On the Behavior of Stochastic Local Search Within Parameter Dependent MOPs

Víctor Adrián Sosa Hernández[1(✉)], Oliver Schütze[1], Heike Trautmann[2], and Günter Rudolph[3]

[1] Computer Science Department, CINVESTAV-IPN, Av. IPN 2508, C.P. 07360, Col. San Pedro Zacatenco, Mexico City, Mexico
msosa@computacion.cs.cinvestav.mx, schuetze@cs.cinvestav.mx

[2] Department of Information Systems, University of Münster, Leonardo-Campus 3, 48149 Münster, Germany
trautmann@uni-muenster.de

[3] Fakultät für Informatik, Technische Universität Dortmund, 44221 Dortmund, Germany
Guenter.Rudolph@tu-dortmund.de

Abstract. In this paper we investigate some aspects of stochastic local search such as pressure toward and along the set of interest within parameter dependent multi-objective optimization problems. The discussions and initial computations indicate that the problem to compute an approximation of the entire solution set of such a problem via stochastic search algorithms is well-conditioned. The new insights may be helpful for the design of novel stochastic search algorithms such as specialized evolutionary approaches. The discussion in particular indicates that it might be beneficial to integrate the set of external parameters directly into the search instead of computing projections of the solution sets separately by fixing the value of the external parameter.

Keywords: Parameter dependent multi-objective optimization · Stochastic local search · Evolutionary algorithms

1 Introduction

In many applications the problem arises that several objectives have to be optimized concurrently leading to a *multi-objective optimization problem* (MOP). Furthermore, it can happen that the MOP contains one or several external parameters $\lambda \in \Lambda$ such as the environmental temperature of a given mechanical system. Such parameters cannot be 'optimized', but on the other hand the decision maker cannot neglect them. Since for every fixed value of λ the problem acts as a 'classical' MOP, the solution set of such a *parameter dependent MOP* (PMOP) is given by an entire family P_Λ of Pareto sets. One question that arises is to compute a finite size representation of P_Λ which has been addressed so far in some works using specialized evolutionary algorithms ([2,3,5,7,14,16]).

A. Gaspar-Cunha et al. (Eds.): EMO 2015, Part II, LNCS 9019, pp. 126–140, 2015.
DOI: 10.1007/978-3-319-15892-1_9

In this work, we address one facet of this problem via investigating the behavior of stochastic local search (SLS) within PMOPs. By utilizing a certain relation of SLS with line search methods as used in mathematical programming we will see that—under certain (mild) assumptions on the model—both pressure toward and along the set of interest (in objective space) is already inherent in SLS. Initial studies on a simple set based method that includes SLS underline that the problem to compute a finite size representation of the entire solution set via stochastic search methods such as evolutionary algorithms (EAs) is a well-conditioned problem. We hope that the obtained insights will be valuable for future designs of specialized EAs. The results in particular suggest that it might make sense to integrate the entire λ-space into the search which will allow to compute the desired approximation in *one* run of the algorithm which is in contrast to the current works which consider 'λ-slices' in each run.

The remainder of this paper is organized as follows: in Section 2 we briefly state the problem at hand and discuss the related work. In Section 3, we consider some aspects of SLS within PMOPs which we underline by some computations, and finally draw our conclusions in Section 4.

2 Background and Related Work

In the following we consider continuous parameter dependent multi-objective optimization problem (PMOPs) of the form

$$\min_{\boldsymbol{x} \in S} F_\lambda(\boldsymbol{x}). \tag{1}$$

Hereby, F_λ is defined as a vector of objective functions

$$\begin{aligned} &F_\lambda : S \to \mathbb{R}^k, \\ &F_\lambda(\boldsymbol{x}) = (f_{1,\lambda}(\boldsymbol{x}), \cdots, f_{k,\lambda}(\boldsymbol{x})), \end{aligned} \tag{2}$$

where $S \subset \mathbb{R}^n$ is the domain (here we will consider unconstrained problems, i.e., $S = \mathbb{R}^n$) and $\lambda \in \Lambda \subset \mathbb{R}^l$ specifies the external parameters to the objective functions.

Note that for every fixed value of λ problem (1) can be seen as a classical multi-objective problem (MOP) (for the discussion on classical MOPs we refer e.g. to [6]). Thus, the solution set of (1) consists of an entire family of Pareto sets which is defined as follows:

$$P_{S,\Lambda} := \{(\boldsymbol{x}, \lambda) \in \mathbb{R}^{n+l}, \text{ s.t } \boldsymbol{x} \text{ is a Pareto point of } F_\lambda, \ \lambda \in \Lambda\}. \tag{3}$$

The according family of Pareto fronts is denoted by $F(P_{S,\Lambda})$. Both sets typically—i.e., under mild assumption on the model— form $(k-1+l)$-dimensional objects.

Probably the first study in the field of parameter dependent optimization has been published by Manne in the year of 1953 [1]. In the following we summarize some important works in the evolutionary multi-objective optimization literature.

A classification of dynamic MOPs (which are particular PMOPs where the value of λ changes in time) is presented in [13]. This work focuses on the components that lead to the observed dynamic behavior. The work of Farina, Deb, and Amato [7] also deals with dynamic MOPs. It contains some test case applications as well as many results related to problems which depend of an external parameter. Further, a classification of dynamic MOPs is established. The work in [15] gives a good insight into PMOPs, but only treats problems with uniquely one external parameter by using numerical path following algorithms. Also some geometrical properties of the solution sets are discussed as well as connections to bifurcation theory are provided. In [2] the authors provide a survey over the evolutionary techniques that tackle dynamic optimization problems. They mention four main ways to master such problems: (i) increasing the diversity after the change of the solution set, (ii) maintaining the diversity over the complete run of the evolutionary algorithm to detect the changes in the solution set, (iii) memory based approaches, and finally, (iv) multi-population approaches which are the ones that reduce the main problem into subproblems or 'slices' in order to maintain a small population until the family of solution sets is reached.

