POLITECNICO DI TORINO
Repository ISTITUZIONALE

Stochastic Optimization of Service Provision with Selfish Users

(Article begins on next page)

07 May 2024

Stochastic Optimization of Service Provision with Selfish Users

F. Altarelli*, A. Braunstein*, C.F. Chiasserini†, L. Dall'Asta*, P. Giaccone†, E. Leonardi†, R. Zecchina*.

* Department of Applied Sciences and Technology, Politecnico di Torino, Torino, Italy – {firstname.lastname}@polito.it

† Department of Electronics and Telecommunications, Politecnico di Torino, Torino, Italy – {lastname}@tlc.polito.it

Abstract—We develop a computationally efficient technique to solve a fairly general distributed service provision problem with selfish users and imperfect information. In particular, in a context in which the service capacity of the existing infrastructure can be partially adapted to the user load by activating just some of the service units, we aim at finding the configuration of active service units that achieves the best trade-off between maintenance (e.g. energetic) costs for the provider and user satisfaction. The core of our technique resides in the implementation of a belief-propagation (BP) algorithm to evaluate the cost configurations. Numerical results confirm the effectiveness of our approach.

I. INTRODUCTION

[1] Mathematical models of the distributed service provision problem has been studied thoroughly in computer science under the name of *selfish load balancing* [1] and *congestion games* [2, 3]. Most results apply concepts borrowed from Game Theory and concern worst-case analysis, in particular the computation of the so-called "price of anarchy", i.e. the ratio between the cost of the worst Nash Equilibrium (NE) and the optimal social cost [4]. Several works also address algorithmic issues, such as the question of designing distributed dynamics that converge to NEs, their convergence time [5], or computational complexity [6]. In many practical problems, service providers should be more interested in the average-case scenario, in particular in the average cost of service/resource allocations determined by the selfish behavior of users. In order to be able to compare the expected cost of different service allocations, a service provider is called to the arduous computational task of evaluating an average over the possibly huge number of different NEs that are obtained as result of the allocation. In addition, service providers do not always have perfect information about the user behavior – a fact that is usually modeled by including some stochastic parameter into the problem [7, 8, 9, 10]. In the presence of stochasticity, algorithms based on Monte Carlo methods are extremely inefficient already for moderately large problem sizes, whereas recent works [11] have shown that much better results can be obtained using message-passing algorithms inspired by statistical physics methods.

In our formulation, we assume that the total service capacity of the existing infrastructure can be adapted in part, by activating some of the service units. Our goal is to find the configuration of active server units that achieves the best trade-off between maintenance costs for the provider and user satisfaction. For the sake of example, we assume maintenance

costs expressed in terms of energy costs to keep the service unit active. For any given configuration of service units and users, we propose to use a belief-propagation (BP) algorithm to evaluate the cost of every service configuration. Moreover, we put forward an approximate method, also based on BP, which allows to perform the average over the stochastic parameters within the same message-passing algorithm used to average over the NEs. The information obtained is then used to optimize the service units allocation. This can be done easily either exhaustively or by means of decimation methods.

II. SYSTEM MODEL

The service provision system is represented by a bipartite graph $G = (\mathcal{U}, \mathcal{S}, \mathcal{E})$, in which $\mathcal{U} = \{1, \ldots, U\}$ and $\mathcal{S} = \{1, \ldots, S\}$ are the sets of nodes, *users* and *service units*, respectively, and $\mathcal{E}$ is the set of indirect edges (u, s) connecting elements $u \in \mathcal{U}$ and $s \in \mathcal{S}$. In general the graph is not complete, i.e., users cannot connect to any service unit. In addition, every service unit s has an operational energy cost, r_s. Thus, in certain time periods, it may be convenient to keep active only part of the existing service units ($x_s = 1$) and deactivate the others ($x_s = 0$).

