Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 915 – 920
ISBN 978-83-60810-14-9
ISSN 1896-7094

Communication Network Design Using Particle Swarm Optimization

C. Papagianni
School of Electrical and Computer Engineering, National Technical University of Athens, Iroon Polytechneioy 9 Str. 15773 Zografou, Athens, Greece

Email: chrisap@telecom.ntua.gr

K. Papadopoulos, C. Pappas,
School of Electrical and Computer Engineering, National Technical University of Athens, Iroon Polytechneioy 9 Str. 15773 Zografou, Athens, Greece

Email: kpapadop@esd.ece.ntua.gr, chrispap@telecom.ntua.gr

N. D. Tselikas,
D. T. Kaklamani, I. S Venieris
School of Electrical and Computer Engineering, National Technical University of Athens, Iroon Polytechneioy 9 Str. 15773 Zografou, Athens, Greece

Email: ntsel@telecom.ntua.gr, dkaklam@cc.ece.ntua.gr, venieris@cs.ntua.gr

Abstract—**Particle Swarm Optimization is applied on an instance of single and multi criteria network design problem. The primary goal of this study is to present the efficiency of a simple hybrid particle swarm optimization algorithm on the design of a network infrastructure including decisions concerning the locations and sizes of links. A complementary goal is to also address Quality of Service issues in the design process. Optimization objectives in this case are the network layout cost and the average packet delay in the network. Therefore a multi-objective instance of the hybrid PSO algorithm is applied. The particular hybrid PSO includes mutation to avoid premature convergence. For the same reason repulsion/attraction mechanisms are also applied on the single objective case. Mutation is passed on to the multi-objective instance of the algorithm. Obtained results are compared with corresponding evolutionary approaches.**

I. Introduction

The increasing complexity of the network design problems calls for advanced optimization techniques. Network design problems where even a single cost function is optimized are often NP-hard [1]. In addition communication network design problems are not time critical. Therefore approaches have been designed to address these problems based in meta-heuristics such as simulated annealing, taboo search, evolutionary computing, nature inspired algorithms or both [2][3]. Concerning evolutionary computing in telecommunication network design, a comprehensive study is presented in [4] up to 2005 containing relevant research study references, where network design problems are classified in node location problems, topology design, tree design, routing, restoration, network dimensioning, admission control and frequency assignment/wavelength allocation. The optimization techniques employed are mainly variations of Genetic Algorithms. Additional work on telecommunication network optimization has followed in the last three years.

Real world network design problems normally involve the simultaneous optimization of multiple and usually partially contradicting objectives. Therefore more often than not, there is not a single optimal solution, given the diversity of the set of objectives, but a set of congruent solutions, known as Pareto-optimal. The topological design of communication networks is usually a multi-objective problem involving simultaneous optimization of the cost concerning network deployment as well as various performance criteria (e.g. average delay, throughput) subject to additional constraints (e.g. reliability, bandwidth). These problem specific objectives are often opposing; for example a way to reduce average delay in the network is over provisioning; that is to increase available link capacities which will consequently result in the increase of the total network deployment cost.

In this paper we will use Particle Swarm Optimization algorithm for the Topological Network Design problem, including capacity allocation, considering shortest path routing. Therefore the target is to design a near optimal network infrastructure, including decisions concerning the locations and sizes of links. For that purpose, a hybrid version of the PSO algorithm will be applied to the real network problem introduced by Rothlauf [5] and its efficiency will be evaluated against GAs. In addition a bi –criteria communication network topology problem is considered to address Quality of Service issues in the design process. For the corresponding delay function, a Poisson traffic model is utilized [1][3][7]. This real world application is addressed using multi-objective PSO. The Pareto front obtained by the MOPSO application is compared to the results obtained by a multi-objective GA (NSGA-II [6]). Relevant work on the subject has been presented in [1][7] among others using EAs. An alternative approach is also proposed in [8] where the relevant Delay Constrained Least Cost Path problem is addressed, utilizing the principle of Lagrangian relaxation based aggregated cost, where a PSO and noising metaheuristic are used for minimizing the modified cost function.