The idea to use slices, or multi-population approaches, in the evolution of an evolutionary algorithm is used for example in [3]. There, an algorithm is proposed that solves the problem by dividing the objective landscape into subpopulations in order to reach all the solutions over the external parameter (in this case time). Another work related to PMOPs can be found in [12]. Here the authors use a parallel version of the NSGA-II in order to solve a dynamic optimization problems to reduce the energy consumption when solving this kind of problem. They divide the complete problem into nodes and then the algorithm NSGA-II is executed in each node to compute the solution set. Finally, in [5] the authors present a taxonomy of the ways to treat PMOPs and also mention several similarities and differences between PMOPs and MOPs. Here again the multi-population idea is used and adapted by using migration methods.

In the current literature, the investigation of stochastic local search for continuous PMOPs is neglected so far. This paper makes a first attempt to fill this gap.

3 Behavior of Stochastic Local Search Within PMOPs

For our considerations it is advantageous to treat λ —at least formally— within PMOPs as a 'normal' parameter leading to the following problem:

$$\begin{aligned} F &: \mathbb{R}^{n+l} \to \mathbb{R}^{k+l} \\ F(\boldsymbol{x}, \lambda) &= \begin{pmatrix} f_1(\boldsymbol{x}, \lambda) \\ \vdots \\ f_k(\boldsymbol{x}, \lambda) \\ \lambda \end{pmatrix} = \begin{pmatrix} g_1(\boldsymbol{x}, \lambda) \\ \vdots \\ g_{k+l}(\boldsymbol{x}, \lambda) \end{pmatrix}, \end{aligned} \tag{4}$$

where $g_i : \mathbb{R}^{n+l} \to \mathbb{R}$, $i = 1, \ldots, k+l$. The Jacobian is given by

$$J(\boldsymbol{x}, \lambda) = \begin{pmatrix} \nabla_x f_1(\boldsymbol{x}, \lambda)^T & \nabla_\lambda f_1(\boldsymbol{x}, \lambda)^T \\ \vdots & \vdots \\ \nabla_x f_k(\boldsymbol{x}, \lambda)^T & \nabla_\lambda f_k(\boldsymbol{x}, \lambda)^T \\ 0 & I_l \end{pmatrix} := \begin{pmatrix} J_x & J_\lambda \\ 0 & I_l \end{pmatrix} \in \mathbb{R}^{(k+l)\times(n+l)}, \tag{5}$$

where

$$J_x = \begin{pmatrix} \nabla_x f_1(\boldsymbol{x}, \lambda)^T \\ \vdots \\ \nabla_x f_k(\boldsymbol{x}, \lambda)^T \end{pmatrix} \in \mathbb{R}^{k\times n}, \quad J_\lambda = \begin{pmatrix} \nabla_\lambda f_1(\boldsymbol{x}, \lambda)^T \\ \vdots \\ \nabla_\lambda f_k(\boldsymbol{x}, \lambda)^T \end{pmatrix} \in \mathbb{R}^{k\times l}, \tag{6}$$

and where I_l denotes the $(l \times l)$-identity matrix.

To understand the behavior of SLS it is advantageous to see its relation to line search as it is used in mathematical programming: if a point $z_1 = (\boldsymbol{x}_1, \lambda_1)$ is chosen (at random) from a small neighborhood of $z_0 = (\boldsymbol{x}_0, \lambda_0)$, then z_1 can be written as

$$z_1 = z_0 + 1(z_1 - z_0) = z_0 + ||z_1 - z_0|| \frac{z_1 - z_0}{||z_1 - z_0||}. \tag{7}$$

Thus, the selection of z_1 can be viewed as a search in direction $v := (z_1 - z_0)/||z_1 - z_0||$. For infinitesimal steps in a direction $\nu \in \mathbb{R}^{n+l}$ (in decision space) the related change in objective space is given by $J(\boldsymbol{x}_0, \lambda_0)\nu$. To see this, consider the i-th component of $J(\boldsymbol{x}_0, \lambda_0)\nu$:

$$(J(\boldsymbol{x}_0, \lambda_0)\nu)_i = \lim_{t\to 0} \frac{g_i((\boldsymbol{x}_0, \lambda_0) + t\nu) - g_i(\boldsymbol{x}_0, \lambda_0)}{t} = \langle \nabla g_i(\boldsymbol{x}_0, \lambda_0), \nu \rangle, \quad i = 1, \ldots, k+l. \tag{8}$$

For problem (4) this direction is given by

$$J\nu = \begin{pmatrix} J_x & J_\lambda \\ 0 & I_l \end{pmatrix} \begin{pmatrix} \nu_x \\ \nu_\lambda \end{pmatrix} = \begin{pmatrix} J_x\nu_x + J_\lambda\nu_\lambda \\ \nu_\lambda \end{pmatrix}, \tag{9}$$

where $J = J(\boldsymbol{x}_0, \lambda_0)$ and $\nu = (\nu_x, \nu_\lambda)$ with $\nu_x \in \mathbb{R}^n$ and $\nu_\lambda \in \mathbb{R}^l$.

Based on these considerations, we now consider different scenarios for SLS that occur in different stages within an evolutionary algorithm.

