Synergetic Population Search Strategies

(CS447 SP2011 Final Report)

Raghavendra Kotikalapudi
Department of Computer Science
Missouri University of Science and Technology
Rolla, Missouri, USA
Email: rkyvb@mst.edu

***Abstract*—Population search strategies are amongst the most robust and powerful tools available to solve complex optimization problems. Unfortunately, results such as the No Free Lunch (NFL) theorem establishes that it is impossible for a single search strategy to work well across all search spaces; however, it does not refute the possibility of a strategy that works well on a diverse class of problems – especially of practical significance. The main focus of this paper is to develop a strategy that automates the hybridization process by dynamically combining the strengths of various search strategies, resulting in a more general strategy. In particular, Synergetic Population Search Strategies ($SPSS$) utilizes k search strategies in parallel, each evolving their own regional sub-populations. After every generation, all sub-populations compete though a global selection operator for faster convergence. In order to prevent the early extinction of potential sub-populations, and to preserve diversity, random individuals from fitter sub-populations migrate to weaker sub-populations, resulting in a cooperative behavior. Numerical experiments evaluating SPSS (using DE and PSO) with individual DE and PSO on six popular benchmark problems shows that the performance improvement of SPSS is statistically significant on unimodal and non noisy multimodal cost functions with relatively low number of local optima, while performing no worse on deceptive problems.**

I. INTRODUCTION

The quest for general optimization techniques is of paramount importance due to their wide spread applicability in many areas such as Engineering, Mechanics, Economics, Operations Research, and Modeling. Amongst the myriad of optimization techniques, population search strategies are more robust and general[1] than gradient or derivative based methods due to the use of multiple candidate solutions, cooperating to find the global optima. In spite of various successful applications, all search algorithms are fundamentally limited by the No Free Lunch (NFL) theorem, which establishes that it is impossible for a universal search strategy to work well on all conceivable cost functions [1]. However, the set of all possible cost functions includes ones with no particular order or pattern. Fig 1 shows one such example. It is easy to imagine why random search is equivalent to hill climbing or any other evolutionary algorithm on such landscapes.

Unlike search spaces with no pattern or order, many real world optimization problems have a definitive structure to them. The main goal of this research is to identify a suitable search strategy that works well on a *diverse* class of problems – especially of practical significance. Currently, there are two ways to approach this problem: 1) Invent new algorithms; 2) Hybridize existing algorithms.

[1]Works efficiently for a wider class of cost functions

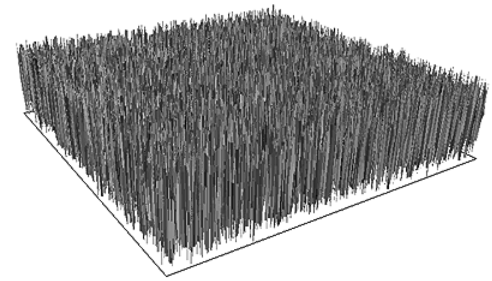


Fig. 1. A Random and Chaotic Fitness Landscape

Following the first approach, the last few decades witnessed the advent of new optimization algorithms such as Particle Swarm Optimization (PSO) [2], Ant Colony Optimization (ACO) [3], Differential Evolution (DE) [4], Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [5], Simulated Annealing (SA) [6], Bee Colony Optimization (BCO) [7] and many others. Although not truly general, some algorithms are better suited on certain problems when compared to others. For example, ACO is known to work well with traveling salesman problem [8], while simple methods such as gradient descent is very efficient on unimodal cost functions.

The second alternative of hybridizing existing algorithms capitalizes on the strengths of individual search algorithms and aims to combine them in a single search strategy. In many cases, the resulting hybrid algorithm is more general than the constituents as demonstrated by the success of GA-PSO [9], GA-ACO [10], and DE-PSO [11].

While increasingly general hybrids are being discovered, manual research methods for the development of new or hybrid search strategies take a significant amount of time, money and effort. Currently, there are at least 15 popular search strategies, yielding approximately 1.3×10^{12} possible hybrids. Using manual research methods, it is not feasible to explore all possible hybrid algorithms – much less identify a hybrid suited to the given optimization task.

