Participant selection for crowdsourcing disaster information

E. T.-H. Chu[1], C.-Y. Lin[1], P. H. Tsai[2] & J. W. S. Liu[3]
[1]*Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology, Taiwan, ROC*
[2]*Institute of Manufacturing Information and Systems, National Cheng Kung University, Taiwan, ROC*
[3]*Institute of Information Science, Academia Sinica, Taiwan, ROC*

Abstract

Experiences with past major disasters tell us that people with wireless devices and social network services can serve effectively as mobile human sensors. A disaster warning and response system can solicit eye-witness reports from selected participants and use information provided by them to supplement surveillance sensor coverage. This paper describes a natural formulation of the participant selection problem that the system needs to solve in order to select participants from available people given their qualities as human sensors and the costs of deploying them. For this, we developed a greedy algorithm, named PSP-G, that first calculates the benefit-to-cost (B2C) factor of each participant. It then dispatches participants to regions according to participants' B2C. We compared PSP-G with the two well-known optimization methods, BARON and BONMIN. The results show that PSP-G delivers a near optimal solution with a low time complexity. In particular, the time PSP-G needs can be merely one tenth of the execution time of the existing optimization methods, which makes PSP-G a practical solution for emergency needs in disaster areas.

Keywords: crowdsourcing, social network, disaster management.

1 Introduction

Despite advances in sensor technologies, disaster surveillance and response systems cannot always rely solely on sensors and sensor networks/systems for surveillance data to acquire situation awareness and support decisions.

WIT Transactions on The Built Environment, Vol 133,
www.witpress.com, ISSN 1743-3509 (on-line)
doi:10.2495/DMAN130211

Deployment costs may limit the coverage and density of sensors (e.g. [1, 2]). In-situ sensors in disaster affected areas may be damaged, and thick clouds, vegetation, buildings, etc. can render remote sensors (e.g. surveillance satellites and unmanned aerial vehicles) ineffective. Fragmented sensor coverage can leave decision makers and responders ill-informed of imminent dangers to hundreds of people. This was what happened during Typhoon Morakat in 2009 in Taiwan [3].

Using people armed with wireless devices and Web 2.0 services as mobile *human sensors* is a way for a system to enhance its surveillance capability. Eye-witness reports of conditions at selected locations can complement data from physical sensors to eliminate blind spots and mend fragmentation in sensor coverage. A disaster surveillance system designed to make use of human sensors triggers a *human sensor data collection* (HSDC) process under specified conditions. By selecting individuals from available human sensors to participate and directing the selected participants to explore the threatened area, the system aims to collect eye-witness reports needed for it to acquire situation awareness in the shortest time.

We call the problem of selecting individual human sensors to collect data in different regions of the threatened area in order to optimize some specified objective subject to constraints in the number and costs of human sensors the *participant selection problem* (PSP). Solutions to variants of the problem are bases of the participant selection strategy used by the system. This paper presents a natural formulation of the PSP and approximation and a heuristic algorithm for solving the problem.

Following this introduction, Section 2 presents models of the threatened area and human sensors available for selection. Section 3 presents formulations of the participant selection problem and variations of the problem. As it will become evident, PSP is an extension of a special case of the well-known maximum *generalized assignment problem* (GAP) [4, 5], which is known to be NP-hard and APX-hard to approximate it. Section 4 compares the problems and presents an overview of existing algorithms and solutions of *GAP* and its variants. Section 5 presents approximation and heuristic algorithms for solving variants of the PSP. Section 6 summarizes the paper and discusses future works.

2 Participant selection problem (PSP)

Specifically, the solution of a *participant selection problem* (PSP) is a selection of participants of a HSCD process and an assignment of the selected participants to regions to optimize some objective function, referred to as (total) *value*, subject to constraints in terms of the number, quality and costs of participants available for selection. Table 1 provides a summary. The formulations of PSP described in subsequent sections focus primarily on how to make best use of participants of types I and M.

Table 1: Model of participants.

