2005 International Symposium on Nonlinear
Theory and its Applications (NOLTA2005)
Bruges, Belgium, October 18-21, 2005

Improving Local Optima of Local Search with Adjustable Multipliers

Zongmei Zhang[†‡], Hiroki Tamura[‡], Zheng Tang[‡]and Jun Ma[†]

† Faculty of Computer Science, Shandong University, Jinan 250061 China
‡Faculty of Engineering, Toyama University, Toyama-shi 930-8555 Japan
Email: zongmeizhang@hotmail.com, tamura@iis.toyama-u.ac.jp

Abstract– In this paper, we give two improved algorithms of Guided Local Search (GLS) to improve the local optima of local search. In the GLS-like algorithm, a new penalty principle is proposed to further improve the effectiveness of GLS. The Objective function Adjustment (OA) algorithm is an improved algorithm of GLS-like using multipliers which can be adjusted during the search process. The simulation results based on some TSPLIB benchmark problems showed that the OA algorithm could find better solutions than the local search, guided local search, Tabu Search and GLS-like.

1. Introduction

Local search algorithms have been widely applied to many optimization problems for finding near-optimal solutions. For example, it can be applied to the traveling salesman problem (TSP) for finding near-optimal solutions, and 2-opt and Or-opt are two classical local optimization algorithms for the TSP. A problem of the local search algorithm difficult to avoid is the local optimum problem. Tabu Search (TS) [1] and Guided Local Search (GLS) [2], two meta-heuristic algorithms, showed that they could help local search to escape local optima.

In this paper we introduce two improved algorithms of GLS called GLS-like algorithm and Objective function Adjustment (OA) algorithm. In addition, the proposed algorithms are applied to the traveling salesman problem and the simulation results based on some TSPLIB benchmark problems showed that they could find better solutions than the local search and guided local search, which means that the local optima have been improved.

2. Technique of escaping local optima

2.1. Basic principles of local search

Local search works as follows: starts from a complete initial solution and tries to find a better solution among neighbors of the current solution. If such a solution is found, it replaces the current solution and the search continues. When no improving neighbor solution can be found anymore, the procedure ends and a local optimum is obtained.

2.2. Tabu Search

Tabu search starts from a local optimum and moves to the best neighbor of the local optimum, which helps local search escape the local optimum. In order to avoid a previous move being repeated, one or more Tabu lists are used to record recent moves. The Tabu lists are historical in nature and form the Tabu search memory.

2.3. Guided Local Search

Guided Local Search (GLS) [2, 3] is another method to help local search escape local optima. The basic principle of GLS is to penalize features of the candidate solutions when local search settles in a local optimum. Features for the set of the candidate solutions are defined. Given a set of features $M=\{1, \ldots, m\}$, a feature $i \in M$ in a candidate solution s is represented by an indicator function in the following way (Eq(1)):

$$I_i(s) = \begin{cases} 1 & \text{if solution } s \text{ has feature } i \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Through augmenting the objective function *g*(*s*) to include a set of penalty terms, constraints on features can be performed. The augmented objective function *h*(*s*) is defined in (Eq(2)):

$$h(s) = g(s) + \alpha \sum_{i=1}^{m} p_i \cdot I_i(s) \qquad (2)$$

where penalty p_i is an integer that counts how many times feature *i* has been penalized. α is a parameter to the GLS algorithm (constant throughout the search). p_i is initialized to 0 . During the search a utility measure $util_i(s)$ (Eq(3)) for solution *s* and each of its feature *i* is defined.

$$util_i(s) = I_i(s) \frac{c_i}{1 + p_i} \qquad (3)$$

where c_i is the cost associated to each feature $i \in M$. During the search, the features with the highest *util* value will be penalized. Its penalty value is incremented by 1 as(Eq(4)):

$$p_i := p_i + 1 \qquad (4)$$

When local search settles on a local optimum, the penalty of some of the features associated to the local optimum is increased. Therefore, the objective function is changed which driving the search towards other candidate solutions. The general GLS procedure is given in Fig.1.

```
Step1: initialize all penalty terms $p_i$ to 0;
       generate a candidate solution s randomly;
Step2: perform local search algorithm until a local optimum is reached;
Step3: calculate util values of all features in s according to Eq4;
       penalize the feature i with maximum util value: $p_i := p_i+1$;
Step4: go to step2 until a termination condition is reached;
Step5: return the best solution;
```

