

PARALLEL GUIDED LOCAL SEARCH AND SOME PRELIMINARY EXPERIMENTAL RESULTS FOR CONTINUOUS OPTIMIZATION

Nasser Tairan¹, Muhammad Asif Jan² and Rashida Adeeb Khanum³

¹ College of Computer Science, King Khalid University, Abha, KSA
nmtairan@kku.edu.sa

² Department of Mathematics, Kohat University of Science & Technology,
Kohat 26000, KPK, Pakistan
majan@kust.edu.pk

³ Jinnah College for Women, University of Peshawar, Peshawar, KPK, Pakistan
adeeb_maths@yahoo.com

ABSTRACT

This paper proposes a Parallel Guided Local Search (PGLS) framework for continuous optimization. In PGLS, several guided local search (GLS) procedures (agents) are run for solving the optimization problem. The agents exchange information for speeding up the search. For example, the information exchanged could be knowledge about the landscape obtained by the agents. The proposed algorithm is applied to continuous optimization problems. The preliminary experimental results show that the algorithm is very promising.

KEYWORDS

Guided Local Search, Continuous Optimization Problems, Parallel Algorithms, Cooperative algorithms

1. INTRODUCTION

Many real-world applications can be modeled as optimization problems [1][2]. These problems may have many local optima and/or have no analytical form available. Traditional mathematical programming methods are not suitable for dealing with them. During the last two decades, many heuristic methods have been developed. Many of them such as simulated annealing [3], tabu search[4], and guided local search [5] are single point iteration based methods. The research work in the Evolutionary Computation and Particle Swarm Intelligence has demonstrated that a population-based search scheme often has some advantages over single-point based methods on efficiently utilizing computer memory and parallel computing process units. Therefore, a natural issue arises: how can a single point based heuristic be populationized? Mistunori and Ogura [6], E-Abd and Kamel[7]and Blum and Roli [8]have made some attempts on this issue.

GLS, proposed by Voudouris and Tsang in 1995 [5], is a single point based meta-heuristic approach. It escapes from local optima by penalizing some bad solution features. GLS has been applied successfully to a number of optimization problems [5]. This paper proposes a population-based guided local search (PGLS). In PGLS, several agents run the GLS and exchange

information, learned from the previous search about the problem, with each other periodically for speeding up the search. Some preliminary experiments have been carried out to study the effectiveness of the proposed PGLS up the search.

The rest of the paper is organized as follows: the proposed PGLS is described in Section 2. Sections 3 and 4 list the test instances and report experimental results. Section 5 concludes this paper.

2. PARALLEL GUIDED LOCAL SEARCH

2.1. Guided Local Search

Guided Local Search (GLS) is a very general strategy for guiding a local search algorithm to escape from local optima. In GLS, solution features are defined to distinguish between solutions with different characteristics, so that bad characteristics can be penalized and hopefully removed. The solution features depend on the problem and the local search. Features can be defined in a very natural way. For example, in the travelling salesman problem (TSP), a feature can be a link between two cities. When the search is trapped in a local optimum, GLS will use the following function as the objective function:

$$h(s) = g(s) + \lambda \sum (p_i \times I_i(s)), \quad (1)$$

where s is a candidate solution and $g(s)$ is the original objective to minimize, λ is a control parameter, I ranges over the features. p_i is the penalty for feature i . $I_i(s)$ indicates whether or not solution s exhibits feature i :

$$I_i(s) = \begin{cases} 1 & \text{if } s \text{ exhibits feature } i; \\ 0 & \text{otherwise.} \end{cases}$$

All the p_i are initialized to be 0. When the local search traps at a local optimum s^* , GLS computes the utility function of penalizing feature i :

$$\text{util}_i(s^*) = I_i(s^*) \times \frac{c_i}{1+p_i} \quad (2)$$

Where c_i is the cost for the feature i (in TSP, the cost for a feature is the distance for that feature), p_i is the current penalty value for feature i . GLS penalizes the feature with the highest utility value, i.e., increase its p_i value by 1. The LS will then continue with the new augmented objective function.

