Exposing Metaheuristics as Web Services in Distributed Systems using OpenCF

Melquíades Pérez, Francisco Almeida, J.M. Moreno Vega, Vicente Blanco.

Computer Science Department
Universidad de La Laguna
La Laguna, Spain

Adrián Santos
Galotecnia R.S.S.
La Laguna
Tenerife, Spain

Abstract
Web Services (WS) have emerged as an industry standard attracting the attention of the scientific community as technological alternative for implementing computational portals. Using the OpenCF computational framework, we develop the p-hub Web Service Portal (p-HubWSP) to provide a platform with capabilities for the efficient execution of metaheuristics for p-hub problems through the Internet. The technology used eases the implementation of a web-accessed computing system. The p-HubWSP portal greatly eases the access and use of sequential and parallel codes and platforms.

Keywords: Web Services, Distributed Systems, Metaheuristics, p-hub problems
1 Introduction

Hosting methods for complex optimization problems on specialized server computers, accessible from the Internet, allow the scientific community to share the use of sophisticated solution algorithms for non-specialized users, and can save considerable resources (e.g., processing time, memory, wattage, etc.). Sequential, distributed or parallel resources can be exposed together into common interfaces where techniques for optimization problems are operative to client devices for collection and/or presentation of processing data.

Solution strategies founded on Client/Server-based applications have risen as technological alternatives. A friendly graphical interface enables access to non-local resources where codes can be executed remotely in geographically distributed sequential or parallel machines. The approach allows the use of the system by a larger number of users, including those without any knowledge of programming. Details of the target architecture are transparent to the user and codes can be efficiently executed in the remote system.

A WS basically consists of the use of open standards to connect applications through a communication network. This approach allows to homogenize the access to those services and eases the development at the clients. Since the information exchange employs Extensible Markup Language (XML) documents, the application is independent of the transport layer protocols and interfaces, enabling Internet-wide, implementation-independent distributed computing. The services are commonly associated to Internet but this is not strictly necessary. Well known examples of WS are the Google API or the Amazon WS.

Note that WS technology has the ability to communicate through any common firewall security measure without requiring changes to the firewall filtering rules. The solutions presented vary from static ad-hoc WS for specific applications on specific parallel machines, to more general and complete solutions. It is worth noticing that a WS based design adds, by itself, new facilities to those provided by some other projects. Moreover, it enables the use of the system by a larger number of users, including those without any

---

1 This paper has been partially founded by the EC (FEDER) and the Spanish MEC (Plan Nacional de I+D+I, TIN2005-09037-C02, TIN2005-08404-C04-03, TIN2008-06570-C04-03 and TIN2009-13363.
2 Email: melperez@ull.es
3 Email: falmeida@ull.es
4 Email: jmmoreno@ull.es
5 Email: vicente.blanco@ull.es
6 Email: adrian@galotecnia.com
knowledge in programming (both sequential or parallel).

The *p-hub* median problems are NP hard location-allocation problems that find the *p* points to establish facilities and the assignment of the users to those points. This is an important class of problems since they model many real situations, in particular, those scenarios where several nodes must interact among each other by sending and receiving traffic flow of some nature. The flow exchanged may represent data, passengers, merchandises, express packages, etc. Generally, in these situations is useful to find an optimal location of several points called hubs. The hubs act like exchanging points and the objective is to find the location of the hubs and the allocation of the remaining non hub nodes to the hubs minimizing an objective function.

In this paper, we describe a computing service for *p-hub* problems that has been implemented according to the W3C (World Wide Web Consortium) recommendations for WS development. The service, p-HubWSP (for p-Hub Web Service Portal), enables hosting metaheuristics to solve various model *p-hub* problems (capacitated and uncapacitated) as WS. While our work was focused on methods for *p-hub* problems, the approach is very easily extensible to any other algorithm for any other problem. At the back-end, our computational service currently implements constructive and evolutive metaheuristics, as well as hybrid methods, for *p-hub* problems. Our goal is to enable the use of the routines via a web interface as part of a free service to the scientific community. Users registered at the service can execute precompiled routines providing their own input data. However, since the proposed methodology is generic, the approach is applicable to any other sequential or parallel library.