(a) $(\boldsymbol{x}, \lambda)$ *'far away' from* $P_{S,\Lambda}$. Here we use an observation made in [4] for classical MOPs namely that the 'objectives gradients' may point into similar directions when the decision point $(\boldsymbol{x}, \lambda)$ is far from the Pareto set. We assume here the extreme case namely that all gradients point into the same direction. For this, let $g := \nabla_x f_i(\boldsymbol{x}, \lambda)$ and assume that

$$\nabla_x f_i(\boldsymbol{x}, \lambda) = \mu_i g, \; i = 1, \ldots, k, \tag{10}$$

where $\mu_i > 0$ for $i = 1, \dots, k$. Then

$$J_x \nu_x = \begin{pmatrix} \nabla_x f_1(\boldsymbol{x}, \lambda)^T \nu_x \\ \vdots \\ \nabla_x f_k(\boldsymbol{x}, \lambda)^T \nu_x \end{pmatrix} = g^T \nu_x \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_k \end{pmatrix}. \tag{11}$$

That is, the movement is 1-dimensional regardless of ν_x which is n-dimensional. Since $J_x \nu_x = 0$ iff $\nu_x \perp g$, the probability is one that for a randomly chosen ν_x either dominated or dominating solutions are found (and in case a dominated solution is found, the search has simply to be flipped to find dominating solutions).

Thus, for $\nu_\lambda = 0$, which means that the value of λ is not changed in the local search, we obtain for $\mu = (\mu_1, \dots, \mu_k)^T$

$$J\nu = \begin{pmatrix} g^T \nu_x \mu \\ 0 \end{pmatrix}. \tag{12}$$

For $\nu_\lambda \neq 0$, i.e., in the case that the value of λ is changed within the local search, no such physical meaning exists to the best of our knowledge. Nevertheless, the investigation of this problem will be one topic for future research.

As a general example we consider here the following PMOP ([9]):

$$\begin{aligned} F_\lambda &: \mathbb{R}^2 \to \mathbb{R}^2 \\ F_\lambda(x) &:= (1-\lambda) F_1(\boldsymbol{x}) + \lambda F_2(\boldsymbol{x}), \end{aligned} \tag{13}$$

where $\lambda \in [0, 1]$ and $F_1, F_2 : \mathbb{R}^2 \to \mathbb{R}^2$,

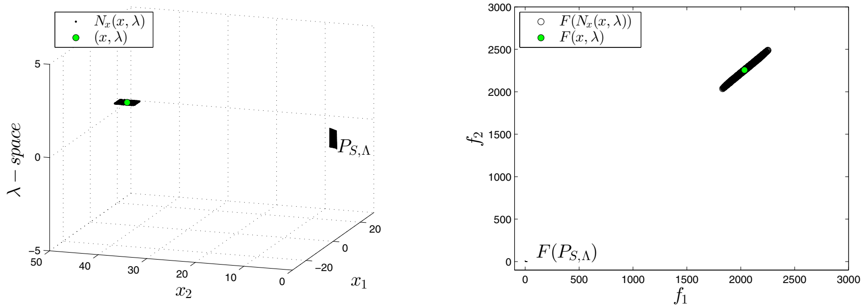
$$F_1(x_1, x_2) = \begin{pmatrix} (x_1 - 1)^4 + (x_2 - 1)^2 \\ (x_1 + 1)^2 + (x_2 + 1)^2 \end{pmatrix},$$

$$F_2(x_1, x_2) = \begin{pmatrix} (x_1 - 1)^2 + (x_2 - 1)^2 \\ (x_1 + 1)^2 + (x_2 + 1)^2 \end{pmatrix}.$$

This problem is a convex homotopy of the MOPs F_1 and F_2 which have both convex Pareto fronts. Figures 1 and 2 show the behavior of SLS for 100 uniformly randomly chosen points near $(\boldsymbol{x}, \lambda) = (10, 45.2, 0.7)$ for $\nu_\lambda \neq 0$ and $\nu_\lambda = 0$. As neighborhood we have chosen the infinity norm with radius $r_x = 2$ in $\boldsymbol{x}$-space and $r_\lambda = 0.3$ (respectively $r_\lambda = 0$) in λ-space. For the case $\nu_\lambda = 0$ a clear movement toward/against $F(P_{S,\Lambda})$ can be observed while this is not the case for $\nu_\lambda \neq 0$. Thus, it may make sense to exclude the change of the value of λ in early stages of the search process where the individuals of the populations are supposed to be far away from the set of interest.

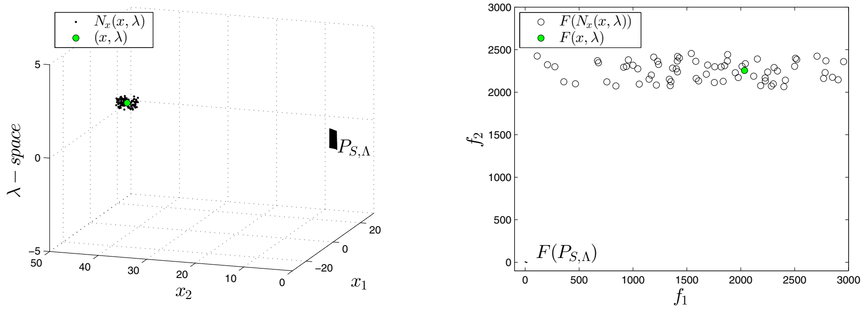
(b) $(\boldsymbol{x}, \lambda)$ *'near' to* $P_{S,\Lambda}$. Here we consider again the extreme case, namely that $\boldsymbol{x}$ is a Karush-Kuhn-Tucker (KKT) point of F_λ. That is, assume that there exists a convex weight $\alpha \in \mathbb{R}^k$ such that

$$\sum_{i=1}^{k} \alpha_i \nabla_x f_i(\boldsymbol{x}, \lambda) = J_x^T \alpha = 0. \tag{14}$$