Fig.1. Program flow of GLS algorithm

3. Escaping local optima with multipliers

It is known that the Guided Local Search algorithm can find better solutions than local search via adding penalty terms to objective function. Here, we propose two improved methods of GLS, GLS-like and Objective function Adjustment (OA) algorithm, to further improve the effectiveness of GLS using multipliers that can be adjusted during the search process.

The conceptual graph of objective function value transition in search process of the proposed algorithms is as Fig.2 shows. The Energy function value is reflected in the height of the graph. Each position on the energy landscape corresponds to a possible state of the local

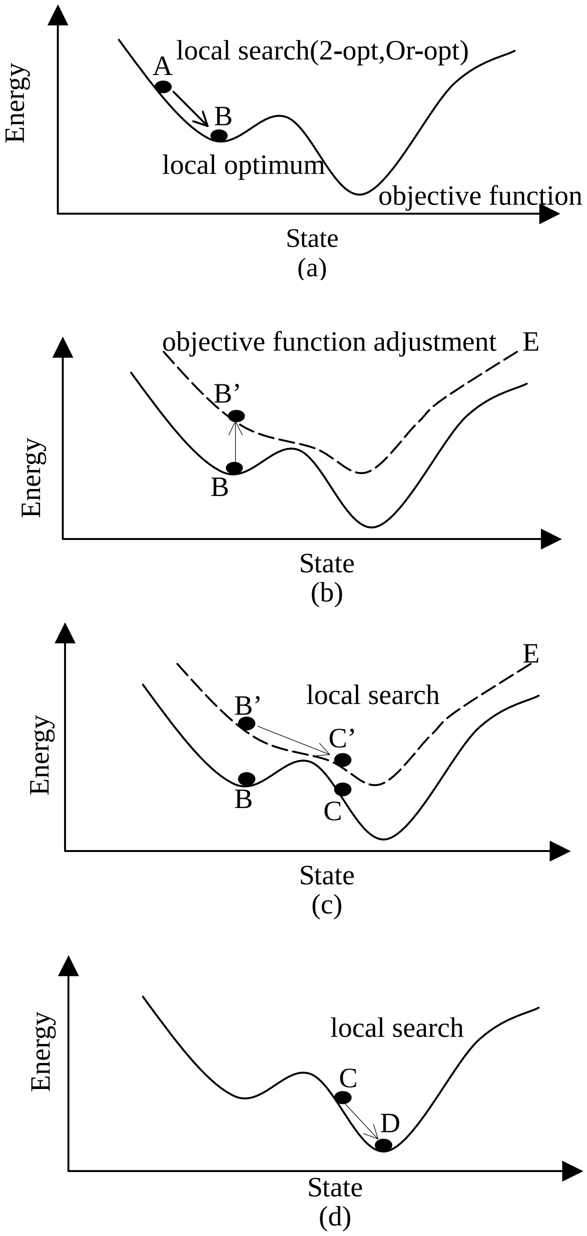

Fig.2. The conceptual graph of objective function value transition in search process of proposed algorithms

search. For example, if the solution of the local search is initialized onto point A (Fig.2(a)), because the updating procedure makes the state move towards negative gradient direction, the solution of the local search will first reach the local optimum B (Fig.2(a)). Then we adjust the multipliers in gradient ascent direction so as to increase the energy temporarily, and point B will become a new point B' of the new energy landscape (Fig.2(b)).

After updating energy function with the new multipliers again, point B' goes down the slope of the valley and reaches a new stable state C' (Fig.2(c)). Then, all multipliers are reset and the energy of C' will decrease and reach a new state C. The search process continues until a new optimum D is reached (Fig.2(d)). Details of the proposed methods and their application are given in the following section.

