Bet-and-Run Strategy and MOEA/D An Alternative Resource Management

First Author[1][0000−1111−2222−3333], Second Author[2,3][1111−2222−3333−4444], and Third Author[3][2222−−3333−4444−5555]

[1] Princeton University, Princeton NJ 08544, USA
[2] Springer Heidelberg, Tiergartenstr. 17, 69121 Heidelberg, Germany
lncs@springer.com
http://www.springer.com/gp/computer-science/lncs
[3] ABC Institute, Rupert-Karls-University Heidelberg, Heidelberg, Germany
{abc,lncs}@uni-heidelberg.de

Abstract. Bet-and-run strategies have been shown, both experimentally and theoretically, to be beneficial on single objective problems. Given this success and the fact that they do not take any problem knowledge into account and are not tailored to the optimization algorithms, we propose to integrate a bet-and-run strategy into the Multiobjective Evolutionary Algorithm based on Decomposition framework, (MOEA/D). MOEA/D represent a class of population-based metaheuristics for the solution of multicriteria optimizarion problems. It decomposes a multiobjective optimization problem into a set of scalar objective subproblems and solve this set in a collaborative way.

1 Introduction

A Multiobjective Optimization Problem (MOP) are box-constrained problems that have m multiple objective functions that must be optimized simultaneously:

$$minf(x) = (f_1(x), f_2(x), ..., f_{n_f}(x)), \text{ subject to } x \text{ in } \Omega, \tag{1}$$

where n_f is the number of objective functions, $x \in \mathbb{R}^{n_v}$, the decision vector, represents a candidate solution with n_v variables, $f : \mathbb{R}^{n_v} \to \mathbb{R}^{n_v}$ is a vector of objective functions and Ω is the feasible decision space, such that:

$$\Omega = \{x \text{ in } \mathbb{R}^{n_v} | g_i(x) \leq 0 \ \forall_i \text{ and } h_i(x) = 0 \ \forall_j\}, \tag{2}$$

Objectives often conflict with each other, therefore, no point in Ω minimizes all the objectives at the same time. Consequently, the goal of MOP solvers is to find the best trade-off that balances the different objectives in an optimal way.

Given two feasible solutions u, v in Ω, u Pareto-dominates v, denoted by $f(y) \succ f(v)$, if and only if $f_k(u) \leq f_k(v), \forall_k \in \{1, ..., n_f\}$ and $f(u) \neq f(x)$. A

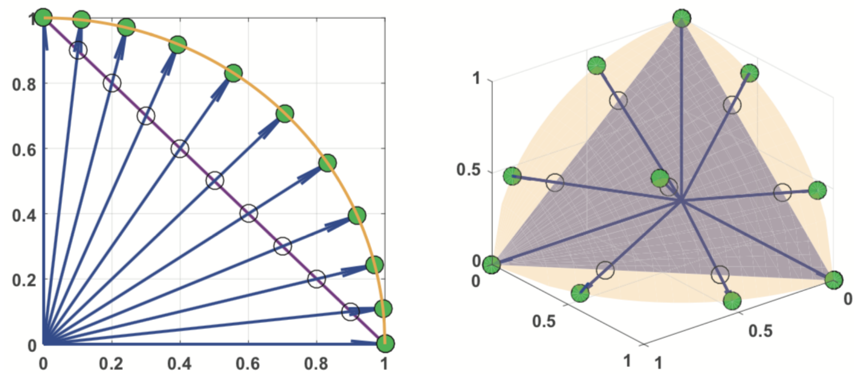

Fig. 1. Decomposition - 2 and 3 objectives - Figure from [2]

solution $x^* \in \Omega$ is considered Poreto-Optimal if there exists no other solution $y \in \Omega$ such that $f(y) \succ f(x^*)$, i.e., if x^* is nondominated in the feasible decision space. A point is called nondominated if no other point dominates it. That is, no single solution provides a better trade-off in all objectives.

