Proceedings of the ASME 2021
15th International Conference on Energy Sustainability
ES2021
June 16-18, 2021, Virtual, USA

ES2021-62982

DRAFT: RUBIK'S CUBE TOPOLOGY BASED PARTICLE SWARM ALGORITHM FOR BILEVEL BUILDING ENERGY TRANSACTION

Xiaochun Feng
School of Economics and Management
Northwest A&F University
Yangling, China

Yang Chen[*]
Environmental Sciences Division
Oak Ridge National Laboratory
Oak Ridge, U.S.

Jian Zhang
Richard J. Resch School of Engineering
University of Wisconsin-Green Bay
Green Bay, U.S.

Heejin Cho
Department of Mechanical Engineering
Mississippi State University
Mississippi State, U.S.

Xin Shi
Department of Industrial and
Systems Engineering, Lehigh University
Bethlehem, U.S.

ABSTRACT

Following the rapid growth of distributed energy resources (e.g. renewables, battery), localized peer-to-peer energy transactions are receiving more attention for multiple benefits, such as, reducing power loss, stabilizing the main power grid, etc. To promote distributed renewables locally, the local trading price is usually set to be within the external energy purchasing and selling price range. Consequently, building prosumers are motivated to trade energy through a local transaction center. This local energy transaction is modeled in bilevel optimization game. A selfish upper level agent is assumed with the privilege to set the internal energy transaction price with an objective of maximizing its arbitrage profit. Meanwhile, the building prosumers at the lower level will response to this transaction price and make decisions on electricity transaction amount. Therefore, this non-cooperative leader-follower trading game is seeking for equilibrium solutions on the energy transaction amount and prices. In addition, a uniform local transaction price structure (purchase price equals selling price) is considered here. Aiming at reducing the computational burden from classical Karush-Kuhn-Tucker (KKT) transformation and protecting the private information of each stakeholder (e.g., building), swarm intelligence based solution approach is employed for upper level agent to generate trading price and coordinate the transactive operations. On one hand, to decrease the chance of premature convergence in global-best topology, Rubik's Cube topology is proposed in this study based on further improvement of a two-dimensional square lattice model (i.e., one local-best topology-Von Neumann topology). Rotating operation of the cube is introduced to dynamically changing the neighborhood and enhancing information flow at the later searching state. Several groups of experiments are designed to evaluate the performance of proposed Rubik's Cube topology based particle swarm algorithm. The results have validated the effectiveness of proposed topology and operators comparing with global-best version PSO and Von Neumann topology based PSO and its scalability on larger scale applications.

1. INTRODUCTION

Renewable energy is the fastest-growing energy source in the U.S., increasing 100 percent from 2000 to 2018, and the consumption of renewable energy in 2019 was nearly three times greater than in 2000. In 2019, renewable energy supported 17% of total U.S. electricity consumption and it surpasses coal generation for the first time in over 130 years [1]. Increasing renewable energy start to impose burden on current grid infras-

[*] Corresponding Email: cheny2@ornl.gov

tructure, and it could be a grand challenge in terms of power system stability to integrate such amount of intermittent renewables. To reduce the burden, local microgrids and energy market is a promising option in absorbing more renewables locally, by providing a wide range of benefits, such as reducing power loss over long distance transmission, more choices and more cheaper energy, preventing large scale blackouts due to extreme storms, etc. Community level microgrid practice in U.S. could refer to Residential Solar Microgrid in California, Long Island Community Microgrid, North Bay Community Resilience Initiative in San Francisco Bay area [2].

With smart grid/building technologies, traditional building consumer has transformed into prosumers integrated with distributed energy resources (DERs). Along with more prosumers (players) involved in the local energy trading game, Stackelberg game or bilevel optimization is very much relied to obtain equilibrium bidding strategies for all players. For Stackelberg game, the common method to find subgame-perfect nash equilibrium is to substitute the follower's model by deriving Backward Induction. Bilevel optimization is a more general framework of Stackelberg game, and similarly, it can be solved by Karush-Kuhn-Tucker (KKT) based necessary conditions [3]. For instance, a bilevel community energy transaction market is designed as potential use case for a new modular pump hydro storage patented in oak ridge national laboratory [4] and the storage is assumed as the game leader and price maker, then detailed KKT conditions are derived for solving process. With the conflict objective of distribution company and DERs aggregators, bilevel optimization approach is proposed for the coordination between transmission, distribution, and DERs aggregators that interact in a local market model. Then non-linear bilevel model is transformed into a linear single level model through KKT conditions and duality theory [5].