The main focus of this paper is to automate the hybridization process by facilitating synergetic interactions among various population search algorithms, combining their strengths. In particular, the goal is to identify a hybridization strategy H that utilizes k constituent population search strategies such that it finds better quality solutions faster than any of the individual search strategies, if run separately. Given a set of population search strategies $S = \{S_1, S_2, ..., S_k\}$, the hybrid search strategy H should satisfy Equations 1, 2 simultaneously.

$$SolQuality\,(H) > SolQuality\,(S_i)\,\forall S_i \in S \quad (1)$$

$$Evals\,(H) < Evals\,(S_i)\,\forall S_i \in S \quad (2)$$

II. Background

The idea of combining multiple algorithms dates back to the success of ensemble neural networks performing classification tasks [12]–[14]. In the evolutionary optimization community, the ensemble effect is usually achieved by running multiple search strategies in parallel. In addition to running multiple search strategies in parallel, the ensemble strategy described in [15] incorporates cooperation between heterogeneous search paradigms by interrupting the search process periodically and exchanging positive or negative information about the search space. This approach, however, mandates an explicit communication format for information exchange.

Pohlheim used a similar approach with the regional population model but goes a step further by dynamically allocating computational resources to various sub-populations based on their relative performance [16]. Unfortunately, as this work was published before 2005, performance evaluations on the standard CEC2005 benchmark problems [17] is not available, yielding little of no means of comparison with current state of the art approaches. Furthermore, there is no explicit information exchange between constituent search strategies, potentially resulting in a lot of redundant searches.

In 2007, Jasper A. Vrugt *et al.* coined the term "multi-method search" to evoke the notion of using ensembles of multi-objective search algorithms. In AMALGAM, the population size of several multi-objective search algorithms, exploring the fitness landscape concurrently, are adapted to favor individual algorithms that exhibit the highest reproductive success [18]. Though similar to work in [16], the competition aspect was enforced differently. AMALGAM-SO extends this idea to optimize single objective, real parameter spaces [19] by using a restart strategy. With each restart, the population size of promising sub-populations (islands) is heuristically increased, receiving a larger share of computational resources. In both algorithms, a single regional population is used. Instead, the use of multiple regional populations is less risky as the search space is explored from multiple vantage points. The effect is even more significant in high dimensional spaces.

Lai used a similar approach where a single population is evolved separately using GA and PSO. After every epoch, fit individuals are separately selected from both sub-populations and are combined to replace the public population [20]. Experimental results showed that the ensemble is twice as efficient as the individual GA or PSO. More recently, Peng et al. proposed the population algorithm portfolios (PAP) as a general framework for regional migration and competition [21]. Their major contribution includes the use of *risk metrics* to define the risk associated with different sub-populations. In spite of competitive results on standard benchmark problems, the use of a static resource allocation scheme is somewhat limiting.

III. Motivation

In the evolutionary optimization community, almost all ensemble approaches are based on the idea of running multiple search strategies simultaneously. In the regional population model, multiple regions are used. This has advantages as the search space is explored from multiple vantage points. However, the use of migration is parametric and can be sensitive in some problems. Another approach is to use a single sub-population with each search strategy contributing towards individuals in the next generation. Although the choice of migration parameters is mitigated, the use of single sub-population results in a rapid loss of diversity, something that could have been avoided with regional sub-populations. Furthermore, the loss of diversity is even more of an issue in high dimensional spaces.

Synergetic Population Search Strategies ($SPSS$) aims to preserve both the advantages: 1) The use of regional population model without migration parameters; 2) Rapid convergence without the loss of diversity. In order to promote rapid convergence, various sub-populations can compete with each other, diminishing the weaker, unpromising sub-populations. This approach has some similarities to competitive coevolution [22] as the survival of an individual within a species also depends on the fitness of other individuals in different sub-populations. However, unlike competitive coevolution, fitness measure is absolute, determined by using an objective function. This scheme, however, exacerbates the loss of diversity. To avoid the rapid loss of diversity, and to prevent the extinction of potential sub-population strategies, random migration scheme can be used to renew the weak individuals. With this approach, migration can be emergent rather than being parametric.

IV. Methodology

$SPSS$ begins by initializing k sub-populations $\{P_1, P_2, ..., P_k\}$, one for each population search strategy S_i provided by the user. All sub-populations are initialized using the maximum Euclidean distance Latin hypercube sampling (LHS) method to ensure maximum spread from each other [23]. During the optimization phase, each sub-population P_i is updated using the corresponding search strategy S_i for one generation, resulting in an updated population P_i'. Thereafter, various sub-populations compete through the use of a global selection operator on $\bigcup_{i=1}^{k} P_i \cup P_i'$. Prior to selection, individuals are tagged so that their correspondence to the individual search strategies can be identified at a later stage. After the selection process, individual tags are used to split the surviving population into their corresponding sub-populations.

