2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
17-20 October, 2021. Melbourne, Australia

Reference Point-Based Particle Sub-Swarm Optimization

Benjamin DeBoer, Conor McDermott, Ali Hosseini, and Carlos Rossa

2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC) | 978-1-6654-4207-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/SMC52423.2021.9659146

Abstract— **In this paper, a novel optimization method named reference point-based particle sub-swarm optimization (RPB-PSWO) is presented. RPB-PSWO utilizes the particle position update method of PSO and with the non-dominance and diversity selection methods of NSGA-II. The multi-objective optimizer utilizes a reference point-based system to allocate particles into an equidistant sub-swarm, in which particles are attracted to a pareto optimal solution in that sub-swarm. To encourage diversity and avoid local minima, density and turbulence factors are included. RPB-PSWO is capable of optimizing problems with many dependent variables, as the position update method of PSO inherently preserves dependent relationships, but suffers from an increased computation cost compared to NSGA-II. The proposed algorithm, although less computationally efficient, is capable of creating diverse pareto front solutions for standardized and custom optimization problems.**

I. Introduction

Optimization is implemented in various facets of modern engineering design, with the focus on minimizing cost, weight, time, etc. In 1998, Altshuler *et al.* were the first to optimize an antenna configuration by means of the genetic algorithm developed in the 1960's by John Holland [1], [2]. The optimization process randomly altered a set of design variables known as a gene, and swapped values between genes, mimicking evolution. Since the development of the genetic algorithm, new methods such as Particle Swarm Optimization (PSO) [3] and Non-dominated Sorting Genetic Algorithm II (NSGA-II) [4] have been created.

PSO was designed as a mono-objective optimization solver by Kennedy *et al.* in 1995, and proved effective for a plethora of optimization problems as explored by Shi *et al.* [3], [5], [6]. Further segmentation of the particle swarm, known as sub-swarms, was designed for multi-modal functions, in which the sub-swarms compete for dominance in local and global solutions [7]. NSGA-II is a multi-objective optimizer based on the genetic algorithm, in which the parent generation is selected based on pareto dominance and diversity [4]. Difficulties arises when objective functions rely on dependent decision variables. The random nature of genetic algorithms (NSGA-II) does not conserve the relationship found between the dependent variables, where the inherent nature of PSO does. To alleviate this issue, the multi-objective benefits of NSGA-II and position update methods of PSO can be combined to create a strong multi-objective optimization method.

Faculty of Engineering and Applied Science, Ontario Tech University, Oshawa, Ontario, Canada. Email: benjamin.deboer@ontariotechu.ca; conor.mcdermott@ontariotechu.ca; sayyedali.hosseini@ontariotechu.ca; rossa@sce.carleton.ca

Multi-objective particle swarm optimization algorithms have been extensively explored. Coelle *et al.* developed a Multi-Objective PSO (MOPSO) in which the particles are attracted to the best solution in the hyper-cube (i.e. a sub-swarm) it occupies, allowing for a pareto front to be developed [8]. Sub-swarms are groups of candidates solutions attracted to the closest non-dominated solutions, conducting a local search around the non-dominated solution. When compared to hybrid multi-objective evolution algorithms, MOPSO performed better for problems with a large amount of decision variables [9]. Speed constrained Multi-objective PSO presented by [10] extends on MOPSO by applying a speed limitation to the particles in the swarm based on a constriction coefficient, to force the particle swarm to conduct a local search while traveling throughout the decision space. The introduction of turbulence and sub-swarms further improved MOPSO [8], [9]. A turbulence factor randomly alters the particles position in order to avoid local minima and encourage diversity [9], [11].

Multi-objective PSO's have been developed with tools to aid convergence and diversity, such as reference points. Where Allmendinger *et al.*, developed a reference point-based particle swarm optimizer that requires the decision maker to select one or many desired solutions (reference points) in the objective space [12]. The method adjusts one particle's position per iteration and analyzes its dominance against the particles previous and swarms global best positions. In the case where all three solutions are non-dominated the two closest to the reference point are selected, generating a localized pareto front. In [13], Figueiredo *et al.* developed a many objective PSO based on equidistant reference points to encourage diversity. The swarm is separated into two sub-swarms, the first focuses on identifying extreme solutions while the other focuses on convergence and swarm diversity. To identify all extreme solutions the first sub-swarm is further broken down into small sub-swarms equal to the number of objective functions.