Of crucial importance to the success of the optimization procedure is the choice of candidate solutions representation. Especially for evolutionary algorithms a variety of encodings have been proposed as characteristic vectors, predecessors, Prufer numbers, link and node biasing, edge sets etc. In [8] [9] a tree based encoding/decoding scheme, based on heuristics has been devised for representing the paths as particles. In the presented work a tree is encoded with the network random keys (NetKeys) scheme introduced in [10].

In [11] Random keys were adapted to represent spanning trees. In this coding, a tree is represented by a string of real-valued weights, one for each edge of the complete graph. Therefore the size of the encoded string for a graph representing N nodes is $N*(N-1)/2$. In order to decode the string, the edges are sorted by their weights and Kruskal's minimum spanning tree algorithm considers the edges in sorted order. The mapping is one to one since any string of weights is a valid tree. Using NetKeys, that represents a tree with normalized real values, allows the optimization algorithms to avoid making binary and hard decisions on whether to establish a link or not.

In Section II a short overview of the PSO and multi-objective PSO algorithm is provided, alterations to the basic PSO algorithm for the single and multi objective cases are introduced and general notes on GA-PSO comparison are made. In Section III the network design problems are formulated (as single and multi-objective problems) and the corresponding fitness functions are defined. In Section IV, the obtained results of the proposed methodology are critically evaluated against the evolutionary algorithms' performance in terms of design goals satisfaction and convergence behavior, while in Section V the most important conclusions and future work are briefly discussed.

I. Particle Swarm Optimization

A. Hybrid Simple PSO

Particle Swarm Optimization (PSO) is a population based algorithm that exploits a set of potential solutions to the optimization problem. Each potential solution is called a particle and their aggregation, in each iteration step, forms the swarm. Swarm particles fly through the multi-dimensional problem space subject to both deterministic and stochastic update rules to new positions, which are subsequently scored by a fitness function. Each particle knows the best position $\vec{p}_l$ that it has ever found, called the local best and is also aware of the best position $\vec{p}_g$ found by any neighbor, called the global best.

$$\begin{aligned} \upsilon_{k+1} &= \upsilon_k + c_1\rho_1\left(p_l - x_k\right) + c_2\rho_2\left(p_g - x_k\right) \\ \upsilon_{k+1} &= sign\left(\upsilon_{k+1}\right)\ \min\left\{\left|\upsilon_{k+1}\right|, \upsilon_{\max}\right\} \\ x_{k+1} &= x_k + \upsilon_{k+1} \end{aligned} \quad (1)$$

Consequently the individual particles are drawn stochastically toward the positions of their own previous best performance and the best previous performance of their neighbors, in accordance to velocity update equations (1) that should be applied to each dimensional component of the velocity vectors, where k denotes the generation number, $\vec{x}_k$ and $\vec{\upsilon}_k$ represent the particle's position and velocity, and ρ_1, ρ_2 are uniformly distributed random numbers between 0 and 1. The parameter c_1, associates the particle's own experience with its current position and is called individuality. The parameter c_2 is associated with social interaction between the particles of the neighborhood and is called sociality. The velocity clamping parameter υ_{max} controls the algorithm's step size and is applied to all dimensional components of the particle's velocity, thereby limiting the chances that particles will leave the boundaries of the search space. A more refined way of constraining particles is by using "hard boundary conditions" [12], such as the reflective or absorbing boundary conditions. These do not enforce velocity clipping but rather inverse (RBC) or nullify (ABC) the velocity vector.

When the whole swarm is considered as a neighborhood, the global variant of the PSO is employed. Local neighborhoods may also be used to avoid entrapment of the algorithm on local minima but at the cost of slower convergence. For the network design problem specified in the following section, the Inertia Weight variance of PSO was used [13]. A global as well as a grid neighborhood was defined for improved convergence results and the RBC was enforced along with small υ_{max} values.

A major problem with PSO is premature convergence, which results in great performance loss and sub-optimal solutions. In order to avoid premature convergence, diversity guided PSO was utilized (Attractive and Repulsive PSO [14]) whereas mutation was also applied to the particles—hence the term hybrid simple PSO. ARPSO evaluates the global diversity of the swarm, triggering modes of global attraction or repulsion when predefined thresholds are crossed. Mutation has been demonstrated to successfully complement PSO and improve its performance [15][16][17]. In this study uniform mutation is applied to each particle prior to fitness evaluation with a probability of p_m, alternating a predefined percentage of the particle. The application of the mutation operator on the swarm improved significantly the convergence behavior of the algorithm.