The first ingredient of the model is the rational behavior of the users. In many problems, such as selfish load balancing [1], this is introduced by assuming that the quality of service received by a user, when selecting a service unit, depends on the load of the unit at the time of service, defining a correlation between users' utilities. Here, we simplify the model assuming that users' satisfaction in selecting a service unit does not depend on the state of the service unit itself (provided it is available). Each edge (u, s) has a weight, $w_{us} \in \mathbb{R}$, that represents the satisfaction obtained by user u when selecting service unit s. However, users' decisions are not independent, as there is a limitation to the number of individuals that can be served at the same time by the same service unit. The weight in the opposite direction, $w_{su} \in \mathbb{R}^+$, is the workload sustained by the service unit s when providing the service to user u. If we assume that each service unit s can sustain a maximum load c_s, the sum of the workloads w_{su} of all users u selecting unit s should not exceed c_s. This set of hard *capacity constraints* introduces an indirect competition between users. More precisely, suppose that user u considers service unit s to be the most preferable one (i.e., $w_{us} \geq w_{us'} \ \forall s' \neq s$), but adding the workload w_{su} of u to the total load already faced by unit s, it would exceed c_s. Then we say that service unit s is *saturated* for user u and the latter has to access another of the service units accessible to her. She will thus turn to the unit with the second best weight. If this one is *available*, user

u will make use of it, otherwise she will try the third one on her list and so on. Note that multiple connections from the same user to many service units are not allowed.

The second ingredient is stochasticity. We imagine that in any realistic situation the activity of the users could follow very complex temporal patterns. Users could leave the system and come back, using different service units depending on their preference and the current availability. The stochastic nature of the problem is summarized into a set of stochastic parameters $\{t_u\}_{u\in\mathcal{U}}$. With probability $P(t_u = 0)$, the user u is absent from the service system and $t_u = 0$, whereas with probability $P(t_u = 1)$, she is present and $t_u = 1$. For the moment, we can assume that the actual realization t_u, $\forall u \in \mathcal{U}$, is known.

Given the bipartite graph, the configuration of active service units $\{x_s\}_{s\in\mathcal{S}}$ and the set of parameters $\{t_u\}_{u\in\mathcal{U}}$, every user tries to maximize her own utility using the best service unit available (i.e., among those that are not saturated or inactive). Such a system model can represent several different application scenarios. For example, we can represent videoconferencing, including several Multipoint Control Units (MCUs) or a heterogeneous wireless access network, where points of access, possibly using different technologies, are available (e.g., 3G/LTE, WiFi, WiMax) to the users. In this scenarios, indeed, it would be convenient to switch off service units when underloaded.

III. PROBLEM DEFINITION

The system outlined above poses the following service provision problem: *at any given time period, which service units should be activated by a central controller, so as to maximize the users' satisfaction and minimize the system energy consumption?* Since the decision of the central controller has to account for the rational behavior of the users, we address the optimization problem as follows.

First, we consider a system configuration, where the active service units are given, and model the users' association process as a game. The players of the game are the users that have to select a service unit among the active ones. We solve the game so that, for each user pattern, $\{t_u\}_{u\in\mathcal{U}}$, the corresponding NE strategy profiles can be identified; note that, given $\{t_u\}_{u\in\mathcal{U}}$, there may exist multiple NEs. Then, in order to evaluate the performance of the system configuration, we define an objective function, which accounts for the energy cost of the active units and for the users' satisfaction. Since, for a given $\{t_u\}_{u\in\mathcal{U}}$, different NEs are reached depending on the order of arrival of the users, we average the objective function first over all NEs corresponding to $\{t_u\}_{u\in\mathcal{U}}$, and then over all possible realizations of the users arrival process. Finally, we use the obtained result as an indication of the system configuration performance, and we select the system configuration that optimizes such an index.

Let us now detail the procedure outlined above. We denote the tagged system configuration by $\mathbf{x}$, and define $\mathcal{S}_u$ as the set of service units that can be selected by user $u \in \mathcal{U}$. Also, let $\mathcal{U}_s$ be the set of users that can select service unit $s \in \mathcal{S}$ and let $\mathcal{U}_{s,u}$ be the set of nodes $v \in \mathcal{U}_s \setminus u$ with $s \in \mathcal{S}_u$.