In this paper, a novel optimization method is described, using the NSGA-III reference point distribution by [14] to equally distribute sub-swarms of particles over the objective space. The results of the proposed method, called Reference Point-Based Particle Sub-Swarm Optimization, (RPB-PSWO) is capable of quickly achieving diverse global pareto fronts for various multi-objective optimization problems. The algorithm is compared with NSGA-II by analyzing the Zitzler–Deb–Thiele (ZDT) test set and a real world optimization problem, in which an actuator for an ankle foot orthosis is optimized for size, power, and total motor range utilization.

The paper is structured as follows: Section II describes the implemented portions of NSGA-II, section III demonstrates the workings of the mono-objective PSO, the novel RPB-PSWO method is proposed in section IV , section V outlines the comparison between NSGA-II and RPB-PSWO on a standardized test, section VI applies RPB-PSWO and NSGA-II to an optimization problem with dependant variables, and finally section VII concludes and states future work.

II. Non-Dominated Sorting-Based Genetic Algorithm II

NSGA-II is a multi-objective optimizer that was developed to ensure the preservation of non-dominated solutions and diversity, by implementing non-dominated sorting and crowding distance methods, respectively [4].

Non-dominated sorting is the process of separating each attribute in the population to a non-dominated front. Solution $\vec{x_1} \in Pop_t$ dominates $\vec{x_2} \in Pop_t$ if all the objective values for $\vec{x_1}$ is equal or better than $\vec{x_2}$, and $\vec{x_1}$ is better than $\vec{x_2}$ in at least one objective [9]. NSGA-II non-dominated sorting assigns each attribute to a respective pareto front, in which front 1 contains all the globally non-dominated solutions. Elarbi *et al.*, proposed a reference point-based dominance in which each attribute in a population is assigned to an equally distributed point on a hyper-plane [15]. The attributes are assigned to the reference point with the minimal distance between the normalized solution and reference point directional vector. The number of reference points in the objective space ($Count_{RP}$) is presented as a binomial coefficient as:

$$Count_{RP} = \binom{n+d-1}{d} \tag{1}$$

where n is the number of objectives and d is the number of divisions between two objectives [14].

Crowding distance analyzes the space between the normalized objective values, as described in [4]. The solutions with the highest crowding distance are selected to be in the parent population to preserve pareto front diversity. This paper intends to implement the inherent strengths of NSGA-II to create a multi-objective PSO.

III. Particle Swarm Optimization

PSO was inspired by the social psychology of schooling fish and flocking birds and is a computationally inexpensive optimizer in mono-objective problems [3], [16]. Consider n particles, analogous to n fish or birds, that have their own position and velocity in the k-dimensional decision space denoted $\vec{p_i}$ and $\vec{v_i}$, respectively. Let:

$$i = \{1, 2, \cdots, n-1, n\}$$

be the index of the a single particle, in which the position of the particle in the decision space is:

$$\vec{p_i} = \{x_1, x_2, \cdots, x_{k-1}, x_k\}$$

The values x_1 and x_k correspond to a position of the particle in the 1^{st} and k^{th} dimension, respectively. Where:

$$\vec{v_i} = \{v_1, v_2, \cdots, v_{k-1}, v_k\}$$

is the velocity of the particle in the decision space, updated at each iteration using:

$$\vec{v_i} = w \cdot \vec{v_i} + c_1 r_1 (\vec{p}_{p_{best}} - \vec{p_i}) + c_2 r_2 (\vec{p}_{g_{best}} - \vec{p_i}). \tag{2}$$

Positions $\vec{p}_{p_{best}}$ and $\vec{p}_{g_{best}}$ are the best solutions for the individual particle (personal) and whole swarm (global), respectively. Scalars r_1 and r_2 are random numbers in $R \in [0, 1]$, and c_1, c_2, and w are set scalar values [5], [6]. The inertial weight (w) of the particle allows the range of the search space to be altered; a larger w promotes exploration, where a smaller value focuses on local exploitation [6]. The cognitive factor (c_1) and social factor (c_2) are known as acceleration constraints, where scalars r_1 and r_2 randomly adjust the social and cognitive effects, altering the particles velocity to reach the p_{best} and g_{best} solutions [5]. To limit the speed of the particles over the decision space, a maximum velocity (V_{max}) constraint is applied to the system. The constraint forces the particle to conduct a local search as it travels through the search space [5]. The V_{max} values are calculated as

$$V_{max} = \rho \cdot (max(\vec{p}(:,:)) - min(\vec{p}(:,:))) \tag{3}$$

determining a unique maximum velocity for each dimension in the objective space. In (3), $\rho \in [0, 1]$ is a user selected scalar with the V_{max} constraint being applied to each particles velocity as follows

$$V_{ik} = \begin{cases} V_{ik} = V_{max} \cdot \text{sign}(V_{ik}) & \text{if } |V_{ik}| > V_{max} \\ V_{ik} & \text{otherwise} \end{cases}. \tag{4}$$

