Tabu assisted guided local search approaches for freight service network design

Ruibin Bai · **Graham Kendall**

1 Introduction

Service network design involves determination of the most cost-effective transportation network and service characteristics subject to various constraints. Good progress has been made in developing metaheuristic approaches that can compete or even outperform some commercial software packages (Ghamlouche et al 2004; Pedersen et al 2009). However, since most of these metaheuristic methods involve solving many capacitated multi-commodity minimum cost flow problems, computational time tends to be a bottleneck. In this research, we intend to build on the success of a guided local search metaheuristic (Bai et al 2010) in reducing computational time, without compromising solution quality, and carry out a set of experiments and analyses in an attempt to discover elements and mechanisms that could improve the algorithmic performance further.

2 Freight Service Network Design

We focus on a specific service network design formulation that has been studied recently in (Pedersen et al 2009). For the purpose of completeness, we also present it here. Let $\mathscr{G} = (\mathscr{N}, \mathscr{A})$ denote a directed graph with nodes $\mathscr{N}$ and arcs $\mathscr{A}$. Denote (i, j) be the arc from node i to node j. Let $\mathscr{K}$ be the set of commodities. For each commodity $k \in \mathscr{K}$, let $o(k)$ and $s(k)$ stand for its origin and destination respectively. Let y_{ij} be boolean design variables and y_{ij} equals 1 if arc (i, j) is used in the final design and 0 otherwise. Denote x_{ij}^k be flow of commodity k on arc (i, j). Let u_{ij} and f_{ij} be the capacity and fixed cost of arc (i, j). Denote c_{ij}^k be the variable cost of moving one unit of commodity k along arc (i, j). The service network design problem can be formulated as follows:

Ruibin Bai
Division of Computer Science, University of Nottingham Ningbo China
Tel.: +86-574-88180278
Fax: +86-574-88180125
E-mail: ruibin.bai@nottingham.edu.cn

Graham Kendall
School of Computer Science, University of Nottingham, Nottingham, UK
E-mail: gxk@cs.nott.ac.uk

$$\min \sum_{(i,j)\in\mathscr{A}} f_{ij}y_{ij} + \sum_{k\in\mathscr{K}} \sum_{(i,j)\in\mathscr{A}} c_{ij}^k x_{ij}^k \tag{1}$$

subject to

$$\sum_{k\in\mathscr{K}} x_{ij}^k \leq u_{ij}y_{ij} \quad \forall (i,j)\in\mathscr{A} \tag{2}$$

$$\sum_{j\in\mathscr{N}^+(i)} x_{ij}^k - \sum_{j\in\mathscr{N}^-(i)} x_{ji}^k = b_i^k, \quad \forall i\in\mathscr{N}, \forall k\in\mathscr{K} \tag{3}$$

$$\sum_{j\in\mathscr{N}^-(i)} y_{ji} - \sum_{j\in\mathscr{N}^+(i)} y_{ij} = 0 \quad \forall i\in\mathscr{N} \tag{4}$$

where $x_{ij}^k \geq 0$ and $y_{ij} \in \{0,1\}$ are decision variables. $\mathscr{N}^+(i)$ (respectively $\mathscr{N}^-(i)$) stands for the set of outward (respectively inward) neighbours of node i. We set $b_i^k = d^k$ if $i = o(k)$, and $b_i^k = -d^k$ if $i = s(k)$, and 0 otherwise. Note that for a given set of design variables $\bar{y}_{ij}$, the problem becomes a capacitated multicommodity minimum cost flow problem (CMMCF).

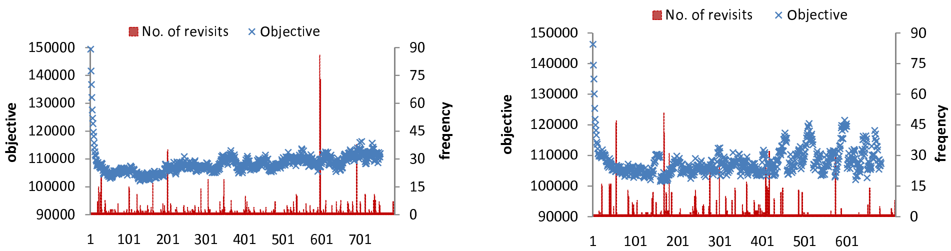
In (Bai et al 2010), we have shown that a variant of the guided local search (GLS) approach is able to produce competitive results with much less computational time than a recently proposed tabu search method (Pedersen et al 2009). Based on this initial success, this research aims to investigate, in detail, components and mechanisms that may lead to further improvement either in terms of computational time or solution quality. In particular, we intend to investigate; a) how effectively the current GLS escapes from a local optimum. b) whether more efficient mechanisms can be found and integrated within GLS.

3 Guided Local Search

Guided local search (GLS) is a metaheuristic for constraint satisfaction and combinatorial optimization problems (Voudouris and Tsang 2003). Its main idea is augmenting the original objective function so that the search not only escapes from local optima but also obtains high quality solutions. In our implementation, the neighbourhood is defined by either closing or opening an arc. The flow variables (x_{ij}) are then determined by solving the corresponding CMMCF problem using a free LP solver, `LP_Solve`. More implementation details of this approach can be found in (Bai et al 2010).