(a) Decision space (b) Objective space

Fig. 1. SLS for a point that is 'far away' from $P_{S,\Lambda}$ using $\nu_\lambda = 0$



(a) Decision space (b) Objective space

Fig. 2. SLS for a point that is 'far away' from $P_{S,\Lambda}$ using $\nu_\lambda \neq 0$

It can be shown ([8]) that the normal vector to the linearized set $F(P_{(S,\Lambda)})$ at $(\boldsymbol{x}, \lambda)$ is given by

$$\eta = \begin{pmatrix} \alpha \\ -{J_\lambda}^T \alpha \end{pmatrix}. \tag{15}$$

We obtain

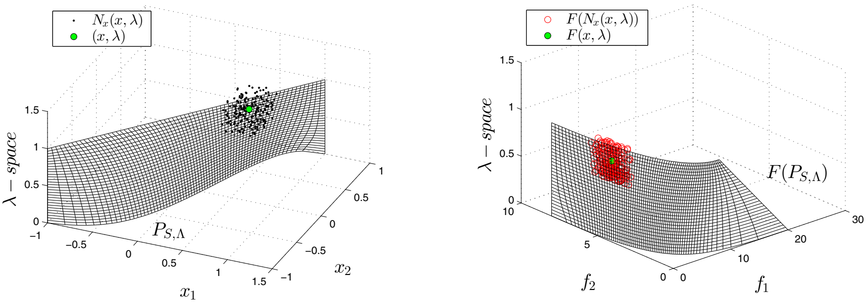
$$\langle J\nu, \eta \rangle = \langle \nu, J^T \eta \rangle = \langle \nu, \begin{pmatrix} J_x^T & 0 \\ J_\lambda^T & I_l \end{pmatrix} \begin{pmatrix} \alpha \\ -{J_\lambda}^T \alpha \end{pmatrix} \rangle = \langle \nu, \begin{pmatrix} J_x^T \alpha \\ J_\lambda^T \alpha - J_\lambda^T \alpha \end{pmatrix} \rangle = 0. \tag{16}$$

That is, it is either (i) $J\nu = 0$ or (ii) $J\nu$ is a movement orthogonal to η and thus along the linearized set at $F(\boldsymbol{x}, \lambda)$. If we assume that the rank of J_x is $k - 1$, then the rank of J is $k - 1 + l$ and the dimension of the kernel of J is $n - k + l$. Hence, for a randomly chosen ν the probability is 1 that event (ii) happens.

Equation (16) tells us that the movement is orthogonal to the normal vector, but it remains to investigate in which direction of the tangent space the movement is performed. For this, let

$$\eta = QR = (q_1, q_2, \dots, q_{k+l})R \tag{17}$$

be a QR factorization of η. Then, the vectors $q_2, \dots, q_{k+l}$ form an orthonormal basis of the tangent space. If we assume again that the rank of J_x is $k-1$, then the rank of J is $k-1+l$. Since by Equation (16) η is not in the image of J, there exist vectors $\nu_q, \dots, \nu_{k+l}$ such that $J\nu_i = q_i$, $i = 2, \dots, k+l$. Thus, a movement via SLS can be performed in all directions of the linearized family of Pareto fronts (i.e., both in $\boldsymbol{x}$- and λ-direction). Figure 3 shows an example for $(\boldsymbol{x}, \lambda) = (0.44, 0.47, 0.84)^T$ and $r_x = r_\lambda = 0.2$. Again, by construction, no structure in decision space can be observed, but a clear movement along the set of interest can be seen in objective space.

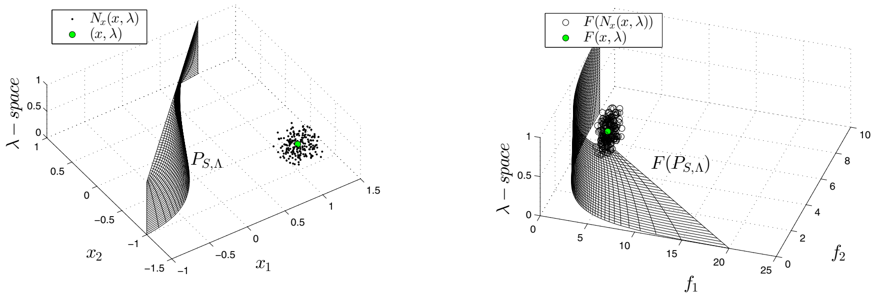


(a) Decision space (b) Objective space

Fig. 3. SLS for a point that is 'near' to $P_{S,\Lambda}$

(c) $(\boldsymbol{x}, \lambda)$ *'in between'.* Apparently, points $(\boldsymbol{x}, \lambda)$ do not have to be far away from nor near to the set of interest but can be 'in between'. In this case, no clear preference of the movement in objective space can be detected. However, this 'opening' of the search compared to the 1-dimensional movement in early stages of the search is a very important aspect since it allows in principle to find (in the set based context and given a suitable selection mechanism) and spread the solutions. In this case for finding multiple connected components. Figure 4 depicts such a scenario for $(\boldsymbol{x}, \lambda) = (1, -1, 0.5)^T$ and $r_x = r_\lambda = 0.2$.