With the velocity evaluated and bounded the updated particle position becomes

$$\vec{x}_i = \vec{x}_i + \vec{v}_i \tag{5}$$

in which the velocity is treated as the changed in particle position per iteration of the PSO algorithm. Once computed the objective function is evaluated and the g_{Best} and p_{Best} solutions are updated. The process is then repeated for a set number of iterations or until an acceptable solution is achieved.

PSO inherently preserves the relationships between the dependant variables during each iteration, as the new position of the particle is based on the best solutions in which the same dependencies are already present. The incorporation of the PSO particle position manipulation into a strong multi-objective optimization method could improve the optimization performance of highly dependant problems.

IV. Reference Point-Based Particle Sub-Swarm Optimization

The proposed RPB-PSWO method uses the reference point distribution and assignment method suggested in [14], the non-dominated sorting and crowding distance methods from [4], with the particle sub-swarm technique described in [9]. To start, a set of y reference points are equally distributed about the normalized k-dimensional hyper plane (See. (1)). An initial particle swarm is randomly distributed across the

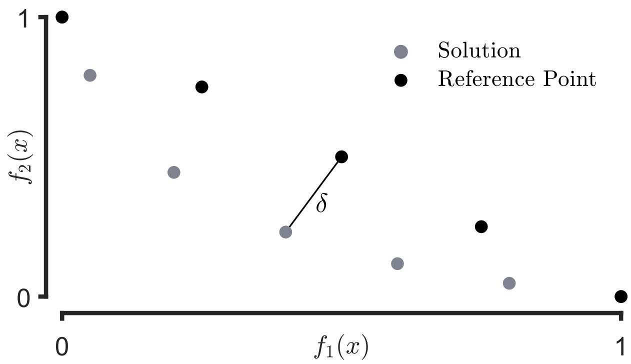


Fig. 1. Reference Point Assignment: In which δ is calculated Euclidean distance between each solution and each reference point. A total of 4 divison resulted in the 5 reference points between the objectives.

decision space and the objective functions evaluated. Each particle is then assigned to the closest reference point based on the lowest Euclidean distance between its normalized objective solution and the distributed reference points, as shown in Fig. 1, where each reference point now presents a particle sub-swarm.

In order to evaluate the PSO iteration, the values of p_{Best} and g_{Best} must be obtained. The p_{Best} of each particle is updated upon each iteration by comparing each of the objectives to the particles previous best value, resulting in n p_{Best} options, where n is the number of objective functions. To find the g_{Best} solution for each sub-swarm, non-dominated sorting is conducted over the whole swarm, and the one or more g_{Best} particles located on the best pareto front for that sub-swarm are selected. If no non-dominated solutions are identified, the sub-swarm particle with the smallest distance to the origin in the objective space is used.

To evaluate the PSO algorithm described in Section III, p_{Best} is randomly selected from the particles n best solutions for each objective function, and g_{Best} is the pareto optimal solution in the sub-swarm the particle belongs to. When multiple pareto optimal solutions are present in the sub-swarm, the optimal solution with the lowest Euclidean distance to the particles previous position in the objective space is chosen. An additional density factor is introduced to encourage swarm diversity and populate the pareto front. The density factor replaces the g_{Best} solution of a particle with the g_{Best} solution of the lowest populous reference point ($g_{Best-div}$) for the current iteration. The method is implemented as follows. First, a g_{Best} solution is retrieved from the lowest populous reference point to become $g_{Best-div}$, followed by the g_{Best} evaluation for each particle determined by:

$$g_{Best}(k) = \begin{cases} g_{Best-div} & \text{if } R < D_p \\ g_{Best}(k) & \text{otherwise} \end{cases} \tag{6}$$

where $R \in [0, 1]$ is a random number and $D_p \in [0, 1]$ is the selected density probability. With the p_{best} and g_{best} selected for each particle, the velocity and position of each particle can be calculated according to (2) and (5).