3.1 Local optima trap

In order to analyse how efficiently GLS escapes from local optima, we carried out experiments based on some of the 24 C-set benchmark problems used in (Pedersen et al 2009). All local optima that have been identified during the search are recorded together with its visit frequency (i.e. the number of times a local optimum is visited). All algorithms are tested on a same machine with the same amount of computational time. Initially we tested a simple GLS approach and the multi-start GLS (denoted by M-GLS) proposed in (Bai et al 2010). Figure 1 (a) and (b) show the corresponding information for instance C37 (similar results were obtained for other instances). The horizontal axis represents the list of local optima found during the search. One can see that both approaches revisited the same local optima many times. For the simple GLS, a few local optima are revisited over 80 times. On average, it visits each local optimum 2.3 times. For M-GLS, to our surprise, the average number of visits per local optimum is even higher (2.9). Nevertheless, high quality local optima by M-GLS tend to attract higher visit frequencies which may be one of the reasons that lead to better performance than the simple GLS. Overall, both versions of GLS wasted significant time when the same solution is evaluated many times. It is also interesting to note that GLS seems to converge to a good local optimum very quickly but is not so efficient when escaping from some local optima.



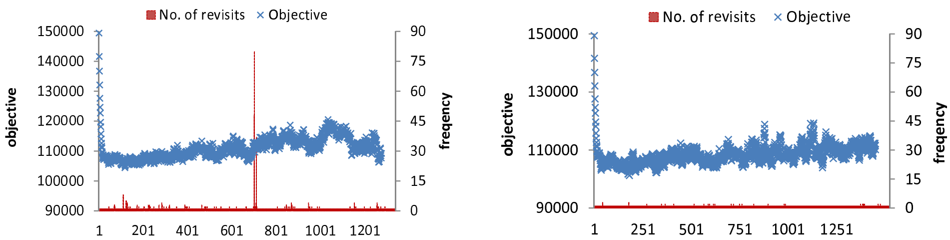
(a) Simple GLS (b) Multi-start GLS

Fig. 1 The number of visits to each local optima by the simple GLS and M-GLS (C37)

3.2 Tabu assisted GLS (T-GLS)

Since M-GLS cannot effectively prevent local optima revisits, we borrow the idea of the tabu search metaheuristic and introduced a tabu list into the simple GLS. The tabu list contains a list of arcs that have been modified recently in the current solution. The length of the tabu list is fixed to a predefined parameter *TabuLen*. The list is then maintained on a first-in-first-out basis. Figure 2 (a) and (b) plot the objective values and the revisit frequency by GLS with $TabuLen = 2$ and $TabuLen = 9$ respectively. It can be seen that even a tabu list of length 2 is effective in reducing GLS visiting local optima many times. When we increase *TabuLen* to 9, the majority of local optima are visited only once.

In order to measure the performance of these variants of GLS (namely simple GLS, M-GLS and T-GLS), we have tested them on 24 widely used benchmark problem instances. Details of the results are not included here but we will present them during the conference. The general observation is that T-GLS obtains better results than the simple GLS for all instances but is outperformed by M-GLS for the majority of instances. It seems, based on our observations, that T-GLS lacks necessary random elements to diversify the search.



(a) TabuLen=2 (b) TabuLen=9

Fig. 2 Revisit frequency of local optima by tabu assisted GLS (C37)

4 Discussions and future work

In this research, we carried out experiments to monitor the revisit frequency of local optima by a GLS metaheuristic. Results show that in both the simple GLS and its multi-start version, time is wasted due to revisiting. A simple tabu assisted GLS schema is implemented to prevent this problem. Although improvements have been obtained when compared against the simple GLS, this simple hybridisation fails to produce better results than the multi-start GLS. One of the possibilities that causes this problem may be that the current tabu assisted GLS does not contain random elements and it lacks efficient mechanisms to "jump" to a distant point in the search space. We are currently trying to implement various schemata to combine the multi-start GLS with a tabu list. Early experiments on a limited number of instances have shown very promising results. Comprehensive tests will be carried out and results will be reported during the conference.

In addition, our observations show that `LP_Solve` struggles on some problem instances, for which the majority of computational time is used when solving CMMCF problems (more than 85%). For some instances, LP_Solve even fails to solve one single CMMCF instance within 300 seconds. In future, we will look at other more efficient LP solvers, including `NAG` and `CPLEX`. It is hoped that a faster LP solver can improve the proposed algorithm further.

References

Bai R, Kendall G, Li J (2010) A guided local search approach for service network design problem with asset balancing. In: 2010 International Conference on Logistics Systems and Intelligent Management (ICLSIM 2010), January 9-10, Harbin, China, pp 110–115

Ghamlouche I, Crainic TG, Gendreau M (2004) Path relinking, cycle-based neighbourhoods and capacitated multicommodity network design. Annals of Operations Research 131:109–133

Pedersen MB, Crainic TG, Madsen OB (2009) Models and tabu search metaheuristics for service network design with asset-balance requirements. Transportation Science 43(4):432–454

Voudouris C, Tsang EP (2003) Guided local search. In: Glover F, Kochenberger G (eds) Handbook of Metaheuristics, Kluwer, pp 185–218