

Local Search Operators Based on Linear-Quadratic Approximations for Multiobjective Problems on a Budget Scenario

André R. da Cruz^{1,2}, Rodrigo T. N. Cardoso¹, Elizabeth F. Wanner³, Ricardo H. C. Takahashi⁴

¹Centro Federal de Educação Tecnológica de Minas Gerais – Belo Horizonte, Brasil

²Instituto Federal de São Paulo – São Paulo, Brasil

³Aston University – Birmingham, United Kingdom

⁴Universidade Federal de Minas Gerais – Belo Horizonte, Brasil

ABSTRACT

This paper presents three local search procedures based on linear-quadratic approximations for multiobjective problems, which consider a goal attainment model. The approximation is constructed for each objective using a current solution, a queue of evaluated solutions, and a nearest neighborhood algorithm. The operators differ by the approximation technique: linear regression, linear programming or linear matrix inequalities. Depending on the size of the queue, the approximation is done using a linear, a quadratic with diagonal or full symmetric Hessian function. In order to compare the local searches, an experimental procedure is applied using two benchmarks. For this, each local search operator is coupled with NSGA-II for solving problems on a limited budget scenario. The quality of the final solutions via the hypervolume indicator is assessed as well as the runtime for each method. A statistical test reveals that the version with local search based on linear regression produces superior nondominated sets.

KEYWORDS. Expensive multiobjective optimization. Local search operator. Linear-quadratic approximation.

Paper topics: MH – Metaheuristics. OA – Other Applications in OR.

1. Introduction

Population-based techniques have become an important tool to solve multiobjective problems since a set of nondominated solution can be obtained in a single run. These techniques can handle objectives with difficult characteristics for a deterministic method, like nonconvexity, non-differentiability, discontinuity and multimodality [Deb, 2001]. There are many research works dealing with the development of powerful Multiobjective Evolutionary Algorithms (MOEAs). However, they usually require a large number of objective evaluations. When the evaluation is expensive or the number of function evaluations is small, these algorithms usually show a limited performance. This situation is typical in real optimization problems which use computational simulation process, for example automotive industry [Regis, 2014] and control of disease spread [da Cruz et al., 2017].

A common strategy when optimizing costly objectives with a limited budget is to use a surrogate assisted MOEAs [Jin, 2011]. The incorporation of surrogate models aims to reduce the number of required objective evaluations to approximate the nondominated solutions set of expensive multiobjective problems. The main idea is to use a sample of previously evaluated solutions to generate an approximated model which is cheaper to evaluate, easier to optimize and its returned optimum is a promising solution for the original problem. In this way, the algorithm convergence can be improved at spending less time than using only the original objective functions.

Since the first reported work in 1985 [Grefenstette e Fitzpatric, 1985], many successful cases applying surrogate assisted MOEAs have been applied [Jin, 2011] in different domains. The surrogate models can be classified according to its working procedure, for example, polynomial approximation [da Cruz et al., 2017], radial basis function [Regis, 2014] or neural networks [Kourakos e Mantoglou, 2009], to name a few.

The surrogate model can also be used as a local search operator inside the MOEAs. The model can be built using previously evaluated solutions and optimized to provide a good solution for the real optimization problem. The surrogate model is continuously updated and is expected to keep the optimization towards the nondominated set. In [Wanner et al., 2008], a local search operator using quadratic approximation and linear matrix inequality is coupled with a MOEA and applied to costly black-box optimization problems. In [da Cruz et al., 2011], a local search operator for multiobjective problems based on convex quadratic approximations is proposed. Surrogate models such as quadratic with diagonal or symmetric Hessian are built and the approximated problem can be solved via linear programming or linear matrix inequality.

This work presents an improved local search procedure to be coupled with any multiobjective evolutionary algorithm for solving computationally expensive optimization problems. Due to the expensive nature of evaluating the solutions, a budget of function evaluations is imposed to the problem. Each local search operator approximates the original objective functions using linear or quadratic functions (with a diagonal or full symmetric Hessian matrix) depending on the number of previously evaluated solutions. These solutions are stored in queue with a limited size and are selected using a nearest neighborhood search, given an initial solution. The approximated multiobjective problem is optimized via the goal attainment [Gembicki e Haimes, 1975] subject to the original variable bounds and the hypersphere in which the given initial solution is the center and a radius is chosen at random in a interval. The approximated optimum replaces the previous given solution in the current population before the evaluation step.

Three linear-quadratic approximation models are considered to approximate the objective functions: Linear Regression (LRG) Montgomery et al. [2012], Linear Programming (LIP) Bazaraa et al. [2011] and Linear Matrix Inequalities (LMI) [Boyd et al., 1994]. These distinctive ways of approximations can return discrepant approximations with particular computational costs. In this way, it investigates here how the local search approximations can contribute with the final nondominated set quality, whether one can be picked up as the best choice, with a smaller computational cost. In this way, an experiment is designed to compare the local search efficiency using the hypervolume indicator [Zitzler et al., 2003], and the runtime, in seconds, for the different approximation models.