B. Comparison to Genetic Algorithms

The PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms. Both techniques begin with a group of a randomly generated population; utilize a fitness value to evaluate the population and search for the optimum in a partially stochastic manner by updating generations. There are however important differences. Although PSO algorithm retains the conceptual simplicity of the GAs, its' evolutionary process does not create new population members from parent ones; all swarm individuals survive. The original PSO does not employ genetic operators such as crossover and mutation and particles only evolve through their social behavior. Particles' velocities are adjusted, while evolutionary individuals' positions are acted upon. In GAs chromosomes share information with each other, thus the whole population moves like one group towards an optimal area whereas in PSO the information exchange information is a one-way process since only the (local) global best provides information to members of the (sub) swarm. PSO in comparison with GAs, has less complicated operations and it is much easier to implement and apply to design problems with continuous parameters. Moreover in [18] it is shown that binary PSO outperforms GAs in terms of convergence, results and scalability of the problems at hand. In the same study it is suggested that a hybrid model combining characteristics of GA and PSO is preferable. In [19] PSO is shown to exhibit faster convergence compared to GAs. In this paper, the performance of the two algorithms will be compared, for network design problems, since GAs is the EC technique that has been mostly applied to such problems [14].

C. Multi Objective PSO

The main approaches to multi objective optimization consist of weighted aggregation multi objective schemes and population Pareto-based techniques. Pareto-based techniques maintain the set of Pareto-optimal solutions. Pareto-optimal solutions are non dominated solutions in the sense that there are no other superior solutions given the particular search space and set of objectives. Using mathematical notation, a multi-objective optimization problem given a set of n objectives and m decision parameters is denoted as:

$$\text{Optimize } F(\vec{x})=(f(x), f(x), \ldots, f(x)) \quad (2)$$

Where $\vec{x}\in S$ is a vector satisfying a set of inequality constraints $g_i(\vec{x})\leq 0,\ i=0,\ldots,k$. A vector $\vec{u}=(u_{1,}u_{2,}\ldots u_m)$ is said to dominate $\vec{v}=(v_{1,}v_{2,}\ldots v_m)$ if the solution $\vec{u}$ is no worse than $\vec{v}$ in all objectives $(\forall i\in\{1,\ldots,m\},\ F(u_i)\leq F(v_i))$ and the solution vector $\vec{u}$ is better than $\vec{v}$ in at least one objective $(\exists i\in\{1,\ldots,m\},\ F(u_i)<F(v_i))$. A solution $\vec{x}\in\Omega$ is said to be Pareto-optimal with respect to Ω, if there is no $\vec{z}\in\Omega$ for which $F(\vec{z})$ dominates $F(\vec{x})$.

During the past decade, several Multi-Objective Particle Swarm Optimization (MOPSO) methods have been proposed. However there are inherent drawbacks of the PSO algorithm concerning its application on multi-objective optimization. The tendency of the particles to converge to the (single) best solution in the global variant of the PSO is inappropriate for multi-objective optimization, whereas the local variant provides just a refinement near the local optima [20]. Weighted aggregation multi objective schemes were described in [21][22] and Pareto-ranking techniques [23][24][25]. In this study the MOPSO proposed in [25] is used. The particular technique is inspired from MOEA; therefore an external fixed repository is used in which every particle deposits its flight experiences after each flight cycle. The updates to the repository are performed considering a geographically-based system defined in terms of the objective function values of each particle. The search space is divided in hyper-cubes that are appointed a fitness value based on the containing number of particles (fitness sharing). Roulette-wheel is applied to select the hypercube from which a leader for a particle of the swarm will be selected randomly. Mutation as in (II-A) is also applied to the particles with possibility p_m.