In the game, the action of the generic user (player) u consists in choosing one of the service units connected to her, e.g., $z_u = s$ with $s \in \mathcal{S}_u$. The payoff is $w_{us} \geq 0$ if unit s is active and not saturated otherwise it is $-\infty$. If no unit is chosen, $z_u = \emptyset$ and the payoff is $-\omega$, being ω a penalty value. More precisely:

$$\pi_u(z_u, \{z_v\}_{v\in\mathcal{U}_{s,u}}|t_u = 1) =$$
$$= \begin{cases} -\omega, & \text{if } z_u = \emptyset \\ w_{us}, & \text{if } z_u = s, \text{ and} \\ & w_{su} + \sum_{v\in\mathcal{U}_{s,u}} \delta(z_v, s)w_{sv} \leq c_s x_s \quad (1) \\ -\infty, & \text{if } z_u = s, \text{ and} \\ & w_{su} + \sum_{v\in\mathcal{U}_{s,u}} \delta(z_v, s)w_{sv} > c_s x_s \end{cases}$$

If instead user u is absent, $z_u = \emptyset$ is the only possible value and we set

$$\pi_u(z_u, \{z_v\}_{v\in\mathcal{U}_{s,u}}|t_u = 0) = \begin{cases} 0, & \text{if } z_u = \emptyset \\ -\infty, & \text{otherwise.} \end{cases}$$

Note that, at every perturbation in the system, e.g. due to the departure of a user, a player may decide to connect to another service unit than the currently selected one, if she can increase her payoff.

It is useful to represent the NE conditions in terms of *best-response relations*: a strategy profile $\mathbf{z}^* = (z_1^*, \ldots, z_N^*)$ is a pure NE if and only if, for each user u, z_u^* is the best response to the actions of the others, i.e.,

$$z_u^* = \arg\max_{z_u} \pi_u(z_u, \{z_v^*\}_{v\in\mathcal{U}_{s,u}}|t_u), \quad \forall u \in \mathcal{U}.$$

In principle, the weight given to each NE should depend on specific details of the dynamics of the users (e.g. on the order of arrival of the users and on the order according to which users unilaterally deviate from the current strategy profile if they find it convenient). Unfortunately these details are largely unknown in any realistic setup. It is thus worth considering a simplifying hypothesis in which all the NEs are weighted uniformly and the complex user dynamics is summarized in the average over the realizations of the stochastic parameters $\mathbf{t}$. In general we do not know which users are actually present in the system, but we assume to know the probability p_u that user u is present, $\forall u \in \mathcal{U}$.

The problem consists in optimizing the trade-off between the system energy cost and the expected users' satisfaction, i.e. in finding the configuration of active service units $\{x_s\}$ which maximizes the following objective function:

$$\mathcal{F}(\mathbf{x}) = \mathbb{E}_{\mathbf{t}}\left[\left\langle \sum_{u\in\mathcal{U}} \pi_u(z_u, \{z_v\}_{v\in\mathcal{U}_{s,u}}|t_u)\right\rangle\right] - \alpha \sum_s r_s x_s$$

where $\langle\cdots\rangle$ represents the average over the values of $\mathbf{z} \in \text{NASH}(\mathbf{x}, \mathbf{t})$ that satisfy the NE conditions (which depend implicitly on $\mathbf{x}$ and $\mathbf{t}$). The objective function is composed of two contrasting terms: a first contribution that measures the overall quality of the service, and a second term that quantifies the total cost of active service units (alternatively, the service provider's revenue could depend explicitly on the perceived quality of the service). The parameter α is used to set the relative importance of the two objectives.

We can finally formulate the optimization goal of the central controller which is, given $G = (\mathcal{U}, \mathcal{S}, \mathcal{E})$, the vector of unit capacities $\mathbf{c}$, the payoff matrix $\mathbf{W}$, the vector of presence probabilities $\mathbf{t}$, and the parameter α, to find a minimizing $\mathbf{x}^*$

such that $\mathcal{F}(\mathbf{x}^*) = \min_\mathbf{x} \mathcal{F}(\mathbf{x})$. In conclusion, the vector $\mathbf{x}^*$ represents the system configuration that corresponds to the best tradeoff between the system energy cost and the user satisfaction.

