d_{p(i)p(j)} + \sum_{i=1}^{n} b_{ip(i)} \\ \text{subject to } p \in \mathrm{P}_n, \end{array} \tag{7}$$

where:

P_n - the set of all permutations of numbers $1, \ldots, n$,

$b_{ip(i)}$ - a linear term that represents the cost of making the assignment according to permutation p .

The linear term $b_{ip(i)}$ is omitted in some formulations of the problem. A multiobjective version of the QAP problem can be generated by using a different matrix for each objective. A common practice is to use the same distance matrix D and m different flow matrices F_j, $j = 1, \ldots, m$ [21]. In this paper biobjective QAP instances with correlated flow matrices introduced in [30] were used. The tests were performed on unstructured instances of size 25, 50 and 75 and with correlations between the flow matrices of $\rho \in \{-0.75, -0.50, -0.25, 0.0, 0.25, 0.50, 0.75\}$. Similarly as with the TSP, permutation-based objective functions were used for 3- and 4-objective test instances, in which the flow matrix used for the first objective was taken from [30] and the remaining objectives were calculated using permutations in the same manner as with the TSP. The random generator seeds used for calculating the permutations are presented in Table 2.

Table 2. Random generator seeds used for calculating the permutations used for calculating the objectives in a 3- and 4-objective QAP

Instance size	Seeds
25	13270, 21272, 15280
50	4099, 8101, 3109
75	3119, 2121, 27121

4. Experiments and Results

The experiments were aimed at comparing the performance of the Directional Local Search (DirLS) with the Decomposition Local Search (DLS) presented in the Algorithm 1 and the Pareto Local Search (PLS). The Pareto Local Search was implemented similarly as in the Algorithm 1, except a pareto dominance criterion was used for determining if the solution has been improved. The NSGA-II algorithm [8] was used as the evolutionary algorithm, based on which all three compared local search methods worked, because it is currently one of the best performing evolutionary multiobjective optimization algorithms, especially for combinatorial optimization. Solving of the test problems using the SPEA2 algorithm has also been attempted, but the results produced by the SPEA2 (with all the local search methods) were much worse than those attained by the NSGA-II with DLS and DirLS. The experiments were conducted on test problems described in Section 3. The performance of both algorithms was measured using the hypervolume (HV) indicator [40]. The hypervolume is the function of a set of points equal to the Lebesgue measure of the portion of the objective space that is dominated by solutions in P collectively:

$$HV(P) = L\left(\bigcup_{x\in P} [f_1(x), r_1] \times \ldots \times [f_m(x), r_m]\right), \tag{8}$$

where:
m - the dimensionality of the objective space,
$f_i(\cdot)$, $i = 1, \ldots m$ - the objective functions,
$R = (r_1, \ldots, r_m)$ - a reference point,
$L(\cdot)$ - the Lebesgue measure on R^m.

In R^2 and R^3 the hypervolume is equal to the area and volume of the objective space covered by the solutions in the set P, respectively. The higher the value of the hypervolume indicator, the better the Pareto front and as shown in [10] maximizing the hypervolume is equivalent to achieving Pareto optimality.

Each test was repeated 30 times and the median values were calculated. Statistical significance of the results was determined using the Wilcoxon test [9] with the null hypothesis that the DirLS method gives the same median hypervolume value than a given reference method. The Wilcoxon test for the equality of medians was chosen, because, contrary to tests that compare means, it does not assume the normality of the distribution of the tested measurements. Since the normality assumption is most often not satisfied for the distribution of the measured hypervolume values, the Wilcoxon test is preferable to tests such as the Student's t-test. Because the comparison of the DirLS method is made against a set of two other methods an adjusted p-value [1] was calculated as:

$$P_{ADJ} = 1 - (1 - P_{DLS})(1 - P_{PLS}) \tag{9}$$

where P_{DLS} and P_{PLS} are p-values obtained for the test comparing DirLS to DLS and DirLS to PLS, and P_{ADJ} is the adjusted p-value. The adjusted p-value gives the probability of incorrectly rejecting the null hypothesis at least once in the set of two comparisons performed (DirLS vs. DLS and DirLS vs. PLS). Therefore, if a low adjusted p-value is obtained it indicates a high statistical significance of the conclusion that the DirLS produces higher hypervolume values than both DLS and PLS.

