Available online at www.sciencedirect.com
ScienceDirect
Computers & Operations Research 36 (2009) 1406 – 1422
computers & operations research
www.elsevier.com/locate/cor

The heuristic concentration-integer and its application to a class of location problems

Vladimir Marianov[a,*,1], Michelle Mizumori[b,1], Charles ReVelle[✠]

[a] College of Engineering, Pontificia Universidad Católica de Chile, Vicuña Mackenna 4860, Santiago, Chile
[b] The Johns Hopkins University, Baltimore, MD, USA

Available online 10 March 2008

Abstract

We propose a new metaheuristic called heuristic concentration-integer (HCI). This metaheuristic is a modified version of the heuristic concentration (HC), oriented to find good solutions for a class of integer programming problems, composed by problems in which p elements must be selected from a larger set, and each element can be selected more than once. These problems are common in location analysis. The heuristic is explained and general instructions for rewriting integer programming formulations are provided, that make the application of HCI to these problems easier. As an example, the heuristic is applied to the maximal availability location problem (MALP), and the solutions are compared to those obtained using linear programming with branch and bound (LP + B&B). For one-third of the instances of MALP, LP + B&B can be allowed to run until the computer is out of memory without termination, while HCI can find good solutions to the same instances in a reasonable time. In one such case, LP-IP was allowed to run for nearly 100 times longer than HCI and HCI still found a better solution. Furthermore, HCI found the optimal solution in 33.3% of cases and had an objective value gap of less than 1% in 76% of cases. In 18% of the cases, HCI found a solution that is better than LP+B&B. Therefore, in cases where LP + B&B is unreasonable due to time or memory constraints, HCI is a valuable tool.

Keywords: Facility location; Heuristics; Integer programming

1. Introduction

In the p-median problem on a network (Hakimi [1,2], ReVelle and Swain [3]), locations must be selected for p facilities on a network, so that the average distance between each demand point and its closest facility is minimized. Heuristic concentration (HC) is a two stage metaheuristic originally designed to approach large instances of the p-median problem (Rosing and ReVelle [4]). This heuristic has been applied to this problem with excellent results, outperforming well-established heuristics such as Tabu Search (Rosing et al. [5]).

When applying the HC to the p-median, its first stage consists of q runs of a fairly simple base heuristic, such as the one-opt heuristic of Teitz and Bart [6], with some added random or varying component, usually a randomly generated initial feasible solution. This variation allows the multiple runs to find different good solutions, whereas if the algorithm

✠ Deceased.
* Corresponding author. Tel.: +562 3544974; fax: +562 5522563.
E-mail addresses: marianov@ing.puc.cl (V. Marianov), mschlossberg@gmail.com (M. Mizumori).
[1] We dedicate this paper to the memory of Professor ReVelle who passed away on August 10, 2005.

0305-0548/$ - see front matter
doi:10.1016/j.cor.2008.02.011

contained no varying element, each run would arrive at the same solution. Each solution contains a set of good locations of the facilities. The multiple good solutions are used to create the Concentration Set, which lies at the heart of HC. This Concentration Set is then used to apply a more complex algorithm in stage 2.

The Concentration Set, CS, is the union of the best m solutions from all of the runs of the base heuristic of the first stage. In other words, it is every item (location) that was selected in any of the top m ($m \leqslant q$) unique solutions. This CS will be the reduced eligible or candidate set for the second stage, since there is reason to believe that the elements of CS must have some reasonably good quality to have been selected to be in the top solutions, and therefore the optimal solution is likely to be a subset of CS.

The second stage of the original version of HC consists of a single run of an exact method, as linear programming plus branch and bound (LP + B&B), applied over the Concentration Set. At the end of this second stage, an optimal solution is found, constrained to choosing elements belonging to the CS. It is important to note, however, that although an exact solution method was used, the optimal solution over the reduced eligible set may not be the same as the optimal solution to the problem solved over the entire eligible set.

A later version of HC, also intended for solving the p-median, was developed by Rosing et al. [7]. In this modified version, called "Gamma Heuristic", the first stage is the same, but an additional Concentration Set is defined after the first stage: the concentration set CS_0, which is the intersection of the best m unique solutions from all of the runs of the first stage of the heuristic. This differs from the Concentration Set CS, because it only includes an item if that item occurred in every single one of the best solutions. Again, it is reasonable to believe that, if an item was selected in all of these good solutions, it will also be selected in the optimal solution. Therefore, the members of CS_0 are required to be selected in every solution considered in the second stage. Naturally, CS_0 contains a feasible number of items, since all of them are present in a number of feasible solutions.

As opposed to HC, where the second stage is an exact method, the second stage of the Gamma Heuristic consists of s runs of a more complex, non-exact algorithm, using the concentration sets described above. The CS is used as the set of all possible items to choose, while the items contained in the CS_0 are forced to belong to all the solutions. Notice that if this algorithm has no variation, for example, when using an exact method as in the original HC, there is no point in repeating s times the algorithm. However, if there is a varying element to the algorithm, repetition can lead to better solutions.

An optional third stage of the Gamma Heuristic takes the best solution from the second stage and tries to improve upon it by performing another heuristic on the eligible set, possibly again the one-opt method. Thus, the third stage provides the opportunity for further improvement.

For examples and analyses of different versions of HC, see Rosing and ReVelle [4], Rosing et al. [5,7], Rosing [8], Rosing and Hodgson [9], and Mizumori et al. [10].

The p-median belongs to a much larger family of combinatorial optimization problems, that can be described as picking at most p elements from a large set of "candidate" elements and putting them into a smaller set P, in such a way as to optimize an objective that depends on the quality of the p elements in the set P. Besides the p-median, this family contains other location problems, as the discrete p-center problem; the maximal covering location problem (MCLP) of Church and ReVelle [11]; the maximum availability location problem (MALP) of ReVelle and Hogan [12] and its derivates, as the queuing maximum availability location problem (QMALP) of Marianov and ReVelle [13]; the maximum expected covering location problem (MEXCLP) of Daskin [14] and many other similar problems.

In the p-median, each demand point is assigned to a single facility, specifically the closest one. In the MCLP, coverage of a demand is achieved if it has at least one facility within a preset neighborhood. In both cases, there is no point in locating more than a facility at the same place. On the other hand, in MALP, coverage frequently requires more than one facility in the neighborhood, and the optimal solution will most likely include sites with more than one facility. In other words, in MALP the same candidate location will be selected more than once and co-location of servers or facilities at the same site will result. The same happens in MEXCLP.

The method we propose here, called heuristic concentration integer (HCI), is a two-stage general procedure that can, in a number of instances, solve integer programming problems as the ones just described, often with optimal solutions. HCI is an outgrowth of the HC of Rosing and ReVelle [4] and the Gamma Heuristic of Rosing et al. [7]. In its second phase, HCI uses a heuristic, as does the Gamma Heuristic. Although it also could have an optional third stage, we use only two stages. In its first stage, HCI repeatedly uses a one-opt procedure, each time using a different initial feasible solution of the problem, and saving the best solution found. In its second stage, HCI uses a two-opt heuristic, using as candidate points all the locations found in any solution in the first stage. The main difference between HCI and