IV. PROBLEM SOLUTION

The NE conditions define a set of local hard constraints on the individual actions, such that finding a pure NE can be translated into finding a solution of a Constraint-Satisfaction Problem (CSP).

Using the node variables $\{z_u\}_{u \in \mathcal{U}}$, we can formulate the associate CSP over a factor graph with many small loops even when the original graph had none, which is not appropriate to develop a solution algorithm based on message passing [12]. In the following we switch to an equivalent representation, using edge variables, that is much more convenient for message passing applications.

A. CSP Representation Using Edge Variables

The actions of the users can be described using three-states variables y_{us} defined on the undirected edges $(u, s) \in E$ (see Figure 1)

$$y_{us} = \begin{cases} -1, & \text{if } s \text{ is inactive or saturated for } u \\ 0, & \text{if } s \text{ is available but not used by } u \quad (2) \\ 1, & \text{if } s \text{ is used by } u \end{cases}$$

where "saturated for u" refers to the case in which if u connects to s, the latter violates the capacity constraint, while "available" refers to the case where s is active and able to accommodate user u. The NEs are the configurations taken by the variables $\{y_{us}\}$ that satisfy the following set of constraints. First, users cannot have access to more than one service unit at the same time: $\sum_{s \in \mathcal{S}_u} \mathbb{1}[y_{us} = 1] \leq t_u \quad (\forall u \in \mathcal{U})$. Second, the capacity of each service unit cannot be exceeded: $\sum_u \mathbb{1}[y_{us} = 1] w_{su} \leq c_s x_s \quad (\forall u \in \mathcal{U}_s, s \in \mathcal{S})$. And third, users try to use the best service unit available: $\{y_{us} = 1, y_{us'} = 0\} \Rightarrow \{w_{us} \geq w_{us'}\} \quad \forall (s, s' \in \mathcal{S}_u, u \in \mathcal{U})$.

The stochastic optimization problem consists in finding the configuration of active service units $\mathbf{x}$ such that it maximizes the objective function

$$\mathcal{F}(\mathbf{x}) = \mathbb{E}_\mathbf{t}\left[\left\langle \sum_{(u,s) \in \mathcal{E}} \mathbb{1}_{[y_{us}=1]} w_{us} \right\rangle\right] - \alpha \sum_s r_s x_s. \quad (3)$$

The most difficult part of this optimization problem is that of performing the average over the NEs in the presence of stochasticity, that is the essential step to be able to evaluate the average costs and benefits from activating/deactivating different service units. Once this is done, the optimization step over the $\{x_s\}$ becomes trivial and it can be done either exhaustively or by means of decimation methods. In the next section we describe an approximate method to perform the average over the NEs and the stochastic parameters in a computationally efficient way.

B. Average over NEs with Stochastic Parameters

In general one should first average over the pure NEs at fixed realization $\mathbf{t} = \{t_u\}$ of the stochastic parameters and then perform the average over the distribution $P(\mathbf{t}) = \prod_i P(t_i)$ of the latter. The double average is extremely costly
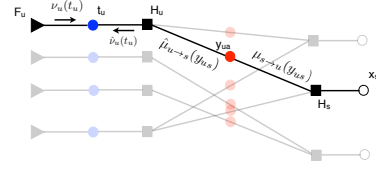

Fig. 1. Factor graph representation of Eqs. (6)-(9) involving the four types of messages. The use of edge variables $\{y_{us}\}$ highly simplifies the representation and allows to get rid of small loops. The other two types of variable nodes are blue and empty nodes, corresponding respectively to the stochastic parameters $\{t_u\}$ and the service provider's variables $\{x_s\}$. Square nodes $\{H_u\}$ and $\{H_s\}$ are standard factor nodes containing the capacity constraints and the best-response conditions.

at a computational level. The message passing approach allows one to perform these two steps together although at the cost of introducing an approximation in the computation.

