Robust Utility Design in Distributed Resource Allocation Problems with Defective Agents

Bryce L. Ferguson & Jason R. Marden

***Abstract*— The use of multi-agent systems to solve large-scale problems can be an effective method to reduce physical and computational burdens; however, these systems should be robust to sub-system failures. In this work, we consider the problem of designing utility functions, which agents seek to maximize, as a method of distributed optimization in resource allocation problems. Though recent work has shown that optimal utility design can bring system operation into a reasonable approximation of optimal, our results extend the existing literature by investigating how robust the system's operation is to defective agents and by quantifying the achievable performance guarantees in this setting. Interestingly, we find that there is a trade-off between improving the robustness of the utility design and offering good nominal performance. We characterize this trade-off in the set of resource covering problems and find that there are considerable gains in robustness that can be made by sacrificing some nominal performance.**

I. Introduction

Multi-agents systems have emerged as a viable method of implementing distributed system operation. In teams of robots [1], [2], resource allocation problems [3], [4], autonomous mobility and delivery [5], [6], and many other large-scale systems, distributing certain decision making processes to individual agents can help reduce computational and communication burdens. Designing local control laws for agents that guarantee good system performance overall can be difficult; a promising method of solving this problem is by using tools from game theory to describe the system operation [7]. A system designer can assign each agent a local objective or utility function which they seek to maximize. By carefully designing these utility functions, the system designer can guarantee that the emergent system behavior is within a good approximation of the system optimal [8], [9]. Though the use of game theoretic techniques in distributed control is encouraging, the robustness of these utility designs is not well understood, and certain sub-system failures may cause significant degradation in the system performance.

In this work, we consider the problem of utility design in resource allocation problems in which each agent selects a set of resources (or tasks) with the objective of maximizing the system welfare through the agents' collective actions. To determine a preference over actions, each agent is assigned a local objective (or utility) function which they seek to maximize by repeatedly updating their action; the equilibria of this process are the Nash equilibria of the game formed between the agents with their payoffs described by their local objectives. By designing the utility rules of the agents, a system operator can alter the Nash equilibria and improve the equilibrium performance guarantees. The problem of utility design has been studied in many settings of resource allocation problem [10]–[13] and has been proven to be effective at offering reasonable approximations of the optimal system welfare. In this work, we seek to understand the robustness of utility rules to sub-system failures by investigating the impact of defective agents on performance guarantees.

Previous work has considered how sub-system failures can affect the operation of a distributed control system. In [14], the authors look at how performance of greedy submodular-maximization degrades as communication channels between agents are removed and find that certain design modifications can be beneficial. The authors of [15] introduce a framework to discuss the robustness properties of log-linear learning: a process in which users' action updates are noisy best responses. They find that the presence of a single heterogeneous agent can alter the long-term group behavior but do not discuss the impact this agent has on system welfare. In [16], the authors consider the effect of heterogeneous agents on opinion dynamics in networked systems and their impact on group consensus. In this work, we not only consider how these types of agents affect the system operation and equilibria but also how they impact the system welfare and achievable performance guarantees of a utility design. We further consider how existing design techniques can be modified to offer improved robustness to defective agents.

The results of this work give insight on the robustness of game theoretic techniques of distributed control in resource allocation problems with defective agents. We consider two types of defective agents: (1) stubborn agents which do not update their actions and do not contribute to the system welfare but do alter the agents' perceived utilities, and (2) failure-prone agents that have a probability of failing and leaving the system entirely. In either setting, the presence of these defective agents will alter the intended system operation and worsen system performance guarantees. Our first result, focused in the setting of resource covering problems, shows that significant performance improvements are attainable by designing utility rules more robustly. Interestingly, these design modifications necessarily reduce the systems nominal performance and Theorem 1 quantifies a trade-off between the achievable nominal and robust performance guarantees. In Section III-B, we find that a similar trade-off between robustness and nominal performance exists when agents are failure prone. Finally, in Section IV, we provide

This research was supported by ONR grant #N00014-20-1-2359 and AFOSR #FA9550-20-1-0054

B. L. Ferguson and J. R. Marden are with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA, {blferguson, jrmarden}@ece.ucsb.edu.

a tractable linear program which can be used to compute the optimal utility rules in the presence of either type of defective agents for a more general class of welfare functions.