II. Real World Telecommunication Tree Network Design Problem

Optimum Communication Spanning Tree Problem (OCSTP) is a special case of the Network Design Problem. Given a connected graph $G=(V,E)$, V represents the node set and E the arc set (links). There are communication demands among the nodes V. The traffic demands are specified by a demand matrix $R=(r_{ij})$ where r_{ij} is the volume of traffic demand among nodes i, j for all $(i,j)\in E$. A distance matrix $D=(d_{ij})$ represents the distance among sites for all $(i,j)\in E$. The problem can be stated as: given a set of node locations $G=(V,E)$ specific traffic demand and distance among every set of nodes i and j , the goal is to construct a spanning tree such that the total cost of communication is minimum among all the spanning trees of G. In the classical OCST problem as proposed by Hu [29] the cost of the tree is calculated as the product of the distance of the edge times the overall traffic over the edge. However in real world network design problems, capacity of links is assigned in discrete increments and this is the case that will be addressed in the particular study.

Simple generational genetic algorithms have been proved to provide adequate solutions for the particular problem [10] . However PSO is a simple and robust to control parameters algorithm, whereas the computational efficiency of the technique in comparison the GAs, renders it attractive for application in large scale network design optimization problems. In [26] the authors prove that PSO outperforms GAs in terms of computational efficiency, although the quality of the solutions found is similar in unconstrained nonlinear problems with continuous design variables. Therefore we utilize NetKeys encoding the importance of the links in a continuous manner (see section I).

In order to evaluate the performance of the PSO on this special case of the network design problem, benchmarking of the algorithm against an instance of the OCST has been performed. Several instances of the OCSTP problem have been presented in literature. In most of them [26][7] the communication cost was given by (3) for a variety of node sets, traffic and distance matrices.

$$min \sum_{i,j\in V} r_{ij}\times d_{p_{(i,j)}} \quad (3)$$

Rothlauf [5] introduced a set of real world telecommunication tree network design problems from a company with nodes located around Germany with modular link capacities. The available capacities for the links are discrete, determined by a fixed cost – for link installation - and a variable cost. The cost of the link per capacity is piecewise linear and monotonically increasing with the length of the link, with decreasing slope (Figure 1).

One instance proposed will be utilised in the particular study. It involves the creation of a communication network including one headquarter and 15 branch offices [10]. It deals with the cost optimization of a rooted spanning tree, where the root of the initial connected graph G is the "headquarter" realizing demands towards branch offices. This instance is being studied since it resembles tree/star hierarchical access networks [3]. The demand matrix and the location of the nodes as well as the modular cost function are provided in [28].

The formulation of the single objective problem is given by (4) where F denotes the set of used links, d_{ij} the distance weights of the links between nodes i and j, $Cap_{i,j}$ the capacity of the links and b_{ij} the traffic flowing directly and indirectly over the link between nodes i and j. The second equation in (4) denotes that the capacity of the link connecting nodes i and j must exceed the overall traffic flowing over that link. The form of one available capacity type function is presented in Figure 1.

$$\min \sum_{i,j \in F} f\left(d_{i,j}, Cap_{i,j}\right) \quad b_{i,j} < Cap_{i,j} \tag{4}$$

Very often topological design of WANs involves determining the links between nodes given the mean or peak inter node traffic so as to optimize certain QoS parameters [3]. In the multi-objective optimization case presented, the total network cost and average link delay is minimized simultaneously to obtain a Pareto front including optimal non-dominated solutions. A typical *M/M/1* queuing delay model is assumed [1][7] for one type of network service. Therefore two objective functions are used; the cost function delineated by the previous paragraph and the following delay fitness function.

$$\min AvgDelay$$

$$AvgDelay = \frac{\sum_i \sum_j \frac{LinkFlow_{i,j}}{Cap_{i,j}} - LinkFlow_{i,j}}{\sum_i \sum_j LinkFlow_{i,j}} \tag{5}$$

III. Results

A global and a local version of the hybrid PSO were applied to the Single Objective (SO) problem regarding optimization of network topology deployment. The presented results were obtained after repeating the optimization process for 100 times in order to obtain more statistically important indicators on the algorithm's convergence behavior over several runs.