Type	Property	Benefit	Cost
I	Professional responders	High	High
M	Registered Volunteers	Medium/Low	Medium/Low
U	Unregistered Volunteers	Low	Low

We use the notations defined below to denote the input parameters of the PSP:

- The area has ρ regions R_1, R_2, … R_ρ, and their values are v_1, v_2, … v_ρ, respectively.
- Among π participants P_1, P_2, … P_π , first π(I) participants are of type I ; the next π(M) participants are of type M; the remaining $\pi - \pi(\mathrm{I}) - \pi(\mathrm{M})$ participants are of type U.
- For $i = 1, 2, \ldots \pi$ and $k = 1, 2, \ldots \rho$
  - b_{ik} ($0 \le b_{ik} \le v_k$) is the *value* (*benefit*) achievable by P_i if he/she is assigned to explore region R_k and
  - c_{ik} ($0 \le c_{ik}$) is the cost of P_i when assigned to explore region R_k.
- B (>0) is the total budget available to be spent on all selected participants.

We assume that values of regions, costs of participants and total budget are positive integers.

In terms of these notations, a variant of the PSP can be stated below:

Maximize $$V = \sum_{k=1}^{\rho} \sum_{i=1}^{\pi} b_{ik} x_{ik} \quad (1)$$

Subject to $$\sum_{i=1}^{\pi} b_{ik} x_{ik} \le v_k \ , \ k = 1, 2, \ldots \pi \quad (2)$$

$$\sum_{k=1}^{\rho} x_{ik} \le 1 \ , \ i = 1, 2, \ldots \rho \quad (3)$$

$$x_{ik} \in \{0, 1\}, \qquad i = 1, 2, \ldots \pi, \qquad k = 1, 2, \ldots \rho \quad (4)$$

$$\sum_{k=1}^{\rho} \sum_{i=1}^{\pi} c_{ik} x_{ik} \le B \quad (5)$$

The variable $x_{ik} = 1$ means that participant P_i is selected and is assigned to region R_k; it is equal to 0 if otherwise. The set $\{x_{ik}\}$ for all $i = 1, 2, \ldots \pi$ and $k = 1, 2, \ldots \rho$ gives an assignment of a subset of participants to regions; the inequality (3) allows $\{x_{ik}\}$ to be a proper subset of the set of all participants. The term V given by Eq. (1) is the total value achievable by all the selected participants when they explore their assigned regions; V is equal to the sum of benefits contributed by all the participants. The inequality (5) says that the total cost incurred by them must not be greater than the budget B. The constraint (2) ensures that the solution $\{x_{ik}\}$ never assigns more participants to any region than needed to achieve the full value of the region. The variant of the PSP is called PSP-frugal, a solution of it will be presented in Section 4.

3 Related works

Problems on managing resources (e.g. devices/equipment, supplies, and human sensors) during disaster preparedness and response phases have been treated to a great extent in literature. As an example, Therese *et al.* [9] presented an Android-based disaster management system, which also handles participant selection and assignment. A major difference between their work and ours is in problem formulations and algorithms. Our work is among the first that apply GPS and knapsack algorithms to resource allocations by disaster surveillance and response applications. One of the most well-known crowdsourcing platforms is SAHANA [10]. Its primary function is to facilitate the collection, filtering, organizing and display social reports. Participant selection is not supported.

Human resource allocation problems have also been treated extensively for many other types of applications. As examples, Taesoo Kwon and Dong-Ho Cho [14], Bartoli *et al.* [15], and Chen Junjie *et al.* [16] proposed human resource allocation algorithms for various scenarios. Compared to the formulations presented in the previous section, their models and problem formulations are more ad hoc.

Returning to Section 2, we note that in the case of infinite budget (i.e. $B = \infty$), PSP-Frugal is a special case of the well-known maximum general assignment problem (GAP) [4, 5]. The GAP is a generalization of the assignment problem [7] that just celebrated its golden anniversary recently. The problem is often stated as that of seeking an optimal placement of objects in bins. For each bin, each object in it has a profit and a weight that are dependent on both the object and the bin. The objective is to find placements of objects in bins so that the total profit is maximized subject to the constraints that the total weight of all objects in every bin is no greater than the weight limit of the bin. The GAP is known to be NP-hard and APX-hard to approximate it. Some algorithms for solving the problem use algorithms for the 0-1 knapsack problem [7] as the basis. An example is the greedy (δ+1)-approximation algorithm in [4]: It finds a solution of the GAP iteratively using a δ-approximation algorithm for the knapsack problem to find a tentative solution of the single-bin sub-problem, one bin at a time.