II. Preliminaries

Before discussing the specific setting of resource allocation, we offer a general overview of the use of utility design as a method of distributed control in multi-agent systems. To define a multi-agent system, let $N = \{1, \ldots, n\}$ denote a finite set of agents whose interactions comprise the operation of the system. Each agent $i \in N$ has a set of allowable actions $\mathcal{A}_i$ that they can employ. When each agent $i \in N$ has selected an action $a_i \in \mathcal{A}_i$, we denote the joint action or allocation of users decisions by the tuple $a = (a_1, \ldots, a_n) \in \mathcal{A} = \mathcal{A}_1 \times \ldots \times \mathcal{A}_n$. To quantify the efficacy of system operation, each joint action is mapped to a global objective value via a system welfare function $W : \mathcal{A} \to \mathbb{R}$.

The objective of a system designer is to select an allocation of agents actions that maximizes the system welfare $a^{\mathrm{opt}} \in \arg\max_{a \in \mathcal{A}} W(a)$; however, finding and executing these strategies via a centralized control process can be difficult and intractable [17]. A popular and effective method of designing these systems to offer good system welfare is to implement distributed control algorithms using game theoretic techniques [7], [8]. With this approach, each agent $i \in N$ is assigned a utility function $U_i : \mathcal{A} \to \mathbb{R}$ which they seek to maximize. By letting agents repeatedly update their actions, the system will reach a steady state when no user can improve their utility by unilaterally altering their action, i.e., a state $a^{\mathrm{Ne}} \in \mathcal{A}$ that satisfies

$$U_i(a_i^{\mathrm{Ne}}, a_{-i}^{\mathrm{Ne}}) \geq U_i(a_i', a_{-i}^{\mathrm{Ne}}), \quad \forall a_i' \in \mathcal{A}_i,\ i \in N, \tag{1}$$

where a_{-i} denotes the action of all players excluding player i. A state a^{Ne} that satisfies (1) is termed a *Nash equilibrium*. Though these states are equilibrium in the described action revision process, they need not maximize the system welfare.

The efficiency of a Nash equilibrium can be described by the ratio between the equilibrium system welfare and the optimal system welfare. For a game described by the tuple $(N, \mathcal{A}, \{U_i\}_{i \in N})$, the performance guarantee for the utility design control scheme $\{U_i\}_{i \in N}$ can be measured by the lower bound on this performance ratio. This performance guarantee is termed the *price of anarchy*, and can formally be defined as

$$\mathrm{PoA}(N, \mathcal{A}, \{U_i\}_{i \in N}) = \min_{a \in \mathrm{NE}(N, \mathcal{A}, \{U_i\}_{i \in N})} \frac{W(a)}{W(a^{\mathrm{opt}})}, \tag{2}$$

where $\mathrm{NE}(N, \mathcal{A}, \{U_i\}_{i \in N})$ is the set of Nash equilibria in the game. A system designer who seeks to improve performance in this multi-agent system can design the utility rules to alter the interactions between agents, with the goal of promoting more efficient Nash equilibria. The effectiveness of utility design has been studied in many settings, and has been shown to be effective at providing a reasonable approximation for the system optimal [4], [10]. This work is focused on understanding the robustness of these game theoretic methods of distributed control to sub-system failures in the form of defective agents. As such, we seek to understand the impact of defective agents on the price of anarchy and what improvements are possible when agents' utilities are designed with the knowledge that these hazards may exist.