Due to the sharply increased constraints (prime constraints, stationary constraints, Complementary Slackness Constraints) and binary variables in KKT conditions, the resulted single level model from bilevel optimization is generally much harder to solve and limited for small case application. Meanwhile, KKT approach is only applicable when lower level problem is convex and it needs all information (e.g. energy storage capacity, customer preference, etc) from lower level agents for central decision making. To reduce the computational burden and protect privacy data of lower level, distributed solution approaches have been explored in bilevel optimization context. In a bilevel price-demand response game between distribution system operator and flexible load aggregator, particle swarm algorithm and pattern search algorithm are investigated as distributed solution approaches and compared with classical centralized backward induction approach [6], where the price signal and optimal demand response are shared. Similary, distributed genetic algorithm is proposed for smart grid with multiple utility companies and multiple users based on expectation bilevel programming. At each iteration of genetic algorithm, real time price from utility company and optimal consumption of users are provided and exchanged [7]. A swarm intelligence base bilevel distributed approach is developed to coordinate transactive operations among buildings, where the only information exchanged between system level and building level is the marginal price of transactive energy which is utilized to guide particles' searching process [8] [9]. For a multiple-leader multiple-follower stackelberg game in community energy trading, the price competition among sellers is modeled as a static non-cooperative game, while the seller selection competition among buyers is modeled by evolutionary game approach. And then two iterative distributed algorithms are designed respectively for sellers' and buyers' competition [10]. A novel generation expansion problem that integrates the renewable energy market is formulated as a bilevel optimization [11], where the offering prices, supplies, and demands in renewable energy credits market are considered as functions of power system optimal operations. The problem is solved effectively by the proposed combination of KKT reformulation and the fixed point iterative algorithm.

For particle swarm algorithm, it uses a social topology for particles to share information among neighbors during solution searching. Since its introduction in 1995, two kinds of interparticle communication network were proposed: Gbest and Lbest networks [12]. In Gbest or global topology, all particles are fully connected and share information, each particle is attracted towards the best solution found by entire swarm. Whereas, in Lbest or local topology, each particle only share information with immediate neighbors. Observed in large amount of studies, the performance of PSO depends on its social topology greatly and there is no outright best topology for all problems. Eight PSO topologies are investigated on their performance in PSO trained extreme learning machine [13], which include Global, Ring, Von Neumann, Wheel, Four cluster, Clan, etc. To provide a guide to topology selection of PSO, a class of deterministic regular topologies with regard to what affect the optimality of algorithmic parameters are analyzed [14]. A ring topology is adopted in exemplar generation to enhance diversity and exploration of particle swarm while a global leaning component with linearly adjusted control parameters is employed to improve adaptability [15]. In order to improve PSO performance, a new topology is proposed based on small-world network, in which, each particle interacts with its cohesive neighbors and by chance to communicate with some distant particles via small-world randomization [16] [17]. In small-world topology, the neighborhood size and the randomization probability are adaptively adjusted based on convergence state. A dynamic hierarchical version (tree-like) topology is introduced [18], where particles with better fitness move up the hierarchy and generate a larger influence on the swarm. By assembling positive algorithmic components, such as time-varying topology, velocity-update mechanism, time-decreasing inertia weight etc, the proposed Frankenstein's PSO is capable of performing better in many cases than the variants from which its components were taken [19]. Less connection in a topology, the more it delays the propagation of the best-so-far solution. Thus, low connected topologies result in more exploratory behavior than highly connected ones.

Of particular interest, Von Neumann topology is a 2D lattice grid and each particle has above, below, to the left, to the right as its neighbors, wrapping around if necessary. It is a static topology with fixed neighbors. In this research, we propose rubik's cube topology for particle swarm algorithm which could be treated as 3D Von Neumann topology, and it is then applied on bilevel energy transaction game with one leader and multiple followers. The energy transaction model for leader-followers and uniform local energy price structure are introduced in Section II; in Section III, the proposed rubik's cube topology and its rotation operator are explained in details with pseudo code for implementation; different cases are designed for experiments in Section V with comparison on different variations of particle swarm algorithms. Finally, conclusions are drawn in Section VI.

2. LOCAL ENERGY TRANSACTION

In this section, a benchmark bilevel local energy transaction model is adopted from our previous study [3]. Shown by the system scheme in Figure 1, at the upper level, an energy storage (battery) is served as local energy market operator with the privilege of setting internal transaction price and transact with external grid if necessary, and the lower level building prosumers can only purchase energy from or sell energy to local energy transaction market.

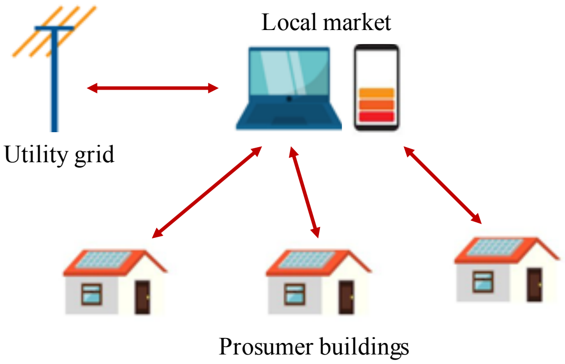

FIGURE 1: Bilevel energy transaction scheme for local prosumers

TABLE 1: Parameter notations for the operation model

Index	
n, n'	Index of prosumers, $n' \neq n$
t, u	Index of time, the upper level agent
Parameter	
Pp, Ps	Purchasing, selling price in power grid
SV, SB	Size of PV panel, power capacity of battery
EB, η_V	Initial battery energy level, PV efficiency
El, Sol	Nominal load profile, solar radiation
$\underline{El}, \overline{El}$	Lower, upper bound for energy demand
$\underline{\alpha}_b, \overline{\alpha}_b$	Min, max state-of-charge of battery
$\overline{\alpha}_c, \overline{\alpha}_d$	Max coefficient of battery charging, discharging
η_c, η_d	Charging, discharging efficiency of battery
γ, δ	Degradation cost, sensitivity in load shifting
Variable	
ep, es	Energy purchased from, sold to power grid
et^i, et^o	Energy transacted into, out from prosumers
xi, xo	In/out binary status for energy transactionU
el, ev, eb	Actual load, PV generation, storage level
ec, ed	Charging, discharging power of battery
pt^+, pt^-, pt	Purchasing, selling, uniform price in local market