The proposed methods do not stop when a local optimum is obtained. Instead, many local optima are calculated via changing the energy function *E*. The proposed methods choose the best one of the local optima as a solution.

4. Application to traveling salesman problem

4.1. Traveling salesman problem

The Traveling Salesman Problem [4] is a well-known optimization problem. It can be stated as follows: given a set $C=\{c_1, c_2, \dots, c_n\}$ of *n* cities and a distance $d(c_i, c_j)$ for each pair $\{c_i, c_j\}$ of distinct cities, the goal is to find a tour (a closed path that starts from a city, and visits each city exactly once) with minimum tour length. That is, given an ordering π of the cities, try to find an ordering π^* that minimizes the distance function as follows (Eq(5)) :

$$f(\pi) = \sum_{i=1}^{n-1} d(c_{\pi(i)}, c_{\pi(i+1)}) + d(c_{\pi(n)}, c_{\pi(1)}) \qquad (5)$$

It is costly to find the optimal solution through global search and some times leads to combinatorial explosion. One way to solve this problem is to sacrifice completeness and use local search to find near-optimal solution.

4.2. GLS-like algorithm

When GLS-like algorithm is applied to the TSP, the objective function *E* is as Eq(6) shows which is identical with the original GLS algorithm.

$$E = f(\pi) + \lambda\left(\sum_{i=1}^{n-1} p(c_{\pi(i)}, c_{\pi(i+1)}) + p(c_{\pi(n)}, c_{\pi(1)})\right) \qquad (6)$$

where $p(c_{\pi(i)}, c_{\pi(i+1)})$ initialized to 0 is the penalty of distance $d(c_{\pi(i)}, c_{\pi(i+1)})$. λ is a parameter (constant throughout the search) and is assigned to 1 for all experiments.. What we improved is the penalty principle. Instead of penalizing the features with highest *util*, we consider the relation between the current distance and the average distance, and penalize features whose values are greater than the average value [5]. See Eq(7).

$$\text{if } \frac{d(c_{\pi(i)}, c_{\pi(i+1)})}{(1/n)\cdot f(\pi)} > 1 \text{ then}$$

$$p(c_{\pi(i)}, c_{\pi(i+1)}) := p(c_{\pi(i)}, c_{\pi(i+1)}) + \alpha\left(\frac{d(c_{\pi(i)}, c_{\pi(i+1)})}{(1/n)\cdot f(\pi)} - 1\right) \qquad (7)$$

where $d(c_{\pi(i)}, c_{\pi(i+1)})$ is the distance between two cities, and α is a parameter to the GLS-like algorithm and is assigned to 1.1 for all the experiments.

When local search fell in a local optimum (Fig.2(a)), we penalize $p(c_{\pi(i)}, c_{\pi(i+1)})$ according to Eq(7) (Fig.2(b)). All penalties $p(c_{\pi(i)}, c_{\pi(i+1)})$ are reset to 0 (Fig2(c),(d)) as soon as a better solution is found.

4.3. Objective function adjustment algorithm

The Objective function Adjustment (OA) algorithm, is a method that considers multiplying a multiplier $\lambda(c_{\pi(i)}, c_{\pi(i+1)})$ for every term of the objective function as follows (Eq(8)):

$$E = \sum_{i=1}^{n-1} \lambda(c_{\pi(i)}, c_{\pi(i+1)}) \cdot d(c_{\pi(i)}, c_{\pi(i+1)}) + \lambda(c_{\pi(n)}, c_{\pi(1)}) \cdot d(c_{\pi(n)}, c_{\pi(1)}) \qquad (8)$$

When the local search settled in a local optimum, the OA algorithm can help it out by changing the objective function using multipliers. We adopt the same principle as GLS-like to adjust $\lambda(c_{\pi(i)}, c_{\pi(i+1)})$, see Eq(9).