Finally, turbulence can be implemented into the swarm to allow small changes of the decision variables within the particles, similar to that of NSGA-II mutation. Turbulence is implemented as follows:

$$x_i(k) = \begin{cases} x_i(k) + \beta(k)x_i(k) & \text{if } R(k) < T_p \\ x_i(k) & \text{otherwise} \end{cases} \tag{7}$$

where $\beta \in [0, T_f]$ and $R \in [0, 1]$ are vectors of k random numbers, T_f and $T_p \in [0, 1]$ are the turbulence scalar and probability, respectively. Once the turbulence has been applied to the swarm the multi-objective functions can be evaluated.

The inherent nature of PSO enforces continuous movement of the particle. Therefore, if a pareto optimal solution is identified, the result and position of the particle must be stored in an archive for each reference point, preserving the identified pareto front. Upon evaluation of the next iteration, the archived solutions are re-evaluated for non-dominance. In the case an archived solution is dominated it is removed. A size limit (RP_{save}) is imposed on the archive to minimize the computational time for non-dominated sorting and g_{Best} selection [9]. If the number of non-dominated solutions exceeds RP_{save} for the sub-swarm, the most diverse solutions are selected based on NSGA-II crowding distance. Using crowding distance in multi-objective PSO has been validated by numerous authors [11], [17].

The introduction of reference point-based particle swarms, non-dominated sorting and crowding distance created a unique method of selecting the g_{best} solution for Multi-objective PSO. The inclusion of reference points, a pareto archive, density, and turbulence factors allows the new multi-objective optimizer to achieve diverse pareto fronts while maintaining the relationship between dependent variables.

Since the initial submission of this work, Sharma *et al.* [18] published a similar multi-objective PSO by combining NSGA-III with PSO; in which particles are assigned to reference lines (from the origin of the objective space to each reference point), instead of the proposed reference points. The g_{best} selection criteria is similar in which the non-dominated solution for each reference line is used. The density factor presented in this paper, is replaced with the niche count of each reference line. The implementation of turbulence in the work is not present, however, the g_{best} solution of [18] undergoes evolutionary search in which crossover and mutation is conducted.

V. Optimizer Evaluation and Comparison

To compare RPB-PSWO with the well known NSGA-II, three metrics are used: the convergence for a set number of iterations, pareto diversity by means of Hyper Volume (HV), and computational time per iteration. All computations were conducted in MATLAB on a dedicated computer (Intel Core i5-7200U CPU, 16 GB DDR4 RAM). The respective control parameters of each optimizer is shown in Table I, where NSGA-II encompassed polynomial mutation with a distribution index of 20 and parent tournament selection of size 2.

TABLE I
OPTIMIZER PARAMETERS

NSGA Parameter	Value	RPB-PSWO Parameter	Value
Population Size	100	Swarm Size	100
Mutation Rate	0.8	c_1	2
Crossover Rate	0.8	c_2	2
		$w \in [0, \alpha]$	$\alpha \in [0, 0.8]$
		RP_{save}	10
		T_p	0.4
		T_f	decreasing $\in [0.2, 0]$
		D_p	0.1