III. Resource Covering Problems

We focus our study on the robustness of utility design to a setting of resource allocation problems. In these problems, agents actions are to utilize or cover a portion of a larger group of resources $\mathcal{R}$; each agents set of allowable actions are subsets of the set of resources, i.e., $\mathcal{A}_i \subset 2^{\mathcal{R}}$ for each $i \in N$. Each resource has a value v_r when being utilized by at least one agent, thus the system welfare can be written as

$$W(a) = \sum_{\{r \in \mathcal{R} : |a|_r > 0\}} v_r, \tag{3}$$

where $|a|_r$ denotes the number of agents utilizing resource r in allocation a. A resource covering problem can be described by the tuple $G = (N, \mathcal{A}, \mathcal{R}, \{v_r\}_{r \in \mathcal{R}})$. In a resource covering problem G, a system designer thus desires to find an allocation that maximizes the value of covered resources, however, finding these allocations is an NP-hard problem in general [17]. Thus, a system designer may choose to implement a utility-based distributed control scheme to provide adequate, attainable system performance.

One effective and computationally efficient method of designing agents' utility functions is to implement a *local utility rule* $f : \{0, \ldots, n\} \to \mathbb{R}_{\geq 0}$ that describes how agents should evaluate overlap with other agents. When a system designer chooses a local utility rule f, an agent i has payoff function

$$U_i(a) = \sum_{r \in a_i} v_r \cdot f(|a|_r). \tag{4}$$

By designing the local utility rule f, the system designer can alter how agents perceive local interactions with other agents.

When agents payoffs are described as in (4), the tuple (G, f) comprises a game as discussed in Section II, the Nash equilibria of which are denoted $\mathrm{NE}(G, f)$. The performance of the local utility rule f at maximizing the welfare in the resource covering problem G can be measured by the price of anarchy of the game (G, f), written formally as $\mathrm{PoA}(G, f) = \min_{a \in \mathrm{NE}(G,f)} W(a)/W(a^{\mathrm{opt}})$. To quantify the performance guarantee of a local utility rule f in general, we let $\mathcal{G}$ denote the set of all possible resource covering problems; the price of anarchy is extended to be the worst-case inefficiency of the local utility rule f, i.e., $\mathrm{PoA}(\mathcal{G}, f) = \min_{G \in \mathcal{G}} \mathrm{PoA}(G, f)$. The local utility design problem of maximizing the worst-case inefficiency guarantees can be described as

$$f^{\mathrm{opt}} \in \underset{f : [n] \to \mathbb{R}_{\geq 0}}{\arg\max} \min_{G \in \mathcal{G}} \mathrm{PoA}(G, f), \tag{5}$$

where $[n] = \{0, \ldots, n\}$. Finding the optimal local utility rule is non-trivial. However, in [10] the authors find that the local utility rule

$$f^0(j) := (j-1)! \frac{\frac{1}{(n-1)(n-1)!} + \sum_{i=j}^{n-1} \frac{1}{i!}}{\frac{1}{(n-1)(n-1)!} + \sum_{i=1}^{n-1} \frac{1}{i!}}, \tag{6}$$

is a solution to (5) and provides a price of anarchy guarantee of $1 - \frac{1}{e}$. In this work, we investigate how performance guarantees like this are impacted by the presence of defective agents. Additionally, we ask if appropriate design modifications to the local utility rule can make the system more robust to defective agents and what impact these design modifications have on the systems nominal performance.

A. Stubborn Agents

The first type of defective agent we consider are stubborn agents, where the agent no longer has the ability to update their action and does not contribute to the system operation but does influence the utilities of other agents in the system. In this work, we investigate how these agents affect the achievable system welfare guarantees of local utility design.