A. Lower Level Model

At lower level, each prosumer is trying to minimize its operation cost in Eq.(1), which consist of trading cost in local market (first term), degradation cost of battery storage (second term) and inconvenience cost of load shifting (third term).

$$\min C_n = \sum_t \left[(pt^+ \cdot et^i_{n,t} - pt^- \cdot et^o_{n,t}) + \gamma \cdot (ec_{n,t} + ed_{n,t}) + \delta \cdot (el_{n,t} - El_{n,t})^2 \right] \quad (1)$$

Electricity demand is balanced for each prosumer in Eq.(2). On the left side, power generation includes solar panel, battery discharging and local purchasing, while on the right side, power is needed for actual load consumption, battery charging and local selling.

$$ev_{n,t} + ed_{n,t} \cdot \eta_d + et^i_{n,t} = el_{n,t} + \frac{ec_{n,t}}{\eta_c} + et^o_{n,t} \quad (2)$$

Solar power for each prosumer is determined by its solar panel size, solar radiation level and solar panel electricity generation efficiency, in Eq.(3)

$$0 \leq ev_{n,t} \leq SV_n \cdot Sol_t \cdot \eta_V \quad (3)$$

The behavior of battery storage is modeled by constraints Eq.(4)-(9). The charging/discharging power are limited in the range of Eq.(4)-(5) defined by its capacity and according coefficients. Energy level in battery in each time period is constrained in the allowable range in Eq.(6), and it is determined by the charging/discharging activities in Eq.(7)-(8). To mitigate the operation dependency of two consecutive days, Eq.(9) is added to ensure that the power level in battery at the end time period T of operation is equal to initial level. Δt is the time step length with one hour as unit (e.g. 1 hour, 15/60 hour, etc).

$$0 \leq ec_{n,t} \leq SB_n \cdot \overline{\alpha}_c \tag{4}$$

$$0 \leq ed_{n,t} \leq SB_n \cdot \overline{\alpha}_d \tag{5}$$

$$SB_n \cdot \underline{\alpha}_b \leq eb_{n,t} \leq SB_n \cdot \overline{\alpha}_b \tag{6}$$

$$eb_{n,1} = EB_{n,0} + (ec_{n,1} - ed_{n,1}) \cdot \Delta t \tag{7}$$

$$eb_{n,t} - eb_{n,t-1} = (ec_{n,t} - ed_{n,t}) \cdot \Delta t \tag{8}$$

$$eb_{n,T} = EB_{n,0} \tag{9}$$

Prosumers have the ability to change its demand pattern based on electricity price, but it is still need to make sure that electricity usage does not drop below the base load or exceed the upper limit in each hour by Eq.(10). Also, to ensure normal operation, the total electricity consumption in the operation period (e.g. one day) shouldn't be curtailed beyond a user defined curtail coefficient ρ in Eq.(11).

$$\underline{El}_{n,t} \leq el_{n,t} \leq \overline{El}_{n,t} \tag{10}$$

$$(1-\rho) \cdot \sum_t El_{n,t} \leq \sum_t el_{n,t} \tag{11}$$

B. Upper Level Model

The upper level agent in local energy market has the responsibility to maintain market order, coordinate the trading, etc., and also it is given the privilege to set the internal energy transaction price. With the selfish objective, it aims at maximizing its own profit R in Eq.(12), which equals the profit in the external market (first term) minus battery degradation cost (second term) and plus profit in local market arbitrage (third term).

$$\begin{aligned} \max R = &\sum_t \left[(Ps_t \cdot es_{u,t} - Pp_t \cdot ep_{u,t}) - \gamma \cdot (ec_{u,t} + ed_{u,t}) \right] \\ &+ \sum_{n,t} (pt^+ \cdot et^i_{n,t} - pt^- \cdot et^o_{n,t}) \end{aligned} \tag{12}$$

To encourage energy trading and absorb more distributed renewables locally, local energy transaction prices are set to be within the external price range in Eq.(13). Note that, different from common discriminative price structure ($pt^- \leq pt^+$), the uniform trading price structure ($pt^- = pt^+ = pt$) is considered in this work since it is shown in previous work [3] that a more fairer profit allocation could be achieved using uniform price structure in such a bilevel non-cooperative game. In actual implementation, a very small number (e.g. 0.001) should be used ($pt^- = pt^+ - 0.001$) to avoid the concurrence of $et^i_{n,t}$ and $et^o_{n,t}$.

$$Ps_t \leq pt^- = pt^+ \leq Pp_t \tag{13}$$

Since upper level agent owns a battery storage, the operation constrains in Eq.(4)-(9) also need to be applied similarly for upper level model. In summary, for this bilevel model, the upper level objective is in Eq.(12) with constraints Eq.(4)-(9) and Eq.(13). The lower level agents have objective Eq.(1) and constraints Eq.(2)-(11).