As a result of survival selection, fitter sub-populations increase in size. If evolution were to continue in this manner, the less fit sub-populations will eventually be extinct. There are several possibilities for preventing the early extinction of potential sub-populations: 1) Sub-populations with size greater than $popSize$ can be truncated to $popSize$ and new individuals (randomly initialized or seeded) can be added to sub-populations with size lesser than $popSize$; 2) Randomly selected individuals can be migrated from larger sub-population to smaller sub-populations. In $SPSS$, the latter methodology is adopted as it prevents disengagement[2]. However, frequent migrations can result in regional sub-populations heading towards each other. In order to enforce separation, and to maintain diversity, the selection operator deletes similar individuals and uses fitness uniform selection scheme (FUSS) [24]. The overall framework is illustrated in Algorithm 1.

Algorithm 1 SPSS Algorithm

procedure SPSS($S\,[1..k]\,, popSize,$)
  for $i \leftarrow 1..k$ **do** ▷ Initialization
    Initialize $P\,[i]$ to $Popsize$ using LHS
    Tag all individuals in $P\,[i]$ to i
  end for

  while Not_Termination_Condition **do**
    $combined \leftarrow NULL$
    for $i \leftarrow 1..k$ **do** ▷ For each algorithm
      Add $P\,[i]$ to $combined$
      Update $P\,[i]$ using $S\,[i]$
      Add $P\,[i]$ to $combined$
    end for
    Delete redundant individuals from $combined$
    $combined \leftarrow FUSS\_Select\,(combined)$ ▷ [24]

    $excess \leftarrow NULL$ ▷ Identify emigrants
    for $i \leftarrow 1..k$ **do** ▷ For each sub-population
      if $|P_i| > popSize$ **then**
        Move $(popSize - |P_i|)$
        random individuals to $excess$
      end if
    end for
    for $i \leftarrow 1..k$ **do** ▷ Immigration
      if $|P_i| < popSize$ **then**
        Move $(|P_i| - popSize)$ random
        individuals from $excess$ to P_i
      end if
    end for
    Update tags $\forall P\,[i] \in P$
  end while
end procedure

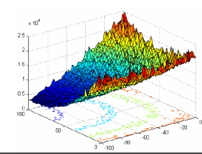
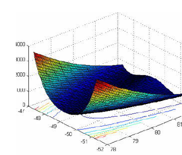
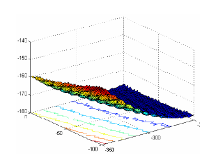
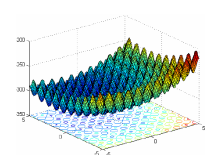
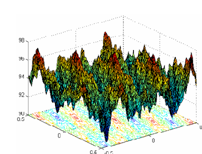
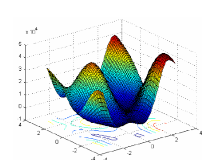
Note that $SPSS$ is not entirely competitive. Random migrations from fitter sub-populations to lower fitness sub-populations signifies cooperation and is equivalent to information sharing. Through the interplay of competition and cooperation, $SPSS$ should be able to escape the local minima, resulting in better resilience to sub-optimal convergence.

[2]Occurs when one population evolves so much faster than the other that all individuals of the other are utterly defeated, making it impossible to differentiate between better and worse individuals without which there can be no evolution

V. Experimental Setup

The performance of the $SPSS$ is evaluated on 6 real parameter optimization functions defined in the CEC2005 benchmark suite [17]. The six functions are chosen to span a diverse set of problem features such as multimodality, separability, scalability, noise and ill-conditioning. Table I provides a listing of all the functions, along with their features.

TABLE I
Test functions for real parameter optimization

Function	Features	3D Landscape
Shifted Schwefel	Unimodal Shifted Non-separable Scalable Noisy	
Shifted Rosenbrock	Multimodal Shifted Non-separable Scalable	
Shifted Rotated Griewank	Multimodal Shifted Non-separable Scalable Rotated	
Shifted Rastrigin	Multimodal Shifted Separable Scalable Many local minima	
Shifted Rotated Weierstrass	Multimodal Shifted Non-separable Scalable Rotated	
Schwefels Problem	Multimodal Shifted Non-separable Scalable	

Each test function is solved in 30 dimensions, using 25 independent trials[3] to obtain statistically meaningful results.

