Towards Standardized and Seamless Integration of Expert Knowledge into Multi-objective Evolutionary Optimization Algorithms

Magdalena A.K. Lang[1] and Christian Grimme[2(✉)]

[1] RWTH Aachen University, Kackertstr. 7, 52072 Aachen, Germany
magdalena.lang@ada.rwth-aachen.de

[2] Westfälische Wilhelms-Universität Münster, Leonardo-Campus 3, 48149 Münster, Germany
christian.grimme@uni-muenster.de

Abstract. Evolutionary algorithms allow for solving a wide range of multi-objective optimization problems. Nevertheless for complex practical problems, including domain knowledge is imperative to achieve good results. In many domains, single-objective expert knowledge is available, but its integration into modern multi-objective evolutionary algorithms (MOEAs) is often not trivial and infeasible for practitioners. In addition to the need of modifying algorithm architectures, the challenge of combining single-objective knowledge to multi-objective rules arises. This contribution takes a step towards a multi-objective optimization framework with defined interfaces for expert knowledge integration. Therefore, multi-objective mutation and local search operators are integrated into the two MOEAs MOEA/D and R-NSGAII. Results from experiments on exemplary machine scheduling problems prove the potential of such a concept and motivate further research in this direction.

Keywords: Multi-objective evolutionary algorithm · Expert knowledge integration · MOEA/D · R-NSGAII · Scheduling

1 Introduction

Multi-Objective Evolutionary Algorithms (MOEAs) aim to obtain useful solutions for real-world optimization problems in a reasonable amount of time. Different quasi-standard approaches have been introduced in research, for instance, the Non-dominated Sorting Genetic Algorithm II (NSGAII) [3] and the Multi-Objective Evolutionary Algorithm Based on Decomposition (MOEA/D) [20].

However, to increase MOEA acceptance in practice, these metaheuristics have to improve regarding reliability and computational efficiency for complex problems. It is widely acknowledged that including problem specific heuristics supports the search and allows faster convergence towards optimal solutions [5, 12, 16]. For instance, in application fields such as scheduling, a broad theoretically founded knowledge base is available which can help to tackle subproblems of multi-objective optimization tasks [5,6].

H. Trautmann et al. (Eds.): EMO 2017, LNCS 10173, pp. 375–389, 2017.
DOI: 10.1007/978-3-319-54157-0_26

In single-objective optimization, expert knowledge integration is from an algorithmic point of view rather simple and commonly applied via special search operators [2]. However, for multi-objective problems, domain experts face the challenge to effectively introduce their mostly subproblem-oriented knowledge. In particular, the question arises of *when* to apply *which problem-specific operators* on *which individuals* in case only heuristics for single objectives are available. Thus, to utilize the full potential of MOEAs in practice, we suggest a reusable and easily deployable optimization framework that allows the integration of problem-specific expert knowledge by simple interfaces without requiring to redesign the underlying algorithm. An internal logic needs to effectively apply the problem specific heuristics in order to leverage the potential of both the metaheuristic and the problem specific heuristics. In the following, we call such a framework *extended or adapted MOEA*.

Several successful applications of MOEAs integrating expert knowledge already exist. Nevertheless, authors like Peng and Zhang [14] or Konstantinidis and Yang [10] apply specific algorithm architectures that are only designed for the given problems. Grimme et al. [5] follow the idea of an extended MOEA and successfully implement a modular integration of single-objective knowledge into a multi-objective predator-prey model via variation operators. Until now, the more popular, dominance-based MOEAs are regarded as rather unsuitable for such a general adaption, due to their monolithic structure and focus on the selection mechanism [5,18]. Nevertheless, more recent research advances, such as introducing reference points [4] or decomposition-based algorithmic structures [20], motivate to check their potential and openness for a respective framework.

This contribution takes a step towards an extended MOEA by proposing to integrate multi-objective mutation and local search modules as reusable interfaces for subproblem-specific heuristics. The modules are integrated into two MOEAs, MOEA/D and R-NSGAII, which especially recommend themselves for a practical framework. That is, they allow both a-priori and a-posteriori optimization. An experimental approach to multiple a-posteriori parallel machine scheduling problems allows to examine and compare the performance of the adapted frameworks in contrast to plain general-purpose MOEAs. The performance indicators consider the quality and reliability of provided solutions as well the as speed of convergence.

2 Scheduling Problems and Dispatching Rules

Scheduling is '*an optimization process by which limited resources are allocated over time among parallel and sequential activities*' [1, p. 1]. In the following, we focus on machine scheduling because most theoretical models and concepts refer to this class and commonly accepted optimization criteria exist [16]. Additionally, we restrict ourselves to offline, non-preemptive, and non-delay settings, where jobs cannot be paused and unforced idleness of machines is not allowed.

2.1 Notation

A schedule comprises both the assignment of jobs to machines and their processing order [15]. A job j is a task mainly characterized by its processing time $p_{i,j}$ on machine i, due date d_j, and release date r_j. For a given schedule, a job has a start time S_j and a completion time $C_j = S_j + p_{i,j}$.