Specifically, for $B = \infty$, the PSP-Frugal is the special case of the GAP where the weight and profit of every object put in every bin are equal. The special case of equal weight and profit knapsack problem is known as the subset sum problem [8]. The functional form of the subset sum problem can be stated as follows: Given a set of N non-negative integers, find a subset of integers with the maximum sum among all subsets with sums equal to or less than the given limit. This problem is known to be NP-hard in general, but can be solved exactly in a reasonable amount of time by exhaustive search when N is small (e.g. less than 20) or by dynamic programming when the precision of the problem is small. For the PSP-Frugal, N is the number π of candidate participants, which can be large. On the other hand, the number of distinct values of $b_{i,k}$ is usually small. In practice, it also makes sense to adjust the unit of $b_{i,k}$'s to reduce the precision of the problem.

4 PSP-G algorithms

The PSP-G (PSP-Greedy) algorithm shown in Table 2 is a greedy algorithm. After initializing related parameters (line 1), PSP-G calculates the benefit-to-cost

(B2C) factor $Q_{i,k}$ (= b_{ik} / c_{ik}) of each participant P_i (line 2) and sorts all B2C factors by their values in non-increasing order (line 3). It then dispatches participants in turn to regions according to participants' B2C (line 4 to line 13). Each time, the participant with the highest B2C is selected first (line 5) and is dispatched to a region where he or she can increase the total value most if this assignment satisfies both the budget and the value constraints (line 6). The participant selection process stops until all participants are dispatched or total values cannot be further increased. Finally, another round of HDCS is issued to solicit more participants if the threatened areas are not fully explored (line14).

As one sees from Table 2, PSP-G first takes O($\rho\pi$) to calculate the B2C factors (line 2). It then takes O($\rho\pi$ log($\rho\pi$)) to sort them (line 3). The time complexity of the selection process (line 4 to 13) is bounded by O($\rho\pi$). Therefore, the time complexity of PSP-G is O($\rho\pi$ ln $\rho\pi$).

Table 2: Algorithm for PSP-G.

Algorithm for PSP-G
β_k: The current benefit region k gets from the selected participants
ψ: The remaining budget
Q: A ordering set to record all B2C factors $Q_{i,k}$ of participant i to region k.

1:	Set $\psi = B$ and $\beta_k = 0,\ k = 1, 2, \ldots, \rho$;
2:	Calculate all B2C factors (i.e., $Q_{i,k} = b_{ik} / c_{ik}$);
3:	Sort $Q_{i,k}$ by their values in descending order;
4:	**while (**Q is not empty**)**
5:	$H_{i,k}$ = the first element in Q;
6:	**if (**P_i is not selected and ($\beta_k + b_{ik} \le v_k$) and ($\psi - c_{ik} \ge 0$))
7:	β_k += b_{ik} ; ψ −= c_{ik} ;
8:	Dispatch P_i to R_k;
9:	Remove all P_i 's B2C factors from Q;
10:	Mark P_i as a selected participant;
11:	**end if**
12:	Remove $H_{i,k}$ from Q;
13:	**end while**
14:	If participants are not enough, broadcast participant collecting message on social network again.

Example:
To illustrate the PSP-G algorithm, we consider here a simple example. There are two regions R_1 and R_2 and three participants P_1, P_2, and P_3. The region value v_1 is 60 and v_2 is 50. The total budget is 100.

As Table 3 shows, the B2C factor list is 1.31, 1.29, 0.71, 0.69, 0.65, and 0.48. Hence, PSP-G first selects P_1 for examination. Because the total budget is not

Table 3: Profiles of participant.