A canonical implementation of the NSGA-II [Deb et al., 2002] is used as a baseline to be compared to the hybrid approaches. The ZDT and the DTLZ test functions are used as a benchmark functions with a budget of 500 function evaluations. A nonparametric Quade test [Derrac et al., 2011] is applied to assess the results.

It is important to highlight some differences between the paper [da Cruz et al., 2011] and the present work. Here, a queue of previously evaluated solutions is created using a nearest neighborhood search from each randomly selected solution, in order to avoid extra function evaluations. In the previous paper, solutions were always randomly generated around the selected nondominated solution. Secondly, the approximated problems are here simultaneously minimized via goal attainment constrained to a hypersphere and the variable bounds. It is done to preserve the locality of the search procedure and to allow a better usage of nonconvex quadratic approximations. In the aforementioned paper, since the surrogate models were convex and quadratic ones, the approximated problem was always solved via a quadratic programming technique. Finally, another difference is that here one child solution used as a starting point for the local search is replaced for the obtained optimal vector of the approximated problem in each operator run, instead of generating many similar solutions, as could happen using the previous procedure.

The remaining of this paper is organized as follows. Section 2 presents how the function approximations are constructed. Section 3 explains how the proposed local search operator is executed inside the multiobjective evolutionary algorithm. Section 4 presents the methodology and explains the computational experiment to be applied. Section 5 discusses the obtained results. Finally, Section 6 concludes this paper and presents future works.

2. Linear-Quadratic Approximations

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a real-valued function and $B = \{(\mathbf{x}^1, f(\mathbf{x}^1)), \dots, (\mathbf{x}^k, f(\mathbf{x}^k))\}$ be a set k previously evaluated solutions. These solutions are used to approximate f by another function $q : \mathbb{R}^n \rightarrow \mathbb{R}$ so that $q(\mathbf{x}) \approx f(\mathbf{x})$ for each $(\mathbf{x}^i, f(\mathbf{x}^i)) \in B$, $i \in \{1, \dots, k\}$. The function q is represented by Equation 1, in which $\mathbf{h} = [h_1 \dots h_n]'$ can be the Gradient vector representing the linear terms and

$$\mathbf{H} = \begin{bmatrix} h_{n+1} & h_{2n+1} & \dots & h_{3n-2} & h_{3n-1} \\ h_{2n+1} & h_{n+2} & \dots & h_{4n-4} & h_{4n-3} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ h_{3n-2} & h_{4n-4} & \dots & h_{2n-1} & h_{(n^2+3n)/2} \\ h_{3n-1} & h_{4n-3} & \dots & h_{(n^2+3n)/2} & h_{2n} \end{bmatrix}$$

can be the Hessian matrix representing the quadratic terms.

$$q(\mathbf{x}) = h_0 + \mathbf{h}'\mathbf{x} + 0.5\mathbf{x}'\mathbf{H}\mathbf{x} \quad (1)$$

The goal is to determine $\mathbf{p} = [h_0 \dots h_n h_{n+1} \dots h_{2n} h_{2n+1} \dots h_{(n^2+3n)/2}]'$, the best parameter vector which best fits q in f for a given subset of evaluated solutions. If $\mathbf{H} = \mathbf{0}$, q is a linear function with $p = 1 + n$ parameters ($h_{n+1}, h_{n+2}, \dots$ are discarded). If $\mathbf{H}$ is a diagonal matrix, q has $p = 1 + 2n$ parameters and represents a quadratic function without mixed terms ($h_{2n+1}, h_{2n+2}, \dots$ are discarded). If $\mathbf{H}$ is a full symmetric matrix, q has $p = 1 + (n^2 + 3n)/2$ parameters. It must be noted that the problem dimension, p , grows with the sample size from $1 + n$ to $(n^2 + 3n)/2$.

A queue of previously evaluated solutions is implemented for the local search operator. When the local search is performed, the number of samples in the queue is considered in order to select which kind of approximation will be executed. If the selected sample size is greater than $1 + (n^2 + 3n)/2$, then a quadratic approximation with full symmetric Hessian is done. If the sample size is greater than $1 + 2n$, then a quadratic approximation with diagonal Hessian is performed. If

When the surrogate model is approximated via a linear function, then multivariate linear regression technique is applied. For a higher level approximation, one particular technique amongst multivariate linear regression, linear programming or linear matrix inequality is performed. The next subsections explain how these methods can be applied to construct the approximated function.

The multivariate linear regression model, LRG, is used here to represent a k -dimensional continuous response vector $\mathbf{y}$ as a linear combination of predictor terms plus a vector of error terms $\boldsymbol{\epsilon}$ which follows a multivariate normal distribution [Montgomery et al., 2012].