A scheduling problem is stated via the popular three-field notation $\alpha|\beta|\gamma$, with α being the machine environment of the problem, β describing problem constraints, and γ enumerating the considered objectives. This study focuses on the P_m machine environment, i.e. identical parallel machines, for which a job consists of a single operation and can be processed on any of the available machines with processing time p_j [15]. As objectives, this study considers the minimization of four regular criteria, which are among the most popular in literature and important in practice. The *makespan* C_{max} denotes the total duration of a schedule, i.e. $C_{max} = \max_{j=1,...,n}(C_j)$ with n as the number of jobs. It aims at balancing the load on the identical parallel machines in P_m [15]. In contrast, the *total completion time objective* $C_{sum} = \sum_{j=1}^{n} C_j$ strives for completing all jobs as fast as possible. The due date-related *maximum lateness* objective $L_{max} = \max_{j=1,...,n}(L_j)$ with $L_j = C_j - d_j$ minimizes the largest due date violation. When considering $T_{sum} = \sum_{j=1}^{n} T_j$, the *total tardiness* with $T_j = \max(0, L_j)$ needs to be minimized. It demands that no delay of any job is unacceptably long [15].

Scheduling problems usually strike for combinatorial solutions. For multi-objective problems with regular criteria without preemption, the number of optimal points is finite and rather small in contrast to continuous Pareto fronts [1,7]. Most scheduling problems are (weakly or strongly) NP-hard or open [7]. References such as [15] provide details about complexity hierarchies.

2.2 Dispatching Rules

In practice, dispatching jobs to machines often follows specific rules, which require little computational time and exactly or approximately solve basic scheduling problems. These rules usually rely on job attributes or machines conditions. The *Shortest Processing Time* (SPT) rule sorts the jobs in ascending order of processing time p_j, leading to an optimal solution for the $1||C_{sum}$ problem [17]. In conjunction with the *First Available Machine* (FAM) rule, which assigns the next job to the first available machine in a parallel machine environment, also $P_m||C_{sum}$ is solved to optimality [15]. The reverse of SPT, the *Longest Processing Time* (LPT) rule, heuristically balances the load on parallel machines under the FAM assignment [15]. It is easy to show that the upper bound for this heuristic is $4/3 - 1/(3m)$ times the optimal value of C_{max}.

The *Earliest Due Date* (EDD) rule and *Minimal Slack* (MS) rule aim on solving due date related criteria. The EDD rule sorts jobs in ascending order of due dates and optimally solves the $1||L_{max}$ problem [15]. In contrast to the static rules mentioned before, the MS rule is a so-called dynamic rule [15]. Each time a job needs to be assigned, the job j with the minimum $slack_j = max(0, d_j - p_j)$ is chosen next.

If release dates are present, the *Earliest Release Date* (ERD) rule sorts the jobs according to increasing release dates. So-called *domination results* can help to reduce the search space of a scheduling problem. When considering $1||T_{sum}$, we know that *if* $p_j \leq p_k$ *and* $d_j \leq d_k$*, then there exists an optimal sequence in which job* j *is scheduled before job* k [15, p. 51]. This is further on called DR rule.

For multi-objective scheduling problems, only few efficient algorithms exist. Wassenhove and Gelders [19] introduced an approach to optimally solve the $1||C_{sum}, L_{max}$ problem with $\mathcal{O}(n^3 log(n))$ complexity [7]. The approach leverages respective single-objective dispatching rules to acquire the Pareto front.

3 Methodology

The extended MOEA approach aims to find a better approximation of the Pareto front, i.e. the *approximated front*, for the minimization or maximization of $F(x) = (f_1(x), f_2(x), ..., f_k(x))$ with k objectives, objective functions $f_i(x)$, $i = 1, ..., k$, and x as feasible solution. Thereby, the common quality criteria *convergence to the Pareto front* as well as *diversity* of the approximated front, including the number of solutions, are considered. Secondary criteria are the *speed of convergence* towards good solutions as well as the *reliability* of an approach regarding robustness of the result quality.

3.1 Multi-objective Modules

Similar as for existing successful approaches to integrate multi-objective expert knowledge [6,10,14], subproblem-specific expertise is combined by a successive application of alternate subproblem dependent heuristics. They manipulate parts of an individual's genotype. The proposed *multi-objective modules* (MOM) extend the approach by [10].[1] For each objective, the module provides adapters for registering subproblem-specific operators. Figure 1 illustrates the concept for a mutation module (MOM-M). The domain expert can provide single-objective mutation operators to facilitate the local improvement of a temporary solution to the respective objective by small changes in the genotype of the considered individual. If an operator fits multiple objectives, it can be registered on several adapters. Thus, this approach has the advantage that it allows to integrate both single-objective knowledge as well as problem specific improvement mechanisms supporting combinations of objectives. Thereby, constraint-related rules as the ERD rule, concerning multiple objectives, can easily be added.

Furthermore, the module has an interface to the base MOEA. It receives an individual of the population as well as the information needed for the so-called *assignment strategy*. The latter is a logic within the module that decides

[1] Note, however, that we focus on the more general context of applicability in MOEAs. Thus, the proposed methodology is not restricted to specific applications but strive for identifying additional integration points of expertise.