2.2. PGLS

In PGLS, several agents run GLS in a parallel way and exchange information periodically for speeding up the search. Different agents start their searches from different points in the solution space. PGLS (manager) records the best solution x^* found so far to the problem under consideration by these agents. After every K penalizations, each agent conducts a crossover operator on x^* and its current solution to generate a new solution and then performs its search

from this new solution. The p_i values for each agent will be reset to zero. The algorithm with n agents works as follows:

Step1: Initialization

Generate randomly n points x_1, \dots, x_n in the search space.

Step2: Guided Local Search

2.1 For $i=1$ to n

Agent i does GLS from x_i until K penalizations have been made and returns its best solution z_i found so far and its current solution y_i .

2.2 Set z to be the best solution to the problem among all the z_i 's.

Step 3

3.1 If the stopping condition is met, return z .

3.2 Otherwise,

3.2.1 For $i=1$ to n , do crossover on z and y_i to produce a new solution

x_i .

3.2.2 Go to Step 2.

2.2.1. Local Search

Following Voudris [9], each real-valued variable in a continuous optimization problem is represented by a number of bits. Therefore, the flip bit mutation is used as a local search method. The local search moves from one solution to another by flipping the value of a bit in the solution. It starts from a random bit and then examines all the possible bits.

2.2.2. Crossover

In the crossover, $2n$ crossover sites are first randomly selected under the constraint that there are two crossover sites in each variable substring, and then 2 child solutions are generated by swapping along these sites. The child solution with lower original objective function value is returned as a new solution (Step 3.2.1).

3. EXPERIMENTAL STUDIES

We have tested the above PGLS algorithm to minimize Rastriging function, Rosenbrock function, and Schwefel function. These three functions are:

$$f_{Ra} = 10n + \sum_{j=1}^n (x_j^2 - 10 \cos(2\pi x_j)) \quad (3)$$

$$f_{\min} = 0, [x_j = 0], \quad -5.12 \leq x_j \leq 5.12$$

$$f_{Ro} = \sum_{j=1}^{n-1} [100(x_{j+1} - x_j^2)^2 + (x_j - 1)^2] \quad (4)$$

$$f_{\min} = 0, [x_j = 1], -2.048 \leq x_j \leq 2.048$$

$$f_{Sh} = 418.9829n - \sum_{j=1}^n (x_j \sin \sqrt{|x_j|}) \quad (5)$$

$$f_{\min} = 0, [x_j = 1], -500 \leq x_j \leq 500$$

In our experiments, each real-valued variable is represented in 22 bits as in [9]. Therefore, each solution is represented in 22n bits when n is the number of variables.

The GLS used in our experiments is the same as in [9]. The number of agents=2. The value of K varies based on the problem as it is shown in Table 1. The algorithm stops after each agent has done $100 \times K$ penalizations.

3. RESULTS

The obtained results show that the proposed cooperative method, PGLS, was the most effective algorithm of the two algorithms. It showed a good performance and produced good results for the lunched functions in every run. Table 1 shows the experimental results for the run algorithms

Table 1. Comparison between the run algorithms

| Test Function | Number of Variables | Known Optimum | Algorithm 1 No cooperation | | Algorithm 2 (PGLS) | | Number of Penalization per generation |
|---------------|---------------------|---------------|-------------------------------|----------|-----------------------|----------|---------------------------------------|
| | | | Average | STDEV | Average | STDEV | |
| f_{Ra} | 2 | 0 | 3.24E-01 | 7.06E-01 | 5.91E-10 | 0.0 | 300 |
| f_{Ra} | 5 | 0 | 4.23E-01 | 5.50E-01 | 1.47E-09 | 0.0 | 800 |
| f_{Ra} | 10 | 0 | 7.93E-01 | 8.06E-01 | 9.95E-02 | 3.15E-01 | 1500 |
| f_{Ro} | 2 | 0 | 1.34E-02 | 3.94E-02 | 1.34E-02 | 3.94E-02 | 500 |
| f_{Ro} | 5 | 0 | 0.2634 | 0.4174 | 0.0002 | 0.0002 | 800 |
| f_{Ro} | 10 | 0 | 0.6281 | 0.7416 | 0.0906 | 0.0498 | 1000 |
| f_{Sh} | 2 | 0 | 6.31E-04 | 4.04E-04 | 4.79E-04 | 2.67E-04 | 100 |
| f_{Sh} | 5 | 0 | 1.00E-02 | 3.88E-04 | 6.60E-04 | 5.32E-04 | 500 |
| f_{Sh} | 10 | 0 | 2.38E-02 | 2.30E-02 | 2.26E-02 | 2.36E-02 | 1000 |