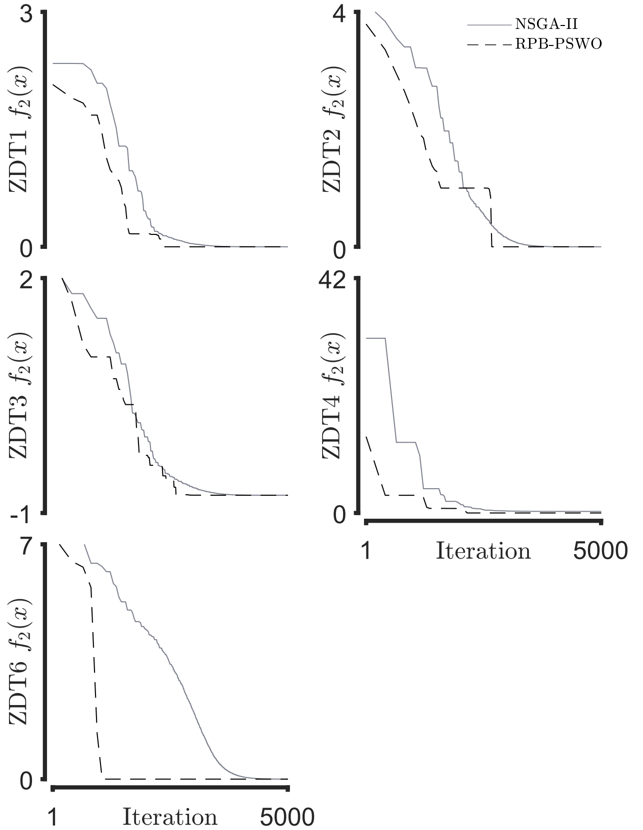


Fig. 2. Convergence of the $f_2(x)$ for each ZDT function, upon a logarithmic x-axis.

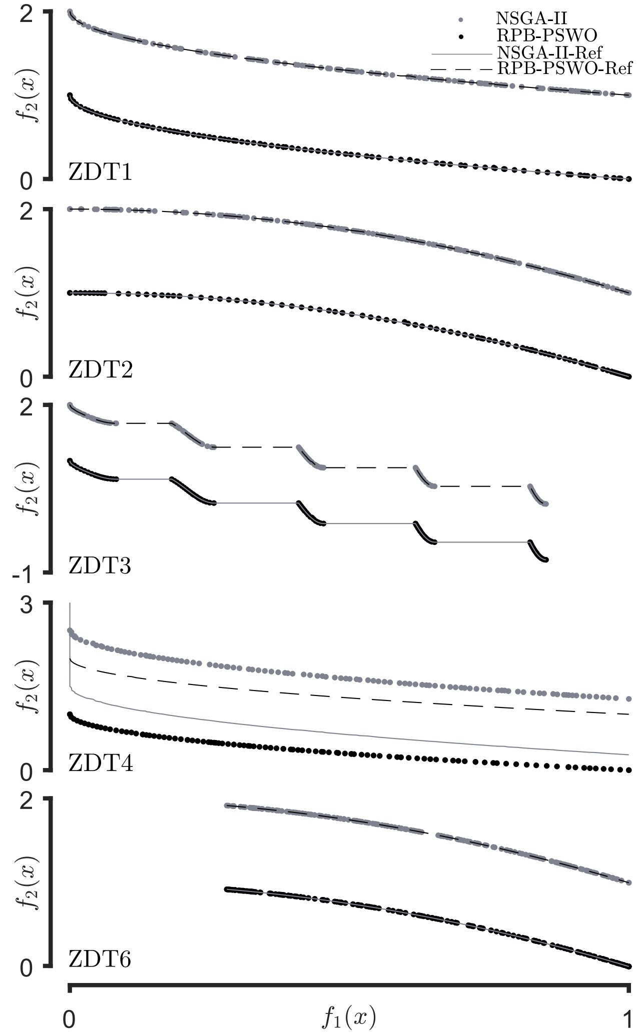


Fig. 3. Pareto Fronts of Respective ZDT Tests: For visualization purposes the value of $f_2(x)$ for NSGA-II is shift up by a value of 1.

Five ZDT tests were conducted (1, 2, 3, 4, 6). Nine divisions (d of (1)) were selected between objectives for RPB-PSWO, creating 10 reference points for the two-dimensional objective space to allow the the final number of archived points for RPB-PSWO and NSGA-II child population size to be equal. The resulting pareto front and convergence of 5000 iterations are presented in Fig. 2 and 3, where the average iteration time and final HV is presented in Table II, and the percent HV is with respect to the point $[1, 1]$ in the objective space.

The ZDT test convergence and pareto fronts (see. Fig 2 and 3) demonstrate that RPB-PSWO is a comparable algorithm to NSGA-II. The convergence of RPB-PSWO exhibits a stair-casing effect, as a new solution can be significantly better than the previous, therefore dominating for longer period of time. NSGA-II converges at a uniform rate, but requires additional iterations to reach the optimal solution. The convergence rate of both NSGA-II and RPB-PSWO can be further optimized; changing the crossover and mutation probability and tournament size for NSGA-II, and RP_{save}, T_p, T_f, d_f for RPB-PSWO. The computational time for NSGA-II is considerably less for all ZDT functions, ZDT4 exempted due to the reduced archived solutions for RPB-PSWO during convergence; thus reducing the computational time for g_{Best} selection and non-dominated sorting. All other ZDT tests had a visible pareto front until convergence, where ZDT4 converged first with a minimum $f_1(x)$ value and was then distributed over the pareto front. The difference in HV between the optimizers is minimal for all test, with the exception of ZDT4. Fig. 3 shows that NSGA-II reached a local pareto front resulting in a reduced percent HV. The pareto fronts generated for both optimizer are diverse, proving that RPB-PSWO is capable of discovering and holding a pareto front.