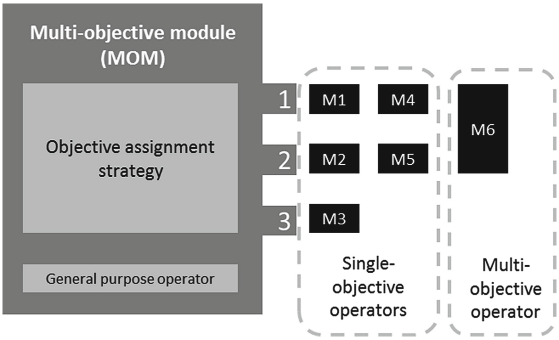

Fig. 1. Schema of the Multi-objective Mutation Module (MOM)

which objective to support for the current individual. Subsequently, the module randomly chooses one operator of the respective queue. The most simple strategy randomly selects one objective. Alternatively, if the individual belongs to a specific weighting of objectives as in a decomposition algorithm like MOEA/D, the module can set selection probabilities according to the weighting of considered objectives. Table 1 lists the assignment strategies considered here. Strategy applicability depends on the underlying algorithm.

Moreover, the MOM can allow a certain probability to apply general purpose operators like swap mutation to facilitate diversity. These do not explicitly bias towards specific objectives but apply undirected random adaption. Similar to [8], this probability is an external parameter.

Table 1. Objective assignment strategies of a MOM

Assignment strategy	Description	MOEA
Random assignment (`random`)	Random selection of the objective	MOEA/D
		R-NSGAII
Probabilistic assignment based on objective weights (`weights prob`)	Weighting of objectives is transferred to selection probabilities for the respective objectives	MOEA/D
Deterministic assignment based on objective weights (`weights det`)	Objective with highest weight is selected (random draw on ties)	MOEA/D
Worst objective assignment (`worst obj`)	The currently worst performing objective is selected, either by looking at	
	1. The worst weighted objective value	MOEA/D
	2. The largest distance to the assigned reference point	R-NSGAII

Such a modular interface combines low maintenance effort and flexible reconfiguration of individual modules, without a need to adapt the overall algorithm structure. Therefore, different implementations of modules using various assignment strategies can be tested. A MOM can be integrated at different parts of the MOEA structure to serve as mediator between the metaheuristic and the problem specific heuristics provided by the user. We evaluate such a MOM as mutation operator (MOM-M) as well as for an additional local search (MOM-L) starting from individuals of the population.

A local search module facilitates intensification. Next to choosing problem specific techniques, it applies an internal procedural logic to decide for further actions after producing a new candidate solution. This study uses a greedy strategy [8,9]. Therein, the local search module receives a solution from the population as a starting point. A neighbor of this solution is produced via a problem specific heuristic, which biases the neighborhood structure of a solution. Thereafter, the module compares fitness values of the old and the new solution based on a fitness function it receives from the underlying algorithm. If the new solution is better or as good as the original one, it serves as new starting point for the next local search step. The overall number of steps as well as the allowed number of steps without improvement are external termination parameters. Our preliminary experiments detected an advantage for accepting new solutions as intermediate step towards better individuals, if they have the same fitness. Note that the problem specific heuristic is selected only once at the beginning of each local search process and pursued for all following steps.

3.2 Adapting MOEA/D

MOEA/D [20] was already successfully extended by single-objective expert knowledge (e.g. in [10,14,18]). It is a decentralized MOEA dividing a MOP into scalar sub-problems each explicitly addressing a specific weighting of the objectives. Per generation, the sub-problems are subsequently handled in a single-objective manner and interact in terms of crossover and selection mechanism with their neighboring subproblems to benefit from their findings.

The weights of the scalar functions can be interpreted as relative prioritization of the respective objectives and determine the search direction [18]. A method like the weighted sum approach (see [11]) or the Tchebycheff approach (see [20]) can serve to determine the fitness of an individual. The latter uses a reference point like the ideal vector, consisting of the best values of the individual objectives, as a goal and aims on minimizing the maximum weighted distance between the objectives of a solution and this point. As in the original MOEA/D, we use the Tchebycheff method as scalarizing function, especially because it provides more connecting points for the assignment of expert knowledge.

The MOMs are integrated as follows. For MOM-M, the algorithm initiates the respective module as mutation operator within its reproduction cycle. Still, the module needs to know about the sub-problem under consideration to decide which single-objective operator to choose. Different possibilities are considered here. First, the module assigns operators randomly (`random`).

Second, the algorithm informs the module about the respective weight vector and the module decides either based on the highest weight (`weights det`) or randomly (`weights prob`) for a sub-problem specific operator. Last, the algorithm passes on the worst objective based on the Tchebycheff rule (`worst obj`), see Table 1.

For better exploitation, local search is periodically initialized after a complete generation was determined. All sub-problems are traversed and each individual is used as possible start solution. If the local search module returns a better performing individual for the subproblem, the old solution is replaced. Therefore, the algorithm provides a comparator which evaluates the fitness of the solutions based on the respective weight vector. The assignment strategies are analog to the mutation module.