Consider $\mathbf{y} = \left[f(\mathbf{x}) \dots f(\mathbf{x}) \right]' \in \mathbb{R}^k$ as the response vector and $\mathbf{D} \in \mathbb{R}^{k \times p}$ as the design

$$\mathbf{D} = \begin{bmatrix} 1 & \frac{1}{x_1} & \dots & \frac{1}{x_n} & \frac{1}{x_1 x_1} & \dots & \frac{1}{x_n x_n} & \frac{1}{x_1 x_2} & \dots & \frac{1}{x_{n-1} x_n} \\ & & \ddots & & & \ddots & & & \ddots & \\ 1 & \frac{k}{x_1} & \dots & \frac{k}{x_n} & \frac{k}{x_1 x_1} & \dots & \frac{k}{x_n x_n} & \frac{k}{x_1 x_2} & \dots & \frac{k}{x_{n-1} x_n} \end{bmatrix} \quad (3)$$

$$\hat{\mathbf{p}} = (\mathbf{D}'\mathbf{D})^{-1}\mathbf{D}\mathbf{y} \quad (4)$$

2.2. Linear Programming

$$\begin{aligned} \min_{h_0, \mathbf{h}, \mathbf{H}, \beta_{ij}} \quad & z = \sum_{i=1}^k \left(f(\mathbf{x})^i - q(h_0, \mathbf{h}, \mathbf{H}; \mathbf{x})^i \right) \\ \text{subject to:} \quad & \begin{cases} q(h_0, \mathbf{h}, \mathbf{H}; \mathbf{x})^i \leq f(\mathbf{x})^i & \forall i \in \{1, \dots, k\} \\ h_{jj} \geq \sum_{\substack{i=1, \\ i \neq j}}^n \beta_{ij} & \forall j \in \{1, \dots, n\} \\ -\beta_{ij} \leq h_{ij} & \forall (i, j) \in \{1, \dots, n\}^2 \wedge i < j \\ h_{ij} \leq \beta_{ij} & \forall (i, j) \in \{1, \dots, n\}^2 \wedge i < j \end{cases} \end{aligned} \quad (5)$$

The $n^2 + n + 1$ variables are unrestricted ones and there are $k + n^2$ constraints. The first constraint set ensures that the approximation underestimates the sample points, providing a lower bound and a local trend indication for the original function. The next three constraints guarantee a diagonally dominant Hessian which implies convexity [De Klerk, 2006].

The LIP model for a diagonal quadratic approximation is presented in Equation 6. This model has $1+2n$ variables and $k+n$ constraints. Once there is no mixed terms, it is only necessary to ensure that the Hessian diagonal values must be greater than 0 to guarantee the convexity property.

$$\begin{aligned} \min_{h_0, \mathbf{h}, \mathbf{H}} z &= \sum_{i=1}^k \left(f(\mathbf{x}^i) - q(h_0, \mathbf{h}, \mathbf{H}; \mathbf{x}^i) \right) \\ \text{subject to:} & \\ \begin{cases} q(h_0, \mathbf{h}, \mathbf{H}; \mathbf{x}^i) & \leq f(\mathbf{x}^i) & \forall i \in \{1, \dots, k\} \\ h_{jj} & \geq 0 & \forall j \in \{1, \dots, n\} \end{cases} \end{aligned} \quad (6)$$

2.3. Linear Matrix Inequalities

A linear matrix inequality, LMI, can be represented by Equation 7, in which $\mathbf{x} \in \mathbb{R}^p$ are the decision variables and $\mathbf{F}_i \in \mathbb{R}^{m \times m}$ are symmetric matrices, for all $i \in \{0, 1, \dots, p\}$. $\mathbf{F}(\mathbf{x}) \succcurlyeq 0$ means that the matrix is positive semidefinite, i.e., $\mathbf{u}'\mathbf{F}(\mathbf{x})\mathbf{u} \geq 0$ for all nonzero $\mathbf{u} \in \mathbb{R}^m$ [Boyd et al., 1994].

$$\mathbf{F}(\mathbf{x}) = \mathbf{F}_0 + \sum_{i=1}^p x_i \mathbf{F}_i \succcurlyeq 0 \quad (7)$$

The problem of finding a convex quadratic function q (i.e., with $\mathbf{H} \succcurlyeq 0$) can be done by solving the semidefinite model [Vandenberghe e Boyd, 1996], presented in Equation 8, by implicitly minimizing the quadratic error considering the given sample. This model is applied for both quadratic functions with diagonal or symmetric Hessian. The difference is given, respectively, by the number of parameters to be determined ($1 + 2n$ or $1 + (n^2 + 3n)/2$) and the required number of samples to fit it.

$$\begin{aligned} \min_{h_0, \mathbf{h}, \mathbf{H}, t} z &= t \\ \text{subject to:} & \\ \begin{cases} \sqrt{\sum_{i=1}^k \left(f(\mathbf{x}^i) - q(h_0, \mathbf{h}, \mathbf{H}; \mathbf{x}^i) \right)^2} & \leq t \\ \mathbf{H} & \succcurlyeq 0 \end{cases} \end{aligned} \quad (8)$$

This optimization model is solved here using the SeDuMi package [Sturm, 1999], which has an interior point self-dual convex cone optimization algorithm.