4. CONCLUSIONS

A framework called PGLS for solving continuous optimization problems has been proposed. The idea is to build a cooperative learning environment through running a number of agents of GLS to efficiency explore the search space. After predetermined iterations the acquired information through the search is exchanged between the agents to speed up the search. The genetic crossover was exploited as an exchanging information operator between the agents. In order to allow different components (variables) from the continuous spaces to be selected we apply the crossover operation in each variable substring. Our experimental results on test continuous optimization problems show that our method is the most effective and performed better than the other mean of parallel –GLS: no coordination. Our future work is to expand the experiments with more number of agents and variables. Furthermore, theoretical analysis for the proposed algorithm should be conducted in order to study its convergence and computation time and thus to perform further improvement to the algorithm and subsequently make comparisons with others algorithms.

REFERENCES

- [1] R. Horst, N.V. Thoai and P.M. Pardalos, Introduction to Global Optimization, Second Edition, Kluwer Academic Publishers, Boston, MA, 2000.
- [2] P.M. Pardalos and H. E. Romeijn (Eds.), Handbook of Global Optimization, Kluwer Academic Publishers, Boston, MA, 2002.
- [3] Kirkpatrick, S., C. D. Gelatt Jr., M. P. Vecchi, "Optimization by Simulated Annealing", Science, 220, 4598, 671-680, 1983.
- [4] Fred Glover, Fred Laguna, Tabu Search, Kluwer Academic Publishers, Norwell, MA, 1997.
- [5] Voudouris, C, Guided local search for combinatorial optimization problems, PhD Thesis, Department of Computer Science, University of Essex, Colchester, UK, July, 1997.
- [6] Hiroyasu, T., Miki, M. and Ogura M., "Parallel simulated annealing using genetic crossover", Proceedings of the IASTED International Conference on Parallel and Distributed Computing Systems, Las Vegas, pp. 145-150, 2000.
- [7] El-Abd, M. and Kamel, M., "A Taxonomy of Cooperative Search Algorithms". Hybrid Metaheuristics (Springer Berlin - Lectures in Computer science), pp. 32-41, 2005.
- [8] Blum, C. and Roli, A. "Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison". ACM Computing Surveys, vol. 35, no. 3, pp. 268- 308, 2003.
- [9] Voudouris, C., Guided Local Search -- An illustrative example in function optimisation, BT Technology Journal, Vol.16, No.3, July 1998, 46-50.

AUTHORS

Nasser Tairan received B.Sc. degree in computer science from King Abdulaziz University, KSA in 1999, M.Sc. degree in Software Engineering from University of Bradford, UK in 2005, M.Sc. degree in Computer Science (AI) from University of Essex, UK in 2007 and P.hD. degree from University of Essex 2012. He is currently Assistance Professor in King Khalid University, College of Computer Science, KSA. His main research areas are evolutionary computation, single and multiobjective optimization and metaheuristics,

Muhammad Asif Jan received the M.Sc. degree in mathematics from University of Peshawar, Khyber Paktunkhwa, Pakistan in 1997 and the Ph.D. in mathematics from University of Essex, UK, in 2011. He is currently the Chairperson of the Department of Mathematics, Kohat University of Science & Technology (KUST), Khyber Paktunkhwa, Pakistan. His main research areas are evolutionary computation, unconstrained/constrained single/multi- objective optimization by using evolutionary algorithms, decomposition based evolutionary methods for constrained multi-objective optimization, mathematical programming, and numerical analysis.



Rashida Adeb Khanum received the M.Sc. degree from University of Peshawar, Pakistan in 1997 and the Ph.D. from University of Essex, Colchester, United Kingdom in 2012.