3. RUBIK'S CUBE TOPOLOGY BASED PSO

Particle swarm algorithm is a widely used swarm intelligence approach, which mimics a flock of birds flying through the solution space while communicating with each other. Extensive studies have demonstrated that PSO usually performs better than other population-based algorithms including genetic algorithms, ant-colony algorithm, shuffled frog leaping algorithms in terms of solution quality and computational efficiency on high-dimensional nonlinear continuous solution space [20] [8]. Meanwhile, canonical PSO has its own disadvantages, such as the high speed of convergence which often implies a rapid loss of diversity during iterations and lead to undesirable premature convergence [21]. To overcome the premature convergence and avoid local optimality effectively, lots of PSO variants have been proposed from the perspective of improved social topology [13] [15], mutation mechanism [22] [23], hybrid with other complimentary algorithms [24] [25], etc.

In the standard or Gbest version of PSO, the velocity vector $\boldsymbol{v}$ of each particle is updated according to its own best experience as well as to the best particle in the swarm. The current positions of the particles can be described as a set of candidate solution in the space, and in each iteration new positions will be updated by adding the velocity, which can be see as step size, to current position coordinate vector $\boldsymbol{x}$, see Eq.(14)-(15).

$$\boldsymbol{v}_p^{i+1} = \omega \cdot \boldsymbol{v}_p^i + c1 \cdot r^i_{1,p} \cdot (\boldsymbol{p}^i_p - \boldsymbol{x}^i_p) + c2 \cdot (\boldsymbol{p}^i_g - \boldsymbol{x}^i_p) \tag{14}$$

$$\boldsymbol{x}_p^{i+1} = \boldsymbol{x}_p^i + \boldsymbol{v}_p^{i+1} \tag{15}$$

where $\boldsymbol{p}^i_p$ is the best position found so far by the pth particle and $\boldsymbol{p}^i_g$ is the best position found by the whole swarm. ω is the inertia weight which control the relationship between exploration and exploitation by determining how much inertia it has on previous velocity. $c1, c2$ are cognition and social factors. $r^i_{1,p}, r^i_{2,p}$ are two random values in the range of [0,1].

$$\boldsymbol{v}_p^{i+1} = \omega \cdot \boldsymbol{v}_p^i + c1 \cdot r^i_{1,p} \cdot (\boldsymbol{p}^i_p - \boldsymbol{x}^i_p) + c2 \cdot (\boldsymbol{p}^i_l - \boldsymbol{x}^i_p) \tag{16}$$

In a local topology or Lbest version of PSO, the velocity updation of each particle follows Eq.(16), where global best position $\boldsymbol{p}^i_g$ is changed to the best position $\boldsymbol{p}^i_l$ founded by its connected neighbors. Local topology usually works better when problems contain cliffs, variable interactions, and other features that are not typified by smooth gradients [26]. For instance, if all particles forms a circle in ring topology, it will only communicate with its two adjacent neighbors. In local topology version PSO, the particles will be spread out and explore different regions simultaneously.

The number of neighbors influences the information propagation speed. Here the proposed Rubik's cube topology is inspired in the Von Neumann topology. In Figure 2, three slices $(x,0,z)$ $(x,y,0)$ $(0,y,z)$ of a 4-order cubic topology (totally 16 slices) are shown. As mentioned, Von Neumann topology is a 2D lattice grid, thus each slice is a Von Neumann topology grid. Depends on the coordinate location of particles, it may has three, four, five or six adjacent neighbors, e.g, corner particle $(0,0,0)$ has $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$ three adjacent neighbors. Similarly, the edge particle $(1,0,0)$ has four, outer face particle $(1,0,1)$ has five and inner particle $(1,1,1)$ has six adjacent neighbors. The particles are typically included in their own neighbor set as they may influence themselves. Note that, the coordinate (x,y,z) used here is not the particle position in Eq.(14), it is only used to determine neighbor set more easily in implementation. To enhance the information flow and improve the chance of jumping out of local minimum at the later stage of searching, the *rotation* operator is introduced. The criteria of starting rotating and the selection of rotating slides may depend on the fitness improvement ranking in several successive runs or other heuristic rules. After *rotation*, the neighbor set of rotated particles is re-assigned based on their coordinate location.