3. Proposed Local Search Procedure

First of all, consider the bounded constraint multiobjective optimization problem presented in Equation 9. The column vector of n real decision variables $\mathbf{x} = [x_1, \dots, x_n]'$ belongs to the feasible set (region) $\mathbf{X} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}$ at which $\mathbf{l} = [l_1, \dots, l_n]' \in \mathbb{R}^n$ and $\mathbf{u} = [u_1, \dots, u_n]' \in \mathbb{R}^n$ are the lower and upper bound vectors, respectively ($l_i \leq x_i \leq u_i, \forall i \in \{1, \dots, n\}$). Each objective function f_i of the multiobjective problem will be approximated by q_i , i.e., $q_i(\mathbf{x}) \approx f_i(\mathbf{x})$ in relation to B for all $i \in \{1, \dots, m\}$.

$$\min_{\mathbf{x}} \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_m(\mathbf{x})]' \quad \text{subject to: } \mathbf{x} \in \mathbf{X} \quad (9)$$

To cope with this problem, with a limited number of function evaluations, it proposes here a local search procedure, which will be explained below.

Given a new offspring solution $\tilde{\mathbf{x}}^c \in \mathbb{R}^n$ and queue $Q = \{(\tilde{\mathbf{x}}^1, f(\tilde{\mathbf{x}}^1)), \dots, (\tilde{\mathbf{x}}^{qs}, f(\tilde{\mathbf{x}}^{qs}))\}$ of previously evaluated solutions ($qs \leq qs_{max}$), a nearest neighborhood search [Friedman et al., 1977] in relation to $\tilde{\mathbf{x}}^c \in \mathbb{R}^n$ is performed in order to select $k = \max(k_0, qs)$ samples $B = \{(\tilde{\mathbf{x}}^1, f(\tilde{\mathbf{x}}^1)), \dots, (\tilde{\mathbf{x}}^k, f(\tilde{\mathbf{x}}^k))\}$ which will be used in the function approximation process. Here, qs represents the current queue size, qs_{max} , the queue maximum size, k_0 the biggest possible sample size and k the current cardinality of B .

Depending on the k , a distinct function approximation may occur. If $k > 1 + (n^2 + 3n)/2$, then a quadratic approximation with full symmetric Hessian will be done. If $k > 1 + 2n$, then the sample will be used to build a quadratic approximation with a diagonal Hessian. And, if $k > 1 + n$, the original function will be approximated by a linear one. Otherwise, the approximation will not occur, the local search will not be applied, and the algorithm flow returns to the evolutionary cycle.

With all the approximations, the auxiliary problem presented by Equation 10 is solved via goal attainment [Gembicki e Haimes, 1975], using adequate vectors for a goal $\mathbf{g}$, and weights $\mathbf{w}$, for the model in Equation 11. The new feasible set $\tilde{\mathbf{X}} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \wedge \|\mathbf{x} - \tilde{\mathbf{x}}^c\| \leq r\}$ corresponds to the region limited by the original lower and upper bounds $\mathbf{l}$ and $\mathbf{u}$, respectively, and a hypersphere with a center located at $\tilde{\mathbf{x}}^c$ and a positive uniform random radius r , such that $r_{min} \leq r \leq r_{max}$.

$$\min_{\mathbf{x}} \mathbf{q}(\mathbf{x}) = [q_1(\mathbf{x}), \dots, q_m(\mathbf{x})]' \text{ subject to: } \mathbf{x} \in \tilde{\mathbf{X}} \quad (10)$$

$$\min_{\mathbf{x}, \gamma} \gamma \text{ subject to: } (\mathbf{x} \in \tilde{\mathbf{X}}) \wedge (\mathbf{q}(\mathbf{x}) - \gamma \mathbf{w} \leq \mathbf{g}) \quad (11)$$

This model can be solved by a Sequential Quadratic Programming algorithm [Fletcher, 2013] and, in the offspring population, the given vector $\tilde{\mathbf{x}}$ is replaced by the optimal solution of this process, $\tilde{\mathbf{x}}^*$. It is expected that this optimal solution be a better solution for the original problem. Hereafter, the algorithm flow comes back to the evolutionary operators and the offspring population is evaluated according to original objective functions $\mathbf{f}$.

This local search procedure can be coupled with any MOEA. In this work, the NSGA-II [Deb et al., 2002] is used as the baseline algorithm and the local search is applied at every ls_{step} generations. Four different versions of the NSGA-II are applied and analyzed. The first is a canonical version without local search, designated by CAN, which is used as a baseline to be compared. The other three versions are the ones with the local search approximations as described above. The algorithms follow the same acronyms of their quadratic approximation methods: LRG, LIP and LMI. As an example, LIP represents the NSGA-II with a local search which approximates quadratically the objective functions solving Equations (5) or (6) via linear programming.

Initially, NSGA-II creates, evaluates, ranks and assigns a crowding distance according to the nondominance levels, a feasible initial random population P with ps solutions. During the evolution loop, which ends when the number of objective function evaluations reaches $eval_{max}$, ps pairs of parents are selected via binary tournament, considering the rank and crowding distance value as tiebreaker, in order to create a new offspring population C with ps solutions via SBX crossover with parameter eta_c . Some solutions in C suffer, with probability p_m , a polynomial mutation with parameter eta_m .