Participant number	Cost in R_1	Cost in R_2	b_{i1}	b_{i2}	b_{i1} / c_{i1}	b_{i2} / c_{i2}
1	19	36	25	25	1.31	0.69
2	23	31	15	15	0.65	0.48
3	56	31	40	40	0.71	1.29

used up and the achieved value of region R_2 does not exceed the upper bound 50, P_1 is dispatched to region R_2. Then P_1's B2C factors 1.31 and 0.69 are removed from the B2C factor list. The achieved value V now becomes 25 and the available budget decreases to 81. PSP-G next selects P_2 for examination since its B2C factor 1.29 is now the largest in the list. Because both value and budget constraints can be satisfied, P2 is dispatched to R2. The total achieved value V becomes 65 and the available budget decreases to 50. Similarly, P_3 is dispatched to R_1 and the final V is 80, which is the optimal solution of this problem.

5 Experiment setup

We evaluated the PSP-G algorithm via simulation. Our simulation experiments were conducted on an Intel i7 processor with CPU speed 3.3GHz and the total RAM is 6Gb. The algorithm PSP-G was written in Java with Eclipse. We considered a big earthquake that seriously damaged Yunlin county, Taiwan. A HSDC process was triggered so as to collect data in different regions of the threatened area. The total budget B is 100. As Figure 1 shows, we have 9 regions: Mailiao, Lunbei, Erlun, Xiluo, Citomg, Taixi, Dongshi, Bauzhong and Tuku. Their values are set at 100, 100, 80, 60, 80, 60, 60, 60 and 60, respectively. Also, the total number of participants is 1000, and the number of each type participant is one-third of the total participants. Initially, all participants are uniformly distributed in each region. We determine the value of each b_{ik} by

$$b_{ik} = basic\ benenfit/distance,$$

where the basic benefit of a type-I participant is 10, of a type-M participant is 5 and of a type-U participant is 3. Our formulation indicates that the farther the participant P_i is away from the region R_k, the less R_k can be benefited by P_i. For a participant, the distance between any locations inside his/her original region is set at one. Whenever the participant moves across a region, the distance he moves increases one. In other words, the distance between any two nearby regions is set to one. For example, if we have a type-I participant P_1 in R_1 (Mailiao), then b_{11} is 10(=10/1), b_{12} is 5(=10/2), b_{16} is 5(=10/2) and b_{19} is 3.3(=10/3). Similarly, the value of each c_{ik} is determined by

$$c_{ik} = basic\ cost * distance,$$

where the basic distance of a type-I participant is 3, of a type-M participant is 2 and of a type-U participant is 1. The farther the participant P_i away from the region R_k, the higher the cost will be. For example, if we have a type-I participant P_1 in R_1 (Mailiao), then c_{11} is 10(=10*1), b_{12} is 20 (=10*2), b_{16} is 20 (=10*2) and b_{19} is 30(=10*3).

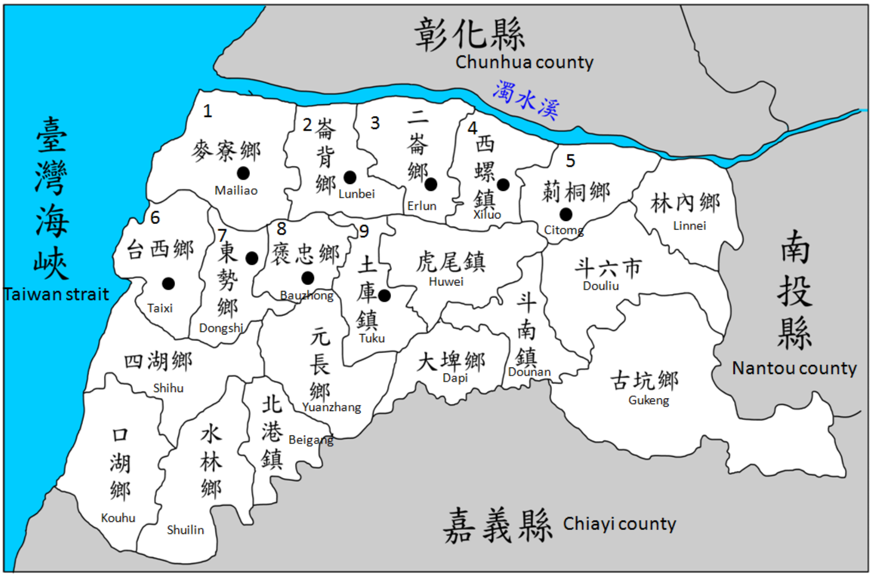

Figure 1: Map of Yunlin County, Taiwan.