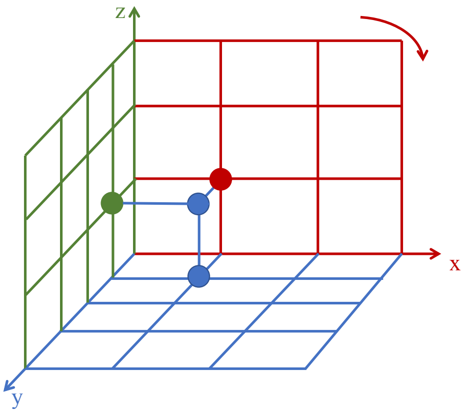

FIGURE 2: 4-order cubic social topology for particle swarm

The procedure of solving bilevel building energy transaction in local market using Rubik's cube topology based PSO is described in Algorithm 1. The particles' position represent local energy transaction price $pt^{+} = pt^{-}$ in the upper level model in section 2, which is randomly initialized (line 2 in Algorithm 1) in the range bounded by external purchasing and selling prices in Eq.(13). Therefore, for each particle, its position/price information will be passed to each prosumer at lower level, and each prosumer solves its own operation model individually (line 10). Then the transaction request et^{i} and et^{o} are collected to upper level for coordinated scheduling (line 11, 12). The objective value of upper level model is used as fitness as it complies that the leader moves with the consideration of the followers' reaction in Stackelberg game. In this work, the upper and lower level model is solved using CPLEX mixed integer solver, it may also be solved by meta-heuristic with constraint handling techniques. The Algorithm 1 could be implemented in a parallel way at particle level. Since the swarm size is not always the required number to fully fulfill the cubic topology, the cubic order maybe modified or some coordinates maybe left to blank but still need to keep the connectivity.

```
Algorithm 1: Rubik's cube topology based PSO
  Input:
     swarm size $m$, inertia weight $\omega$
     learning factors $c1, c2$
  Output:
     $\vec{x}^{*}$, the best particle position
     $f(\vec{x}^{*})$, the best fitness of upper level
1 for particle $m = 1, \cdots, M$ do
2     Assign $m$ with coordinates in cubic topology
3     Initialize $m$ with random $\vec{x} \in [Ps_t, Pp_t]$ and velocity
4 while stop criterion is not met do
5     for particle $m = 1, \cdots, M$ do
6         $\vec{p}_m$ = best position particle $m$ has found so far
7         $\vec{l}_m$ = best position in neighbor set of $m$ so far
8         $\vec{v}_m$ = velocity of $m$ updated from Eq.16
9         $\vec{x}_m$ = position of $m$ updated from Eq.15
10        for prosumer $n = 1, \cdots, N$ do
11            Solve lower level model Eq.(1)-(11)
12        Collect $et^{i}_{n,t}$ and $et^{o}_{n,t}$ from all prosumers
13        Solve upper model Eq.(12)-(13), Eq.(4)-(9)
14        Calculate $f(\vec{x})$ and update $\vec{p}_m, \vec{l}_m, \vec{x}^{*}$
15    if rotation criterion is met then
16        Randomly pick slice $s$ of out 12 slices
17        Rotate the slice $s$ counter/- clockwise
18 Function rotate($s$):
      /* clockwise or counter clockwise */
19    for particle $m = 1, \cdots, 16$ on slice $s$ do
20        Update 3D coordinate of $m$
21        Re-assign neighbor set to $m$ based on coordinate
22        return new neighbor set of particle $m$
```

4. NUMERICAL EXPERIMENTS

In order to evaluate the effectiveness of proposed cubic topology with *rotation* operator, two case studies of bilevel building energy transaction with different time scale are designed in this section and totally five solution approaches are considered: 1) standard Gbest particle swarm (g-PSO), 2) particle swarm with Von Neumann topology (vn-PSO), 3) particle swarm with rubik's cube topology (rc-PSO), 4) rc-PSO with rotation operator (rrc-PSO) , 5) KKT based transformation which provides centralized optimum solution.

- Case 1: With one hour time scale, the time steps from hour 6 to hour 20 are selected for one day's optimization as lower level building are more active in energy transaction when there is solar power. The hourly nominal load and solar power profile of three prosumers $n1, n2, n3$ are shown in Figure 3.

- Case 2: With fifteen minute time scale, the time steps from 21 (hour 6) to 80 (hour 20) are selected for the same reason as in Case 1. When time scale is increased, the solution space dimension becomes much higher. 15-minute data is generated by interpolation based on hourly data.

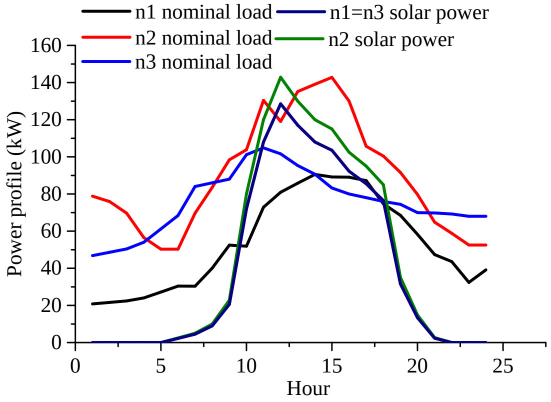

FIGURE 3: Nominal load and solar power for building prosumers

All experiments are conducted on 64-bit Windows PC with i7-3537 CPU @2.00GHz, 2.50GHz and 8GB RAM. Parameters in Table 1 are set as same in Ref [3]: $\eta_V = 0.25$, $\eta_c = \eta_d = 0.95$, $\underline{\alpha}_b = 0.05$, $\overline{\alpha}_b = 1$, $\underline{\alpha}_c = 0.25$, $\overline{\alpha}_d = 0.1$, $\gamma = 0.008$, $\delta = 0.002$, $\rho = 0$. For each time step, $\underline{El} = 0.8 \cdot El$, $\overline{El} = 1.2 \cdot El$. For lower level prosumers in name order $SV = [90, 100, 90](m^2)$, $SB = [45, 60, 40](kW)$. Upper level agent has a battery storage with $160kW$ capacity and same efficiency parameters with building prosumers. The stop criterion of PSO algorithms is set as same: stop when fitness improvement in continuous 20 iterations is less than 1. For the sake of simplicity, rotation criterion for rrc-PSO is set as: rotate when fitness improvement in continuous 5 iterations is less than 1. All PSO algorithms are run continuously for 10 times with same parameter settings. Swarm size is 64, initial inertia weight ω is 0.6 and is adjusted based on random generation strategy in each iteration [27] [28]. Initial learning factor $c1 = c2 = 1.496$ and is adjusted in each iteration according to nonlinear acceleration strategy [27].