3.3 Adapting R-NSGAII

From a practical point of view, the reference point based adaption R-NSGAII [4] of NSGAII [3] plays an important role. It allows to integrate preferences in form of one or more reference points guiding search within the objective space.

Two assignment strategies are most obvious for R-NSGAII. First, the random assignment (`random`) and second, analogously to the worst objective assignment in MOEA/D, an allocation based on the objective which performs worst in comparison to the reference point assigned to an individual (`worst obj`). As the reference points do not need to be optimal, an individual can be better in one or more objectives than the respective reference point $\bar{z}$. Therefore, the objective solving $max_i(f_i(x) - \bar{z}_i)$ for $i = 1, ..., k$ is targeted. As main difference to MOEA/D, the individuals are not constantly assigned to one reference point. Instead, the allocation can change over time. Moreover, several individuals can be allocated to the same point and the selection mechanisms do not only compare individuals in similar search directions but the population as a whole.

As MOEA/D, R-NSGAII applies local search periodically to all individuals after a predefined number of generations. The base population and the new individuals are collected in one combined set and the usual generational selection mechanism is applied to determine the population for the next recreation loop. The algorithm provides the module with a comparator which compares individuals based on non-domination and distances to the respective reference points.

3.4 Subproblem-Specific Operators

The following expert knowledge based operators help to test the performance of the multi-objective procedure. Note that they are not themselves in focus of the current analysis. In principle, they build on generic mechanisms allowing to integrate order based rules into a permutation encoding of an individual.

First, the permutation based mutation operator proposed by Grimme et al. [6] allows to integrate scheduling knowledge into the MOM-M. The operator randomly selects a position within the permutation and sorts all jobs within a range δ around this position according to an assigned dispatching rule. Thereby, an overall amount of $2\delta + 1$ positions is in focus during one mutation operation. The concrete value of δ is determined randomly based on a normal distribution, with $\mu = 0$ and σ as external parameter.

For local search, the neighborhood is biased towards individuals which are likely to improve in the respective objective by considering the appropriate dispatching rule. Thereby, a random position j within the permutation is selected and used as starting point to traverse the following job indices for finding a job which can be inserted at position j such that all following jobs are shifted backwards by one position. For instance, the SPT-biased neighborhood approach looks for a job with a lower processing time than the one at random position j. The number of checks can be implemented as predefined parameter which influences the probability to swap jobs and thus, the possible impact of the operator similar to the σ parameter in the mutation operator. While such a general approach is used for the SPT, MS, DR, and ERD rules, more effective, deterministic methods are adopted for the C_{max} and the L_{max} objectives because their respective values are determined by single jobs. Therefore, the phenotype of the individual is used to identify the job causing the value of the objective in focus. The phenotype is the schedule with assignment of jobs to machines, obtained by applying the FAM rule to the permutation. The approach tries to reposition this job by backward traversion, searching for a job with lower processing time or due date, respectively, to prepend it.

The new individual adopting the resulting job order serves as neighbor of the individual in focus. It is evaluated by the local search module to decide on its acceptance as basis for another local search round or as replacement of the start individual in the population.

4 Experimental Analysis

Computational experiments on machine scheduling problems help to evaluate extended MOEA/Ds and R-NSGAIIs.

4.1 Experimental Setup

For each scheduling problem, 40 test instances serve as the basis for a statistical analysis of MOEA performances. Half of the instances consist of 50 jobs and the other half of 100 jobs. The processing times are randomly drawn realizations of a uniform distribution, i.e. $p_j \sim [\mathcal{U}(1, 49)]$. Half of both 50 job and 100 job instances consist of jobs with a due date offset d_j^o between 0 and 100 ($d_j = r_j + d_j^o$ with $d_j^o \sim [\mathcal{U}(0, 100)]$). The other half is less restrictive with $d_j^o \sim [\mathcal{U}(0, 200)]$. In addition, a special 50 job instance results from due date offsets between 0 and 999 to obtain a wider and more populated Pareto front for the $1||C_{sum}, L_{max}$ problem. Individual genotypes are permutation encoded. Each element of the permutation resembles a job. Jobs are assigned in a first available machine (FAM) manner to parallel machines during evaluation.

Table 2. Experimental settings and algorithmic parameters.

Attribute	Final setting	Attribute	Final setting
Individual representation	Permutation encoding	Variation operators	Order crossover and swap mutation
Initial population	Random permutations	Crossover probability	0.9
Population size μ	100 (2 objectives) or 300 (3 objectives)	Mutation probability	0.8
Termination criterion	25,000 (2 objectives) and 75,000 (3 objectives) individuals	Specific parameters	Neighorhood size: 20 (MOEA/D); Epsilon: 0.001 (R-NSGAII)
Selection operators	Binary tournament (R-NSGAII)	Maximum number of checks in local search	10 % of the number of jobs
Number of instances	20 instances with 50 jobs, 20 instances with 100 jobs	Number of runs per experiment	30

Table 2 summarizes relevant settings like variation operators and population size. Most settings result from a preliminary parameter tuning, suggesting same best settings for both algorithms. They are used throughout all experiments unless otherwise explicitly stated.

