# Decentralized Control for Power Distribution with Ancillary Lines in the Smart Grid

Michele Roncalli and Alessandro Farinelli(✉)

Computer Science Department, University of Verona, Verona, Italy
alessandro.farinelli@univr.it

**Abstract.** Energy management is a key topic for today's society, and a crucial challenge is to shift from a production system based on fossil fuel to sustainable energy. A key ingredients for this important step is the use of a highly automated power delivery network, where intelligent devices can communicate and collaborate to optimize energy management.

This paper investigates a specific model for smart power grids initially proposed by Zdeborov and colleagues [12] where back up power lines connect a subset of loads to generators so to meet the demand of the whole network. Specifically, we extend such model to minimize $CO_2$ emissions related to energy production.

In more detail, we propose a formalization for this problem based on the Distributed Constraint Optimization Problem (DCOP) framework and a solution approach based on the min-sum algorithm. We empirically evaluate our approach on a set of benchmarking power grid instances comparing our proposed solution to simulated annealing. Our results, shows that min-sum favorably compares with simulated annealing and it represents a promising solution method for this model.

**Keywords:** Smart grid · Decentralized constraint optimization · Factor graphs

## 1 Introduction

Energy management is a key topic for today's society, and a crucial challenge that governments and societies are facing is to shift from a production system based on fossil fuel to sustainable energy. A key point for sustainable energy is the use of renewable energy sources such as solar, wind and tidal power, biomass, geothermal energy etc. Many of the renewable energy sources (e.g., solar power, wind power and biomass) can be exploited in a decentralized fashion, dramatically changing the current centralised production system. More specifically, decentralized energy is energy generated near the point of use, and is typically produced by small generating plants connected to a local network distribution rather than to the high-voltage transmission system required by the centralised energy production scheme. Technologies for producing decentralized energy are already mature enough for large scale deployment, and in many countries, such as Finland, the Netherlands and Denmark, a significant percentage

E. Sucar et al. (Eds.): AFI 2016, LNICST 179, pp. 39–50, 2017.
DOI: 10.1007/978-3-319-49622-1_6

of the national electricity production is provided through decentralized energy (respectively 35%, 40% and 50%).

In this context, the vision of an *intelligent electricity delivery network*, commonly called **smart grid**, has been advocated as a key element to achieve decentralized sustainable energy provisioning. The smart grid is a highly automated distribution network that incorporates many different devices, such as smart meters and smart generators. A key element to realize the long term vision of the smart grid is the development of proper ICT infrastructures that enable data transfer and interoperability among all the core components of the smart grid. In this perspective, the internet of things (IoT) provides crucial enabling technologies for the smart grid by proposing a clear set of standard and effective communication protocols that foster the interoperability between different devices [5].

The long term goal of the smart grid is to exploit such ICT infrastructure to optimize the energy management and distribution process, hence minimising carbon emissions and reducing costs to generate electricity. In this perspective a crucial topic is to avoid the overloads of power generation units that may be caused by the fluctuation in demand and by the intermittent generation typical of renewable technologies (e.g., wind).

Within this framework, we will study a specific model for power grids initially proposed by Zdeborov and colleagues [12] where back up power lines (called ancillary lines) connect a subset of loads to several generators. Such loads can then choose which generator to use to meet their demand. In the model proposed in [12] authors focus on a *satisfaction problem*, i.e., the solution is a configuration of loads (i.e., a mapping from load to generators) where no generator in the grid is overloaded (i.e., the demand of the connected load does not exceed the generator maximum production). Figure 1 shows an exemplar situation where 6 loads ($L_{1,1}, \cdots, L_{2,3}$) are connected to 2 generators ($G_1, G_2$). Dashed arrows represents ancillary lines that could be used by a subset of the loads (i.e., $\{L_{1,1}, L_{1,2}, L_{2,1}, L_{2,3}\}$).