$$\lambda(c_{\pi(i)}, c_{\pi(i+1)}) := \lambda(c_{\pi(i)}, c_{\pi(i+1)}) + \alpha\left(\frac{d(c_{\pi(i)}, c_{\pi(i+1)})}{(1/n)\cdot f(\pi)}\right) \quad (9)$$

where α is a parameter and is assigned to 1.1 for all the experiments. Just as GLS-like algorithm, when local search fell in a local optimum (Fig.2(a)), we adjust multipliers $\lambda(c_{\pi(i)}, c_{\pi(i+1)})$ (Fig.2(b)). $\lambda(c_{\pi(i)}, c_{\pi(i+1)})$ is reset to 1 as soon as a better solution is found (Fig.2(c), (d)). This processing is very important for the objective function adjustment algorithm. . Due to this initialization processing, the proposed method is able to search the objective function exactly.

5. Simulation results

In order to test the effectiveness of the proposed methods, we compared their performances with Local Search (LS) and GLS and Tabu Search (TS) on several benchmark problems [6]. For all methods, 2-opt and Or-opt moves are performed as the local search mechanism. According to the experiments, 100 near-optimal solutions are found for each method. We compare the performances of the four methods through calculating the values that:

$$\frac{\text{average tour length - optimal length}}{\text{optimal length}} \times 100\% \quad \text{and}$$

$$\frac{\text{best tour length - optimal length}}{\text{optimal length}} \times 100\% \cdot$$

Table 1 gives the statistics based on att48 (48-city), att532 (532-city) and pr1002 (1002-city) problems. We can see that OA algorithm can find better solutions than other four methods. Running time of GLS, TS, GLS-like and OA are almost the same. However, OA algorithm has lower average error value than other methods. The best solution which has not been obtained by other methods can be obtained by OA algorithm.

Table 1. The results of the proposed methods

	LS	***GLS***	***TS***	***GLS-like***	***OA***
att48 **(best result)**	1.50 (0.00)	0.72 (0.00)	0.46 (0.00)	0.66 (0.00)	**0.38** (0.00)
att532 **(best result)**	4.69 (2.64)	3.95 (2.48)	4.40 (2.48)	**3.51** (2.02)	3.58 (**1.78**)
pr1002 **(best result)**	6.06 (3.99)	5.75 (3.94)	5.82 (3.64)	5.54 (3.52)	**5.13** (**2.50**)

6. Conclusion

In this paper, two improved methods of GLS are proposed to improve the local optima of local search. We have conducted many experiments to evaluate the effectiveness of the proposed methods. The simulation results based on some TSPLIB benchmark problems showed that the OA algorithm could find better solutions than LS, GLS, TS and GLS-like. A future work is to test OA algorithm by Lin-kernighan heuristic.

OA algorithm is applicable by adding and adjusting multipliers. Its calculation process is not difficult. Therefore, we hope the proposed method can provide a useful result to those who are engaging in solving the combinatorial optimization problems.

References

[1] F. Glover and M. Laguna, “Tabu Search”, *Kluwer Academic Publishers*, pp.1-50, 1997.

[2] C. Voudouris and E.P.K. Tsang, “Guided local search”, *Technical Report CSM-247, Department of Computer Science, University of Essex* , 1995.

[3] C. Voudouris and E.P.K. Tsang, “Guided local search joins the elite in discrete optimization”, *Proceedings of DIMACS Workshop on Constraint Programming and Large Scale Discrete Optimisation*, 1998.

[4] L. Wang, A. A. Maciejewski, H. J. Siegel and V. P. Roychowdhury, “A study of five parallel approaches in a genetic algorithm for the traveling salesman problem”, *Intelligent Automation and Soft Computing*.

[5] Joshua D. Knowles, Richard A. Watson and David W. Corne, “Reducing local optima in single-objective problems by multi-objectivization”, *Proceedings of the First International Conference on Evolutionary Multi-criterion Optimization (EMO'01)*, Springer.

[6] G. Reinelt, “TSPLIB-A Traveling Salesman Problem Library”, *ORSA Journal on Computing*, Vol.3, No.4, pp.376-384, 1991.