For Case 1, the iterated mean fitness value in 10 runs of all PSO algorithms is plotted in Figure 4. Since PSO algorithms doesn't end at the same iteration number in multiple runs due to the stop criterion, the converged value is filled to maximum iteration number to calculate the mean value. KKT approach solves this small case in 4 seconds and gives optimum revenue solution \$3.074 for upper level agent, which is the upper bound for other algorithms. g-PSO performs best in this small case since it converges to near optimum solution faster than others because the best solution is shared within the swarm. Other PSO with local topologies slowly converges following very similar mean fitness trace. The resulted mean energy transaction prices are given in Figure 5. The energy purchasing price Pp and selling price Ps in external grid serves as upper and lower bound. It is shown that, before building prosumers have extra electricity before hour 10 and afterr hour 19 due to less solar, the upper level agent tends to set local energy transaction price as high as possible to make more profit. Between hour 10 and 18, the local transaction price is much lower than external purchasing price, and since local purchasing price equals selling price, the arbitrage revenue of upper level depends on difference of purchasing and selling amount, see Eq.(12). The mean price signals obtained in Figure 5 are used to calculate operation cost of lower level prosumers, recorded in Table 2. Without knowing private information on building prosumers, all PSO algorithms solves the lower level model in a distributed way and performance well in this small case.

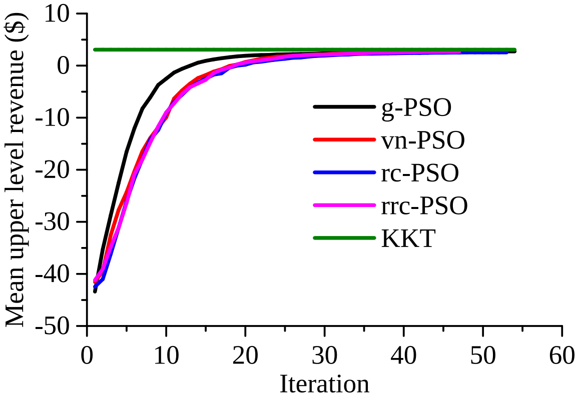

FIGURE 4: Mean fitness in 10 runs of PSO algorithms in Case 1

TABLE 2: Mean operation cost (\$) of prosumers in Case 1

Prosumers	KKT	g-PSO	vn-PSO	rc-PSO	rrc-PSO
n1	8.791	8.773	8.517	8.704	8.588
n2	46.742	46.187	46.898	46.760	46.875
n3	31.306	31.542	31.148	31.200	31.183