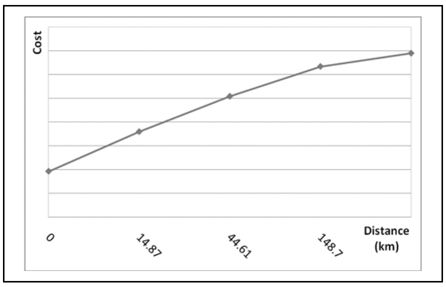

Fig. 1 Link Cost per Capacity

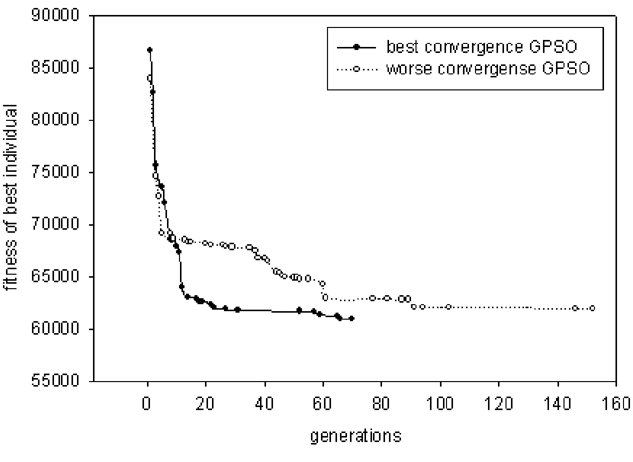

Fig. 2 Convergence Behaviour of Global PSO

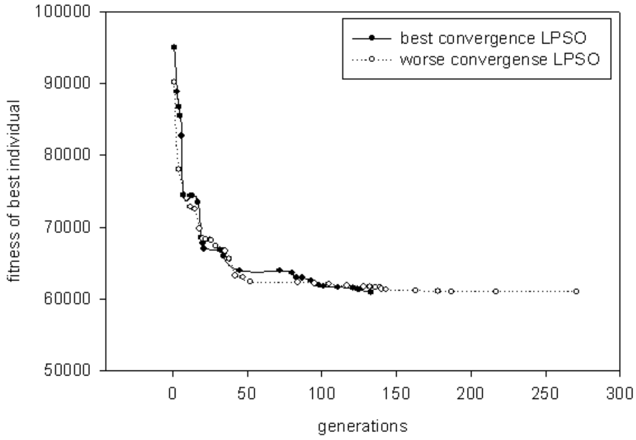

Fig. 3 Convergence Behaviour of Local PSO

The inertia weight PSO was set to sociality c_1=0.9, individuality c_2=0.9 and a velocity clamping of υ_{max}= 0.11. The inertia weight parameter was gradually reduced from w_1=0.7 to w_2=0.2 during the optimization run. Repulsion threshold was set to 0.1 whereas attraction threshold to 0.2. Mutation was enforced with a probability of 0.005 alternating 10% of the particle (12 edges of the complete graph represented by a particle of 120 edges for 16 nodes). The PSO parameters were found to produce optimal results for the given problem formulation after several trial runs.

In Figure 2 the convergence behavior of the PSO with a global neighborhood is presented while solving the SO network design problem. Indicators included in Table I represent obtained mean fitness value, standard deviation of fitness values and normalized deviation of the mean value from the best solution which is depicted in Figure 4. The best run was able to locate the global best configuration of the network within 70 generations, requiring a total of 17920 fitness function evaluations (for a swarm population of 256 particles). In terms of convergence the worse run was able to obtain a fitness value of 61934.27 on the 152th generation and afterwards stagnated on this local minimum position. Overly, when the algorithm was allowed to evolve for a maximum of 390 generations, it failed to obtain the global minimum 46 times thus yielding an effective success rate of 54%. Although the mean fitness value obtained for the 100 runs (61051.91) is numerically close to the absolute minimum value of 60884.2 for the presented problem, the above results indicate that the use of a global neighborhood can often lead to premature convergence on non-globally optimal solutions.