4.2 Performance Evaluation

The *Hypervolume Indicator (HV)* [21], the *Unary Additive Epsilon Indicator (Epsilon)* [21], and the *number of non-dominated solutions (Count)* serve as performance indicators for the quality of the approximated front. Additionally, an indicator *Endpoint* measures the number of generations after which no significant improvement in HV (<2%) exists anymore, compared to the mean final HV of all runs. The indicator *Comparison Reached Point (CRP)* states when the mean final HV of the respective base MOEA was reached, while the *Comparison Reached Rate (CRR)* measures the quota of runs reaching this HV value. The latter indicators mainly measure the speed of convergence of algorithms.

Every algorithm was applied to each test instance 30 times to account for stochastic effects. The average value of each indicator over the 30 runs serves as result per instance. We look at mean and standard deviation over the 40 test instances to present the overall performance of the algorithm. Thereby, we present the indicators as quota over the respective base algorithm outcome, with values above one indicating a larger indicator value. As the Epsilon and CRR indicators can be zero, we present differences instead of quotas.

We conduct non-parametric tests to check the statistical significance of differences over the 40 instances (significance level of $p \leq 0.05$). The *Friedman test* and the paired *Wilcoxon Rank Sum test (Wilcoxon test)* with *Bonferroni correction* check for significant differences in mean, the *Fligner-Killeen test (Fligner test)* helps to identify whether two algorithm outcomes differ significantly in variance.

Normalization balances the influence of incommensurable objectives on quality indicator values. Per instance, the best and worst objective values of all approximated fronts serve as boundary values. Furthermore, the combined best approximated front serves as reference front for the Epsilon indicator and the worst objective values form the reference point for the HV indicator. Note that this approach circumvents the comparability of results across experiments.

For the implementation of algorithms, we adopted the Java-based jMetal framework [13] (Java 8 Update 65). Data analysis approaches as significance tests and visualizations were mainly conducted via R (version 3.2.2).

4.3 Results

We present representative results with main focus on MOEA/D that sufficiently provide relevant insights for both algorithms with and without multi-objective modules applied. For illustration purposes, the computational easy $1||C_{sum}, L_{max}$ is used as for this problem approximated solutions can be compared to the true Pareto front. Then, the analysis is supported by considering an NP-hard parallel machine problem $5||C_{max}, C_{sum}$ and the also NP-hard three objective problem $1||C_{sum}, L_{max}, T_{sum}$.

Multi-objective Mutation Module (MOM-M). For $1||C_{sum}, L_{max}$, SPT and EDD rule based single-objective mutation operators are used to sort jobs within a randomly selected range of the permutation. Figure 2 illustrates the best and worst approximated fronts over 30 runs for MOEA/D and MOEA/D-MOM-M (`random`) on a 50 jobs instance. The results look similar for the other assignment strategies. While MOEA/D focuses on the region that primarily supports the C_{sum}-objective, the extended algorithms find solutions spread over the whole front but can have some problems to converge to optimal extreme solutions regarding L_{max} values. The HV development over the runtime is exemplarily visualized in Fig. 3(a). Apparently, the different assignment strategies are

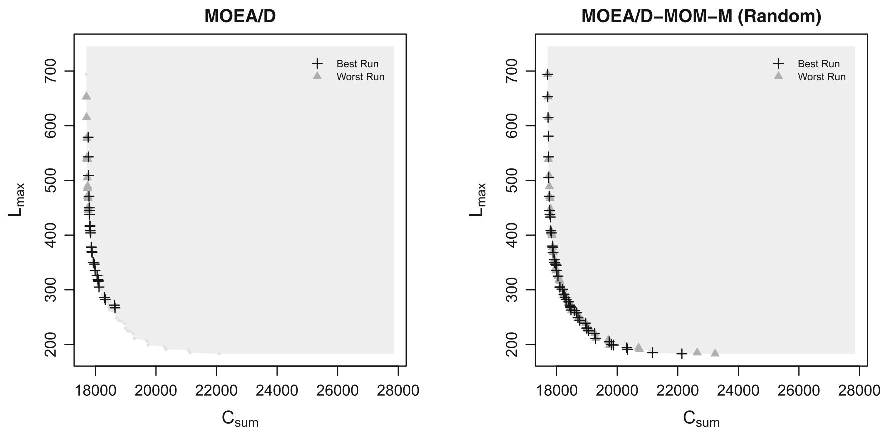

Fig. 2. Pareto front and approximated fronts for $1||C_{sum}, L_{max}$ with 50 jobs, solved by MOEA/D and MOEA/D-MOM-M (random) ($\sigma = 10$, general purpose operators: 0%)