In Case 2, the time dimension or solution space dimension

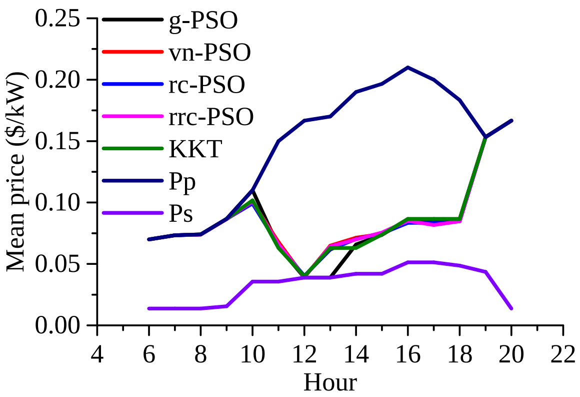

FIGURE 5: Mean transaction price of PSO algorithms in Case 1

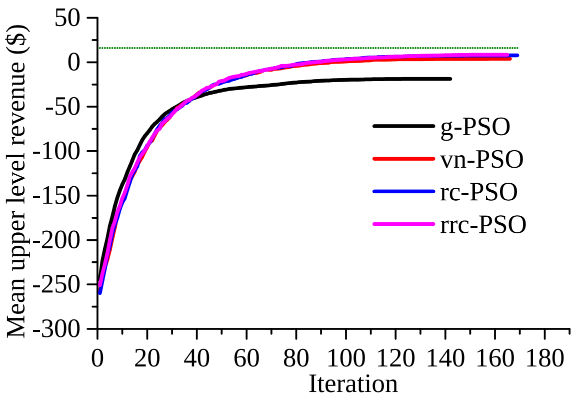

FIGURE 6: Mean fitness in 10 runs of PSO algorithms in Case 2

increases from 15 to 60, about 4 times. As mentioned, constraints and variables number increases sharply for KKT approach, and it takes more than 1 hour to reach \$16.06 with an relative optimality gap 18.76% and was stopped due to time budget. Since it is not optimal solution, the 16.06 is plotted as possible upper bound using dot dash line in following figures. Same as in Case 1, the converged value of PSO algorithms after stopping is filled to maximum iteration number to calculate the mean value for Case 2 shown in Figure 6. Hence the iteration number in Figure 6 is the maximum iteration number of multiple runs for each PSO. Similar mean fitness pattern can be observed as in Figure 4, except that mean converged value of g-PSO is much far away from possible optimum. The iteration fitness value in 10 runs of g-PSO is shown in Figure 7, which indicates that the searching is trapped into several local optimal solution easily in a higher dimension Case 2 and its performance is very much versatile in multiple runs. In contrast, shown by Figure 8, proposed rrc-PSO could jump out of premature local optimal neighborhood and is very reliable for each continuous run. The performance of PSO algorithms in Case 2 is summarized in Table 3. The final fitness range of g-PSO in 10 runs is about 48.7 and have a very high variance of 170, but with a very low mean iteration number due to premature convergence. The Von Neumann topology has greatly improved g-PSO average final performance on this case with a higher mean iteration number. Proposed Rubic's cube topology is superior to vn-PSO on the fitness and reliability. Proposed *rotation* operator could further enhance the final fitness performance of rc-PSO by 10% but converges much slower with the highest mean iteration number.

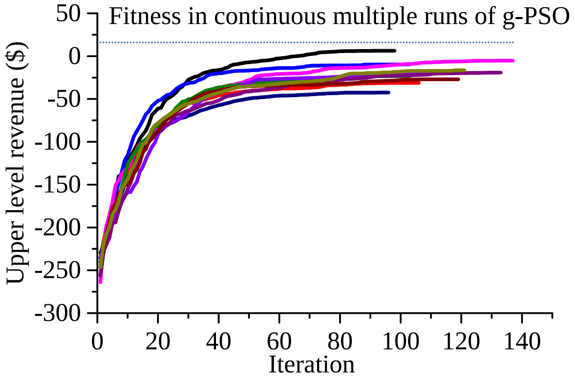

FIGURE 7: Fitness iteration in 10 runs of g-PSO in Case 2

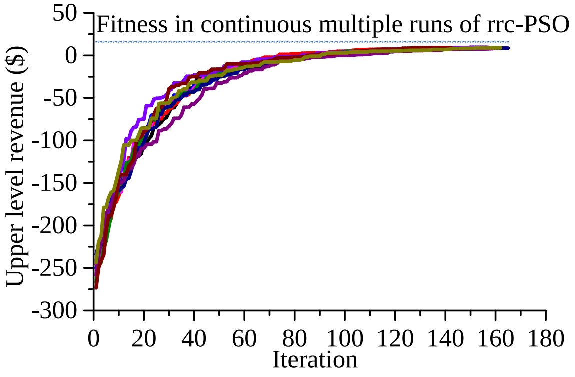

FIGURE 8: Fitness iteration in 10 runs of rrc-PSO in Case 2

5. CONCLUSION

In this research, solution approaches for localized bilevel energy transaction among building prosumers are focused. To overcome the sharply increased computational burden from traditional KKT transformation, particle swarm intelligence based distributed solution approaches are studied for bilevel optimization model, where the upper level agent observes the transaction responses of lower level by passing down the price signal coded in particles' position. In this trial and learn process, upper level agent maximizes its selfish objective in terms of over-

TABLE 3: Performance summary of PSO algorithms in Case 2

Algorithm	Mean fitness	Best	Worst	Var	Mean Iter.
g-PSO	-18.714	6.369	-42.436	170.570	114.0
vn-PSO	3.808	7.687	0.754	3.737	129.4
rc-PSO	7.715	9.120	5.068	1.339	139.0
rrc-PSO	8.476	9.474	7.546	0.391	149.7

all revenue without knowing sensitive information from lower level prosumers. Only the information of prices and transaction amount are exchanged. Aiming at maintaining population diversity and decreasing the chance of premature convergence of Gbest topology, a Rubik's cube topology with rotation operators is proposed based on Von Neumann topology. Comparison experiments on two cases have demonstrated the effectiveness and reliability of proposed topology and operators.

REFERENCES

[1] U.S. Energy Information Administration, 2019. U.S. total energy production and consumption. `https://www.eia.gov/todayinenergy/detail.php?id=43515`.

[2] Clean Coalition, 2020. Community microgrid initiative-designing and staging community microgrids for resilience. `https://clean-coalition.org/community-microgrid-initiative/`.

[3] Chen, Y., Park, B., Kou, X., Hu, M., Dong, J., Li, F., Amasyali, K., and Olama, M., 2020. "A comparison study on trading behavior and profit distribution in local energy transaction games". *Applied Energy,* ***280***, p. 115941.

[4] Chen, Y., Kou, X., Olama, M., Zandi, H., Liu, C., Kassaee, S., Smith, B. T., Abu-Heiba, A., and Momen, A. M., 2020. "Bi-level optimization for electricity transaction in smart community with modular pump hydro storage". In Proceedings of the ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, p. V006T06A016.