(d) Simple Neighborhood Search (SNS) within set based search. As next step we investigate the influence of SNS within set based methods. In order to prevent interferences with other effects we have thus to omit all other operators (as, e.g., crossover). The Simple Neighborhood Search for PMOPs takes this into consideration: initially, a generation $A_0 \subset \mathbb{R}^{n+l}$ is chosen at random, where Λ



(a) Decision space (b) Objective space

Fig. 4. SLS for a point that is 'in between' using $\nu_\lambda \neq 0$

is discretized into $\tilde{\Lambda} = \{\lambda_1, \ldots, \lambda_s\}$. In the iteration process, for every element $(a_x, a_\lambda) \in A_i$, a new element (b_x, b_λ) is chosen via SLS, where b_λ has to take one of the values of $\tilde{\Lambda}$. The given archive A_i and the set of newly created solutions B_i are the basis for the sequences of candidate solutions A_i^l, $l = 1, \ldots, s$, and the new archive A_{i+1}: for A_i^l the non-dominated solutions from $A_i \cup B_i$ with λ-value λ_l are taken, and A_{i+1} is the union of these sets (plus the respective λ values). Algorithm 1 shows the pseudo code of SNS. Hereby, $nondom(A)$ denotes the non-dominated elements of a set A, $\pi(A, \lambda_i) := \{a \ : \ (a, \lambda_i) \in A\}$ denotes the $\boldsymbol{x}$-values of the elements of A with λ-value λ_i, and $(A, \lambda) := \{(a, \lambda) \ : \ a \in A\}$.

Algorithm 1. SNS for PMOPs

Require: Neighborhood $N_i(x, \lambda)$ of a given point (x, λ) in iteration i.
Ensure: Sequence A_i^l of candidate solutions for F_{λ_l}, $l = 1, \ldots, s$
1: Generate $A_0 \subset {}^{n+l}$ at random
2: **for** $i = 0, 1, 2, \ldots$ **do**
3: $\quad B_i^l := \emptyset$
4: $\quad$ **for** all $(a_x, a_\lambda) \in A_i$ **do**
5: $\quad\quad$ choose $(b_x, b_\lambda) \in N_i(a_x, a_\lambda)$
6: $\quad\quad$ $B_i := B_i \cup (b_x, b_\lambda)$
7: $\quad$ **end for**
8: $\quad A_{i+1}^l := nondom(\pi(A_i \cup B_i, \lambda_l))$, $l = 1, \ldots, s$
9: $\quad A_{i+1} := \bigcup_{l=1}^s (A_{i+1}^l, \lambda_l)$
10: **end for**

For sake of a small comparison we also investigate here the global counterpart of SNS, the Simple Global Search (GS), where all points are chosen uniformly at random from the entire domain. That is, GS can be viewed as an application of SNS where the neighborhood N_i in Line 5 of Algorithm 1 is chosen as the entire domain. In order to reduce the overall number of candidate solutions we have not stored all non-dominated solutions but have used $ArchiveUpdateTight2$ ([11])

to update the archives A_i^l. The archiver *ArchiveUpdateTight*2 aims, roughly speaking, for gap free ϵ-Pareto sets. In our computations, we have used $\epsilon = (0.05, 0.05)^T$. Further, in each computation we have used 10 equally spaced divisions in λ-space and have generated one random element for the initial archive A_0 (i.e., $|A_0| = 10$).

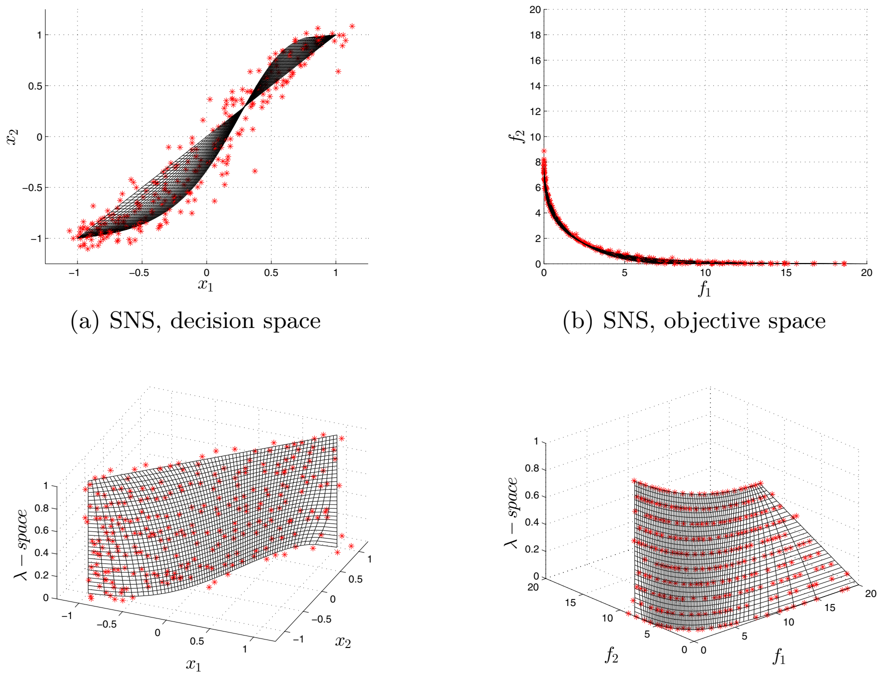
Figure 5 shows some numerical results by solving PMOP (13) in two different angles. In this example we have used a budget of 3,000 function evaluations (FEs) for SNS. Figure 6 shows the respective result for GS for a budget of 3,000 and 10,000 FEs. As domain we have chosen $S = [-10, 10]^2$. The superiority of SNS can be detected visually since those final archives are evenly spread around the solution sets. Compared to this, the result of GS lacks both in spread and convergence, though more than 3 times the number of FEs has been spent to get this result. This observation is confirmed by the values in Table 1 where we show the distances (measured in terms of Δ_2 [10]) between the outcome sets and the union of the 10 Pareto fronts (i.e., our discretized set of interest).