We compare PSP-G with two commonly-used optimization methods. They are BARON [17] and BONMIN [18], which were executed by NEOS servers online [19]. BARON (Branch-And-Reduce Optimization Navigator) is a global optimization solver for convex optimization problems. It solves both linear programming and nonlinear programming problem by using branch and bound strategies. BONMIN (Basic Open-source Nonlinear Mixed Integer programming), also a global optimizer, adopts six different strategies (i.e., B-BB, B-OA, B-QG, B-Hyb, B-ECP and B-iFP) to have better performance in optimization. These two optimization solvers well represent the state of the art in optimization software. In our experiments, we first evaluated the performance of PSP-G in maximizing the total benefits contributed by all participants. We then compared the execution time of PSP-G with that of BARON and BONMIN respectively.

6 Performance measures and simulation results

The performance data obtained from our experiments is summarized by Figure 2. Let V_g represent the total benefits achieved by PSP-G, V_{br} represent that by BRARON, V_{bo} represent that by BONMIN. Performance ratio refers to the ratio of the total benefits achieved by two different methods. In Figure 2 the x-axis is the number of threatened regions and y-axis is the performance ratio. Although BARON and BONMIN are global optimizers, V_{bo} and V_{br} are not the same. This is because each of them has a different converging speed of searching the optimal solution and different stop conditions. In all test cases, both performance ratios V_g/V_{br} and V_g/V_{bo} are closed to 1. It particular, PSP-G delivers almost the same result as BARON and BONMIN when the number of region is nine. Our numerical results indicate that PSP-G, a polynomial time algorithm, can deliver a near optimal solution with less time complexity.

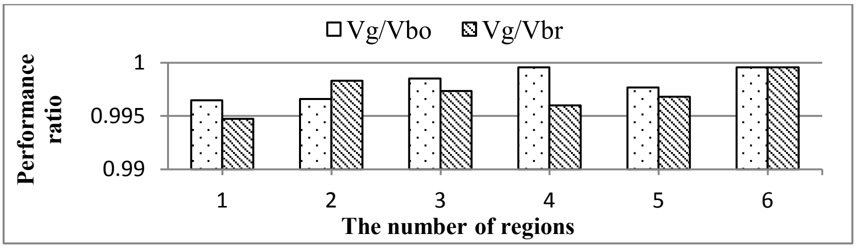

Figure 2: The performance ratio.

We further measured the execution times of PSP-G, BAROM and BONMIN respectively. We set the number of regions at nine and varied the number of participants from 1,000 to 8,000. As Figure 3 shows, compared to BAROM and BONMIN, the execution time of PSP-G increases slightly when the number of participants increases. Most of the experiment configurations can be finished by PSP-G in a few seconds. In contrast, the execution time of BONMIN and BONMIN increases significantly when the number of participants becomes large. In particular, when the number of participants comes to 8,000, the execution time BARON takes is almost 10 times longer than PSP-G. According to our experiment results, we argue that PSP-G is a practical solution for dispatching participants, especially for emergency cases which cannot be postponed until the threatened areas are fully explored.

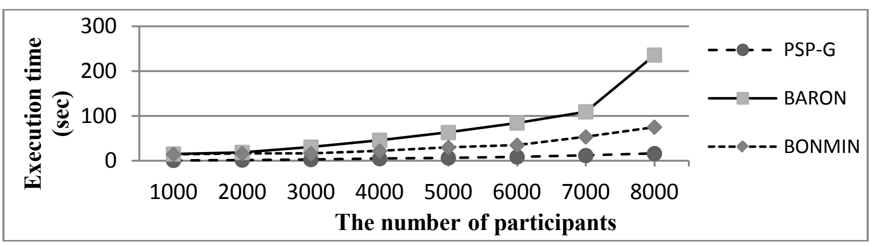

Figure 3: The execution time.