The set of all Pareto-optimal solutions is known as the Pareto-Optimal set (PS), while the image of this set is referred to as the Pareto-optimal front (PF).

$$PS = x^* \in \Omega | \nexists y \in \Omega : f(y) \succ f(x^*), \tag{3}$$

$$PF = f(x^*) | x^* \in PS. \tag{4}$$

2 MOEA/D

.

Here, MOEA/D [12] is briefly explained. MOEA/D represents a class of population-based meta-heuristics for solving Multi Objective Problems (MOPs). It is based on decomposition, which is one kind of scalarizing function, where one multi-objective problem becomes various single-objective sub-problem. That is, MOEA/D decomposes an MOP with n_f objectives, as defined in equation 1, into N sub-problems using a set of uniformly distributed weight vectors. To define the sub-problem a weight vector is used as a decomposition strategy to define the sub-problems. All these N sub-problems are expected to represent a good approximation to the PF, as it is shown in Figure 2.

MOEA/D solves these subproblems simultaneously by evolving a set of solutions in a single run by using a aggregation function. This function allied with the weight vectors need to guarantee that a optimal solution to a sub-problem is Pareto-optimal for the MOP and also need to guarantee that the solutions are well distributed in the space of objectives. Therefore, the set of solutions may provide a fair approximation of the Pareto Front.

The relationship between the set of solutions and the sub-problems is fomulated as for every sub-problem a unique solution is associated to it, thus the size of the set of solutions is equal to the number of sub-problems. At each iteraction, one new candidate solution is generated by applying a sequence of variation operators to the existing solutions. This new solution is then compared with the old one, and the best solution after is maintained as the incumbent solution given the procidure defined by the update strategy.

To compare the solutions or to generate a new one, the algorithm utilizes a neighborhood that defines the limits the exchange of information of a subproblem between candidates solutions. This neighborhood is defined based on the distance among their weights.

Algorithm 1: General procedure of MOEA/D framework

Input: Objective functions **f**; constraint functions **g**; input parameters

t$\rightarrow$ 0
run $\rightarrow$ TRUE
Initialize the solution set $X^{(t)} = x^1, ..., x^{n_f}$ by random sampling from Ω Generate weights Λ
while *run* **do**
    Define or update neighborhood $B^i = 1_i, ..., i_{n_f}$
    Copy incumbent solution set $X'^{(t)}$ into $X'^{(t)}$
    for *each variation operator* $v \in V$ **do**
        $X^{(t)} \leftarrow v(X'^{(t)})$
    end
    Evaluate solutions in $X^{(t)}$ and $X'^{(t)}$
    Define next population $X^{(t+1)}$ Update *run* flag t$\rightarrow$ t + 1
end
return $X^{(t)}$, f($X^{(t)}$)

3 Bet-and-Run

3.1 Restart Strategy

Restart Strategies are a mechanism helps the algorithm to explore more in the solution area [11]. For instance, stochastic algorithms and randomized search heuristics may encouter some stagnation before finding a high quality solution. One way to overcome such stagnation is to introduce a restart strategy, since it forcibly changes the search points by restoring the algorithm to its beginning [7]. Also, Restart Strategy might be used to avoid heavy-tailed running time distributions [6], because if a execution of an algorithm does not conclude within a pre-determined limit or if the solution quality is unsatisfactory, the algorithm is restarted [9]. Finally, it may be considered as an additional speed-up [5].

3.2 Bet-and-Run framework

ischetti and Monaci [4] investigated the Bet-and-Run framework. They defined it as a number of short runs with randomized initial conditions (the bet-phase) and then bet on the most promising run(the bet-phase) and bring it to completion. In their work, they studied the following Bet-and-Run framework:

Phase 1 performs k runs of the algorithm for some short time limit t_1 with $t_1 \leq t/k$.

Phase 2 uses remaining time $t_2 = t - k * t_1$ to continue *only the best run* from the first phase until time out.

In 2017, Lissovoi and Sudholt [9] analysed this framework theoretically. They investigate it in the context of single objective problems and found that new initialisations can have a small beneficial effect even on very easy functions, that this restart strategy might be an effective countermeasure when problems with promising and deceptive regions are encountered.