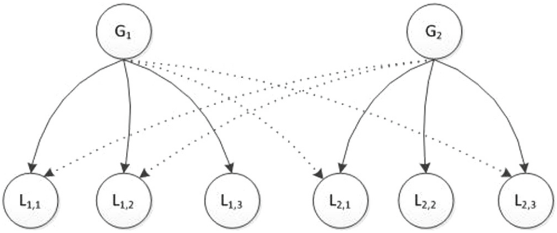


**Fig. 1.** Diagram depicting a power grid with 2 generators, 6 loads and 4 ancillary lines.

Here, we extend such model by considering that the $CO_2$ emissions of the generators depend on their production level. Thus, the problem becomes a constrained *optimization problem*, where the goal is to find the configuration of loads that minimizes the total $CO_2$ emissions in the grid ensuring that no generator is overloaded.

Following the distributed energy management paradigm that underpins the smart grid vision, we propose a decentralized solution approach for this problem. Specifically, we represent the problem discussed above as a Distributed Constraint Optimization Problem (DCOP) which is a widely used framework for decentralized decision making [9]. The DCOP literature offers a wealth of solution approaches ranging from exact algorithms [9,10] to heuristic techniques [3,7,13]. While exact algorithms are guaranteed to return the optimal solution, they all suffer from an exponential element in the computation (and message size/number) that hinders their applicability for the large scale scenarios we consider here (i.e., thousands of variables). In contrast, while heuristic approaches can not guarantee the quality of the retrieved solutions, they have been successfully used in several scenarios [2,4]. Hence, here we employ a standard heuristic approach (i.e., the max-sum algorithm [3]) that has been shown to provide high quality solutions in various application domains. In more detail, following previous work [3,11] we represent the DCOP problem by using factor graphs [6] and we run a close variation of the max-sum algorithm, i.e., the min-sum as we address a minimization problem.

We empirically evaluate our approach on a set of benchmarking power grid problems, created with the procedure proposed in [12]. Results obtained over a wide range of problem instance have been analyzed by considering different measures such the final cost (i.e., $CO_2$ emissions), steps required to converge to a solution and total run time. Our proposed approach has been compared to simulated annealing, a well known centralized method for optimization. Overall the empirical analysis shows that the min-sum algorithm can solve large instances (i.e., up to 20000 generators) in seconds and that it provides a significantly smaller cost when compared to simulated annealing, hence being a promising approach for our model.

## 2 Background

In this section we discuss necessary background detailing the DCOP formalism, factor graphs and the max-sum approach.

### 2.1 Distributed Constraint Optimization Problems

Distributed constraint optimization problems ($DCOP$) are a generalization of COP for distributed frameworks. A DCOP is a tuple $\langle \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{R} \rangle$, where $\mathcal{A} = \{a_1, \dots, a_s\}$ is a set of agents and $\mathcal{X} = \{x_1, \dots, x_n\}$ is a set of variables, each variable $x_i$ is owned by exactly one agent $a_i$, but an agent can potentially own more than one variable. The agent $a_i$ is responsible for assigning values to the variables it owns. $\mathcal{D} = \{D_1, \cdots, D_n\}$ is a set of discrete and finite variable domains, and each variable $x_i$ can take values in the domain $D_i$. Then, $\mathcal{R} = \{r_1, \dots, r_m\}$ is a set of cost functions that describe the constraints among variables. Each cost function $r_i : D_{i_1} \times \cdots \times D_{i_{k_i}} \rightarrow \Re \cup \{\infty\}$ depends on a set of variables $\mathbf{x_i} \subseteq \mathcal{X}$, where $k_i = |\mathbf{x_i}|$ is the arity of the function and $\infty$ is used to represent hard constraints. Each cost function assigns a real value to each possible valid assignment of the variables it depends on and $\infty$ to non valid assignments.

For a minimization problem, the goal is then to find a variable assignment that minimizes the sum of cost functions:

$$\arg\max_{\mathbf{x}} \sum_i r_i(\mathbf{x_i}) \tag{1}$$

### 2.2 Factor Graphs and Min-Sum

A *Factor Graph* is a bipartite graph that encodes a factored function, e.g. functions that can be expressed as a sum of components, such as the function reported in Eq. 1. A factor graph has a *variable node* for each variable $x_i$, a *factor node* $F_j$ for each local function $r_j$, and an edge connecting variable node $x_i$ to factor node $r_j$ if and only if $x_i$ is an argument of $r_j$.