[3]As suggested by the authors of the CEC2005 benchmark

Additionally, the optimization process is restricted to a maximum number of function evaluations (Max-FES) given by $10000 \times D$, where D is the dimensionality of the problem, or when the error in the function value is 10^{-8} or less.

In principle, $SPSS$ is a very general framework and can incorporate *any* population based search strategy. However, for the purposes of this study, DE and PSO are chosen due to their widespread popularity and applicability to real parameter optimization. Other popular strategies such as CMA-ES, GA, and Hill climbing will be included as a part of the future work.

In order to validate if Equations 1 and 2 hold, SPSS using DE and PSO as the constituent search strategies is compared with individual DE and PSO using the same parameter settings. Both DE and PSO use the default parameter settings as suggested by the corresponding authors and is listed in Tables II and III. The population size is arbitrarily set to 50. Note that most parameter settings are not important as the goal of the experiments is to validate if Equations 1, 2 are satisfied $SPSS$.

TABLE II
PARAMETER SETTINGS FOR DE

Parameter	**Value**
DE-Type	Current-to-Best/1 Scheme
F (Amplification)	0.8
K (Alteration probability)	0.6
Lambda	0.6
Mutation	Trig. mutation with 0.05 probability

TABLE III
PARAMETER SETTINGS FOR PSO

Parameter	**Value**
Φ_1	2.05
Φ_2	2.05
Topology	Grid
Topology range	1

VI. RESULTS

Convergence graphs for all six functions are shown in Fig 2 - 7. To further analyze the fitness progression, the box and whisker plot displaying the minimum value, lower quartile, median, upper quartile and the maximum value is used.

On the Unimodal Shifted Schwefel, Fig 2 shows that SPSS exhibits faster convergence when compared to DE or PSO. This is likely due to the fact that DE and PSO populations in SPSS compete, driving each other to increasing levels of fitness. Analysis of variance (ANOVA) with $\alpha = 0.05$ on the median values of SPSS-PSO and SPSS-DE groups revealed that the performance improvement of SPSS is statistically significant.

Convergence on the Shifted Rosenbrock function is shown in Fig 3. While DE and PSO converge – possibly due to low population sizes – SPSS was able to escape the local minima. Similar characteristics were observed on Shift Rotated Griewank (Fig 4). In both cases, in spite of escaping the local minima, the improvement is not statistically significant.