In the hybrid versions, at each ls_{step} generations, an offspring solution $\tilde{\mathbf{x}}^c$ with a random index $i_r \in \{1, \dots, ps\}$ is selected for the local search phase. If the function approximations occur, Problem 10 will be solved using Equation 11, in which $\mathbf{w} \in [0, 1]^m$ is a random uniform vector and $\mathbf{g} \in \mathbb{R}^m$ is the result of a component-wise multiplication of the i_r -th parent objective values by a random uniform vector $\mathbf{d} \in [-1000, 0]^m$. The obtained solution $\tilde{\mathbf{x}}^*$ replaces $\tilde{\mathbf{x}}^c$ in C .

The next step in NSGA-II is the evaluation of C , which is merged in sequence with P to be classified in relation to the nondominance levels. Hereafter, the best ps solutions from $P \cup C$,

according to the nondominance levels and crowding distance values, are extracted to be the parent population P of the next generation.

It must be noted that in hybrid versions, a queue of previously evaluated solutions Q with capacity qs_{max} is maintained. At each evaluation, the new pair $(\mathbf{x}, f(\mathbf{x}))$ is deterministically stored. If the queue is full, following a First-In First-Out method, one point must be removed before the new one can be inserted. In this way, the evolution information is updated and used to make new approximations.

4. Experiment Design

This work intends to study the effect of the proposed local search in the NSGA-II when dealing with multiobjective problems with a limited budget. As a case study, it considers the 5 ZDTs with $m = 2$ objectives and the 6 DTLZs with $m = 3$ functions, both with 10 variables.

The goal of this experiment is to observe possible differences in the final nondominated set. The quality of the nondominated set is assessed by the hypervolume indicator values, and the computational runtime cost is also analyzed, in seconds, to examine the effort to achieve the final nondominated solutions. Therefore, two null hypothesis are tested with a significance level $\alpha = 0.05$ using the nonparametric Quade test [Derrac et al., 2011]. The first null hypothesis, H_0^{hv} , states that, for all considered objective functions, all versions of NSGA-II, CAN, LRG, LIP and LMI have identical effects according to their hypervolume median. The second null hypothesis H_0^{rt} , states that all NSGA-II versions spend the same time, in seconds, in relation to their median runtime.

In order to investigate these issues, a set of computational experiments is designed. Each NSGA-II version run 21 times with different seeds for each test function having a budget of $eval_{max} = 500$ objective function evaluations. Since the underline assumption is that the objective functions are computationally expensive, the population size is set to $ps = 20$. Information about hypervolume values by each ps evaluations in relation to a fixed and adequate reference point is stored. In the end of all experiments, for each objective function, the hypervolume values will be divided by the maximum found in order to obtain a relatively normalized value between 0 and 1. The final runtime is also saved to be compared posteriorly. In order to compare all treatments, only the final performance values are used in the nonparametric Quade test.

The other parameter values for the NSGA-II are: $eta_c = 10$ for SBX crossover, $p_m = 0.01$ and $eta_m = 10$, respectively, as the probability value and the parameter value for the polynomial mutation. For the local search procedure, the parameters are: $ls_{step} = 5$ as number of generations to perform it, $k_0 = 2(1 + (n^2 + 3n)/2)$ as maximum sample size to select in a nearest neighborhood search, $qs_{max} = 5(1 + (n^2 + 3n)/2)$ as the maximum capacity of the evaluated solutions queue. The parameters of the goal attainment are: $r_{min} = 0.051$ and $r_{max} = 0.201$. For the Sequential Quadratic Programming algorithm, it is used: at most 1000 evaluations of $\mathbf{q}$ and $\epsilon = 1e - 6$ as a tolerance in the function values and variable differences.

5. Results

The three hybrid NSGA-II: LRG, LIP and LMI, are compared with the pure NSGA-II. The comparison takes into account the quality of the final Pareto front and the computational complexity of each local search approach. The Hypervolume indicator is used to assess the Pareto front quality while the runtime is used to estimate the local search complexity.

Figure 1 shows the hypervolume median evolution by the number of objective functions in relation to all 11 test functions for the all runs. It is possible to notice that all local search versions became superior than the canonical one after 220 evaluations. It is also possible to note that the hypervolume value of CAN version stagnated after 400 evaluations while the other versions kept improving. It can be stated that the local search operator can lead to a better final nondominated set when compared with the pure algorithm.