To analyze the impact of an increased number of divisions

TABLE II
ZDT TEST FINAL HYPER VOLUME & AVERAGE COMPUTATIONAL TIME RESULTS

Function	Time (ms)		HV (%)	
	NSGA-II	RPB-PSWO	NSGA-II	RPB-PSWO
ZDT1	38.4	59.1	67.2	67.0
ZDT2	36.1	51.8	31.6	33.0
ZDT3	41.8	58.6	79.9	80.8
ZDT4	67.0	64.7	35.8	67.1
ZDT6	34.9	48.6	33.4	33.6

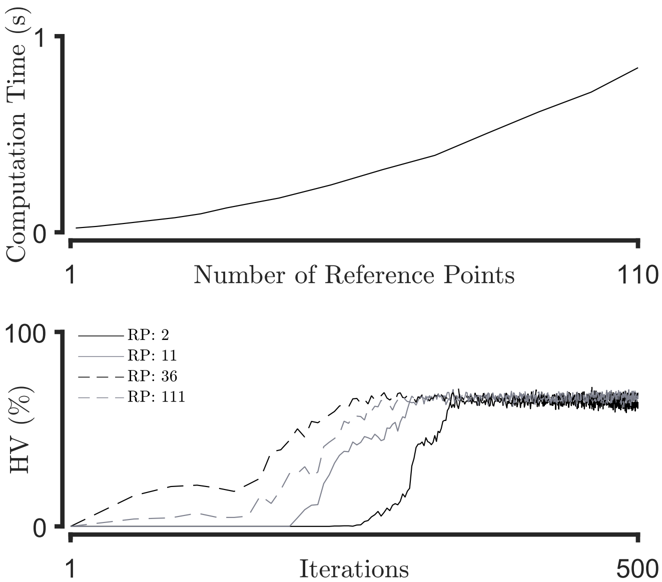
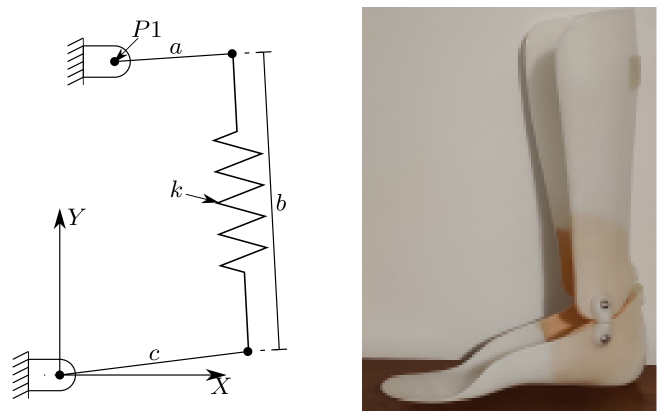


Fig. 4. Computation Time & HV comparison based on the number of reference point (RP) used. The ZDT1 test function was executed with various reference points for 500 iterations.



(a) Series Elastic Crank Rocker Design: Where link b is comprised of a linear spring with constant k.

(b) Common polypropylene ankle foot orthosis, comprised of a the upright shank and horizontal foot-bed.

Fig. 5. A series elastic crank rocker for actuating an ankle foot orthosis, to provide dorsiflexion and plantarflexion assistance to a patient.

(d) in the objective space on computational time, the ZDT1 test was analyzed with $d \in [1, 110]$ and a RP_{save} value of 10. The results (see Fig. 4) show a exponential increase in computation time with respect to divisions. The number of divisions per objective function increases computation time in two ways. First, the g_{Best} sorting method is expensive, as each particle is compared to each archived pareto solution for each reference point. Second, there is a positive correlation between the number of reference points and total archived points ($RP_{save}Count_{RP}$) and thus an increase computation time for non-dominated sorting. In contrast, Fig. 4 shows that convergence and diversity of the pareto front is dependent on the number reference points. Therefore, a trade off is seen between convergence and computational time based on the selected number of reference points.