Table 1. Comparative results for PMOPs (13) and (18) using Δ_2 value between the final archives and the union of the 10 Pareto fronts for both SNS and GS for a budget of 3,000 and 5,000 FEs. Shown are worst, average, and best value for 20 different runs.

Algorithm	$\lambda = 0.0$	$\lambda = 0.11$	$\lambda = 0.22$	$\lambda = 0.33$	$\lambda = 0.44$	$\lambda = 0.55$	$\lambda = 0.66$	$\lambda = 0.77$	$\lambda = 0.88$	$\lambda = 1.0$
SNS	1.934	2.434	2.453	1.743	1.234	1.673	1.563	0.984	0.546	0.986
	1.494	**1.977**	**2.126**	**1.526**	**1.032**	**1.402**	**1.220**	**0.550**	**0.381**	**0.385**
PMOP (13)	1.101	1.103	1.132	1.341	0.532	0.643	0.992	0.314	0.249	0.148
GS	25.342	14.349	24.342	6.342	11.433	7.322	6.424	3.221	9.534	5.213
	17.220	7.349	22.440	5.826	9.356	5.963	3.226	2.217	6.241	3.320
PMOP (13)	14.425	7.023	20.213	4.342	9.003	4.095	2.657	2.043	5.141	2.134
SNS	0.632	0.567	0.453	0.578	0.375	0.198	0.297	0.246	0.186	0.123
	0.365	**0.221**	**0.136**	**0.104**	**0.049**	**0.047**	**0.069**	**0.056**	**0.077**	**0.045**
PMOP (18)	0.242	0.201	0.103	0.098	0.019	0.043	0.034	0.022	0.062	0.032
GS	0.992	0.832	0.567	0.693	0.700	0.422	0.596	0.834	0.532	0.423
	0.692	0.620	0.454	0.496	0.684	0.329	0.552	0.821	0.446	0.351
PMOP (18)	0.432	0.597	0.353	0.394	0.592	0.239	0.539	0.739	0.422	0.311

Next we consider a second PMOP which is again a convex homotopy of two MOPs. In this case, one of the Pareto fronts is convex while the other one is concave.

$$
\begin{aligned}
F_\lambda &: \mathbb{R}^2 \to \mathbb{R}^2 \\
F_\lambda(x) &:= (1-\lambda)F_1(x) + \lambda F_2(x),
\end{aligned}
\tag{18}
$$



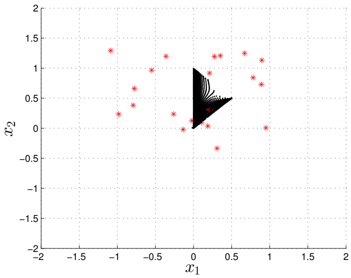
(a) SNS, decision space

(b) SNS, objective space

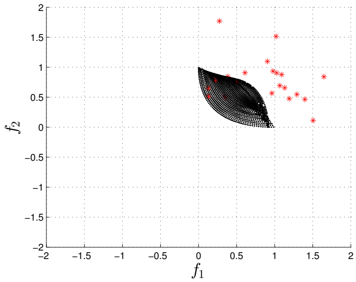
(c) SNS, decision space

(d) SNS, objective space

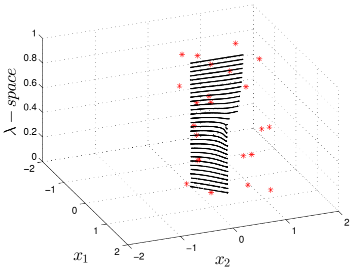
Fig. 5. Numerical result of SNS with a budget of 3,000 FEs for PMOP (13)