The results of the experiments are summarized in Table 3 which presents for how many test instances each algorithm has produced the highest median and for how many test instances the DirLS has been significantly better than the two other algorithms.

Table 3. Summary of the results of the experiments (the number of test instances for which each algorithm has produced the highest median and the number of test instances for which the DirLS has been significantly better than the two other algorithms)

Problem	Number of instances	Number of highest medians			DirLS significantly better	
		DirLS	DLS	PLS	$\alpha = 0.01$	$\alpha = 0.05$
2-objective TSP	7	7	0	0	7	7
3-objective TSP	5	5	0	0	5	5
4-objective TSP	5	5	0	0	5	5
2-objective QAP ($n = 25$)	7	6	0	1	5	6
2-objective QAP ($n = 50$)	7	7	0	0	6	6
2-objective QAP ($n = 75$)	7	7	0	0	6	6
3-objective QAP ($n = 25$)	7	7	0	0	7	7
3-objective QAP ($n = 50$)	7	7	0	0	7	7
3-objective QAP ($n = 75$)	7	7	0	0	7	7
4-objective QAP ($n = 25$)	7	7	0	0	7	7
4-objective QAP ($n = 50$)	7	7	0	0	7	7
4-objective QAP ($n = 75$)	7	7	0	0	7	7

Numerical values and graphs are presented in tables in section S.1 and in figures in section S.2 in a supplemental file available online. In the tables median hypervolume values are presented, along with p-values obtained in the Wilcoxon statistical test for the equality of medians. The timings presented in the tables in secion S.1. of the supplemental file were obtained on machines with a 2.4 GHz Intel Core 2 Quad Q6600 CPU and 4 GB of RAM. In the case of QAP instances the local search has been performed on a GeForce 470 GTX graphics cards with 1.25 GB (1280 MB) of RAM. The amount of memory did not seem, however, to play a significant role in the experiments as the maximum memory allocation was far below the available maximum.

The Travelling Salesman Problem

Each test was run for 200 generations. The population size was equal to the number of cities in each test problem. The inver-over operator [36] was used with the random inverse rate 0.02. All three local search procedures were based on the 2-opt heuristic and were implemented on a regular CPU.

The values of the hypervolume obtained for the TSP are presented in Tables S.1.1-S.1.3 and in Figures S.2.1-S.2.7 in a supplemental file available online. From the figures it can be seen that DirLS and DLS behave similarly, while PLS produces much worse results, which also improve more slowly than in the case of the other two algorithms. For brevity of the presentation, graphs for 3 and 4 objectives (qualitatively similar to those for 2 objectives) have been omitted. Values in the tables show that there is a statistically significant difference between DirLS and DLS in favour of the DirLS which produced the best results for all 2-, 3- and 4-objective TSP instances. As shown in Table S.1.4, the running times of all three algorithms for the 2-objective TSP were similar, with DLS a few percent faster, but clearly at the expense of the quality of the results. In the case of 3- and 4-objective TSP the DLS is much slower than DirLS, as shown in Tables S.1.5 and S.1.6 and this difference seems to increase with the increasing number of objectives. The fastest method is the PLS, but this advantage seems to be negligible, because of the much worse results produced by this method.

The Quadratic Assignment Problem

Each test was run for 200 generations. The population size was 100. The inver-over operator [36] was used with the random inverse rate 0.02. All three local search procedures were based on the 2-opt heuristic and were implemented on a many-core graphics processor (GPU).