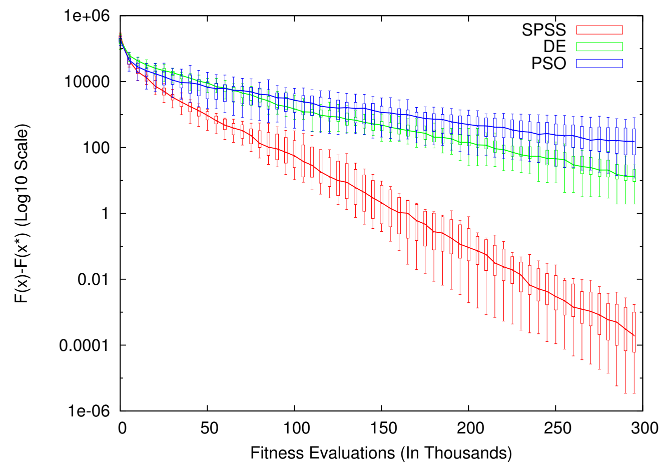


Fig. 2. Convergence progress on 30D Shifted Schwefel

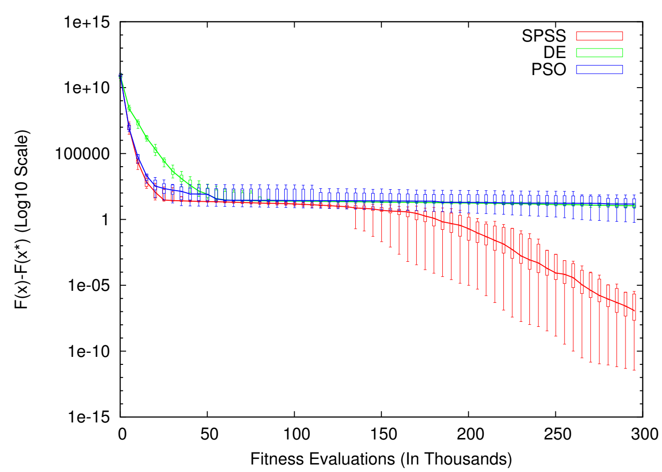


Fig. 3. Convergence progress on 30D Shifted Rosenbrock

The fitness progression on Shifted Rastrigin's function (Fig 5) is particularly interesting. Although SPSS eventually outperforms DE and PSO in terms of median, the variance is comparatively higher. While SPSS is able to preserve diversity, it converges a lot slower. Statistical results indicate almost no improvement of SPSS over DE or PSO. However, it should be noted that SPSS is still no worse than DE or PSO.

Convergence characteristics on Shifted Rotated Weierstrass (Fig 6) and Schwefels Problem (Fig 7) appear similar to that on Shifted Rastrigin's function. However, the performance of SPSS is statistically better than PSO on Weierstrass, while it outperforms both DE and PSO on the Schwefels Problem.

Interestingly, SPSS was able to find a better quality solution on all six benchmark problems when compared with DE or PSO at the end of 3×10^5 fitness evaluations.

