Approximate solution of Simple Plant Location Problem

Ivan Stolov

MIPT DIHT

June 28, 2020

Abstract

This report describes ways of how to get fast approximate solution of simple plant location problem. Our method is based on pseudo-Boolean representation [1, 4] of the goal function of SPLP. We also use well-known heuristics such as Khumawala [3] rules and some new ones. We study the efficiency of chosen combination of heuristics and approaches using J.E.Beasley's SPLP samples [2]. Results of benchmark instances that confirm effectiveness of the approach are also presented.

Introduction

Simple Plant Location Problem (SPLP) also known as warehouse location problem, uncapacitated facility location problem, the location of bank accounts problem etc. We will use the most popular terminology to describe SPLP. Let us consider the set $I = \{1 \dots n\}$ as a set of locations for facilities that can be opened or not opened. Each facility can provide an unlimited amount of commodity. Opening a facility at location $i \in I$ has a nonnegative cost c_i. Let a set $J = \{1 \dots m\}$ assign a set of customers that need to consume goods produced by facilities. For each pair (i, j) of customer and facility nonnegative transportation cost $g_{i,j}$ is defined, which means that if i-th consumer uses the service of j-th facility, it costs $g_{i,j}$. The goal is to come up with a set of facilities to open $\varnothing \subset S \subseteq I$, so that the total cost is as low as possible. Formally,

$$\sum_{i \in S} f_i + \sum_{j \in J} \min_{i \in S} g_{i,j} \to \min_{S \subseteq I}$$

This problem is the generalization of set-covering problem, therefore it is NP hard. Exact algorithms, approximation algorithms with constant performance guarantee, heuristics, randomized algorithms were developed for solving simplest location problem.

1 Theoretical preliminaries

1.1 Pseudo-Boolean representation

Pseudo-Boolean representation of SPLP [1, 4] based on simple idea of representation a loss function as a polynomial depending on m variables, where $m = |I|$. Each variable can take the value of 0 or 1, indicating, whether corresponding facility is open or not.

$$\varnothing \subset S \subseteq I\text{– facilities to open indexes}, 1 \leq i \leq m \text{ Then}$$

$$y_i = \begin{cases} 0, & \text{if } i \in S \\ 1, & \text{if } i \notin S \end{cases}$$

So, the solution can be represented as a vector $(y_1, \dots, y_m)$. Pay attention that y_i is an indicator of a facility i being CLOSED.

Then fixed cost component can be written as $F_c(\mathbf{y}) = \sum\limits_{i=1}^{m} f_i(1 - y_i)$, where f_i is the cost of opening i-th plant.

In order to construct transportation cost representation, let us consider j-th consumer. Their transportation cost vector is $T_k^j, k \in [1 \dots m]$. Obviously, lowest cost of supplying this client is $T_{(1)}^j$, since it is minimal value in T^j. Then, if the client is not supplied by plant with minimal overhead, cost increases at least by $\Delta_1^j := T_2^j - T_1^j$, then by $\Delta_2^j := T_3^j - T_2^j$ and so on. This intuition lead us to this representation of transportation costs of j-th client: (π stands for permutation here, i.e. $y_{\pi_d}^j$ means that $T_{y_{\pi_d^j}} \equiv T_{(d)}^j$)

$$T_{(1)}^j + \sum_{k=1}^{m-1} \Delta_k^j \cdot \prod_{d=1}^{k} y_{\pi_d^j}$$

Combining fixed and transportation costs together, we introduce total cost function:

$$F(\mathbf{y}) = F_c(\mathbf{y}) + F_t(\mathbf{y}) = \sum_{i=1}^{m} f_i(1-y_i) + \sum_{j=1}^{n} \left[T^j_{(1)} + \sum_{k=1}^{m-1} \Delta^j_k \cdot \prod_{d=1}^{k} y_{\pi^j_d}\right]$$

$F(\mathbf{y})$ is called Hammer-Beresnev function. In these terms SPLP can be formulated in this way:

$$F(\mathbf{y}) \to \min_{\mathbf{y}\neq\mathbf{1}}, \mathbf{y} \in \{0,1\}^m.$$

1.2 Heuristics

Since our first priority is speed, it is important to use heuristics. We used the following ones:

1. Khumawala-1 [3]
   If the coefficient of y_j in the linear term of the Hammer function is non-negative, $y_j = 0$.
2. Khumawala-2 [3]
   If the coefficient of y_j in the linear term is negative, and the sum of this coefficient and those of all non-linear terms containing y_j is negative, then $y_j = 1$.
3. New heuristic-1.
   Define two sets of facility indexes. P_+ is the set of indexes k such that there exists a term in Beresnev function, which has y_k as a multiplier and coefficient of this term is positive. P_- is the set of indexes d such that there exists a term in Beresnev function, which has y_d as a multiplier and coefficient of this term is negative. Every variable of Beresnev function lies in at least one of these sets, if all fixed costs are positive. Consider $j \in P_+ \setminus P_-$. According to definitions, it is always better to set $y_j = 0$, since it will decrease the free term of Beresnev function and if we set $y_j = 1$, free term of Beresnev function increases due to addition of positive value of y_j coefficient that becomes free term after $y_j = 1$. On top of that, other terms with y_j have positive coefficients and in every solution their contribution to the final cost is non-negative. So, the rule is the following: if $j \in P_+ \setminus P_-$, then set $y_j = 0$. It is logical to create another rule by analogy: if $j \in P_- \setminus P_+$, then set $y_j = 1$ but is is no use, since second Khumawals's rule covers these instances.
4. New heuristic-2.
   For every $i \in I$ calculate the sum of coefficients of terms with y_i. Then zero $\lfloor \alpha m \rfloor$ variables with biggest sums, if sums are positive. $(0 < \alpha < 1)$ – empiric constant.

1.3 Lower bound

It is important to be able to calculate lower bounds of goal function, since it can be used in computing upper bound of achieved result's error without consideration of all the possible solutions. We used one of the simplest lower bounds. $F(\mathbf{y}) \geq \sum_{j=1}^{n} \min_{i\in I} g_{i,j} + \min_{i\in I} f_i$. This is lower bound, since every customer needs to get service and at least one facility has to be opened.

2 Algorithm

2.1 Algorithm description

Our algorithm consists of two phases – preparation and main phase. Preparation is needed to set up all the matrices that are going to be used and get Hammer-Beresnev function. The main idea of the algorithm is that Khumawala rules work great and we can use them more than once. Usage of heuristic 2 guarantees that process is not endless. After applying any heuristic in the main phase we update Hammer-Beresnev function.

1. Preparation
   - Compute permutation matrix.
     This matrix stores indexes of elements in transportation cost matrix in increasing order. Elements are ordered in every column (for every customer)
   - Compute lower bound for goal function
   - Compute Hammer-Beresnev function
2. Main phase
   While facilities that are neither opened nor closed exist:
   - Apply Khumawala rules
   - Apply heuristic 1
   - Apply heuristic 2
   - Repeat

2.2 Algorithm implementation

Implementation of the algorithm can be found here https://github.com/kLubidorka/SPLP. C++ was used as the main programming language.

3 Computational Experiments

Benchmarks presented here are based on Beasley JE.OR-Library samples [2]. All the benchmark data can be found here. 1,4 GHz Quad-Core Intel Core i5 processor was used. According to the benchmarks, our heuristic algorithm works fast and rather accurate.

Data file	Our solution	Optimal solution	Error, %	Solution wall time, ms
cap71	932616	932616	0	1
cap72	977800	977799	0	0
cap73	1012480	1010641	0	1
cap74	1034980	1034977	0	0
cap102	854704	854704	0	2
cap104	928941	928942	0	3
cap131	820134	793440	3	15
cap132	891395	851495	5	11
cap133	952245	893077	7	16
cap134	998813	928942	8	15

4 Conclusions

We introduced heuristic solution of simple plant location problem and showed its effectiveness. Our goal is reached. However, this approach has some problems. For example, it cannot solve a problem with predefined accuracy, which makes it unsuitable for applications. Our approach and algorithm can be improved. First of all, lower bound can be amplified taking into consideration a relaxation of SPLP. It is important, since lower bound show the highest possible error for obtained solution. Next, other combinations of heuristics can be tested. On top of that, it is worth it to blend introduced heuristics with other approaches in order to find the best combination. Our ultimate goal is to compose a hybrid algorithm that can be set up to solve problem with predefined accuracy or to get solution as fast as possible and examine the dependency between acceptable accuracy and execution time. These are ways of further study.

References

[1] Beresnev VL. On a Problem of Mathematical Standardization Theory. Upravliajemyje Sistemy, USSR 1973

[2] Beasley JE.OR-Library, http://people.brunel.ac.uk/ mastjjb/jeb/info.html

[3] Khumawala B.M. An Efficient Branch-Bound Algorithm for the Warehouse Location Problem. Management Science. v18 (1972), pp 718-731

[4] Hammer PL. Plant Location — A Pseudo-Boolean Approach. Israel Journal of Technology 1968;6:330–332.