During the evolution, the hypervolume behavior is similar when it is analyzed the data for separated objective functions, with some exceptions as expected by no free lunch theorem [Wolpert

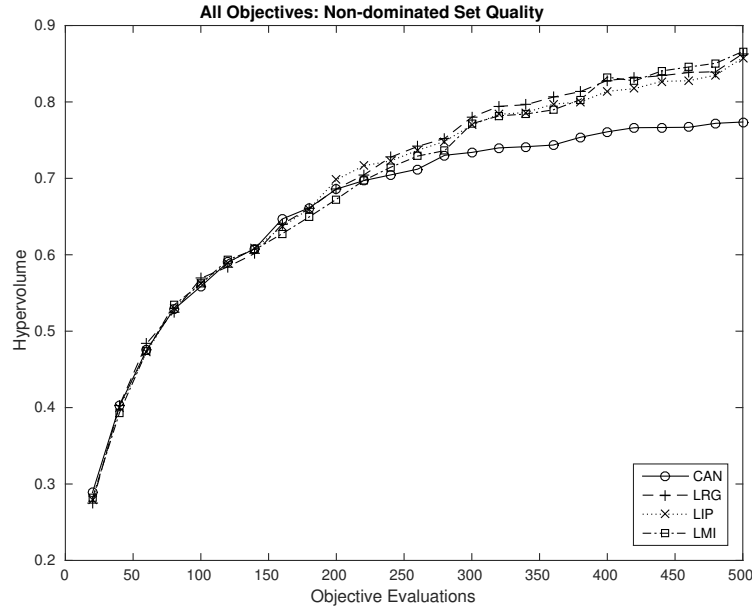


Figure 1: Normalized hypervolume median by the number of evaluated objectives according to all test objective functions.

Table 1: Quantiles of normalized hypervolume values with 500 functions evaluations according to all test objective functions.

Quantile	CAN	LRG	LIP	LMI
Min	0.2915	0.3024	0.3024	0.2990
0.25	0.6394	0.7217	0.6988	0.6735
0.50	0.7737	0.8631	0.8577	0.8660
0.75	0.9258	0.9623	0.9589	0.9542
Max	0.9975	1.0000	1.0000	1.0000

e Macready, 1997]. For example, Figure 2 illustrates how the quality metric behaves with the number of evaluations growth for ZDT2.

Table 1 shows five illustrative quantiles of the final hypervolume value for each hybrid algorithm. The hypervolume value equals 1 indicates that the algorithm reached greatest quality metric for some test function. Observing the data, it is possible to state that the LRG algorithm shows a superiority when compared to others. It is possible to note that all hypervolume quantiles were greater than or equal, with the exception of the sample median which was a little bit less than that obtained by LMI. In other words, it can be said that the hypervolume sample data returned by LRG were much better CAN, superior than LMI (mainly in lower quantiles) and a little bit better than LIP (although very similar).

The nonparametric Quade test was applied in the final returned hypervolume data for all algorithms to validate the null hypothesis H_0^{hv} with significance $\alpha = 0.05$. The test returned a p-value less than $1e - 6$ indicating the rejection of H_0^{hv} and the evidence of a statistical difference in the hypervolume medians among at least one pair of algorithms. The test statistic (F) returned was 24.4476 and the critical value used to compare the algorithms weighted ranks to determine difference between them was $c_{hv} = 17353.8752$. The bigger the weighted rank is, better the algorithm is according to the hypervolume indicator. If the absolute value of the difference between each pair of algorithm weighted rank is less than c_{hv} , then there is a statistical difference. This pairwise comparison is depicted in Figure 3 and summarized in Table 2. These outputs indicate that, with exception of the pair LRG and LIP, there is a statistical difference between each pair of

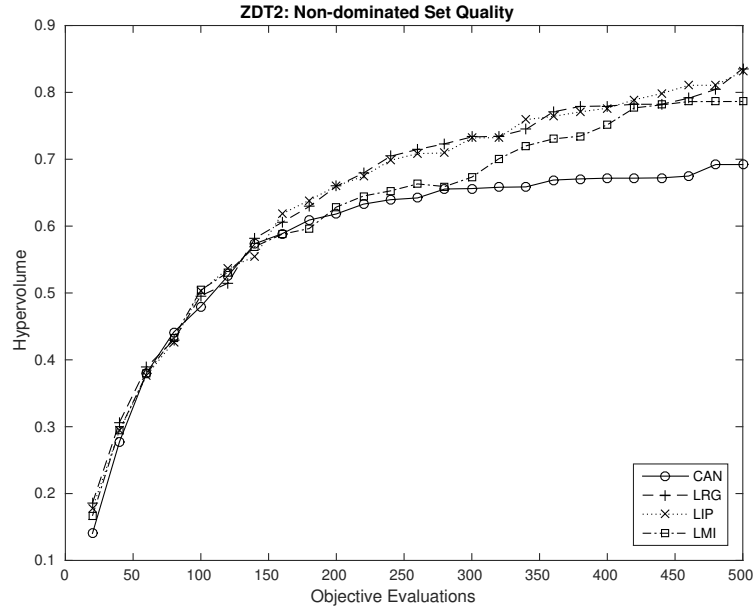


Figure 2: Normalized hypervolume median by the number of evaluated objectives according to ZDT2 test function.

Table 2: Quade test: Hypervolume Pairwise comparison.

Comparison	Difference?	Ranking Difference
CAN × LRG	Yes	−66966
CAN × LIP	Yes	−64088
CAN × LMI	Yes	−45586
LRG × LIP	No	2878
LRG × LMI	Yes	21380
LIP × LMI	Yes	18502

algorithms. In this manner, according to hypervolume median, it can be concluded that the best algorithm was LRG with a statistical draw with LIP, followed by LMI and CAN.