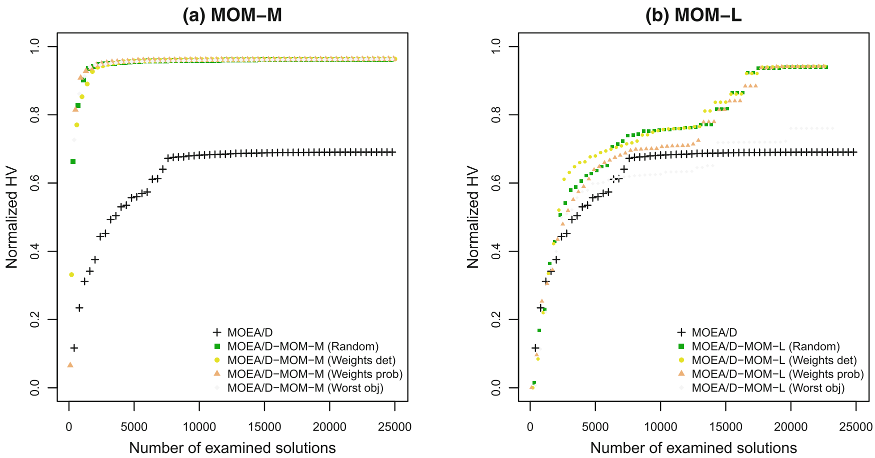

Fig. 3. HV development over the runtime of algorithms for $1||C_{sum}, L_{max}$ problem with 50 jobs for median performing runs of MOEA/D-MOM-M ($\sigma = 10$, general purpose operators: 0%) and MOEA/D-MOM-L

similar in their convergence speed, increasing much faster than the unadapted MOEA/D and ending with higher HV values.

The extended versions outperform MOEA/D in all considered indicators according to Friedman and Wilcoxon tests. The descriptive statistics in Table 3 state, for instance, that HV increases by around 7% and that at most 14% of generations are needed to achieve comparable results to MOEA/D. In general, the assignment strategies do not provide significantly differing results, only `weights det` is outperformed for the Epsilon indicator and converges last. `Worst obj` improves faster than the other adaptions, reaching the mean final HV of MOEA/D and its own endpoint significantly earlier. The Fligner test reveals that, except from HV, the adapted versions perform more robust across different problem instances than MOEA/D.

The R-NSGAII approach applied here uses evenly spread, normalized reference points to allow a search for the whole Pareto front. The analysis over 40 test instances reveals significantly better solutions of the extended R-NSGAII over the base algorithm in all indicators. Table 3 states that the HV values increase by more than 45%. `Worst obj` achieves a significantly faster convergence towards the mean final HV of R-NSGAII and its own endpoint than `random`. Furthermore, both alternatives are more robust than the unadapted R-NSGAII in Epsilon and the time related indicators.

For $5||C_{max}, C_{sum}$, two reverse dispatching rules are needed, i.e. the LPT rule for the first objective and the SPT rule for the second. Because the LPT-rule based mutation is not optimal for the C_{max} objective and the reverse mutations are probably very disruptive, the integration of partly swap mutation instead of problem-specific mutation was advantageous in this case. Table 4 presents the descriptive statistics of the two experiments on the 40 test instances.

Table 3. Descriptive statistics of performance indicators for $1||C_{sum}, L_{max}$ problems (40 instances, $\sigma = 10$, general purpose operators: 0%). Note that MOEA/D and R-NSGAII experiment indicators relate to the respective experiment only and must not be compared.

	Experiment 1: MOEA/D				Experiment 2: R-NSGAII	
	MOEA/D-MOM-M				R-NSGAII-MOM-M	
	Random	Weights det	Weights prob	Worst obj	Random	Worst obj
HV quota mean	1.07028	1.06613	1.06961	1.06999	1.45713	1.45760
HV quota std.	0.11922	0.12237	0.11980	0.11927	0.59446	0.59442
Epsilon diff. mean	−0.05762	−0.05313	−0.05642	−0.05719	−0.21435	−0.21542
Epsilon diff. std.	0.08227	0.08491	0.08262	0.08212	0.23810	0.23786
Count quota mean	4.41897	3.35208	3.80007	4.55546	11.48547	11.74812
Count quota std.	3.57277	1.99379	2.62486	3.55111	12.72062	13.19304
CRR quota mean	1.13997	1.13498	1.14041	1.14398	1.33191	1.33370
CRR quota std.	0.18320	0.18397	0.17829	0.17114	0.36050	0.35854
CRP quota mean	0.10867	0.13612	0.11501	0.10122	0.11892	0.11141
CRP quota std.	0.06440	0.10302	0.07710	0.07895	0.07812	0.06627
Endpoint quota mean	0.12565	0.16001	0.13705	0.11804	0.14360	0.13400
Endpoint quota std.	0.06242	0.10423	0.08664	0.08458	0.07944	0.06991

Table 4. Descriptive statistics of performance indicators for $5||C_{max}, C_{sum}$ problems (40 instances, $\sigma = 10$, general purpose operators: 20%). Note that MOEA/D and R-NSGAII experiment indicators relate to the respective experiment only and must not be compared.