Respectively, the convergence behavior of the PSO when a rectangular grid neighborhood topology (lateral size = 16) is employed while solving the same network design problem, is presented in Figure 3. The convergence indicators are presented in table Table I (100 runs). The best run was able to locate the global optimum of the problem within 133 generations, requiring a total of 35245 fitness function evaluations (for a swarm population of 265 particles). The worse run was able to obtain a fitness value of 61005.19 on the 271th generation and afterwards stagnated on this local minimum position. When the algorithm was allowed to evolve for a maximum of 390 generations, it failed to obtain the global minimum 11 times thus yielding a success rate of 89%. The obtained mean fitness value (60891.59) improves on the mean value obtained from the PSO with a global neighborhood, indicating that the use of a grid neighborhood improves the convergence behavior of the optimization algorithm for the selected problem. Another key aspect highlighting the above conclusion is that the mean fitness value of the non-successful runs is lower compared to the respective values obtained with a global PSO. Nevertheless, the improved convergence behavior comes at the expense of slower convergence (mean number of generations required to obtain best solution).

The proposed methodology shows an improvement in most cases over the results obtained and presented in [7]. Both PSO variations are able to outperform the performance of the GA with a tournament selection scheme with the exception of the obtained fitness deviation indicator value when the global-neighborhood PSO is used (Table I). The latter can be attributed to the already highlighted worst convergence behavior of the global PSO variant. The indicator values obtained when a grid neighborhood is used shows a ten-fold improvement over the respective GA result for the normalized deviation indicator (0.121% vs. 0.77%).

On the other hand, when a GA is employed using (μ+λ) selection, the obtained mean value is better than the respective value of the grid neighborhood PSO (60886.62 vs. 60891.59), although the PSO optimums have a smaller deviation (40 vs. 23.35). The deterministic nature of the (μ+λ) selection helps to preserve the encoding of the so far best solution found in the population but no analogous mechanism exists in the PSO algorithm. The slightly better performance of the GA over the local PSO in terms of p_w (0.0039 vs. 0.0121%) can be attributed to this effect.

Concerning the multi-criteria instance of the network design problem, the only change applied in PSO parameters was velocity clamping of υ_{max}= 0.7. Regarding the multi-objective GA, the algorithm was specified with p_c=0.8 and p_m=0.005 as crossover and mutation probabilities respectively whereas binary tournament selection, uniform crossover and random mutation on each component of the particle with $p_{m.}$ The results were carried with the same instance of the 16 node network design problem for a population of 265 particles and 400 generations.

The target is to compare the Pareto fronts that are obtained with the same number of fitness evaluations, with a relatively small number of iterations and population size. The fronts from the two multi-objective algorithms are depicted in Figure 5. We notice that the MOPSO algorithm with mutation, although it acquires a smaller number of non dominated solutions these are more evenly distributed in comparison to NSGA-II. Moreover a wider range of non dominated solutions were obtained in order to approximate the front with more accuracy. With the NSGA-II algorithm a