To the best of our knowledge, the Bet-and-Run framework was only applied with evolutionary algorithms in the context of single objective problems.

4 MOEA/D and Bet-and-Run

In this work, we propose to integrate both frameworks, the MOEA/D the Bet-and-Run. First, the implementation of MOEA/D is discussed. Then the implementation of the Bet-and-Run followed by the discussion of how to integratate them.

4.1 MOEA/D

In this paper, two different MOEA/D combinations found in the literature were studied. These combinations are the original MOEA/D [12] and MOEA/D-DE [8].

The first modification was to change the parameter control H of the simplex-lattice design (SLD) that is used to generate the weight vectors W. For the 2-objective problem benchmark functions it was set as *199*, while for the 3-objective problem benchmark functions, *19*. Those vales for the H parameter were chosen so that the number of sub-problems and the size of incumbent solutions are equal to *200*, following default settings as in the recent work from Tanabe et. al [10]. Na verdade eles usam varios tamanhos de populacao, e percebem que menor e melhor no inicio e pior no fim. The other modification was to use an archive, that stores all nondominated solutions found during the search process.(????????).

We also studied the integration of On-line Resource Allocation (ONRA), proposed in the context of MOEA/D by [13]. The resource distribution when using ONRA is allocated using an adaptative strategy aiming to adjust the behaviour of an algorithm in on-line manner to suit the problem in question. Although, other strategies were proposed in the work of Zhou, ONRA was the one that perfomed better among all strategies proposed. The ONRA strategy is concerned with the distribution of resources in an execution of MOEA/D. Different amounts of resources are considered to different sub-problems, following the assumption that some sub-problems can be more difficult to approximate that others.

4.2 Bet-and-Run

In this work, the Bet-and-run framework implemented follows the results found by Friedrich et. al [5]. They studied different combinations strategies that are diverse on the amount of resources assigned for phase 1 and 2.

The best overall strategy found is the one that uses 40% of the total budget available on short runs (phase1) and then run the most prominient one (phase 2) with the remaining 60% of the budget found. One adjusment was made to better fit the context of MOP and MOEA/D which is defining the budget as the number of interactions, instead of using time as the budget as Friedrich et. al used.

5 Experiment Design

The DTLZ [3] and the ZDT [14],test problems were used in the analysis. For the first the number of objectives used was two three. According to [3], for the DTLZ problems, the number of position variables D was set to $k = 5$ for the DTLZ1 problem, $k = 7$ for the DTLZ2 problem and $k = 10$ for the other DTLZ problems, where the number of variables $D = n_f + k - 1$. For the ZDT the number of variable $D = 11$.

Our limit budget was set to the maximum of iteractions, with value of 300, which leads to a number of functions evaluations of XXXXX.

The hypervolume (HV) indicator [15] was used. ishibuchi2018specify Compared with their HV - normalized between 0 and 1 (based on the fair comparison paper). 30 repetitions. box-plots Kruskal-Wallis (data non-normal data, used in the literature)

Configurations and Parameters

Control - Based on the common variation: MOEA/D (variation1) and MOEA/D-DE as in preset_moead Control and ONRA - parameters: dt = 20 Ben-and-run Ben-and-run and ONRA - parameters: dt = 20

Dt - interval that control the resources allocation. From the proposal paper, there is no much sensibility. Decomposition method used - SLD, with H being 199 for 2D and 19 for 3D number of dimensions - 60 All other parameters are defined by preset_moead

Bet-and-run

Phase 1 of the bet-and-run strategy is using the epsilon indicator. 40 instances. It needs two Pareto sets. The first is the Pareto set of a bet instance while the other is the Pareto set from the control algorithm executed with 1% of the number of interactions. Phase 2 uses the 60% rest of max interactions.

6 Evaluation Metrics

Evaluation Metrics

Unary Indicators

- Measure Pareto Sets independently. - Power is restricted. - Cannot tell in general if a set is better than another. - Focus on problem dependent and specifics. - Assumptions and knowledge should be specified. 1. Hyper-volume. 2. Error ratio. 3. Distance from reference set.