For an observable $O(\mathbf{y}, \mathbf{x}, \mathbf{t})$, the average over NEs is

$$\begin{aligned}
\mathcal{O}(\mathbf{x}) &= \sum_{\mathbf{y}, \mathbf{t}} P(\mathbf{y}, \mathbf{t} | \mathbf{x}) O(\mathbf{y}, \mathbf{x}, \mathbf{t}) \\
&= \sum_\mathbf{y} \sum_\mathbf{t} P(\mathbf{t}) P(\mathbf{y} | \mathbf{t}, \mathbf{x}) O(\mathbf{y}, \mathbf{x}, \mathbf{t}) \\
&= \sum_\mathbf{y} \sum_\mathbf{t} P(\mathbf{t}) \frac{\mathbb{1}[\mathbf{y} \in \text{NASH}(\mathbf{x}, \mathbf{t})]}{Z(\mathbf{x}, \mathbf{t})} O(\mathbf{y}, \mathbf{x}, \mathbf{t}) \,(4)
\end{aligned}$$

in which $Z(\mathbf{x}, \mathbf{t}) = \sum_\mathbf{y} \mathbb{1}[\mathbf{y} \in \text{NASH}(\mathbf{x}, \mathbf{t})]$. The numerator can be easily expressed in terms of the local constraints for the edge variables $\{y_{us}\}$. If we call $I_s(\{y_{vs}\}_{v \in \mathcal{U}_s} | x_s)$ the hard constraint defined on node $s \in \mathcal{S}$ and $I_u(\{y_{us}\}_{s \in \mathcal{S}_u} | t_u)$ that defined on $u \in \mathcal{U}$, we have $\mathbb{1}[\mathbf{y} \in \text{NASH}(\mathbf{x}, \mathbf{t})] = \prod_u I_u(\{y_{us}\}_{s \in \mathcal{S}_u} | t_u) \prod_s I_s(\{y_{vs}\}_{v \in \mathcal{U}_s} | x_s)$. The main difficulty of performing the quenched average is due to the presence of the normalization factor $Z(\mathbf{x}, \mathbf{t})$ at the denominator of (4). A mean-field approximation, based on the factorization ansatz $Z(\mathbf{x}, \mathbf{t}) = \prod_u Z_u(\mathbf{x}, t_u)$, can be used to transform our quenched average into an easily computable annealed one. In this approximation we get

$$\begin{aligned}
\log P(\mathbf{y}, \mathbf{t} | \mathbf{x}) &= \sum_u \log P(t_u) - \sum_u \log Z_u(t_u) \\
&+ \sum_u \log I_u(\{y_{us}\}_{s \in \mathcal{S}_u} | t_u) + \sum_s \log I_s(\{y_{us}\}_{u \in \mathcal{U}_s} | x_s).
\end{aligned}$$
(5)

The factor graph associated to the problem is shown in Figure 1. In addition to the usual terms $H_u = \log I_u(\{y_{us}\}_{s \in \mathcal{S}_u} | t_u)$ and $H_s = \log I_s(\{y_{us}\}_{u \in \mathcal{U}_s} | x_s)$, corresponding to hard constraints, it also contains energetic terms $\log Z_u(\mathbf{x}, t_u)$ on the nodes $u \in \mathcal{U}$. The energetic terms are unknown but can be computed implicitly introducing a new set of messages $\{\nu_u(t_u)\}_{u \in \mathcal{U}}$ and $\{\hat{\nu}_u(t_u)\}_{u \in \mathcal{U}}$ that must be adjusted in order to have the correct probability marginal $P(t_u)$ on each variable node t_u. On such a factor graph, it is possible to derive the set of message passing equations (6–9) by means of which the observable $\mathcal{O}(\mathbf{x})$ can be approximately evaluated. The proportionality symbol means that the marginal probabilities need to be correctly normalized. A complete description of the method will be presented elsewhere [13].