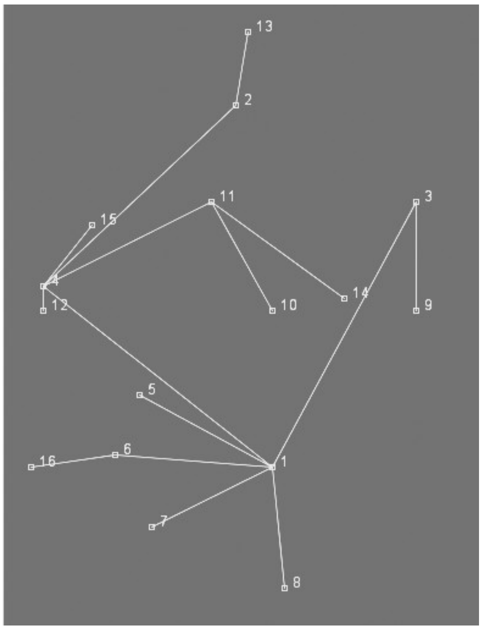

Fig. 4 Optimum Network Topology

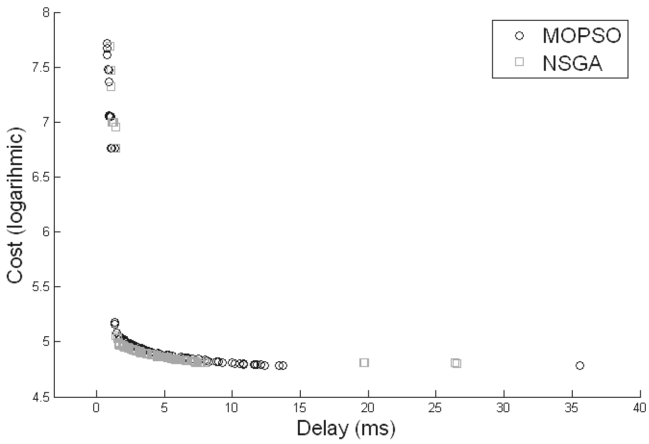

Fig. 5 Pareto Front Comparison among NSGA-II and MOPSO

TABLE I.
COST OF BEST SOLUTION

Cost of best solution		Global PSO [Population=256] [Runs=390]	Local PSO [Population=256] [Runs=390]	GA-Tournament [Population=2000] [Runs=30]	GA-(μ+λ)selection [Population=2000] [Runs=50]
6883.71	μ	61050.91	60890.58	61353.33	60886.62
	σ	267.77	23.35	431	40
	p_w	0.27%	0.01%	0.77%	0.00%

region of the front is acquired. In this region it slightly outperforms the MOPSO in terms of the quality of non dominated solutions.

From the network designer's point of view the solution fronts can be divided into three different areas IV, a Low Cost High Delay (LCHD) region, a Medium Cost Medium Delay (MCMD) region and High Cost Low Delay (HCLD) region. In this case in the LCHD area the cost of infrastructure deployment was chosen arbitrarily to be lower than 65.000 whereas in the HCLD area over 150.000 For the LCHD region the MOPSO algorithm with mutation as well as the NSGA-II obtain similar number of solutions though again the range of the MOPSO solutions is greater. Non dominated solutions in this area are easier to obtain than the HCLD region. However it must be noted that concerning the HLDC region the multi-objective PSO outperforms the NSGA-II both in terms of quality and range. Finally the NSGA-II acquires a slightly better approximation of the Pareto front in the MCMD region.

IV. Conclusions—Future Work

In the particular study, a hybrid version of a single and multi objective PSO algorithm was applied successfully on a communications network topology design problem. The multi-criteria instance of the problem addresses also QoS issues. Optimization objectives in this case are the network layout cost and the average packet delivery delay. NetKeys representation of candidate solutions is utilized. This hybrid version of the particle swarm algorithm applies mutation to the swarm with a probability of p_m whereas additional mechanisms (inertia weight, attraction/repulsion) are utilized to improve convergence behaviour. The proposed methodology shows an improvement in the optimization process in comparison to GAs, concerning both the single and multi objective instance. Future work will include the application of PSO metaheuristic on the topological design of multiservice networks for realistic self similar traffic models.

References

[1] R. Kumar, N. Banerjee, "Multicriteria network design using evolutionary algorithm", *in 2003 GECCO* , pp. 2179-2190.

[2] M. Pioro, D. Medhi, "Routing, flow and capacity design in communication and computer networks", Morgan-Kaufmann, 2004.

[3] M. Resende, P. Pardalos, "Handbook of Optimization in Telecommunications", Springer, 2006.

[4] P. Kampstra, "Evolutionary Computing in telecommunications, A likely EC success story", Thesis, VU university Amsterdam, The Netherlamds,2005.

[5] F. Rothlauf,, "Towards a theory of representations for genetic and evolutionary algorithms: Development of basic concepts and their application to binary and tree representations", Ph.D. dissertation, Department of Information systems, University of Bayreth, Germany, 2001.

[6] Deb K, Pratap A, Agarwal S, Meyarivan T., "A fast and elitist multiobjective genetic algorithm: NSGA-II.", *in Evolutionary Computation, IEEE Transactions on* , vol.6, no.2, pp.182-197, Apr 2002.

[7] N. Banerjee, R. Kumar, "Multiobjective network design for realistic traffic models.", In *Proceedings of GECCO '07* , pp. 1904-1911, 2007.

[8] A.W. Mohemmed, Z. Mengjie, N.C. Sahoo, "Cooperative Particle Swarm Optimization for the Delay Constrained Least Cost Path Problem", *in Proceedings of EvoCOP 2008,* pp. 25-35, 2008.