We consider a setting where there exist a finite number of stubborn agents in the system. If there exist no more than m defective agents, each resource $r \in \mathcal{R}$ can be utilized by at most m stubborn agents $d_r \in [m]$. An allocation of stubborn agents can be denoted by the tuple $d \in [m]^{|\mathcal{R}|}$. These agents do not contribute to the system objective, so the welfare remains the total value of resources covered by regular agents, i.e.,

$$W(a;d) = \sum_{\{r \in \mathcal{R} : |a|_r > 0\}} v_r.$$

The system welfare is thus the same as if there were no stubborn agents, i.e., $W(a;d) = W(a;\emptyset)$. However, the presence of stubborn agents does affect the utility functions of the regular agents and thus their decision making process. The regular agents are unable to differentiate between stubborn agents and other regular agents, thus the utility of a player $i \in N$, under local utility rule f and defective agents allocation d, is determined by the total number of regular and defective agents on each resource, or

$$U_i(a;d) = \sum_{r \in a_i} v_r \cdot f(|a|_r + d_r). \tag{7}$$

The presence of stubborn agents will alter how agents make decisions and can have a detrimental effect on the quality of the system's Nash equilibria. In the game (G, d, f) that is formed when users follow the utility functions in (7), we measure the performance of the local utility rule f by the price of anarchy $\mathrm{PoA}(G, d, f)$ over the new set of equilibria $\mathrm{NE}(G, d, f)$.

The robust utility design problem can be described by finding the local utility rule that maximizes the worst-case price of anarchy guarantee over the class of resource covering problems with defective agents $\mathcal{G}^m = \{(G, d) \mid G \in \mathcal{G},\ d \in [m]^{|\mathcal{R}|}\}$. An optimal utility rule in this setting satisfies

$$f^{\mathrm{opt}} \in \underset{f:[n+m]\to\mathbb{R}_{\geq 0}}{\arg\max} \min_{G \in \mathcal{G}^m} \min_{d \in [m]^{|\mathcal{R}|}} \mathrm{PoA}(G, d, f). \tag{8}$$

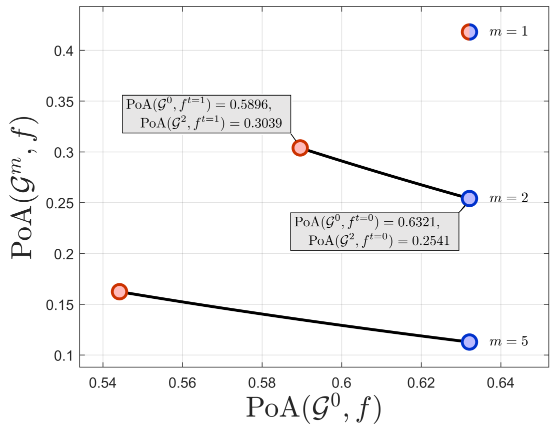


Fig. 1: Achievable price of anarchy guarantees in the nominal setting $\mathcal{G}^0$ and in the presence of m stubborn agents $\mathcal{G}^m$. Each line represents a Pareto-optimal frontier for the achievable performance guarantee in each setting for a specific m. The left (red) endpoints represent the price of anarchy guarantees of the optimal, robust utility rule $f^{t=1}$ and the right (blue) endpoints represent the price of anarchy guarantees of the optimal, nominal utility rule $f^{t=0}$. A system designer is only capable of offering joint performance guarantees that are on the line connecting the endpoints or lower.

In Theorem 1 we show that significant improvements in performance are attainable in the presence of stubborn agents by designing the local utility rule more robustly; however, we also find that these design modifications necessarily reduce the effectiveness of the local utility rule in the nominal setting and that there exists a trade-off between offering good robust and nominal performance. Theorem 1 quantifies this trade-off between robustness to stubborn agents and nominal performance in terms of price of anarchy guarantees.