$$\hat{\mu}_{u \to s}(y_{us}) \quad \propto \quad \sum_{t_u} \nu_u(t_u) \sum_{\{y_{us'}\}, s' \in \mathcal{S}_u \backslash s} I_u(y_{us}, \{y_{us'}\}|t_u) \prod_{s' \in \mathcal{S}_u \backslash s} \mu_{s' \to u}(y_{us'}) \tag{6}$$

$$\mu_{s \to u}(y_{us}) \quad \propto \quad \sum_{\{y_{vs}\}, v \in \mathcal{U}_{s,u}} I_s(y_{us}, \{y_{vs}\}|x_s) \prod_{v \in \mathcal{U}_{s,u}} \hat{\mu}_{v \to s}(y_{vs}) \tag{7}$$

$$\hat{\nu}_u(t_u) \quad \propto \quad \sum_{\{y_{us}\}, s \in \mathcal{S}_u} I_u(\{y_{us}, s \in \mathcal{S}_u\}|t_u) \prod_{s \in \mathcal{S}_u} \mu_{s \to u}(y_{us}) \tag{8}$$

$$\nu_u(t_u) \quad \propto \quad P(t_u)\hat{\nu}_u^{-1}(t_u). \tag{9}$$

TABLE I
SCENARIOS CONSIDERED IN THE SIMULATIONS

Scenario	S1	S2	S3
Number of instances	1842	91	1
Number of users for each instance N_u	12	300	1000
Number of service units N_s	4, 8, 12	30, 60, 90	50
Connectivity of users k_u	2, 3, 4	3, 5, 8	5
Capacity of service units c_s	5, 8, 11	5, 10, 15	20
Maximum weight $w_{\max}$	10	10	15
Penalty for a disconnected user ω	10	10	5

V. NUMERICAL RESULTS ON RANDOM GRAPHS

In this section, we present some numerical results obtained by our algorithm on random graphs generated with the following procedure. Both the users and the service units are placed at random in the unit square of the two-dimensional euclidean space. For each user u, only the k nearest service units are assumed to be accessible, and, for each of these, the workload is $w_{su} = \lceil \gamma\, d(u,s)^2 \rceil$, i.e., an integer proportional to the square of the distance between s and u (the proportionality constant γ is such that the maximum weight is equal to a specified value $w_{\max}$). Recall that the payoff for a disconnected user is $-\omega$ (see (1)), whereas for connected users is $w_{us} = w_{\max} - w_{su}$. Finally, the presence probabilities $p_u = \mathbb{P}[t_u = 1]$ are extracted uniformly in $(0,1)$. We have considered four scenarios, whose parameters are reported in Table I; the ranges for all these parameters are such that the instances are non trivial.

A. Comparison with exhaustive enumeration

As a first test of our message passing approach, in scenario S1 we compared it to an exhaustive enumeration of the NEs for fixed $\mathbf{t}$, averaging the results over a sample of values of $\mathbf{t}$. More specifically, we considered (for a given configuration of the service units $\mathbf{x}$) the following two observables, in terms of which one can compute the objective function we propose to use for the greedy procedure:

$$\mathcal{W}(\mathbf{x}) = \mathbb{E}_{\mathbf{t}} \left[\left\langle \sum_{(u,s) \in \mathcal{E}} \mathbb{1}_{[y_{us}=1]} w_{su} \right\rangle \right] \tag{10}$$

which is the average (over the realization of $\mathbf{t}$ and over the NEs) of the sum of the workloads w_{su} for the users that are

present and connected to some service unit, and

$$\mathcal{N}(\mathbf{x}) = \mathbb{E}_{\mathbf{t}} \left[\left\langle \sum_{u \in \mathcal{U}} \prod_{s \in \partial u} \mathbb{1}_{[y_{us}=-1]} \right\rangle \right] \tag{11}$$

which is the average (again, over the realization of $\mathbf{t}$ and over the NEs) of the number of disconnected users. We compare the value obtained by our algorithm for $\mathcal{W}(\mathbf{x})$ with

$$\bar{\mathcal{W}}(\mathbf{x}) = \frac{1}{|\mathcal{T}|} \sum_{\mathbf{t} \in \mathcal{T}} W(\mathbf{x}, \mathbf{t}) \tag{12}$$

where $\mathcal{T}$ is a random sample of realizations of $\mathbf{t}$ extracted from $P(\mathbf{t})$ (and $|\mathcal{T}|$ is the size of the sample) and where $W(\mathbf{x}, \mathbf{t})$ is the average over the NEs (for fixed $\mathbf{x}$ and $\mathbf{t}$) of the sum of the workloads for connected users, which is computed with an exhaustive enumeration of the possible allocations $\mathbf{y}$. A similar comparison is done for $\mathcal{N}(\mathbf{x})$. Of course, in this scenario the number of users is limited since the exhaustive enumeration is possible only if the size of the instance is very small.