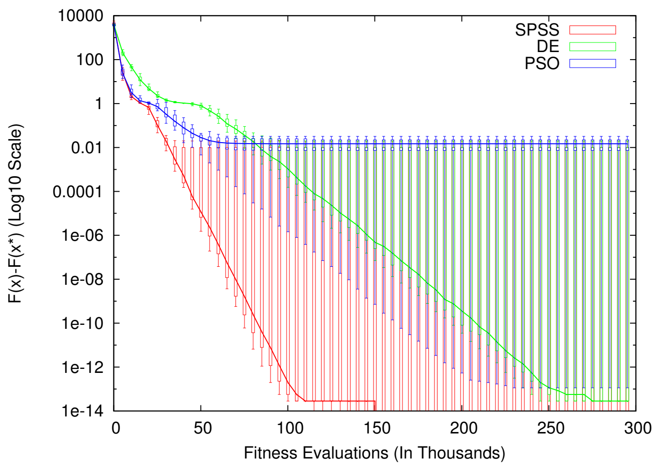


Fig. 4. Convergence progress on 30D Shifted Rotated Griewank

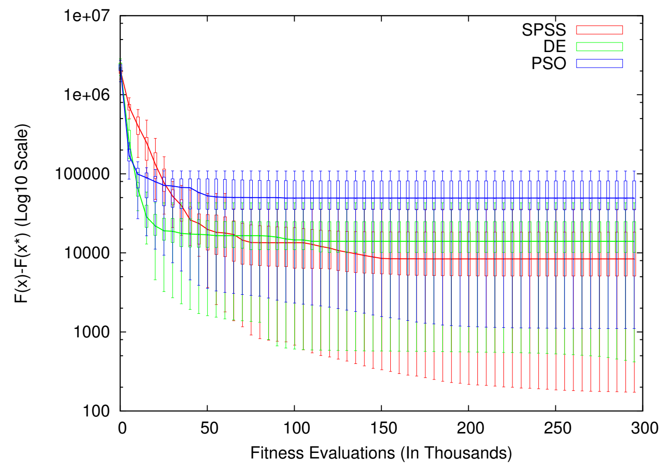


Fig. 7. Convergence progress on 30D Schwefels Problem

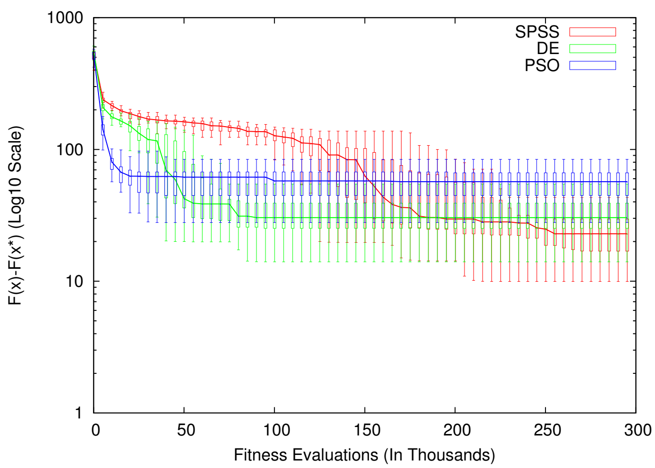


Fig. 5. Convergence progress on 30D Shifted Rastrigin

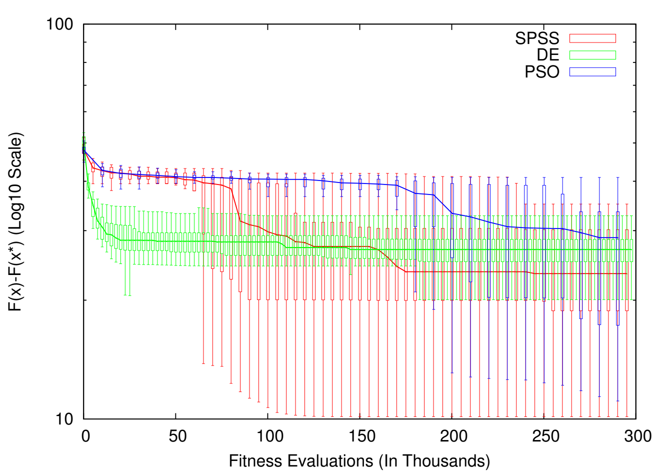


Fig. 6. Convergence progress on 30D Shifted Rotated Weierstrass

VII. Discussion

On unimodal cost functions such as Schwefel's, in spite of the added noise, the performance of SPSS is statistically better than that of an individual DE or PSO. This improvement is very likely attributed to SPSS' ability to engage constituent search strategies in competition.

On multimodal Rosenbrock and Griewank's function, even though SPSS performs better, the improvement is not statistically significant. This is likely due to the diversity preservation mechanisms in SPSS. In multimodal functions, the lack of diversity can cause premature convergence. From Fig 3, it may be noted that SPSS continues to converge while both DE and PSO are stuck at local optima. In this regard, SPSS performs better than PSO on Greiwanks, if not on DE.

Of all the multimodal functions, the performance improvement on Rastrigin's function is the least. While not worst than DE or PSO, the improvement is almost negligible. Interestingly, the performance improvement of SPSS over PSO is statistically significant in Weierstrass' function, while it outperforms (statistically) both PSO and DE in Schwefel's Problem. The only difference between Schewels and Weierstrass is that of noise, indicating that SPSS may perform better on non-noisy functions. The only difference between Rastrigin's to Schewels and Weierstrass is that separability. In addition, Rastrigin's function has a large number of local optima. From these observations, it may be hypothesized that SPSS is not particularly effective on separable and deceptive problems such as Rastrigin's.

All the aforementioned results point to the fact that the diversity preservation mechanism in SPSS is overemphasized. Perhaps, elitist selection could have worked better than FUSS.

VIII. Conclusion

In the past few decades, a number of population search algorithms have emerged, each with their own strengths and weaknesses. This paper showed that both convergence speed and solution quality can be improved by by facilitating synergetic interactions between various population search strategies. In particular, a general hybridization framework (SPSS) was proposed to combine the best features of individual search strategies. In SPSS, individual search populations compete for faster convergence, while cooperating with each other to exchange information and to preserve diversity.

Numerical results on six popular benchmark problem comparing SPSS (DE + PSO) with DE and PSO showed that SPSS was able to find a better quality solution faster than the individual DE or PSO. Preliminary results suggest that SPSS works particularly well on unimodal and non noisy multimodal functions with relatively low number of local optima. On multimodal functions with large number of local optima, SPSS performs no worst than the constituent search strategies. In all cases, SPSS was able to find a better quality solution than the individual DE or PSO at the end of 3×10^5 functions evaluations.

While promising, a more detailed performance evaluation is required. In particular, SPSS needs to be tested on the remaining 19 functions in the CEC2005 benchmark suite. Additionally, other popular search strategies such as the CMA-ES, Ant colony optimization, Firefly optimization algorithm, etc., needs to be tested within the SPSS framework. The author encourages interested researchers to further analyze and evaluate the proposed framework. The source code of SPSS written in JAVA can be obtained from the author upon request.

IX. Future Work

In its current form, SPSS allocates equal number of CPU resources (fitness evaluations) to different search strategies, irrespective of their performance. However, depending on the fitness landscape, some search strategies may work better than the others. For example, in simple unimodal functions, more resources should be allocated to hill climbing as it converges faster. While this approach has the risk of premature convergence, preliminary results indicate that the resulting loss in diversity may not be problematic. In SPSS, there are at least two ways of incorporating these improvements:

1) Depending on the relative performance, variable fitness evaluations could be allocated to different search strategies, resulting in a more equitable distribution of resources. In particular, sub-population fitness evaluations could be made proportional to the amount of fitness gained per unit evaluation, rewarding sub-populations with faster convergence.