Factor graphs represents a very convenient computational framework for several optimization techniques such as max-sum, max-prod, and the min-sum algorithm that we use in this work.

In more detail, the min-sum algorithm belongs to the Generalized Distributive Law (GDL) framework [1], a family of techniques frequently used to solve probabilistic graphical models (e.g. to find the maximum a posteriori assignment in Markov random fields or compute the posterior probabilities) [2]. If applied to constraint networks in tree form, min-sum provides the optimal solution, but when applied to more general networks (i.e. networks which contain loops) optimality (and convergence) can be no longer ensured. However, empirical evidence shows that GDL-based algorithms are able to find solution very close to the optimal in several problems.

Min-sum operates directly on a factor graph representation of the problem iteratively exchanging messages between variable nodes and function nodes. The key idea in the algorithm is that new messages are computed and passed between the nodes in the graph until a stop condition is verified. There can be many convergence criteria (e.g. messages convergence, solution convergence, etc.). In our approach we focused on the messages convergence, as our tests revealed this criterion to be the best choice for our problem scenario. Moreover, to deal with not satisfiable instances we consider a maximum number of message computation steps, if this maximum number of steps is reached the algorithm states that it could not find a valid solution.

## 3 Problem Formulation

In this section we first detail the model for controlling the power distribution with ancillary lines proposed in [12], then we present our proposed extension and our factor graph formalization of such problem.

### 3.1 Model for Power Distribution with Ancillary Lines

The model for power distribution proposed in [12] is composed by two types of elements: a set of $M$ *generators* and a set of *loads*. A generator $G_i$ has a fixed maximum energy production value and without loss of generality, we normalize this to 1, thus no more than one unity of energy can be absorbed by all the loads connected to $G_i$.

A load is a component of the grid that is not able to produce energy for itself or for other nodes in the network. The power consumption rate varies across the loads, but for each load it is a constant that can not be controlled. This value is drawn from a uniform probability distribution with support on the interval $(0, 1)$. The center of the distribution is set to $\bar{x}$ (that represents the *mean value* and $x$ is the consumption rate). The width of the distribution is $\Delta$. Thus, given a load $l_j$, its power consumption rate is a value drawn uniformly in the set $[\bar{x} - \frac{\Delta}{2}, \bar{x} + \frac{\Delta}{2}]$. This power can be absorbed by only one generator at time, i.e. if a load is connected to several generators, only one link can be active. Each generator is connected to $D$ distinct loads, thus the total number of loads is $(M \cdot D)$.

A power grid is then a graph forest composed by $M$ trees. Among this forest, $R$ *ancillary lines* for each generators are added. In more detail, $(R \cdot M)$ new links are created to interconnect the grid. These links are added in two simple steps: (1) $R$ loads are chosen from each generator; (2) for each load chosen in the first step, connect it to another generator in such a way that, at the end of the process, every generator is connected to $(D + R)$ loads.

The final result is a bipartite graph with the following properties: (i) $M$ nodes corresponding to the generator set; (ii) $M \cdot D$ nodes corresponding to the load set; (iii) $M \cdot (D - R)$ loads are connected to only one generator; (iv) $(M \cdot R)$ loads are connected to two generators; (v) every generator is connected to $(D + R)$ loads; (vi) no load in the net is connected to more than 2 generators. Hence the free parameters that define an instance of the power grid model are $M$, $D$ and $R$.

In Fig. 1 shows an exemplar instance of a power grid when $M = 2$, $D = 3$ and $R = 2$. Loads $L_{1,3}$ and $L_{2,2}$ are single connected, so they can use only the generator they are connected to (respectively generator $G_1$ and generator $G_2$). The remaining loads ($L_{1,1}$,$L_{1,2}$,$L_{2,1}$ and $L_{2,3}$) are connected to both the generators, so they can use energy from $G_1$ or from $G_2$ (never concurrently).