Scatter plots A and B in Fig. 2 compare our algorithm with the exhaustive enumeration under scenario S1. As the sample size $S = |\mathcal{T}|$ increases, the data points tend to collapse onto the diagonal, i.e., as the accuracy of the sampling procedure improves, the results obtained by sampling tend to those obtained with the message passing algorithm, except for a small number of "outliers" (less than one percent of the instances). This confirms that, even on very small instances, the two hypotheses on which our method is based, namely the decorrelation assumption of the cavity method and the factorization hypothesis for the partition function $Z(\mathbf{t})$, are a good approximation.

B. Comparison with explicit sampling

In the next scenario S2, we compare the results obtained by our algorithm with those obtained by computing the average over the NEs (for fixed $\mathbf{t}$) with BP, and then averaging over $\mathbf{t}$ with an explicit sampling. This allows us to test, on larger instances, the factorization assumption for the partition function $Z(\mathbf{t})$. Note that our algorithm requires only one convergence of the message passing procedure to perform both averages. The explicit sampling, instead, requires S convergences of a message passing, which is (almost) as complex as ours; thus, it is roughly slower by a factor S. This limits the number of instances that we have been able to analyze to less than 100.

Again, scatter plots C and D of Fig. 2 show that, as the sample size S increases, the data points tend to collapse
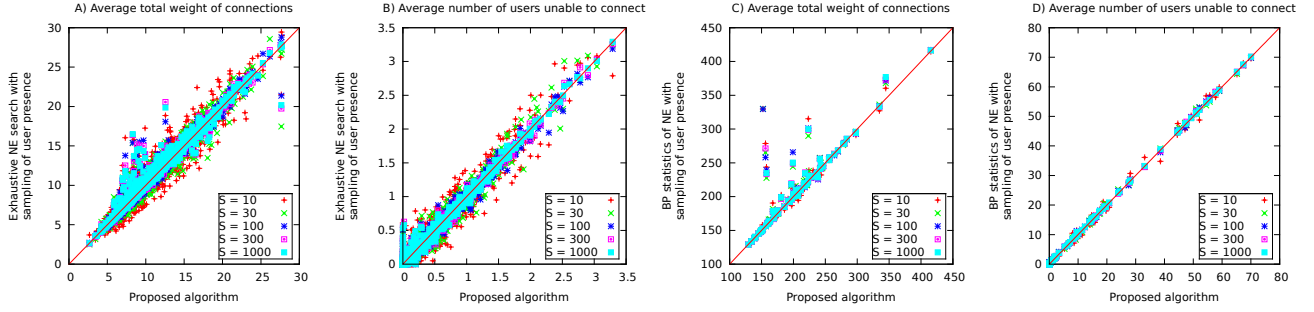
Fig. 2. Comparison between the values of $\mathcal{W}(\mathbf{x})$ (plots A and C for scenario S1 and S2, respectively) and $\mathcal{N}(\mathbf{x})$ (plots B and D for scenarios S1 and S2, respectively), as computed by our message passing algorithm ("Mirror") and by either exhaustive enumeration of the NEs (plots A and B) or BP sampling of the NEs (for fixed $\mathbf{t}$) with explicit sampling over $\mathbf{t}$ (plots C and D). Each data point corresponds to one instance and one sample size S (corresponding to the different symbols/colors).

onto the diagonal, with the exception of a few cases for the estimation of $\mathcal{W}(\mathbf{x})$.