Binary Indicators

- Theoretically have no limitations. - Analysis and presentation of results more difficult.

1. R1, R2, R3 indicators. 2. ε-Indicator. 3. Binary Hyper-volume.

Hypervolume Considerations

- Is complete - If, and only if $HV(A) > HV(B) \implies A$ is not worse than B. - Is weakly compatible - $HV(A) > HV(B) \implies \not{B}$ dominates A. - Assumptions - All points of a Pareto Set under consideration dominate the reference point. - @ishibuchi2018specify proposed a method to specify the reference point from a viewpoint of fair performance comparison.

Considerations - A large population size is **always** more beneficial than a small one. - Measures both the convergence toward the Pareto Front and the diversity of non-dominated solutions. - A monotonic increase of the hyper-volume over time cannot always be ensured. - For MOEA/D that is always true.

ε-Indicator Considerations - It compares 2 Pareto Sets. - It indicates which set is better and how much better - If A is better than B $\implies I_\varepsilon(B, A) > 0$. - If $I_\varepsilon(A, B) \leq 0$ and $I_\varepsilon(B, A) > 0 \implies A$ is better than B.

The benchmark used are the DTLZ and the ZDT group of functions.

DTLZ are easy [1].

7 Results

Analysis are done with

8 Conclusion

References

1. Bezerra, L.C., López-Ibáñez, M., Stützle, T.: Comparing decomposition-based and automatically component-wise designed multi-objective evolutionary algorithms. In: International Conference on Evolutionary Multi-Criterion Optimization. pp. 396–410. Springer (2015)

2. Chugh, T.: Handling expensive multiobjective optimization problems with evolutionary algorithms. Jyväskylä studies in computing 263. (2017)
3. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. In: Evolutionary multiobjective optimization, pp. 105–145. Springer (2005)
4. Fischetti, M., Monaci, M.: Exploiting erraticism in search. Operations Research **62**(1), 114–122 (2014)
5. Friedrich, T., Kötzing, T., Wagner, M.: A generic bet-and-run strategy for speeding up stochastic local search. In: AAAI. pp. 801–807 (2017)
6. Gomes, C.P., Selman, B., Crato, N., Kautz, H.: Heavy-tailed phenomena in satisfiability and constraint satisfaction problems. Journal of automated reasoning **24**(1-2), 67–100 (2000)
7. Kanahara, K., Katayama, K., Okano, T., Kulla, E., Oda, T., Nishihara, N.: A restart diversification strategy for iterated local search to maximum clique problem. In: Conference on Complex, Intelligent, and Software Intensive Systems. pp. 670–680. Springer (2018)
8. Li, H., Zhang, Q.: Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii. IEEE Transactions on evolutionary computation **13**(2), 284–302 (2009)
9. Lissovoi, A., Sudholt, D., Wagner, M., Zarges, C.: Theoretical results on bet-and-run as an initialisation strategy. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 857–864. ACM (2017)
10. Tanabe, R., Ishibuchi, H.: An analysis of control parameters of moea/d under two different optimization scenarios. Applied Soft Computing **70**, 22–40 (2018)
11. Yu, V., Iswari, T., Normasari, N., Asih, A., Ting, H.: Simulated annealing with restart strategy for the blood pickup routing problem. In: IOP Conference Series: Materials Science and Engineering. vol. 337, p. 012007. IOP Publishing (2018)
12. Zhang, Q., Li, H.: Moea/d: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on evolutionary computation **11**(6), 712–731 (2007)
13. Zhou, A., Zhang, Q.: Are all the subproblems equally important? resource allocation in decomposition-based multiobjective evolutionary algorithms. IEEE Transactions on Evolutionary Computation **20**(1), 52–64 (2016)
14. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary computation **8**(2), 173–195 (2000)
15. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms—a comparative case study. In: International conference on parallel problem solving from nature. pp. 292–301. Springer (1998)