The successful behavior of local search based on multiobjective optimization of the linear or quadratic approximations can be explained by its construction. The most recent previously evaluated solutions are used to construct the function approximation. In this way, it is possible to capture a local tendency via linear or quadratic approximation, which is relatively cheaper to obtain. And, optimizing the approximations constrained to a relative small hypersphere, it is possible to generate a good solution. Therefore, the local search fulfills the role of enhancing solutions around the provided original solution vicinity.

Taking into account the computational cost, 3 shows some runtime quantiles, in seconds, for each algorithm in all executions for all versions of the algorithms. It is important to observe the additional cost given by the insertion of the local search procedure in NSGA-II canonical version. It is clear that CAN is the cheaper one, followed by LRG, LIP and LMI. This evidence is supported by the Quade test which returned a p-value less than $1e - 6$ rejecting H_0^{rt} with a test statistic (F) equals to 760.8547 and the critical value $c_{rt} = 10639.0626$. Figure 4 and Table 4 present the pairwise comparison between each algorithm runtime and reinforce the statistical difference among each pair of treatments. The more negative the difference, quicker the method is.

The analysis of the algorithms efficiency and computational cost allows to conclude that LRG is the best hybrid version, since it obtained better hypervolumes with a relative low runtime. The local search using the LRG is able to generate good function approximations when compared

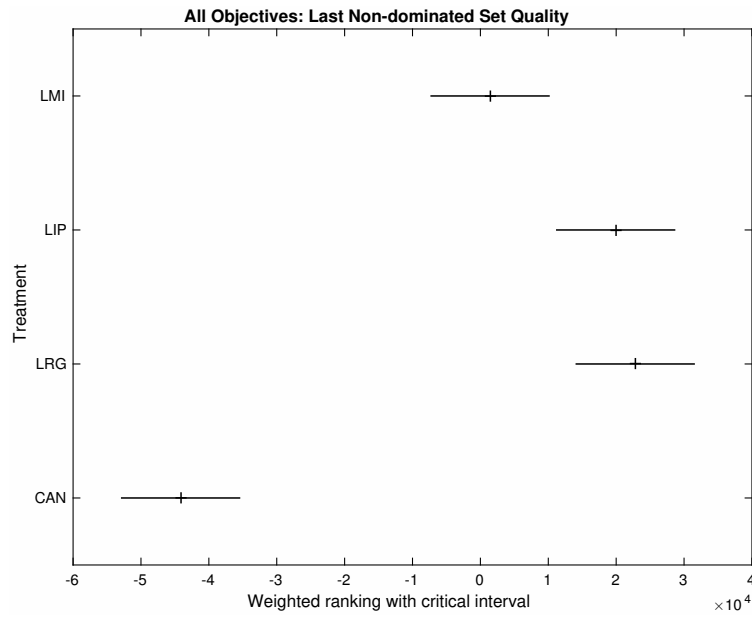


Figure 3: Pairwise comparison determined by the nonparametric Quade test of the algorithms for the normalized hypervolume values with 500 functions evaluations according to all test objective functions.

Table 3: Quantiles of runtime, in seconds, with 500 functions evaluations according to all test objective functions.

Quantile	CAN	LRG	LIP	LMI
Min	7.8869	16.0146	20.9385	30.2592
0.25	8.9751	26.6227	34.3164	51.3469
0.50	12.1679	31.5913	44.6592	60.9847
0.75	13.5785	39.4401	58.3714	68.0712
Max	52.0589	70.6579	125.5823	111.5264

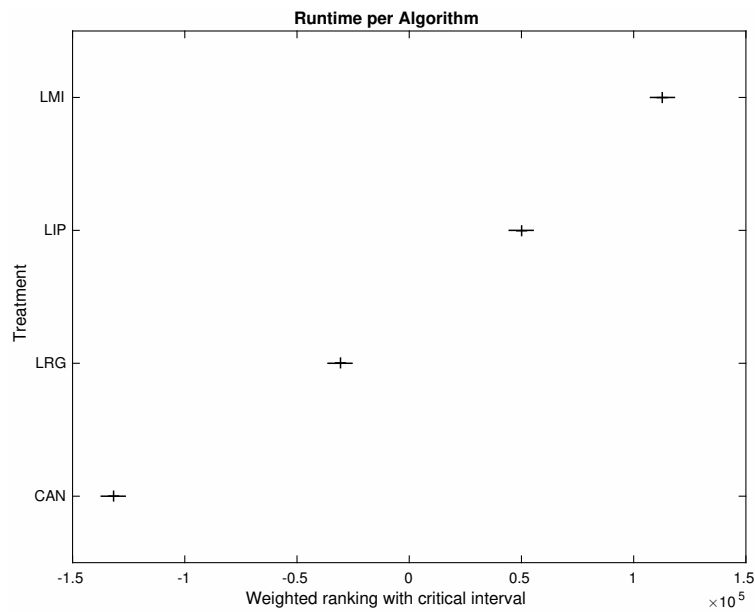


Figure 4: Pairwise comparison determined by the nonparametric Quade test of the algorithms for the runtime, in seconds, with 500 functions evaluations according to all test objective functions.