### 3.2 Formalization of $CO_2$ emissions in the power grid

As mentioned before, Zdeborov and colleagues in [12] focus on a *satisfaction problem*, aiming to find a mapping from loads to generators, where no generator in the grid is overloaded.

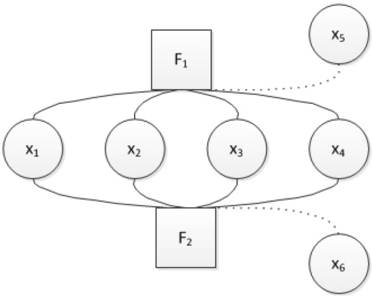
Here, we extended their model considering the $CO_2$ emissions for each generator and aiming to minimize the total emission for the grid.

Specifically, following [8], the $CO_2$ emission function is proportional to the energy produced by the generators. In more detail, the $CO_2$ emission function is: $CO_2 = mult \cdot energy$ where *energy* is the ratio of energy that the generator must produce with respect to its maximum energy production value (which as mentioned above is set to 1); *mult* represent a feature of the generator, that expresses the unit of $CO_2$ emitted by the generator for each unit of energy produced. In our experiments we choose *mult* randomly in a range $[1, 5]$.

### 3.3 From Power Grid to Factor Graph

Our factor graph model of a power grid with ancillary lines consider a variable node for each load $L_i$ and a factor node for each generator $G_j$. In more detail, the variable node $x_i$, that corresponds to load $L_i$, has a domain that contains a value for each possible generator $G_j$ which the load $L_i$ can connect to. Moreover, the scope for the factor node $F_j$, corresponding to generator $G_j$, is the set of variable nodes $\{x_i, \ldots, x_k\}$ that correspond to loads that can get power from $G_j$.

For example, consider the power grid in Fig. 1. The corresponding factor graph is shown in Fig. 2 (left), while the correspondence between nodes and generators/loads is summarized in Fig. 2 (right).



| load or generator | node in the factor graph |
|---|---|
| $G_1$ | $F_1$ |
| $G_2$ | $F_2$ |
| $L_{1,1}$ | $x_1$ |
| $L_{1,2}$ | $x_2$ |
| $L_{2,1}$ | $x_3$ |
| $L_{2,3}$ | $x_4$ |
| $L_{1,3}$ | $x_5$ |
| $L_{2,2}$ | $x_6$ |

**Fig. 2.** Left: example of factor graph for the power grid in Fig. 1. Right: correspondence between nodes and generators

Consider the dotted edge from $x_5$ to $F_1$ and from $x_6$ to $F_2$. In a *naïve* model, $x_5$ would be connected to $F_1$ and $x_6$ to $F_2$ but, since their domains contain only one value, they can be safely removed from $F_1$ and $F_2$. In practice, variable with a single value can be considered meaningless because its value is already known. Specifically, $x_5$ is removed from $F_1$, which is modified to consider the consumption of the load represented in $x_5$ for any possible value of its other arguments. Since $x_5$ can be connected only to $F_1$, it means that for every possible value of $x_1$, $x_2$, $x_3$ and $x_4$ the total energy that generator $G_1$ can produce is $(1 -$ *energy required by* $L_{1,3})$.

## 4 Results and Discussion

In this section we first detail our empirical methodology and then discuss obtained results.

### 4.1 Empirical Methodology

Our goal is to empirically evaluate the proposed min-sum approach in power grid instances that have the same characteristics as the benchmarking test suite proposed in [12]. In that work authors use two different algorithms (i.e., walkgrid and belief propagation), but since here we extend their model considering an optimization problem (and not satisfaction) we do not compare min-sum to such algorithms but instead use a simulated annealing procedure. Simulated annealing is a well known, powerful approach for finding a global minimum of a cost function.

To create the power grid instances, we follow the procedure described in [12], fixing $D = 3$ $R = 2$ and $\Delta = 0.2$. Then we vary $M \in \{200, 1000, 2000, 10000, 20000\}$ and $\bar{x}$ between 0.29 and 0.3 with a step of 0.01.