The ZDT test set proves that RPB-PSWO is a capable alternative to NSGA-II in a multi-objective setting, while suffering from an increased computational time. However, RPB-PSWO should see an increased performance in optimization problems with highly dependent variables.

VI. ACTUATOR OPTIMIZATION

To explore an multi-objective optimization problem with dependent variables, the optimization of an actuation system for an active ankle foot orthosis (AAFO) was explored. The ankle requires a peak maximum power consumption of 3.27 W/kg of weight for a 1.1s gait [19], in which previous works by Hollander *et al.* were able to reduce the required input power by 69% by creating a crank slider mechanism, using a connecting link with a linear spring rate [20]. The proposed mechanical design of the actuator (see Fig. 5a) is a crank rocker configuration in which a motor attached to link a drives the actuator. The output torque is applied to the ankle joint (fixed ground in bottom left of figure) of the user by means of link c, in which a link of linear compliance (b) connects the driven and output links. Both the initial position of links a and c, the length of each link, and the spring rate (linear compliance) of link b can be altered. The proposed crank-rocker configuration increases the dependence's between variables, particularly the link lengths, making the application a good testing ground for RPB-PSWO.

The optimization objectives are to minimize power consumption and total link length, while maximizing the potential rotation of the motor mounted to link a. The optimization problem is constrained by the ability to achieve the correct ankle output angle and torque, while remaining within the maximum output torque of the geared DC motor and abiding by Grashof's criteria for crank-rocker. The convergence results of the optimization can be seen in Fig. 6.

The optimization of the series elastic crank-rocker showed the ability of the RPB-PSWO optimizer to converge or surpass the minimum or maximum objective values identified with NSGA-II (see fig. 6). The stair casing convergence, can be linked to the selection of the control variables for RPB-PSWO such as inertial weight, maximum velocity, turbulence and diversity probability. Overall the crank-rocker case study shows the ability of RPB-PSWO to converge at a faster rate when minimizing the total link length of the system, keeping

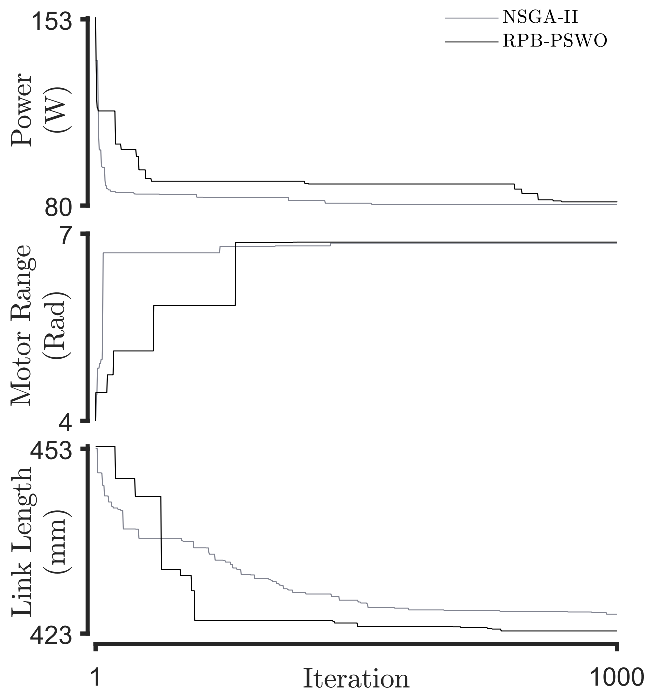


Fig. 6. Ankle Foot Orthosis: Crank-rocker optimization convergence, where RPB-PSWO is able to find the best know minimum of two objectives.

the dependant relationships between the link lengths of a, b, and c; thus proving the effectiveness of RPB-PSWO in highly dependant multi-objective optimization problems.

VII. Conclusion

This paper proposes a novel reference point based multi-objective PSO method combining the position update method of PSO with the multi-objective capabilities of NSGA-II, to preserve variable dependencies in multi-objective problems. RPB-PSWO converges at a faster rate than NSGA-II in the ZDT test cases and is capable of finding better minimums in a real world optimization problem with dependant variables. However, a trade-off is seen between the convergence rate and computation time of RPB-PSWO, as a function of the number of reference points. Therefore, when optimizing a computationally expensive objective function, many reference points should be used, and the inverse applied to an inexpensive function. The design of the proposed crank-rocker AAFO actuation system has high dependencies between the link lengths in order to minimize the size of the system. RPB-PSWO significantly outperform NSGA-II in this optimization metric, showing dominance in highly dependant optimization problems.