Table 4: Quade test: Runtime Pairwise comparison.

Comparison	Difference?	Ranking Difference
CAN × LRG	Yes	−101044
CAN × LIP	Yes	−181772
CAN × LMI	Yes	−244712
LRG × LIP	Yes	−80728
LRG × LMI	Yes	−143668
LIP × LMI	Yes	−62940

with LIP and LMI since the convexity property is not required. In addition, the LRG-based local search is computationally simple to obtain. The second best version is the LIP version, which approximates below the sample data of the original function via linear programming and guarantees a convex Hessian. This version also obtained good results, but has a higher computational cost in order to find the approximation. In the last position, LMI minimizes a quadratic error in order to obtain the surrogate model with a convex Hessian. This local search produces superior results in relation to the canonical version, but it is statistically worse than LRG and LIP. It can be also noted a considerable computational cost to achieve the approximations.

6. Conclusions and Future Works

This work proposed and compared local search techniques based on optimization of linear-quadratic approximations for evolutionary heuristics solving black-box multiobjective models. The budget scenario, with 500 objectives evaluations, is a common situation in real problems with costly objective function evaluation. The proposed operators were coupled with a NSGA-II canonical version in order to obtain a better final nondominated set of solutions.

In order to observe the local search influence in the obtained nondominated solutions, a set of computational experiments was carried out. The versions with and without local search were run 21 times, with 500 objective evaluations per optimization, using 11 test functions with dimension 10. Informations about hypervolume indicator and runtime were extracted and posteriorly analyzed. Statistical differences were detected by Quade test and indicated that the version with local search based on approximation done via linear regression was superior in the context of this work.

For future works, the authors intend to compare the methodologies proposed here with different kind of local searches based on approximations commonly used (for example, those based on radial basis functions and Bayesian optimization techniques [Knowles, 2006]). The application of this work techniques will be applied in other benchmark problems with different characteristics and also on real world problems. It is also desired to analyze the local search effect with the number of variables and objectives variation.

References

- Bazaraa, M. S., Jarvis, J. J., e Sherali, H. D. (2011). *Linear programming and network flows*. John Wiley & Sons.
- Boyd, S., El Ghaoui, L., Feron, E., e Balakrishnan, V. (1994). *Linear matrix inequalities in system and control theory*. SIAM.
- da Cruz, A. R., Cardoso, R. T., e Takahashi, R. H. (2017). Multiobjective synthesis of robust vaccination policies. *Applied Soft Computing*, 50:34–47.
- da Cruz, A. R., Wanner, E. F., Cardoso, R. T., e Takahashi, R. H. (2011). Using convex quadratic approximation as a local search operator in evolutionary multiobjective algorithms. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, p. 1217–1224. IEEE.
- De Klerk, E. (2006). *Aspects of semidefinite programming: interior point algorithms and selected applications*, volume 65. Springer Science & Business Media.

- Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons.
- Deb, K., Pratap, A., Agarwal, S., e Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197.
- Derrac, J., García, S., Molina, D., e Herrera, F. (2011). A practical tutorial on the use of non-parametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1):3–18.
- Fletcher, R. (2013). *Practical methods of optimization*. John Wiley & Sons.
- Friedman, J. H., Bentley, J. L., e Finkel, R. A. (1977). An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)*, 3(3):209–226.
- Gembicki, F. e Haimes, Y. (1975). Approach to performance and sensitivity multiobjective optimization: The goal attainment method. *IEEE Transactions on Automatic control*, 20(6):769–771.
- Grefenstette, J. e Fitzpatrick, J. (1985). Genetic search with approximate fitness evaluations. In *Proceedings of the International Conference on Genetic Algorithms and Their Applications*, p. 112–120.
- Jin, Y. (2011). Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61–70.
- Knowles, J. (2006). Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1): 50–66.
- Kourakos, G. e Mantoglou, A. (2009). Pumping optimization of coastal aquifers based on evolutionary algorithms and surrogate modular neural network models. *Advances in water resources*, 32(4):507–521.
- Montgomery, D. C., Peck, E. A., e Vining, G. G. (2012). *Introduction to linear regression analysis*, volume 821. John Wiley & Sons.
- Regis, R. G. (2014). Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points. *Engineering Optimization*, 46(2):218–243.
- Sturm, J. F. (1999). Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization methods and software*, 11(1-4):625–653.
- Vandenbergh, L. e Boyd, S. (1996). Semidefinite programming. *SIAM review*, 38(1):49–95.
- Wanner, E. F., Guimaraes, F., Takahashi, R., e Fleming, P. J. (2008). Local search with quadratic approximations into memetic algorithms for optimization with multiple criteria. *Evolutionary Computation Journal*, 16(2):185–224.
- Wolpert, D. H. e Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82.
- Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., e Da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on evolutionary computation*, 7(2):117–132.