The contributions of this paper are several: a) The p-HUBWSP computational service for *p-hub* problems presented is a contribution by itself. b) We customize the OpenCF computational framework in our target platform, providing a free computational web service to those users willing to use our tools and who are not provided with the methods or with the machine. The computational portal greatly eases the use of the algorithms both in sequential and in high performance parallel architectures. c) As a contribution to the scientific community, we propose sequential codes implementing efficient methods that are provided as a runtime system that can be freely used from our web site. d) The service brings together a pool of efficient methods under a common execution interface. e) The portal is open to incorporate new sequential or parallel algorithms, is ready to join additional servers providing further metaheuristic methods, and is also available to support new users to access the services.

The paper is structured as follows, section 2 introduces the *p-hub* median
problem and the methods we have implemented to solve it. The portal *p-HubWSP* based on the OpenCF computational framework has been presented in section 3 and some conclusions and future lines of work in section 4.

2 The *p-hub* problem, models and metaheuristics

There are many real situations where several nodes must interact with each other by interchanging traffic flow of some nature. The flow exchanged may represent data, passengers, merchandise, etc. The main objective is to locate and fix some nodes called hubs and to allocate the non-hub nodes to the hubs minimizing an objective function that describes the exchange flow and its cost. The hubs are fully interconnected and the traffic between any pair of nodes is routed through the hubs.

This problem was first formulated by O’Kelly as an integer quadratic problem and, we consider the uncapacitated and capacitated single allocation *p*-hub median problems. These problems can be formulated as follows.

Let $N$ be a set of $n$ demand points. Let also be any pair of nodes, $i$ and $j$:

$W_{ij} =$ number of units sent from $i$ to $j$

$C_{ij} =$ standard cost per unit of flow from $i$ to $j$

Normally, $W_{ii} = 0$ and $C_{ii} = 0$, for all $i$. If one of the points $i$ or $j$ is a hub, the standard cost per unit $C_{ij}$ is assumed. If they are both hubs, then the standard cost per unit of traffic is normally reduced and is equal to $\alpha C_{ij}$, where $\alpha$ is a parameter. In general, $0 \leq \alpha \leq 1$ to reflect the effect of the reduced cost in inter-hub flows. We can also consider parameters $\delta$ and $\gamma$ to capture the influence of reduced or increased costs for non-hub points.

Let $X_{ij}$ and $Y_j$ be decision variables defined as:

$$X_{ij} = \begin{cases} 
1 & \text{if point } i \text{ is allocated to hub } j \\
0 & \text{otherwise}
\end{cases}$$

$$Y_j = \begin{cases} 
1 & \text{if point } j \text{ is a hub} \\
0 & \text{otherwise}
\end{cases}$$

Then, the uncapacitated single allocation *p*-hub median problem can be defined as follows:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} \left( \delta \sum_{k=1}^{n} X_{ik}C_{ik} + \gamma \sum_{m=1}^{n} X_{jm}C_{mj} + \alpha \sum_{k=1}^{n} \sum_{m=1}^{n} X_{ik}X_{jm}C_{km} \right),$$

subject to:

$$(1) \quad X_{ij} \leq Y_j \quad i = 1, \ldots, n; \quad j = 1, \ldots, n$$
The objective function is to minimize the sum of the linear transportation costs (first and second terms) and the quadratic inter hub costs (third term). Constraint (1) ensures that a non-hub point can only be allocated to a location \( j \) if that node is a hub. Constraint (2) guarantees that each point is served by one and only one hub. Constraint (3) generates the correct number of hubs. And finally, (4) is the classical binary constraint.

The capacitated single allocation \( p \)-hub median location problem (CSApHM) can be formulated similarly. Let \( O_i \) and \( D_i \) defined as total flow with:

\[
O_i = \sum_{j \in N} W_{ij}, \quad \text{(origin at node } i) \quad D_i = \sum_{j \in N} W_{ji}, \quad \text{(destiny at node } i).
\]

Also, let \( \xi_k \) be the capacity of each hub \( k \). Then, the CSApHM can be obtained by adding the below constraint:

\[
\sum_{i=1}^{n} (O_i + D_i) X_{ij} \leq \xi_j Y_j \quad j = 1, \ldots, n
\]

The constraint (5) guarantees that the capacity of each hub is not exceeded.

In next subsections we briefly describe the metaheuristics implemented to broach the problem. All the metaheuristics use the same technique to represent the solutions which is described in [7].