2) Instead of using inter-population migrations for maintaining equal population sizes, weak sub-populations could be allowed to go extinct. However, in order to preserve the potential use of weaker strategies at a latter stage, extinct search strategies could be allowed to re-enter the search space by capturing excess individuals from fitter sub-populations. A particularly well suited strategy would thrive, leading to extinction of unsuitable sub-populations. However, when a search strategy loses its advantage, either due to premature convergence, or simply because of its inability to work efficiently on the current fitness landscape, more suitable re-entering strategies would take over. Moreover, with re-entry capturing the most fit individuals, the problem of disengagement can be avoided.

Additionally, the performance of SPSS on high dimensional landscapes ($> 1000D$) remains to be seen.

References

[1] D. Wolpert and W. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 2002.

[2] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, IEEE, 1995.

[3] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *Systems, Man, and Cybernetics, Part B: IEEE Transactions on Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.

[4] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[5] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation," in *IEEE International Conference on Evolutionary Computation, 1999*, pp. 312–317, IEEE, 1996.

[6] S. Brooks and B. Morgan, "Optimization using simulated annealing," *Journal of the Royal Statistical Society. Series D (The Statistician)*, vol. 44, no. 2, pp. 241–257, 1995.

[7] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[8] O. Gomez and B. Baran, "Reasons of ACOs Success in TSP," *Ant Colony, Optimization and Swarm Intelligence*, pp. 344–365, 2004.

[9] K. Premalatha and A. Natarajan, "Hybrid PSO and GA for global maximization," *Int. J. Open Problems Compt. Math*, vol. 2, no. 4, 2009.

[10] A. Acan, "GAACO: A GA+ACO hybrid for faster and better search capability," *Ant Algorithms*, pp. 15–26, 2002.

[11] W. Zhang and X. Xie, "DEPSO: Hybrid particle swarm with differential evolution operator," in *IEEE International Conference on Systems, Man and Cybernetics, 2003.*, vol. 4, pp. 3816–3821, IEEE, 2003.

[12] L. Hansen and P. Salamon, "Neural network ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993–1001, 1990.

[13] Z. Wu and Y. Chen, "Genetic algorithm based selective neural network ensemble," in *IJCAI-01: proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, Seattle, Washington, August 4-10, 2001*, p. 797, 2001.

[14] Z. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: Many could be better than all," *Artificial intelligence*, vol. 137, no. 1-2, pp. 239–263, 2002.

[15] J. Denzinger and T. Offermann, "On cooperation between evolutionary algorithms and other search paradigms," in *Proceedings of 1999 Congress on Evolutionary Computation CEC'99*, vol. 3, IEEE, 1999.

[16] H. Pohlheim, "Competition and cooperation in extended evolutionary algorithms," in *GECCO2001 - Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 331–338, 2001.

[17] P. Suganthan, N. Hansen, J. Liang, K. Deb, Y. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," *KanGAL Report*, 2005.

[18] J. Vrugt and B. Robinson, "Improved evolutionary optimization from genetically adaptive multimethod search," *Proceedings of the National Academy of Sciences*, vol. 104, no. 3, p. 708, 2007.

[19] J. Vrugt, B. Robinson, and J. Hyman, "Self-adaptive multimethod search for global optimization in real-parameter spaces," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 243–259, 2009.

[20] X. Lai and M. Zhang, "An efficient ensemble of GA and PSO for real function optimization," *International Conference on Computer Science and Information Technology*, vol. 0, pp. 651–655, 2009.

[21] F. Peng, K. Tang, G. Chen, and X. Yao, "Population-based algorithm portfolios for numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 5, pp. 782–800, 2010.

[22] W. Hillis, "Co-evolving parasites improve simulated evolution as an optimization procedure," *Physica D: Nonlinear Phenomena*, vol. 42, no. 1-3, pp. 228–234, 1990.

[23] M. McKay, R. Beckman, and W. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.

[24] M. Hutter, "Fitness uniform selection to preserve genetic diversity," in *Proceedings of the World Congress on Computational Intelligence*, pp. 783–788, IEEE, 2002.