7 Summary and future work

In this paper, we presented a formulation of participant selection, in which the total benefits contributed by all participants should be maximized. We developed PSP-G, a greedy algorithm that first calculates the B2C factor of each participant. It then dispatches participants to regions according to participants' B2C. Each time the algorithm tries to maximize the total budget of the partial assignment. According to our experimental results, PSP-G delivers a near optimal solution with less time complexity. In particular, its execution time can be reduced to only one tenth of that of existing optimization methods. The experiment results show that PSP-G is a practical solution for emergency needs in disaster areas. In the future, we plan to integrate PSP-G with an existing open source disaster management system so as to further demonstrate the applicability of PSP-G.

Acknowledgements

This work was partially support by The Academic Sinica Project Open ISDN (Open Information System for Disaster Management) and by the Taiwan National Science Council Grants NSC 100-2219-E-224-001.

References

[1] Deep Horizon Oil Spill, http://en.wikipedia.org/wiki/Deepwater_Horizon_oil_spill and http://www.google.com/crisisresponse/oilspill/
[2] 2009 California Wildfire, http://en.wikipedia.org/wiki/2009_California_wildfires
[3] Typhoon Morakot Aftermath, http://en.wikipedia.org/wiki/Typhoon_Morakot#Taiwan_3
[4] Generalized Assignment Problem, http://en.wikipedia.org/wiki/Generalized_assignment_problem
[5] Oncan, T., "A Survey of Generalized Assignment Problem and its Applications," *Information Systems and Operation Research*, May 2008
[6] Penticoa, D. W., "Assignment Problems: A Golden Anniversary Survey," *European Journal of Operational Research*, Volume 176, Issue 2, 16 January 2007.
[7] Knapsack Problem, http://en.wikipedia.org/wiki/Knapsack_problem
[8] Subset Sum Problem, http://en.wikipedia.org/wiki/Subset_sum_problem
[9] J. Therese, B. Fajardo, and C. M. Oppus, "A mobile disaster management system using the android technology," in World Scientific and Engineering Academy and Society, 2010.
[10] M. Careem, C. De Silva, R. De Silva, L. Raschid, and S. Weerawarana, "Demonstration of Sahana: free and open source disaster management," in Proceedings of the 8th Annual International Conference on Digital Government Research: bridging disciplines and domains, 2007.

www.witpress.com, ISSN 1743-3509 (on-line)

[11] R. McCreadie, C. Macdonald, and I. Ounis, "Crowdsourcing a news query classification dataset," in the Crowdsourcing for Search Evaluation Workshop, 2010.
[12] J.-F. Paiement, J. G. Shanahan, and R. Zajac, "Crowd sourcing local search relevance," in CrowdConf, 2010.
[13] T. P. Walter, "Crowdsourcing as a business model: An exploration of emergent textbooks harnessing the wisdom of crowds," Business, pp. 555-568, 2010.
[14] Taesoo Kwon, and Dong-Ho Cho, "Adaptive-Modulation-and-Coding-Based Transmission of Control Messages for Resource Allocation in Mobile Communication Systems," IEEE Trans. Vehicular Technology, Vol. 58, no. 6, JULY 2009.
[15] Giulio Bartoli, Andrea Tassi, Dania Marabissi, Daniele Tarchi, and Romano Fantacci, "An Optimized Resource Allocation Scheme Based on a Multidimensional Multiple-choice Approach with Reduced Complexity," in Proc. ICC, 2011, pp. 1-6.
[16] Chen Junjie, Wang Shating, and Chen Chen, "Method research for dynamic multi-project human resource allocation based on Multidimension model," in Proc. ICCIII, 2011, pp. 78-81.
[17] BARON. http://neos.mcs.anl.gov/neos/solvers/go:BARON/GAMS.html
[18] BONMIM. http://neos.mcs.anl.gov/neos/solvers/go:Bonmin/GAMS.html
[19] NEOS. http://neos.mcs.anl.gov/neos/solvers/index.html

www.witpress.com, ISSN 1743-3509 (on-line)