Theorem 1. *For a local utility rule f to achieve a robust price of anarchy guarantee of*

$$\mathrm{PoA}(\mathcal{G}^m, f) \geq \frac{\Gamma_m + \frac{e}{e-1}}{1 + t\Gamma_m}, \tag{9}$$

where $\Gamma_m = m! \frac{e - \sum_{i=0}^{m-1} \frac{1}{i!}}{e-1} - 1$ and $t \in [0,1]$, its nominal price of anarchy guarantee will be no better than

$$\mathrm{PoA}(\mathcal{G}^0, f) \leq \frac{(e-1)(1+t\Gamma_m)}{1 + (e-1)(1+t\Gamma_m)}. \tag{10}$$

Further, these price of anarchy guarantees can be jointly realized by a local utility rule

$$f^t(j) = f^0(j) - \max\left\{ t \left(f^0(j) - \frac{m}{j} f^0(m) \right), 0 \right\}. \tag{11}$$

The proof of Theorem 1 appears in the appendix.

The trade-off described in (9) and (10) is depicted in Fig. 1 for several values of m. The horizontal axis measures the nominal price of anarchy and the vertical axis measures the price of anarchy when there are at most m stubborn agents. The line drawn by varying $t \in [0,1]$ constitutes a Pareto-optimal frontier on the multi-objective problem of maximizing the nominal and robust performance guarantees.

By letting $t = 0$, we can evaluate the performance of the optimal, nominal utility rule $f^{t=0} = f^0$ defined in (6).

Clearly, the performance degrades as more stubborn agents are introduced into the problem: the presence of two stubborn agents reduces the performance of the nominal utility rule $f^{t=0}$ by almost 60% down to $\mathrm{PoA}(\mathcal{G}^{m=2}, f^{t=0}) = 0.2541$. By designing the utility rule more robustly, the price of anarchy guarantee in $\mathcal{G}^{m=2}$ can be improved by almost 20% by using $f^{t=1}$; however, this increase in robustness comes at the cost of nominal performance, as the local utility rule $f^{t=1}$ is approximately 7% less efficient than the optimal in the nominal setting. A system designer who would like to optimize both performance metrics can provide guarantees only up to the Pareto-optimal frontier described by (9) and (10) and shown in Fig. 1; these Pareto-optimal performance guarantees can be achieved by using f^t for $t \in [0, 1]$.

B. Failure-Prone Agents

The second type of defective agent we consider are failure prone agents, where each agent operates normally but has a probability of failing and exiting the problem entirely. Every agent will follow the designed utility rule but has a chance of failing and no longer contributing to the system welfare. In a resource allocation problem G and local utility rule f, each agent will follow their best response dynamic until the system reaches a Nash equilibrium $a^{\mathrm{Ne}} \in \mathrm{NE}(G, f)$. After the system settles, each agent independently fails with probability p. In an allocation a, a resource $r \in \mathcal{R}$, utilized by $|a|_r$ agents before failure, has $X_r \leq |a|_r$ non-failed agents remaining with probability

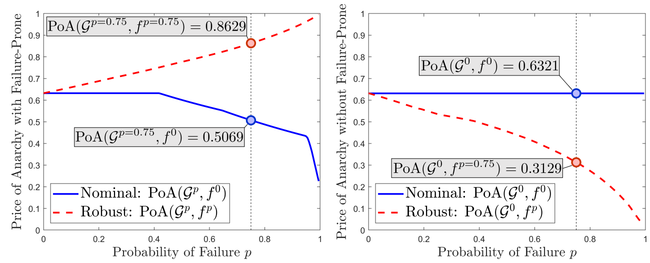
$$\mathbb{P}[X_r = x] = \binom{|a|_r}{x} (1-p)^x \; p^{|a|_r - x}.$$

The expected system welfare in an allocation a is thus

$$W(a) = \sum_{\{r \in \mathcal{R} : |a|_r > 0\}} v_r \cdot \left(1 - p^{|a|_r}\right), \tag{12}$$

and the price of anarchy when agents are failure-prone $\mathrm{PoA}(G, p, f)$ is thus the worst-case ratio between the expected system welfare in a Nash equilibrium and the optimal expected system welfare. The worst-case performance guarantee is a lower bound on the price of anarchy over resource covering games with a probability of failure p, i.e., $\mathcal{G}^p = \{(G, p) \mid G \in \mathcal{G},\ p \in [0, 1]\}$. As described in Section III-A, the optimal local utility design problem can be described as finding an f that maximizes $\mathrm{PoA}(\mathcal{G}^p, f)$; we will denote the optimal local utility rule in the case that agents fail with probability p by f^p.