With this parametrization, the factor graph obtained from a power grid with $M$ generators, has $M$ function nodes and $(M \cdot R)$ variable nodes (as described in Sect. 3.3, only loads connected to two generators are mapped to variable nodes). Hence, at each iteration of the min-sum algorithm $(2R \cdot M)$ messages from variables to functions and $(2R \cdot M)$ messages from functions to nodes are sent.

We create 100 different power grid instances for all possible values of input parameters. For all instances the consumption rate $x$ of each load is drawn from the uniform probability distribution centered in $\bar{x}$ and with width set to $\Delta$.

Our metrics to evaluate the solution techniques are the final cost (the lower the better), the total run time and the amount of iterations required to get a response. The halting condition for min-sum is either message convergence or a maximum number of iterations (set to 300).

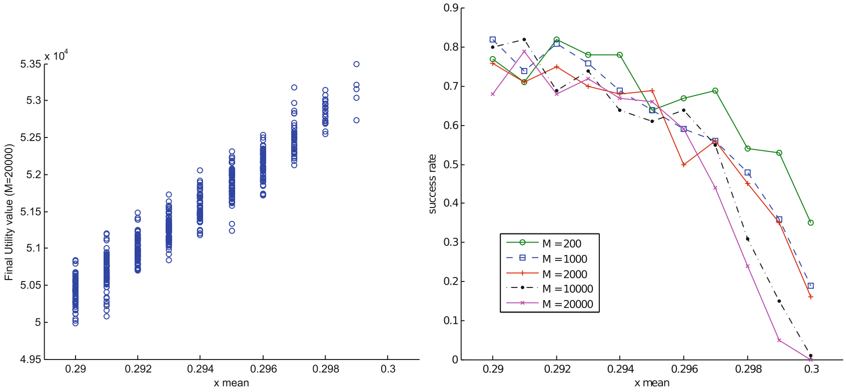


**Fig. 3.** Left: cost values varying $\bar{x}$ and with $M = 20000$, Rigth: success rate varying both $\bar{x}$ and $M$

### 4.2 Results

Figure 3 (left) reports the final cost value when $M = 20000$ (i.e., the biggest value for $M$) varying the value of $\bar{x}$. The graph reports the obtained cost value for all repetitions that result in a valid configuration (i.e., no generator is overloaded given the min-sum mapping from loads to generator). The graph shows that when the values of $\bar{x}$ increases the number of valid solution decreases (i.e., when $\bar{x} = 0.299$ the number of points in the graph is significantly less than the number of points when $\bar{x} = 0.29$). Moreover, there are no points when $\bar{x} = 0.3$, thus there are no solution for the 100 power grid problems created for the maximum value available for $\bar{x}$. Notice that, there is no guarantee that if min-sum is not able to find a solution, then the problem is not-satisfiable. However, bigger values of $\bar{x}$ result in power grid instances where loads generally require more energy. Hence it is more likely that there is no valid configuration for the power grid, this is confirmed by results obtained with simulated annealing (not reported here in the interest of space).

Figure 3 (right) reports the success rates varying both $M$ and $\bar{x}$, where the success rate is number of valid solutions/total number of instances. As mentioned before, the success rate heavily depends on $\bar{x}$, however, there is also a dependency with respect to $M$ when large values of $\bar{x}$ are used (approximately $\bar{x} >= 0.28$). This happens because bigger values of $M$ result in bigger power grids, where the probability of having at least on generator overloaded increases. Thus, when $\bar{x} = 0.3$ the power grid is not-satisfiable most likely when $M = 20000$ then when $M = 200$. Moreover, $M$ has a strong influence on the absolute value of the final cost (i.e., $CO_2$ emissions) because, when all other parameters are fixed, bigger networks will generate more energy and hence create more $CO_2$ emissions. Hence to analyse the quality of the solution returned by min-sum with for we report in Fig. 4 the final cost value normalized with respect to the $M$ parameter (here, and in the following graphs, the error bars represent the confidence interval of a t-test with 95% accuracy). This graph exhibits a similar trend w.r.t the one in Fig. 3. Specifically, the graph shows that while the quality of the optimal solution is strongly correlated with the growth of $\bar{x}$, the min-sum algorithm is able to provide solutions of good quality for large scale systems.