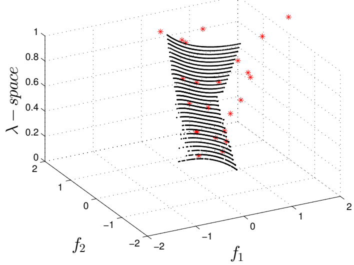
(a) GS 5,000, decision space



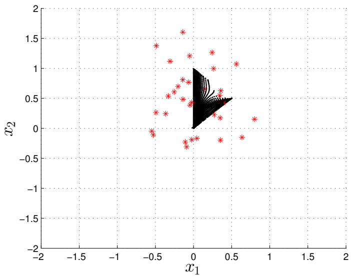
(b) GS 5,000, objective space



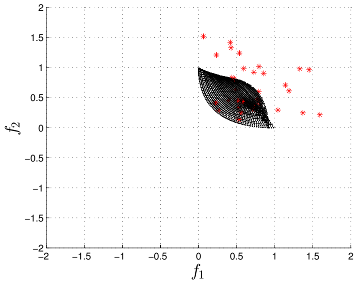
(c) GS 5,000, decision space



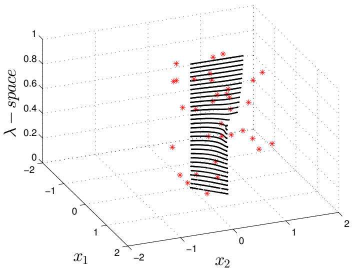
(d) GS 5,000, objective space



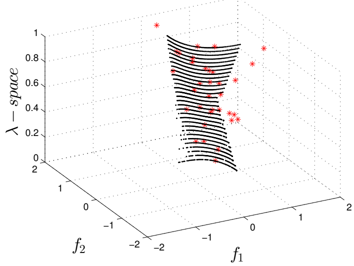
(e) GS 10,000, decision space



(f) GS 10,000, objective space



(g) GS 10,000, decision space



(h) GS 10,000, objective space

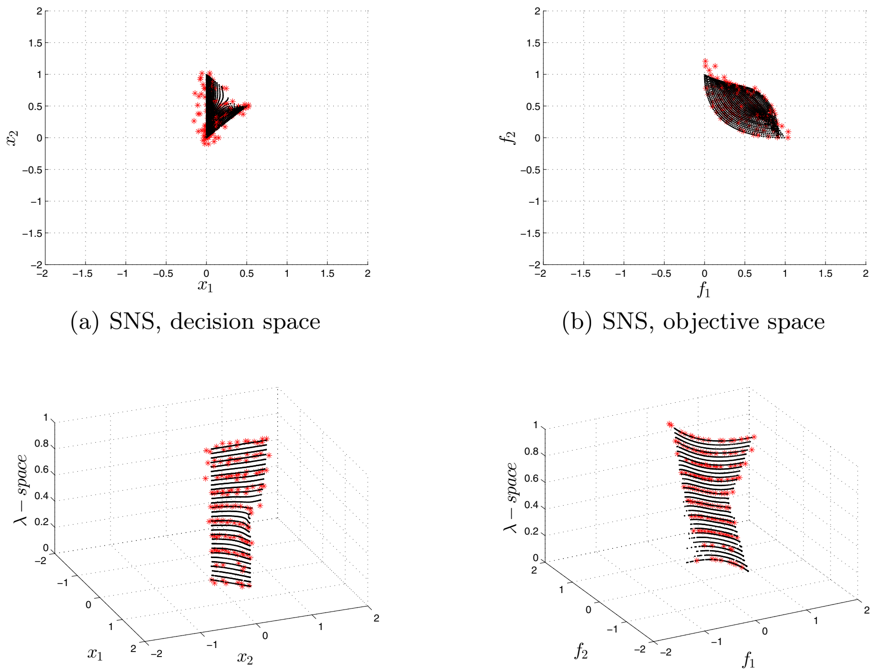
Fig. 6. Numerical result of GS with a budget of 3,000 FEs and 10,000 FEs for PMOP (13)

where $\lambda \in [0,1]$, $a_1 = 0$, $a_2 = 1$ and $F_1, F_2 : \mathbb{R}^2 \to \mathbb{R}^2$,

$$F_1(\boldsymbol{x}) = \begin{pmatrix} (x_1^2 + x_2^2)^{0.125} \\ ((x_1 - 0.5)^2 + (x_2 - 0.5)^2)^{0.25} \end{pmatrix},$$

$$F_2(\boldsymbol{x}) = \begin{pmatrix} x_1^2 + x_2^2 \\ (x_1 - a_1)^2 + (x_2 - a_2)^2 \end{pmatrix}.$$

Figures 7 and 8 show some exemplary numerical results of SNS (5,000 FEs) and GS (5,000 and 10,000 FEs) using $S = [-10, 10]^2$. Again, SNS, though less FEs were used, is superior with respect to spread and convergence according the indicator Δ_2 in Table 1.



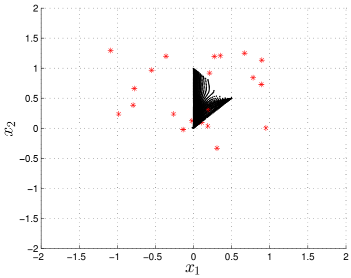
(a) SNS, decision space

(b) SNS, objective space

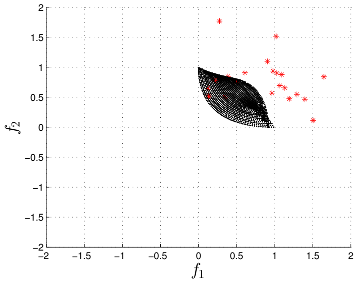
(c) SNS, decision space

(d) SNS, objective space

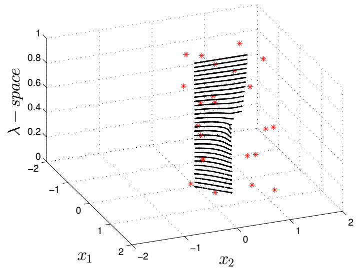
Fig. 7. Numerical results of SNS with a budget of 5,000 FEs for PMOP (18)



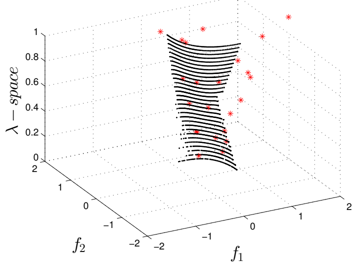
(a) GS 5,000, decision space



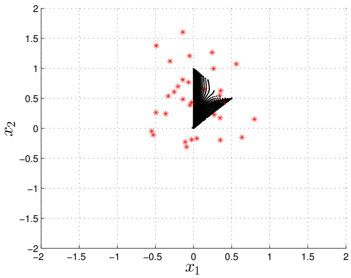
(b) GS 5,000, objective space



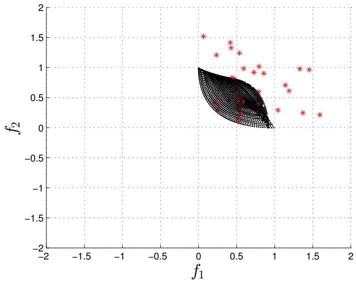
(c) GS 5,000, decision space



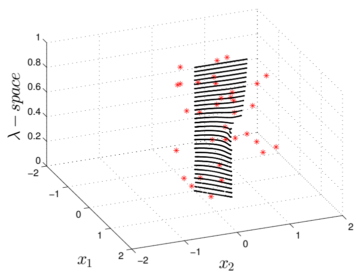
(d) GS 5,000, objective space



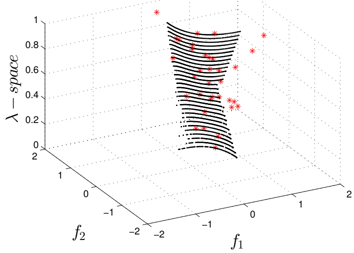
(e) GS 10,000, decision space



(f) GS 10,000, objective space



(g) GS 10,000, decision space



(h) GS 10,000, objective space

Fig. 8. Numerical results of SNS (5,000 FEs) and GS (10,000 FEs) for PMOP (18)