The results obtained for the QAP are presented in Tables S.1.7-§.1.9 (2-objective QAP), S.1.11-S.1.13 (3-objective QAP) and S.1.15-S.1.17 (4-objective QAP) in a supplemental file available online. In almost all cases the Directional Local Search (DirLS) attained better hypervolume values than the Decomposition-Based Local Search (DLS) and the Pareto Local Search (PLS). Only in one case the median hypervolume attained by the DirLS method ($2.971 \cdot 10^{10}$) was lower than the median hypervolume attained by both the DLS ($3.070 \cdot 10^{10}$) and the PLS ($3.086 \cdot 10^{10}$), that is for the biobjective QAP with $n = 25$ and $\rho = +0.75$. Nevertheless, the maximum value of the hypervolume attained in one of the runs of the DirLS ($3.539 \cdot 10^{10}$) was higher than those obtained by DLS ($3.486 \cdot 10^{10}$) and PLS ($3.494 \cdot 10^{10}$). In most cases the difference in favour of the DirLS method has a high statistical significance (adjusted p-value smaller than 0.01).

Running times of the algorithms (for $N_{gen} = 200$ generations) presented in Tables S.1.10, S.1.14 and S.1.18 in a supplemental file available online are very similar. While the fastest algorithm is usually the PLS, as with the TSP, it produces very poor results. The DLS was never the fastest algorithm of the tested three.

Figures S.2.8-S.2.28 show that for the QAP the DirLS improves the results much faster than the other two algorithms. A similar behaviour has been observed for larger dimensionalities, but, similarly as with the TSP, graphs for 3 and 4 dimension have not been included in order not to clutter the paper. The exception is the 2-objective QAP with $n = 25$ and $\rho = +0.75$ (cf. Figure S.2.14), which corresponds to the results presented in Table S.1.7.

Overall, the DirLS seems to improve the results much faster than the other two algorithms in the case of the QAP. For the TSP the performance of the DirLS and DLS is similar, with DirLS producing slightly better results for the 2-objective TSP and about 15-20% better results in the case of 4-objective TSP. The PLS produced the worst results in all the tests, except for the 2-objective QAP with $n = 25$ and $\rho = +0.75$.

The influence of the correlation between flow matrices on the quality of solutions of the biobjective QAP problem

In [30] QAP instances were introduced in which there is a varying level of correlation between flow matrices. The same paper suggests that the level of correlation between flow matrices may influence the quality of results of the biobjective optimization. The results obtained in the experiments involving the DirLS, DLS and PLS were compared for different levels of correlation between flow matrices $\rho \in \{-0.75, -0.50, -0.25, 0, +0.25, +0.50, +0.75\}$. The results are summarized in Figures 2-4 which contain graphs presenting the hypervolume attained versus the correlation between flow matrices.

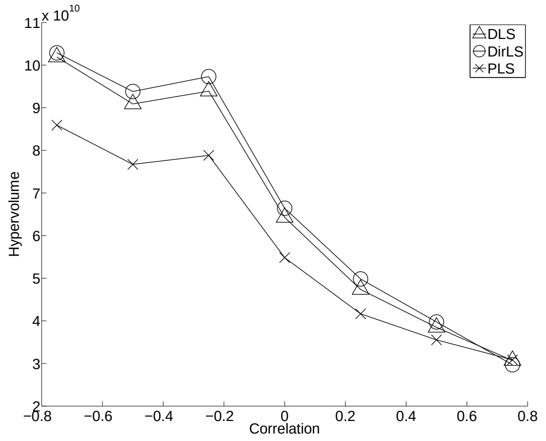

Figure 2. The attained median hypervolume versus the correlation between flow matrices for 2-objective QAP problems of size 25.