Figure 5 (left) reports the run time (in milliseconds) for min-sum to finish (either finding a solution or stating that the problem is not-satisfiable). As shown in the graph, $\bar{x}$ influences the time used. This happens because for smaller values of $\bar{x}$ it is usually easier to find a solution, since there is a smaller probability to have generators overloaded. Moreover, when the problem is *harder* (i.e., the loads mean consumption is higher), min-sum require more iterations to converge, hence the growth of time shown in the graph. When $M = 10000$ and $M = 20000$, from $\bar{x} = 0.298$ time peaks to its maximum value: this is the case for not-satisfiable problems, when the min-sum stops by reaching the maximum number of iterations.

The graph also shows a linear dependency between $M$ and time: this can be easily observed in the cases when $M = 10000$ and $M = 20000$, where when $\bar{x} = 0.3$ the case $M = 20000$ requires approximately double the time required

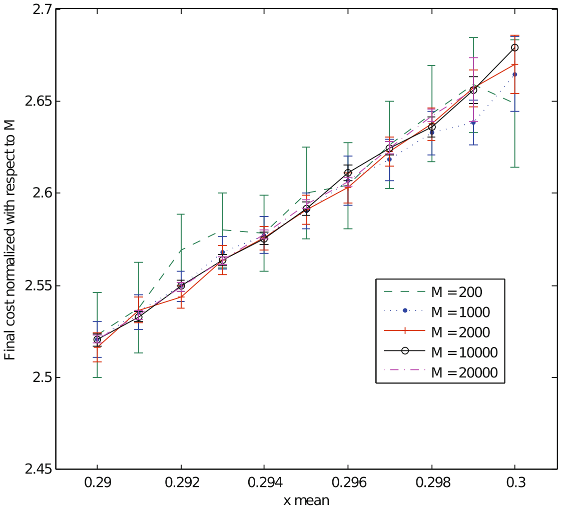


**Fig. 4.** Final cost value (normalized with respect to $M$).

by the case $M = 10000$. This can be explained by considering that the number of messages sent for each iteration by the min-sum algorithm is $4R \cdot M$.

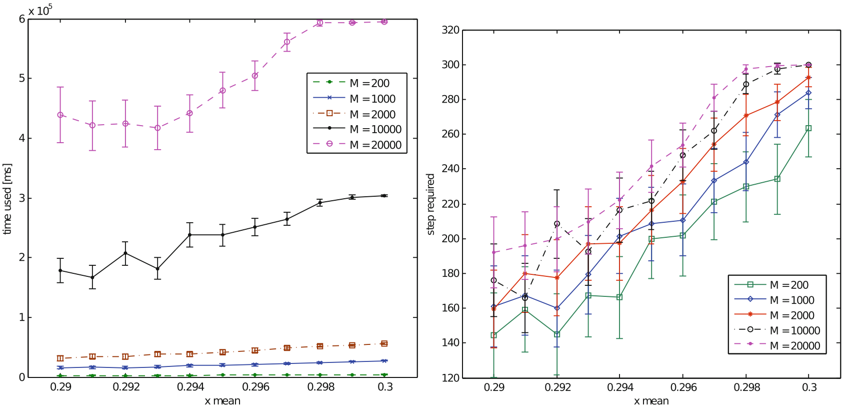


**Fig. 5.** Left: mean values for run time varying both $\bar{x}$ and $M$, Right: steps required.

The steps required by the min-sum algorithm are shown in Fig. 5 (right). The graph shows a similar trend between steps required when $M$ changes. For example consider $M = 200$ and $M = 20000$ when $\bar{x} = 0.296$: $M$ grows 100 times, and the steps required change from 190 to 250. The growth is sub-linear. There is a stronger dependency between step required and $\bar{x}$: thus, when the problem is *harder* a bigger number of iterations is necessary to find a solution.