4 Conclusions and Future Work

In this paper we have investigated some aspects of the behavior of stochastic local search (SLS) within parameter dependent multi-objective optimization problems (PMOPs) which can to a certain extent be explained by considering line search with infinitesimal step sizes. By this, we have seen that both the movement toward as well as along the set of interest (in objective space) are inherent in SLS which we have done by considering the extreme cases in the stages of a search process. Further, we have conjectured that there is also a kind of 'opening' of the search in objective space which allows to find in principle all regions of the solution set. Some first tests on a simple set based neighborhood search, called SNS, confirmed these statements on two PMOPs. Thus, the discussions indicate that the problem to find an approximation of the entire solution set of a PMOP is a well-conditioned problem for set based probabilistic algorithms such as evolutionary algorithms (EAs).

For future work, there are some issues that are interesting to address. For instance, we intend to extend the above considerations to constrained problems. Further, the design of a specialized EA for the approximation of the family of Pareto fronts of a PMOP might be interesting. Based on the above insight it might be in particular advantageous to develop strategies that allow to compute the set of interest in *one* run of the algorithm instead of the consideration in 'λ-slices' as done so far. Finally, for the efficient comparison of methods a better adaption of performance indicators would be desirable to assess the final approximation of the family of Pareto fronts.

Acknowledgments. Víctor Adrián Sosa Hernández acknowledges support by the Consejo Nacional de Ciencia y Tecnología (CONACYT). Oliver Schütze acknowledges support from CONACYT project no. 128554. Heike Trautmann acknowledges support by the European Center of Information Systems (ERCIS). All authors acknowledge support from CONACYT project no. 174443 and DFG project no. TR 891/5-1.

References

1. Bank, B., Guddat, J., Klatte, D., Kummer, B., Tammer, K.: Non-Linear Parametric Optimization. Akademie-Verlag, Berlin (1982)
2. Branke, J.: Evolutionary approaches to dynamic optimization problems-updated survey. In: GECCO Workshop on Evolutionary Algorithms for Dynamic Optimization Problems, pp. 27–30 (2001)
3. Branke, J., Kaußler, T., Smidt, C., Schmeck, H.: A multi-population approach to dynamic optimization problems. In: Evolutionary Design and Manufacture, pp. 299–307. Springer (2000)
4. Brown, M., Smith, R.E.: Directed multi-objective optimization. International Journal of Computers, Systems, and Signals **6**(1), 3–17 (2005)
5. Bu, Z., Zheng, B.: Perspectives in dynamic optimization evolutionary algorithm. In: Cai, Z., Hu, C., Kang, Z., Liu, Y. (eds.) ISICA 2010. LNCS, vol. 6382, pp. 338–348. Springer, Heidelberg (2010)

6. Deb, K.: Multi-objective optimization using evolutionary algorithms, vol. 16. John Wiley & Sons (2001)
7. Farina, M., Deb, K., Amato, P.: Dynamic Multiobjective Optimization Problems: Test Cases, Approximations, and Applications. IEEE Transactions on Evolutionary Computation **8**(5), 425–442 (2004)
8. Hernandez, V.A.S.: On the Numerical Treatment of Parametric Multi-Objective Optimization Problems and Memetic Evolutionary Algorithms. Dissertation, CINVESTAV-IPN, December 2013
9. Schütze, O.: Set Oriented Methods for Global Optimization. Dissertation, Universitt Paderborn, December 2004
10. Schütze, O., Esquivel, X., Lara, A., Coello Coello, C.A.: Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization. IEEE Transactions on Evolutionary Computation **16**(4), 504–522 (2012)
11. Schütze, O., Laumanns, M., Tantar, E., Coello, C., Talbi, E.: Computing gap free pareto front approximations with stochastic search algorithms. Evolutionary Computation **18**(1), 65–96 (2010)
12. Tantar, A.-A., Danoy, G., Bouvry, P., Khan, S.U.: Energy-efficient computing using agent-based multi-objective dynamic optimization. In: Kim, J.H., Lee, M.J. (eds.) Green IT: Technologies and Applications. Non-series, vol. 77, pp. 267–287. Springer, Heidelberg (2011)
13. Tantar, A.-A., Tantar, E., Bouvry, P.: A classification of dynamic multi-objective optimization problems. In: Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation, pp. 105–106. ACM (2011)
14. Tinós, R., Yang, S.: A self-organizing random immigrants genetic algorithm for dynamic optimization problems. Genetic Programming and Evolvable Machines **8**(3), 255–286 (2007)
15. Witting, K.: Numerical Algorithms for the Treatment of Parametric Multiobjective Optimization Problems and Applications. Dissertation, Universität Paderborn, February 2012
16. Yang, S.: Genetic algorithms with elitism-based immigrants for changing optimization problems. In: Giacobini, M. (ed.) EvoWorkshops 2007. LNCS, vol. 4448, pp. 627–636. Springer, Heidelberg (2007)