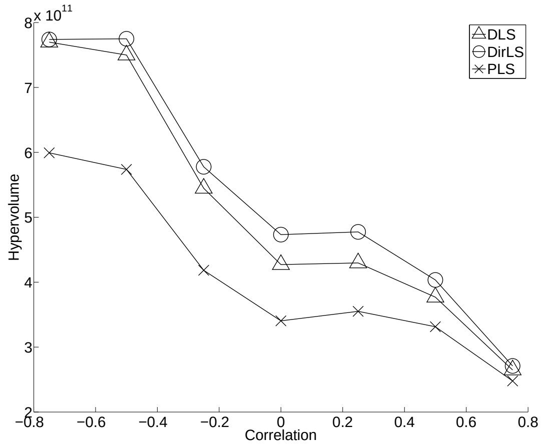

Figure 3. The attained median hypervolume versus the correlation between flow matrices for 2-objective QAP problems of size 50.

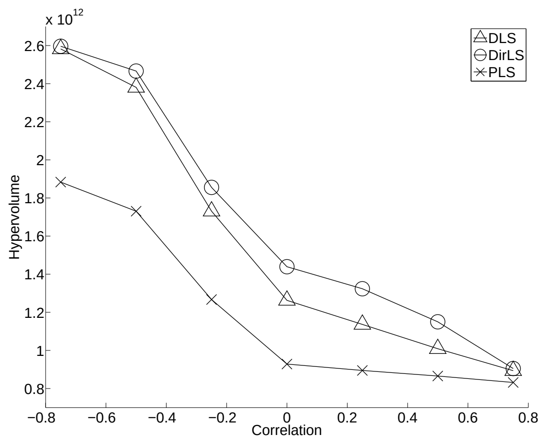

Figure 4. The attained median hypervolume versus the correlation between flow matrices for 2-objective QAP problems of size 75.

From the plotted data it can be concluded that for the instances with a high positive correlation between flow matrices it is harder to attain high hypervolume values. The Directional Local Search method produces better results in most cases, but the results given by all three methods deteriorate with the increasing correlation value. Since the correlation is measured between two flow matrices, the discussion about the influence of the correlation between flow matrices on the quality of solutions concerns a biobjective case only.

5. Conclusion

In this paper a Directional Local Search (DirLS) was presented that utilizes the knowledge concerning promising search directions. The proposed method was compared to two alternative approaches: one based on Pareto dominance (PLS) and the other based on decomposition (DLS). The experiments were carried out on 2-, 3- and 4-objective versions of two well-known combinatorial problems: the Travelling Salesman Problem (TSP) and the Quadratic Assignment Problem (QAP). The quality of solutions found by the compared methods was measured using the hypervolume indicator. The results obtained in the experiments prove that the DirLS method is more effective than the other two local search methods on both tested problems. In most cases it was the DirLS that produced the best results, and the analysis of the attained hypervolume versus the running time shows that DirLS produces better results than the two other approaches in the same computation time.

Further work may include combining search algorithms other than the 2-opt local search with the proposed framework. Also, it may be beneficial to use other decomposition methods, such as the Tchebycheff or boundary-intersection decomposition. The proposed method does not rely heavily on the working of the main metaheuristic algorithm so it may be used with other multiobjective optimization approaches as well.

6. Acknowledgement

This work was supported by the Polish National Science Centre under grant no. DEC-2012/05/D/HS4/01129.

References

[1] H. Abdi, *The Bonferroni and Sidak corrections for multiple comparisons*, in *Encyclopedia of measurement and statistics*, N.J. Salkind, ed., Sage Publications, Thousand Oaks, California, 2007, pp. 103–107.

[2] A. Alsheddy and E.E.P.K. Tsang, *Guided Pareto Local Search based frameworks for biobjective optimization*, in *IEEE Congress on Evolutionary Computation*, IEEE, 2010, pp. 1–8.

[3] E. Angel, E. Bampis, and L. Gourves, *A dynasearch neighborhood for the bicriteria traveling salesman problem*, in *Metaheuristics for Multiobjective Optimisation*, X. Gandibleux, M. Sevaux, K. Sorensen, and V. T'kindt, eds., Lecture Notes in Economics and Mathematical Systems, Vol. 535, Springer Berlin Heidelberg, 2004, pp. 153–176.