We now turn to the comparison of min-sum with simulated annealing. Since simulated annealing is not able to handle hard-constrained problem, every hard-constraint is transformed in a soft-constraint by changing $+\infty$ values to a pre-defined upper bound value. The algorithm implements a mechanism of random restart: for 20 times, the execution of simulated annealing is repeated, and the final value is the lowest (i.e., the best) obtained through the 20 repetitions.

Overall, simulated annealing requires a significant amount of time to solve these power grid problems. In fact, it takes several days to analyze all the 1100 instances when $M = 200$ (that is the smallest value for $M$). The main issue is that simulated annealing changes the value of one variable at each iteration: thus, when $M = 200$ and the variable nodes in the factor graph are 600, simulated annealing needs several steps to change the value of a big part of the variables. We tuned the algorithm to reduce computation time and we found that for $M = 200$ the best value for the number of iterations is 100000 with an initial temperature equals to 1500. These significantly reduced computation time (to few hours), however run time is still prohibitive for instances that have a larger $M$, hence in the following experiments we fixed $M = 200$.

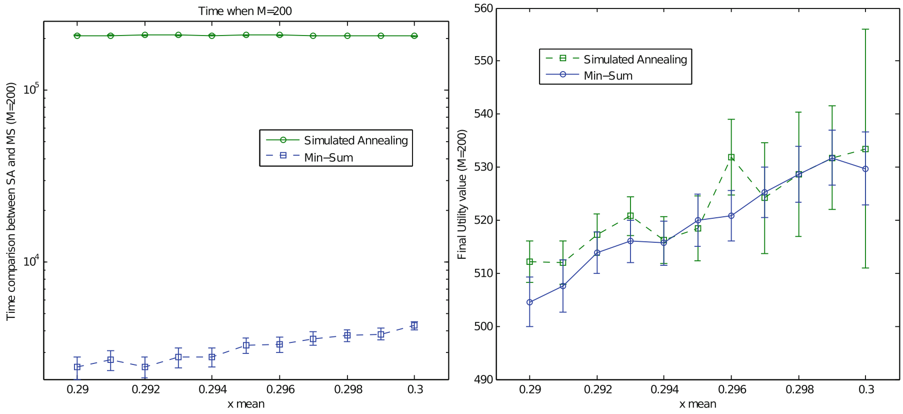


**Fig. 6.** Left: run-time mean value (in milliseconds) $M = 200$, Right: comparison between Simulated Annealing and min-sum $M = 200$

Figure 6 (left) confirms a significant difference in run time in favour of min-sum. Moreover, Fig. 6 (right) reports a comparison between the mean values of the cost obtained by min-sum and Simulated Annealing. The graph shows that min-sum is generally comparable to simulated annealing and sometimes gives better results.

Figure 7 provides a more refined comparison between the performance of min-sum and simulated annealing. This chart (best viewed in colors) reports a percentage of the outcomes for the two algorithms. As previously mentioned, min-sum is in generally better. Moreover, a more detailed analysis of this results reveals that if both min-sum and simulated annealing terminate, than min-sum does never provide a solution of higher cost with respect to simulated annealing.

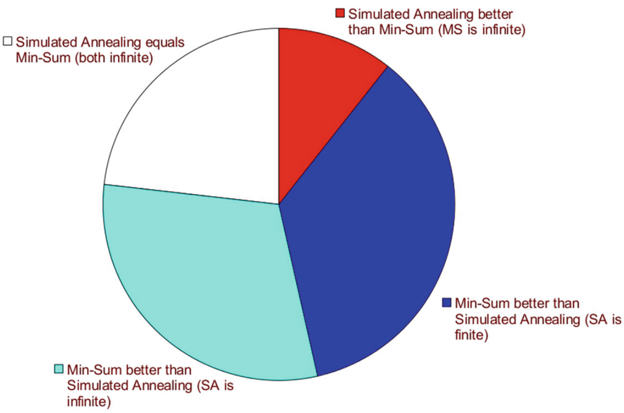


**Fig. 7.** Cake graph for comparing min-sum and Simulated Annealing results (Color figure online)

## 5 Conclusions and Future Work

In this paper we considered the model for power distribution with ancillary lines proposed in [12]. We extend such model to consider $CO_2$ emission and we propose a decentralized solution approach based on a DCOP formalization of the problem and the min-sum algorithm.