Future work can be focused on testing RPB-PSWO in a highly constrained decision space in the presence of various variable dependencies. Further, the effects of the turbulence and density probability can be explored thoroughly to determine the optimal values for various optimization applications, and methods of g_{Best} selection can be tested to minimize the total computational time.

References

[1] E. E. Altshuler and D. S. Linden, "Process for the design of antennas using genetic algorithms," Feb. 17 1998, uS Patent 5,719,794.

[2] M. Mitchell, *An introduction to genetic algorithms*, ser. Complex adaptive systems. Cambridge, Mass: MIT Press, 1996.

[3] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-International Conference on Neural Networks*, vol. 4. IEEE, 1995, pp. 1942–1948.

[4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[5] Y. Shi *et al.*, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546)*, vol. 1. IEEE, 2001, pp. 81–86.

[6] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)*, vol. 3. IEEE, 1999, pp. 1945–1950.

[7] J. Zhang, D.-S. Huang, and K.-H. Liu, "Multi-sub-swarm particle swarm optimization algorithm for multimodal function optimization," in *2007 IEEE Congress on Evolutionary Computation*. IEEE, 2007, pp. 3215–3220.

[8] C. C. Coello and M. S. Lechuga, "Mopso: A proposal for multiple objective particle swarm optimization," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, vol. 2. IEEE, 2002, pp. 1051–1056.

[9] S. Mostaghim and J. Teich, "Covering pareto-optimal fronts by subswarms in multi-objective particle swarm optimization," in *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753)*, vol. 2. IEEE, 2004, pp. 1404–1411.

[10] A. J. Nebro, J. J. Durillo, J. Garcia-Nieto, C. C. Coello, F. Luna, and E. Alba, "Smpso: A new pso-based metaheuristic for multi-objective optimization," in *2009 IEEE Symposium on computational intelligence in multi-criteria decision-making (MCDM)*. IEEE, 2009, pp. 66–73.

[11] R. A. Santana, M. R. Pontes, and C. J. Bastos-Filho, "A multiple objective particle swarm optimization approach using crowding distance and roulette wheel," in *2009 Ninth International Conference on Intelligent Systems Design and Applications*. IEEE, 2009, pp. 237–242.

[12] R. Allmendinger, X. Li, and J. Branke, "Reference point-based particle swarm optimization using a steady-state approach," in *Asia-Pacific Conference on Simulated Evolution and Learning*. Springer, 2008, pp. 200–209.

[13] E. M. Figueiredo, T. B. Ludermir, and C. J. Bastos-Filho, "Many objective particle swarm optimization," *Information Sciences*, vol. 374, pp. 115–134, 2016.

[14] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints," *IEEE transactions on evolutionary computation*, vol. 18, no. 4, pp. 577–601, 2014.

[15] M. Elarbi, S. Bechikh, A. Gupta, L. B. Said, and Y.-S. Ong, "A new decomposition-based nsga-ii for many-objective optimization," *IEEE transactions on systems, man, and cybernetics: systems*, vol. 48, no. 7, pp. 1191–1210, 2017.

[16] Q. Bai, "Analysis of particle swarm optimization algorithm," *Computer and information science*, vol. 3, no. 1, p. 180, 2010.

[17] C. R. Raquel and P. C. Naval Jr, "An effective use of crowding distance in multiobjective particle swarm optimization," in *Proceedings of the 7th Annual conference on Genetic and Evolutionary Computation*, 2005, pp. 257–264.

[18] D. Sharma, D. Agarwal, and S. Kumar, "Reference-lines steered guide assignment and update for pareto-based many-objective particle swarm optimization," *Evolutionary Intelligence*, pp. 1–26, 2021.

[19] G. Bovi, M. Rabuffetti, P. Mazzoleni, and M. Ferrarin, "A multiple-task gait analysis approach: kinematic, kinetic and emg reference data for healthy young and adult subjects," *Gait & posture*, vol. 33, no. 1, pp. 6–13, 2011.

[20] K. W. Hollander, R. Ilg, T. G. Sugar, and D. Herring, "An efficient robotic tendon for gait assistance," *Journal of biomechanical engineering*, vol. 128, no. 5, pp. 788–791, 2006.