	Experiment 1: MOEA/D				Experiment 2: R-NSGAII	
	MOEA/D-MOM-M				R-NSGAII-MOM-M	
	Random	Weights det	Weights prob	Worst obj	Random	Worst obj
HV quota mean	1.05078	1.04538	1.04865	1.05182	1.06638	1.06968
HV quota std.	0.08386	0.08486	0.08377	0.08346	0.07934	0.07875
Epsilon diff. mean	−0.04173	−0.02734	−0.03511	−0.04529	−0.06258	−0.06761
Epsilon diff. std.	0.06339	0.06364	0.06231	0.06215	0.06241	0.06220
Count quota mean	1.99265	1.86450	1.85238	1.98604	3.35462	3.91708
Count quota std.	1.26462	1.16177	1.10300	1.20368	2.60105	3.06321
CRR quota mean	1.15593	1.10869	1.13627	1.16476	1.20999	1.22278
CRR quota std.	0.16691	0.20075	0.18253	0.16551	0.19966	0.19590
CRP quota mean	0.35655	0.45816	0.41022	0.32179	0.35559	0.34094
CRP quota std.	0.13671	0.22237	0.18336	0.10197	0.11080	0.09015
Endpoint quota mean	0.42133	0.52316	0.48718	0.38758	0.44246	0.42077
Endpoint quota std.	0.07969	0.14092	0.12174	0.07047	0.06842	0.07056

The significance tests do not only suggest improvements in all indicators for the extended algorithms, but also a ranking between the different assignment strategies. For MOEA/D, `worst obj` results in best HV and EPSILON values. `Random` is second best and reaches the mean final HV of MOEA/D together with `worst obj` first. The comparatively poor performance of `weights det` possibly results from the missing support of the lower weighted objective. Additionally, all adapted algorithms are more robust in their end HV and Epsilon indicators than MOEA/D.

Multi-objective Local Search Module (MOM-L). We apply local search only after half of the maximum evaluations termination criterion is reached, allowing the MOEA to find promising regions beforehand. Afterwards, local search is activated every ten generations to facilitate intensification. The local search step size is five, while the maximum number of subsequent non-improving individuals is two.[2] To allow a fair comparison between approaches with and without local search, the number of intermediate individuals created during the procedure adds to the overall number of individuals of the algorithm. Figure 3(b) shows the HV development of the median runs for $1||C_{sum}, L_{max}$ and the effects of local search. While MOEA/D-MOM-L `worst obj` falls behind the other extended MOEA/Ds, all extended versions show improvement and perform better than simple MOEA/D.

The Friedman test finds significant improvements in all indicators for the MOEA/D and R-NSGAII experiments. For MOEA/D, no significant differences between the extended algorithms exists according to the Wilcoxon test. All adapted MOEA/Ds are more robust in their comparison reached rate than MOEA/D across problem instances. Additionally, `random` and `worst obj` are less variable in the Epsilon indicator.

For R-NSGAII, both extended algorithms significantly outperform R-NSGAII in all indicators and are more robust across problem instances.

For $5||C_{max}, C_{sum}$, MOEA/D-MOM-L `random` and `weights det/prob` significantly outperform MOEA/D in all indicators. Only in CRR, `worst obj` can outperform the original MOEA/D. According to the Fligner test, the weight-based assignment strategies are more robust in the CRR than MOEA/D and MOEA/D-MOM-L `worst obj`. For R-NSGAII, there are small but significant improvements in all indicators for applying MOM-L. Thereby, the two assignment strategies do not significantly differ.

Three-Objective Problems. Finally, the up to now best performing versions of MOEA/D and R-NSGAII, i.e. MOEA/D-MOM-M `worst obj` and R-NSGAII-MOM-M `worst obj`, are used to analyze their performance on three-objective problems. $1||C_{sum}, L_{max}, T_{sum}$ helps to analyze how two objectives with optimal scheduling rules interact with one objective which is NP-hard to solve in any machine setting. In addition, the domination rule for T_{sum} can be beneficial for the other objectives and is therefore assigned to all objectives. The EDD and the MS rule are applied to both due date related objectives.

For both algorithms, the extended variants can significantly improve in the quality of the approximated front and the speed towards the final end HV of the base algorithm. Additionally, they are more robust in their results.

[2] This configuration was applied without fine tuning to demonstrate the methodology. Future rigorous investigation may consider systematically generated configurations.

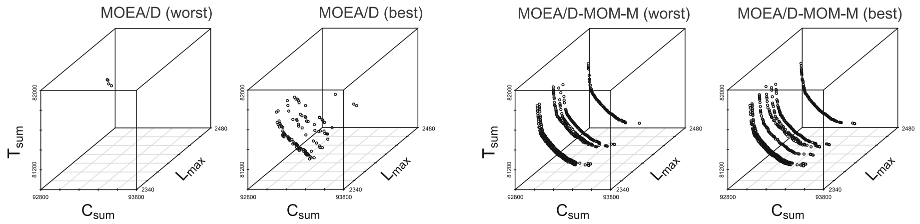

Fig. 4. Approximated fronts of MOEA/D and MOEA/D-MOM-M (`worst obj`) for $1||C_{sum}, L_{max}, T_{sum}$ problem with 100 jobs

A closer look at one of the instances for the MOEA/D experiment provides an insight into the resulting approximated fronts, see Fig. 4. Apparently, the results of the adapted MOEA/D are far more robust whereas MOEA/D performs very poor in the worst run. Additionally, the approximated fronts of the extended MOEA/D are more diverse, here best visible for the combination of C_{sum} and T_{sum} values.