### 2.1 Local Search

A Local Search can be completely stated through the neighbourhood structure used. We use two different neighbourhood structures for the location and another for the allocation. They establish when two solutions are neighbourhood. The neighbourhood structures at the location are exchange procedures where the exchange moves are restricted. To define our neighbourhood structures is useful to make the following observation: every solution for the \( p \)-hub problem induces a partition of the set of demand points. Each set of the partition is constituted by a hub and the points assigned to it.

By combining each one of the neighbourhood structures in location and allocation and, with the application of several greedy search procedures, we obtain two Local Search procedures, \( LS_1 \) and \( LS_2 \).
2.2 GRASP

GRASP (Greedy Randomized Adaptive Search Procedure) is a constructive heuristic consisting of two procedures. In the constructive procedure, a solution is iteratively constructed randomly selecting one element of the restricted candidate list (set of the best elements). Then, in the post-processing, it is attempted to improve this solution using an improved method. The best overall solution is kept as the result (see [6]).

2.3 Variable Neighbourhood Search

Variable Neighbourhood Search (VNS) [4] is a recent metaheuristic proposed by Hansen and Mladenovic. VNS systematically exploits the idea of change of neighbourhood during the search in a dynamic way.

For our implementation ([7]), as local search we use the LS2 procedure explained in section 2.1, and we consider nested recursive neighbourhoods.

2.4 Path Relinking

Path Relinking (PR) [3] is an evolutionary metaheuristic which operates on a Reference Set of solutions selected from an initial population of trial solutions. A Population Creation Method is used to create this population. PR select, combine, improve and generate new and better solutions from the Reference Set. One of the key of the procedure is the Relinking Method combination method used. A detailed description of the algorithm can be found in [7].

2.5 Genetic Algorithms

Genetic Algorithms are probabilistic heuristic techniques that imitate natural evolution. Evolution is produced by selection and genetic transformations of the individuals. During selection the more fit individuals survive, and during genetic transformations new individuals are produced from others. At the end, individuals will be more fit to the environment.

In [5] we defined four genetic operators for the p-hub location problem: location crossover (LC), location mutation (LM), allocation crossover (AC) and allocation mutation (AM).

2.6 Hybrid Approaches

We consider two hybrid approaches: GRASP + PR and VNS – PR, using, respectively, GRASP and VNS as the Population Creation Method of PR.
3 Metaheuristics as Web Services for p-hub problems (p-HubWSP)

We have used the OpenCF framework [2] to expose metaheuristics for p-hub problems in our server which are freely available at [1].

3.1 The Web Service Interface

Figure 1 shows the web interface that we developed for the metaheuristic solvers for p-Hub problems Web Service Portal (p-HubWSP), based on the Open Computational Framework (OpenCF). p-HubWSP offers a WS interface to solve p-hub problems using the metaheuristics described in section 2. Registered users can submit input data files to solve through the WS interface and obtain information about the status of their computations. A user who wants to use the p-HubWSP portal should follow the steps below in order to process her data with a metaheuristics algorithm:

(i) **Register at p-HUBWeb portal.** Access control system for resources.

(ii) **Select an action for the heuristic to be processed.**

(iii) **Input the parameters for the algorithms and/or upload the data to be processed.**

(iv) **Check the status of the job on the WS status page.**

New routines can be easily incorporated into the service just by adding the XML description file with the pre-compiled binary code to the server.

These procedures are described in [6] and [7].
Just as in the same way new OpenCF servers providing further methods can be integrated just by adding the public WS interface WSDL address of the new server using the managing server capabilities of the OpenCF client module. Since servers and services are transparent to the users, they could be used as parallel platforms.

4 Conclusion

We have developed the p-HubWSP portal, the computational portal providing efficient metaheuristics to solve p-hub problems. The platform is founded in OpenCF, a Web Service based framework oriented to remote computation. The platform is best suited to provide an easy access to distributed systems using the web services mechanism. A pool of methods is freely provided as a runtime system ready to use from the Internet. The server system offering the services could be sequential and/or parallel platforms and also the services could be sequential or parallel. In the near future we plan to extend the system with the ability to allow the metaheuristics cooperate between them using the web services interoperability and cooperation mechanisms.

References