C. Optimization results

Finally, in scenario S3 we provide an example of optimization. We used our greedy decimation heuristic based on the message passing algorithm for a single instance. The heuristic we use to find the optimal allocation $\mathbf{x}$ is the following. We start by computing the value of the objective function

$$\mathcal{O}(\mathbf{x}) = \mathbb{E}_{\mathbf{t}}\left[\left\langle \sum_{(u,s)\in\mathcal{E}} \mathbb{1}_{[y_{us}=1]} w_{us} \right\rangle\right] \quad (13)$$

when all the service units are on (i.e. $x_s = 1$ for each s). Then, we compute the same objective function for all the configurations obtained by switching off one service unit. We actually switch off the service unit that corresponds to the smallest drop in the objective function. The same procedure is then iterated, computing the variations in the objective function associated to switching off each of the service units that are still on, and actually switching off the one that minimizes the drop, until all the service units are off (or we decide to stop).

The results of this "greedy decimation" are shown in Fig. 3. We observe that during the first 8 steps of the decimation (i.e. as we switch off the first 8 service units) the value of the objective function decreases very modestly (dropping by 0.18% overall), while for larger number of steps the drops are much greater. We therefore decide to stop the decimation after 8 steps. This allows to switch-off 16% of the service units (i.e. to save 16% of the electric power) without affecting at all the service level.

VI. CONCLUSION

In this paper, we presented a novel computationally efficient optimization approach for distributed resource allocation problems under user behavior uncertainty. We propose a belief propagation scheme to compute the costs of different service configurations. This is obtained by averaging over all the possible Nash equilibrium points associated to a given system configuration.
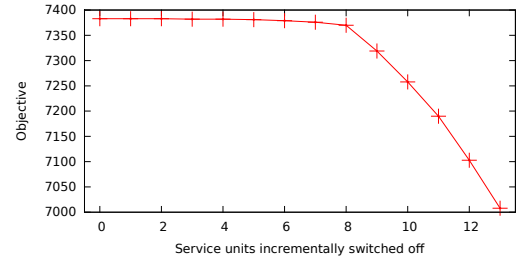

Fig. 3. The results of the greedy decimation. The labels on the abscissa show the number of service units being switched off at each step of the decimation. The last five steps of the decimation are discarded from the solution.

REFERENCES

[1] B. Vocking. Selfish load balancing. In *Algorithmic game theory*. Cambridge Univ. Press, N. Nisan et al. Eds. (2007).

[2] R. W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *Int. Journal of Game Theory*, **2**, 65-67 (1973)

[3] I. Milchtaich. Congestion games with player-specific payoff function. *Games and Economic Behavior*, **13**, 111-124 (1996).

[4] E. Koutsoupias and C. H. Papadimitriou. Worst-case equilibria. *Symp. on Theoretical Aspects of Computer Science* (1999).

[5] E. Even-Dar, A. Kesselman, and Y. Mansour. Convergence time to Nash equilibria. In *Proc. 30th International Colloq. on Automata, Languages and Programming*, pp. 502-513 (2003).

[6] D. Fotakis, et al. The structure and complexity of Nash equilibria for a selfish routing game. 29th ICALP, 123-134 (2002).

[7] A. Goel and P. Indyk. Stochastic Load Balancing and Related Problems. *Symp. on Foundations of Computer Science* (1999).

[8] J. Kleinberg, Y. Rabani, and E. Tardos. Allocating bandwidth for bursty connections. *Proc. 29th ACM Symposium on Theory of Computing* (1997).

[9] E. Nikolova and N. E. Stier-Moses. Stochastic selfish routing. In SAGT (2011).

[10] S. Dye, L. Stougie, and A. Tomasgard. The stochastic single resource service-provision problem. *Naval Research Logistics* **50**(8), 869-887 (2003).

[11] F. Altarelli, A. Braunstein, A. Ramezanpour, and R. Zecchina, Stochastic Matching Problem, *Phys. Rev. Lett.* **106** (2011)

[12] M. Mézard and A. Montanari *Information, Physics and Computation*. Oxford graduate texts (2009).

[13] F. Altarelli, A. Braunstein, L. Dall'Asta and R. Zecchina, *in preparation* (2013).