We empirically evaluate our approach on a set of benchmarking power grid instances built according to the procedure proposed in [12] and we compare the results obtained by min-sum with simulated annealing. Our results, suggests that min-sum favorably compares with simulated annealing and it provides a promising method for a distributed implementation of this model.

This work lays the basis for several interesting future directions. For example an interesting aspect is to further extend the power grid model to better describe realistic situations. A first extension might be to consider loads that can produce energy (e.g., renewable sources) and handle the optimization problems related to storing the excess of power created by such prosumers. Another interesting aspect would be to consider in the optimization process other important factors for energy distribution such as energy prices, government regulations and load/production forecasting.

## References

1. Aji, S., McEliece, R.: The generalized distributive law. IEEE Trans. Inf. Theory **46**(2), 325–343 (2000)

2. Farinelli, A., Rogers, A., Jennings, N.: Agent-based decentralised coordination for sensor networks using the max-sum algorithm. Auton. Agent. Multi-agent Syst. **28**(3), 337–380 (2014)
3. Farinelli, A., Rogers, A., Petcu, A., Jennings, N.R.: Decentralised coordination of low-power embedded devices using the max-sum algorithm. In: Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems, pp. 639–646 (2008)
4. Fitzpatrick, S., Meertens, L.: Distributed coordination through anarchic optimization. In: Lesser, V., Ortiz Jr., C.L., Tambe, M. (eds.) Distributed Sensor Networks: A Multiagent Perspective, pp. 257–293. Kluwer Academic (2003)
5. Karnouskos, S.: The cooperative internet of things enabled smart grid. In: Proceedings of the 14th IEEE International Symposium on Consumer Electronics (ISCE2010), pp. 7–10, June 2010
6. Kschischang, F.R., Frey, B.J., Loeliger, H.A.: Factor graphs and the sum-product algorithm. IEEE Trans. Inf. Theor. **47**(2), 498–519 (2006)
7. Maheswaran, R.T., Pearch, J.P., Tambe, M.: Distributed algorithms for dcop: a graphical game-based approach. In: Proceedings of the Seventeenth International Joint Conference on Autonomous Agents and Multi-agent Systems, pp. 432–439 (2004)
8. Miller, S., Ramchurn, S.D., Rogers, A.: Optimal decentralised dispatch of embedded generation in the smart grid. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, pp. 281–288 (2012)
9. Modi, P., Shen, W.M., Tambe, M., Yokoo, M.: Adopt: asynchronous distributed constraint optimization with quality guarantees. Artif. Intell. **161**(1–2), 149–180 (2005)
10. Petcu, A., Faltings, B.: Dpop: a scalable method for multiagent constraint optimization. In: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, pp. 266–271 (2005)
11. Pujol-Gonzalez, M., Cerquides, J., Meseguer, P., Rodríguez-Aguilar, J.A., Tambe, M.: Engineering the decentralized coordination of UAVs with limited communication range. In: Bielza, C., Salmerón, A., Alonso-Betanzos, A., Hidalgo, J.I., Martínez, L., Troncoso, A., Corchado, E., Corchado, J.M. (eds.) CAEPIA 2013. LNCS (LNAI), vol. 8109, pp. 199–208. Springer, Heidelberg (2013). doi:10.1007/978-3-642-40643-0_21
12. Zdeborov, L., Decelle, A., Chertkov, M.: Message passing for optimization and control of a power grid: model of a distribution system with redundancy. Phys. Rev. E Stat. Nonlinear Soft Matter Phys. **80**(4), 046112 (2009)
13. Zhang, W., Wang, G., Xing, Z., Wittenburg, L.: Distributed stochastic search and distributed breakout: properties, comparison and applications to constraint optimization problems in sensor networks. Artif. Intell. **161**(12), 55–87 (2005)