[4] M. Basseur, *Design of cooperative algorithms for multi-objective optimization: application to the flow-shop scheduling problem*, 4OR 4 (2006), pp. 255–258.

[5] C.W. Chiang, W.P. Lee, and J.S. Heh, *A 2-opt based differential evolution for global optimization*, Applied Soft Computing 10 (2010), pp. 1200 – 1207.

[6] I. Das and J.E. Dennis, *Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems*, SIAM Journal on Optimization 8 (1998), pp. 631–657.

[7] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Inc., New York, NY, USA, 2001.

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

[9] J. Derrac, S. Garca, D. Molina, and F. Herrera, *A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms*, Swarm and Evolutionary Computation 1 (2011), pp. 3–18.

[10] M. Fleischer, *The Measure of Pareto Optima. Applications to Multi-objective Metaheuristics*, in *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, EMO'03, Faro, Portugal, Springer-Verlag, Berlin, Heidelberg, 2003, pp. 519–533.

[11] J. Fliege and B.F. Svaiter, *Steepest descent methods for multicriteria optimization*, Mathematical Methods of Operations Research 51 (2000), pp. 479–494.

[12] A. Gunawan, H.C. Lau, and K. Lu, *An Iterated Local Search Algorithm for Solving the Orienteering Problem with Time Windows*, in *Evolutionary Computation in Combinatorial Optimization - 15th European Conference, EvoCOP 2015, Copenhagen, Denmark, April 8-10, 2015, Proceedings*, Lecture Notes in Computer Science, Vol. 9026, Springer, 2015, pp. 61–73.

[13] G. Gutin and A. Punnen (eds.), *The Traveling Salesman Problem and Its Variations*, Combinatorial Optimization, Springer, 2007.

[14] M.P. Hansen, *Tabu search for multiobjective optimization: MOTS*, in *Proceedings of the 13th International Conference on Multiple Criteria Decision Making*, Springer, 1997, pp. 574–586.

[15] K. Harada, J. Sakuma, and S. Kobayashi, *Local search for multiobjective function optimization: pareto descent method*, in *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, ACM, 2006, pp. 659–666.

[16] H. Ishibuchi and T. Murata, *A multi-objective genetic local search algorithm and its application to flowshop scheduling*, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 28 (1998), pp. 392–403.

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

[18] A. Jaszkiewicz, *Genetic local search for multi-objective combinatorial optimization*, European journal of operational research 137 (2002), pp. 50–71.

[19] Jerry Dallal, *Random permutations*, `http://www.jerrydallal.com/random/random_permutation.htm` (2015), online: accessed 2015.07.09.

[20] L. Ke, Q. Zhang, and R. Battiti, *A simple yet efficient multiobjective combinatorial optimization method using decompostion and pareto local search*, IEEE Trans on Cybernetics accepted (2014).

[21] J. Knowles and D. Corne, *Instance generators and test suites for the multiobjective quadratic assignment problem*, in *Evolutionary Multi-Criterion Optimization*, C.M. Fonseca, P.J. Fleming, E. Zitzler, L. Thiele, and K. Deb, eds., Lecture Notes in Computer Science, Vol. 2632, Springer Berlin Heidelberg, 2003, pp. 295–310.

[22] J. Knowles and D. Corne, *Memetic algorithms for multiobjective optimization: Issues, methods and prospects*, in *Recent Advances in Memetic Algorithms*, W.E. Hart, J. Smith, and N. Krasnogor, eds., Studies in Fuzziness and Soft Computing, Vol. 166, Springer Berlin Heidelberg, 2005, pp. 313–352.

[23] J.D. Knowles and D.W. Corne, *A comparison of diverse approaches to memetic multiobjective combinatorial optimization*, in *Proceedings of the 2000 genetic and evolutionary computation conference workshop program*, ACM, 2000, pp. 103–108.