5 Conclusion

The presented idea of a multi-objective algorithmic framework for ad-hoc integration of expert knowledge uses multi-objective modules that apply subproblem-specific operators based on different assignment strategies. They provide standardized interfaces to allow a seamless integration of domain knowledge by practitioners. Experimental results suggest that the modules can significantly improve the performance of dominance based R-NSGAII and MOEA/D regarding the quality of the approximated front, the convergence speed towards good solutions, as well as the reliability of results. Overall, we prove the potential of such a framework with respect to *usability* and *performance*.

Still, open research issues exist that need to be approached in future work. For instance, the question arises whether the successful integration of expert knowledge is transferable to other scheduling problems or even other application domains of multi-objective optimization. More advanced approaches like adaptive operators can be tested as well. Thereby, other base algorithms and combinations of different integration points may come into play. Furthermore, the underlying forces between the effects seen in the presented experiments should be further investigated to improve the understanding of the problem domain.

References

1. Bagchi, T.P.: Multiobjective Scheduling by Genetic Algorithms. Springer, New York (1999)
2. Burke, E.K., Gendreau, M., Hyde, M.R., Kendall, G., Ochoa, G., Özcan, E., Qu, R.: Hyper-heuristics: a survey of the state of the art. JORS **64**(12), 1695–1724 (2013)
3. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)

4. Deb, K., Sundar, J., Udaya Bhaskara Rao, N., Chaudhuri, S.: Reference point based multi-objective optimization using evolutionary algorithms. Int. J. Comput. Intell. Res. **2**(3), 273–286 (2006)
5. Grimme, C., Kemmerling, M., Lepping, J.: On the integration of theoretical single-objective scheduling results for multi-objective problems. In: Tantar, E., Tantar, A.A., Bouvry, P., Del Moral, P., Legrand, P., Coello Coello, C., Schuetze, O. (eds.) EVOLVE- A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation, Studies in Computational Intelligence, vol. 447, pp. 333–363. Springer, Heidelberg (2013)
6. Grimme, C., Lepping, J., Schwiegelshohn, U.: Multi-criteria scheduling: an agent-based approach for expert knowledge integration. J. Sched. **16**(4), 369–383 (2013)
7. Hoogeveen, H.: Multicriteria scheduling. Eur. J. Oper. Res. **167**(3), 592–623 (2005)
8. Ishibuchi, H., Hitotsuyanagi, Y., Tsukamoto, N., Nojima, Y.: Use of heuristic local search for single-objective optimization in multiobjective memetic algorithms. In: Rudolph, G., Jansen, T., Beume, N., Lucas, S., Poloni, C. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 743–752. Springer, Heidelberg (2008). doi:10.1007/978-3-540-87700-4_74
9. Ishibuchi, H., Hitotsuyanagi, Y., Tsukamoto, N., Nojima, Y.: Use of biased neighborhood structures in multiobjective memetic algorithms. Soft Comput. **13**(8), 795–810 (2009)
10. Konstantinidis, A., Yang, K.: Multi-objective energy-efficient dense deployment in wireless sensor networks using a hybrid problem-specific MOEA/D. Appl. Soft Comput. **11**(6), 4117–4134 (2011)
11. Miettinen, K.: Nonlinear Multiobjective Optimization. International Series in Operations Research and Management Science, vol. 12. Springer, New York (1998)
12. Nagar, A., Haddock, J., Heragu, S.: Multiple and bicriteria scheduling: a literature survey. Eur. J. Oper. Res. **81**(1), 88–104 (1995)
13. Nebro, A.J., Durillo, J.J.: jMetal 4.5 user manual (21 Jan 2014)
14. Peng, W., Zhang, Q.: Network topology planning using MOEA/D with objective-guided operators. In: Coello, C.A.C., Cutello, V., Deb, K., Forrest, S., Nicosia, G., Pavone, M. (eds.) PPSN 2012. LNCS, vol. 7492, pp. 62–71. Springer, Heidelberg (2012). doi:10.1007/978-3-642-32964-7_7
15. Pinedo, M.L.: Scheduling: Theory, Algorithms, and Systems, 4th edn. Springer, New York (2012)
16. Silva, J.D.L., Burke, E.K., Petrovic, S.: An introduction to multiobjective metaheuristics for scheduling and timetabling. In: Gandibleux, X., Sevaux, M., Soerensen, K., T'kindt, V. (eds.) Metaheuristics for Multiobjective Optimisation - Part I, Lecture Notes in Economics and Mathematical Systems, vol. 535, pp. 91–129. Springer, Heidelberg (2004)
17. Smith, W.E.: Various optimizers for single-stage production. Nav. Res. Logistics Q. **3**(1), 59–66 (1956)
18. Wang, J., Cai, Y.: Multiobjective evolutionary algorithm for frequency assignment problem in satellite communications. Soft Comput. **19**(5), 1229–1253 (2015)
19. Wassenhove, L.N.V., Gelders, F.: Solving a bicriterion scheduling problem. Eur. J. Oper. Res. **4**(1), 42–48 (1980)
20. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**(6), 712–731 (2007)
21. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Da Fonseca, V.G.: Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans. Evol. Comput. **7**(2), 117–132 (2003)