[5] Sheikhahmadi, P., Bahramara, S., Mazza, A., Chicco, G., and Catalão, J. P., 2021. "Bi-level optimization model for the coordination between transmission and distribution systems interacting with local energy markets". *International Journal of Electrical Power & Energy Systems,* ***124***, p. 106392.

[6] Chen, Y., Olama, M., Kou, X., Amasyali, K., Dong, J., and Xue, Y., 2020. "Distributed solution approach for a stackelberg pricing game of aggregated demand response". In 2020 IEEE Power Energy Society General Meeting (PESGM), pp. 1–5.

[7] Tao, L., Gao, Y., Zhu, H., and Liu, S., 2019. "Distributed genetic real-time pricing for multiseller-multibuyer smart grid based on bilevel programming considering random fluctuation of electricity consumption". *Computers & Industrial Engineering,* ***135***, pp. 359 – 367.

[8] Chen, Y., and Hu, M., 2018. "A swarm intelligence based distributed decision approach for transactive operation of networked building clusters". *Energy and Buildings,* ***169***, pp. 172 – 184.

[9] Chen, Y., and Hu, M., 2019. "Swarm intelligence–based distributed stochastic model predictive control for transactive operation of networked building clusters". *Energy and Buildings,* ***198***, pp. 207 – 215.

[10] Paudel, A., Chaudhari, K., Long, C., and Gooi, H. B., 2019. "Peer-to-peer energy trading in a prosumer-based community microgrid: A game-theoretic model". *IEEE Transactions on Industrial Electronics,* ***66***(8), pp. 6087–6097.

[11] Nguyen, H. T., and Felder, F. A., 2020. "Generation expansion planning with renewable energy credit markets: A bilevel programming approach". *Applied Energy,* ***276***, p. 115472.

[12] Blackwell, T., and Kennedy, J., 2019. "Impact of communication topology in particle swarm optimization". *IEEE Transactions on Evolutionary Computation,* ***23***(4), pp. 689–702.

[13] Figueiredo, E. M., and Ludermir, T. B., 2014. "Investigating the use of alternative topologies on performance of the pso-elm". *Neurocomputing,* ***127***, pp. 4 – 12. Advances in Intelligent Systems.

[14] Liu, Q., Wei, W., Yuan, H., Zhan, Z.-H., and Li, Y., 2016. "Topology selection for particle swarm optimization". *Information Sciences,* ***363***, pp. 154 – 173.

[15] Lin, A., Sun, W., Yu, H., Wu, G., and Tang, H., 2019. "Global genetic learning particle swarm optimization with diversity enhancement by ring topology". *Swarm and Evolutionary Computation,* ***44***, pp. 571 – 583.

[16] Kennedy, J., 1999. "Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance". In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Vol. 3, pp. 1931–1938 Vol. 3.

[17] Gong, Y.-j., and Zhang, J., 2013. "Small-world particle swarm optimization with topology adaptation". In Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation, GECCO '13, Association for Computing Machinery, p. 25–32.

[18] Janson, S., and Middendorf, M., 2005. "A hierarchical particle swarm optimizer and its adaptive variant". *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* ***35***(6), pp. 1272–1282.

[19] Montes de Oca, M. A., Stutzle, T., Birattari, M., and Dorigo, M., 2009. "Frankenstein's pso: A composite particle swarm optimization algorithm". *IEEE Transactions on*

Evolutionary Computation, ***13***(5), pp. 1120–1132.

[20] Hu, M., Wu, T., and Weir, J. D., 2013. "An adaptive particle swarm optimization with multiple adaptive methods". *IEEE Transactions on Evolutionary Computation,* ***17***(5), pp. 705–720.

[21] Xinchao, Z., 2010. "A perturbed particle swarm algorithm for numerical optimization". *Applied Soft Computing,* ***10***(1), pp. 119 – 124.

[22] Chen, Y., Li, L., Peng, H., Xiao, J., Yang, Y., and Shi, Y., 2017. "Particle swarm optimizer with two differential mutation". *Applied Soft Computing,* ***61***, pp. 314 – 330.

[23] Jana, B., Mitra, S., and Acharyya, S., 2019. "Repository and mutation based particle swarm optimization (rmpso): A new pso variant applied to reconstruction of gene regulatory network". *Applied Soft Computing,* ***74***, pp. 330 – 355.

[24] Wang, F., Zhang, H., Li, K., Lin, Z., Yang, J., and Shen, X.-L., 2018. "A hybrid particle swarm optimization algorithm using adaptive learning strategy". *Information Sciences,* ***436-437***, pp. 162 – 177.

[25] İbrahim Berkan Aydilek, 2018. "A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems". *Applied Soft Computing,* ***66***, pp. 232 – 249.

[26] Kennedy, J., and Mendes, R., 2006. "Neighborhood topologies in fully informed and best-of-neighborhood particle swarms". *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* ***36***(4), pp. 515–519.

[27] Ratnaweera, A., Halgamuge, S. K., and Watson, H. C., 2004. "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients". *IEEE Transactions on Evolutionary Computation,* ***8***(3), pp. 240–255.

[28] Eberhart, R. C., and Yuhui Shi, 2001. "Tracking and optimizing dynamic systems with particle swarms". In Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546), Vol. 1, pp. 94–100 vol. 1.