[9] A.W. Mohemmed, N.C. Sahoo, T.K. Geok, "A new particle swarm optimization based algorithm for solving shortest-paths tree problems.", *in IEEE CEC 2007*, pp. 3221–3225, 2007.

[10] F. Rothlauf, D. Goldberg, A. Heinzl, "Network random keys—a tree network representation scheme for genetic and evolutionary algorithms", *in Evol. Comput,* vol. 10, no1, pp. 75-97, Mar. 2002.

[11] J. C. Bean, "Genetic algorithms and random keys for sequencing and optimization," *in ORSA Journal on Computing*, vol. 6, no. 2, pp. 154–160, 1994.

[12] S. Mikki, A. Kishk, "Improved particle swarm optimization technique using hard boundary conditions", *in IEEE Microwave and Technology Optical Letters*, vol. 46, no. 5, pp. 422-426, Sep. 2005

[13] R. C. Eberhart, Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization", *in Evolutionary Computation, 2000. Proceedings of the 2000 Congress on* , pp.84-88, 2000

[14] J. Riget, J.S. Vesterstroem, "A diversity-guided particle swarm optimizer—the ARPSO", EVALife Technical Report no. 2002-02.

[15] N. Higashi, H. Iba, "Particle swarm optimization with Gaussian mutation", *in. SIS '03. Proceedings of the 2003 IEEE*, pp. 72-79, April 2003.

[16] C. F. Juang, "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design", *in IEEE Transactions on Systems Man and Cybernetics Part B,* vol.34, no.2, pp. 997-1006, April 2004.

[17] Q. Zhang, X. Li, Q. Trana, "Modified Particle Swarm Optimization Algorithm", *in Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, vol.5, pp. 2993-2995, Aug. 2005.

[18] R. C. Eberhart, Y. Shi, "Comparison between Genetic Algorithms and Particle Swarm Optimization", *in Evolutionary Programming VII: Proceedings of the Seventh Annual Conference on Evolutionary Programming*, pp. 611–616, 1998.

[19] R. Hassan, B. K. Cohanim, O. D. Weck, G. A. Venter, "A Comparison of particle swarm optimization and the genetic algorithm". *in Proceedings of the 1st AIAA Multidisciplinary Design Optimization Specialist Conference*, April 2005.

[20] X. Hu, R. Eberhart, "Multiobjective Optimization Using Dynamic Neighborhood Particle Swarm Optimization", *in Proceedings of the 2002 Congress on Evolutionary Computation*, *IEEE World Congress on Computational Intelligence*, pp. 1677-1681, May 2002.

[21] K. E. Parsopoulos, M. N. Vrahatis. "Particle Swarm Optimization Method in Multiobjective Problems", *in Proceedings of the 2002 ACM Symposium on Applied Computing (SAC 2002)*, pp. 603-607, 2002.

[22] U. Baumgartner, Ch. Magele, W. Renhart. "Pareto optimality and particle swarm optimization", *in Magnetics, IEEE Transactions on* , vol. 40, no.2, pp. 1172-1175, March 2004.

[23] J. E Fieldsend, R. M. Everson, S. Singh, "Using unconstrained elite archives for multiobjective optimization", *Evolutionary Computation, IEEE Transactions on* , vol.7, no.3, pp. 305-323, June 2003.

[24] T. Ray, K. M. Liew. "A swarm metaphor for multiobjective design optimization", *in Engineering Optimization*, vol. 34, no. 2, pp. 141–153, March 2002.

[25] C. A. Coello, M. S. Lechunga, "MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization", *in Proceedings of the 2002 Congress on Evolutionary Computation*, *IEEE World Congress on Computational Intelligence,* pp. 1051-1056, May 2002.

[26] C. Palmer, "An Approach to a Problem in Network Design Using Genetic Algorithms" Ph.D. dissertation, Polytechnic University, Brooklyn, New York.

[27] G. R. Raidl, "Various instances of optimal communication spanning tree problems," personal communication, February 2001.

[28] Franz Rothlauf, "Representations for genetic and evolutionary algorithms", Springer-Verlang, 2006.

[29] T. C. Hu, "Optimum communication spanning trees", *in SIAM J. on Computing* , p.p. 188-195, 1974.