[24] J.D. Knowles and D.W. Corne, *M-PAES: A memetic algorithm for multiobjective optimization*, in *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, Vol. 1, IEEE, 2000, pp. 325–332.

[25] H. Li and Q. Zhang, *Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II*, IEEE Transactions on Evolutionary Computation 13 (2009), pp. 284–302.

[26] T. Lust and J. Teghem, *Two-phase pareto local search for the biobjective traveling salesman problem*, Journal of Heuristics 16 (2010), pp. 475–510.

[27] K. Miettinen, *Nonlinear Multiobjective Optimization*, International Series in Operations Research and Management Science, Vol. 12, Kluwer Academic Publishers, Dordrecht, 1999.

[28] P. Mourdjis, P. Cowling, and M. Robinson, *Metaheuristics for the pick-up and delivery problem with contracted orders*, in *Evolutionary Computation in Combinatorial Optimisation*, C. Blum and G. Ochoa, eds., Lecture Notes in Computer Science, Vol. 8600, Springer Berlin Heidelberg, 2014, pp. 170–181.

[29] L. Paquete, M. Chiarandini, and T. Stutzle, *Pareto local optimum sets in the biobjective traveling salesman problem: An experimental study*, in *Metaheuristics for Multiobjective Optimisation*, X. Gandibleux, M. Sevaux, K. Sorensen, and V. Tkindt, eds., Lecture Notes in Economics and Mathematical Systems, Vol. 535, Springer Berlin Heidelberg, 2004, pp. 177–199.

[30] L. Paquete and T. Stutzle, *A study of stochastic local search algorithms for the biobjective {QAP} with correlated flow matrices*, European Journal of Operational Research 169 (2006), pp. 943–959, Available at `http://www.sciencedirect.com/science/article/pii/S0377221704005739`.

[31] L.S. Pitsoulis and P.M. Pardalos, *Quadratic assignment problem*, in *Encyclopedia of Optimization*, C.A. Floudas and P.M. Pardalos, eds., Springer, 2009, pp. 3119–3149.

[32] O. Sassi, W.R. Chérif-Khettaf, and A. Oulamara, *Multi-start Iterated Local Search for the Mixed Fleet Vehicle Routing Problem with Heterogenous Electric Vehicles*, in *Evolutionary Computation in Combinatorial Optimization - 15th European Conference, EvoCOP 2015, Copenhagen, Denmark, April 8-10, 2015, Proceedings*, Lecture Notes in Computer Science, Vol. 9026, Springer, 2015, pp.

138–149.

[33] P. Serafini, *Simulated annealing for multi objective optimization problems*, in *Multiple criteria decision making*, Springer, 1994, pp. 283–292.

[34] N. Srinivas and K. Deb, *Multiobjective optimization using nondominated sorting in genetic algorithms*, Evolutionary Computation 2 (1994), pp. 221–248.

[35] B. Suman, *Study of simulated annealing based algorithms for multiobjective optimization of a constrained problem*, Computers & Chemical Engineering 28 (2004), pp. 1849–1871.

[36] G. Tao and Z. Michalewicz, *Inver-over Operator for the TSP*, in *in Proc. PPSN*, Springer, 1998, pp. 803–812.

[37] Q. Zhang and H. Li, *MOEA/D: A multiobjective evolutionary algorithm based on decomposition*, IEEE Transactions on Evolutionary Computation 11 (2007), pp. 712–731.

[38] E. Zitzler, M. Laumanns, and L. Thiele, *SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization*, in *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*, International Center for Numerical Methods in Engineering (CIMNE), 2002, pp. 95–100.

[39] E. Zitzler and L. Thiele, *Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach*, IEEE Transactions on Evolutionary Computation 3 (1999), pp. 257–271.

[40] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, and V.G. da Fonseca, *Performance assessment of multiobjective optimizers: An analysis and review*, IEEE Transactions on Evolutionary Computation 7 (2002), pp. 117–132.