The form of the expected utility function in (12) motivates the need to look at resource allocation problems beyond covering problems, where the welfare contributed to the system may depend on the number of users utilizing a resource. In [11], the authors provide a linear program whose solution is the optimal utility rule for some local welfare function. Using this result, which is generalized in Theorem 2 to include stubborn agents, we can compute and evaluate the optimal utility rule for the class of resource covering problems failure-prone agents $\mathcal{G}^p$.



(a) Failure-prone (b) Not Failure-Prone

Fig. 2: (Left) Price of anarchy in covering problems with failure prone agents under nominal utility rule f^0 and robust utility rule f^p. The robust utility rule offers significant performance improvements in the presence of failure prone agents. (Right) Price of anarchy in covering problems under nominal utility rule f^0 and robust utility rule f^p designed for agents failing with probability p. The robust utility rule sacrifices notable performance in the original setting.

In Fig. 2, we analyze the performance trade-off between designing utility rules for the nominal and failure-prone settings. For each $p \in [0, 1]$, we compute the optimal utility rule f^p that is robust to agent failure with probability p using (14) with $m = 0$. We then compare the performance of this new, robust utility rule with the performance of the nominal utility rule f^0, defined in (6), when agents are failure-prone and when agents are not failure-prone. Fig. 2 shows the price of anarchy guarantees of the nominal and robust utility rules in presence of failure-prone agents. When the probability of failure is large, the robust utility rule offers large improvements to the expected system welfare; when $p = 0.75$, the robust utility rule offers a price of anarchy guarantee of $\mathrm{PoA}(\mathcal{G}^{p=0.75}, f^{p=0.75}) = 0.8629$ which is a 63% increase from the performance of the nominal utility rule f^0 in this setting. However, as seen in Fig. 2b, if the system designer is incorrect and the agents are not failure-prone, the use of the robust utility rule causes a loss in performance; when $p = 0.75$ the price of anarchy of the robust utility rule without failure prone agents is $\mathrm{PoA}(\mathcal{G}^0, f^{p=0.75}) = 0.3129$, which is a 49% decrease from if the nominal had been used. This difference in price of anarchy guarantees again highlights a trade-off between the achievable nominal and robust performance.

IV. Beyond Covering Problems

In Section III-B, focused on the utility design problem in the face of failure-prone agents, it was shown that consider more general system welfare structures can be useful in describing richer classes resource allocation problems. In this section, we generalize existing results from the literature to compute optimal local utility rules in the presence of stubborn or defective agents for a more general class of system welfare functions. Let $w : \mathbb{Z}_{\geq 0} \to \mathbb{R}_{\geq 0}$ denote a local welfare function such that $v_r \cdot w(k)$ dictates the contribution to the system welfare of the resource r when being utilized by k agents. The system welfare can now be written

$$W(a) = \sum_{r \in \mathcal{R}} v_r \cdot w(|a|_r).$$

In resource covering problems, $w(x)$ can be defined as the indicator function that the arguement is greater than zero, Section III-B showed a local welfare function for maximizing expected welfare in the presence of failure-prone agents, while other examples of these local welfare functions that have been studied in the literature include sub-modular functions [14], [18], which model the diminishing returns of adding agents to a resource.

A class of resource allocation problems with local welfare function w will be denoted $\mathcal{G}(w)$; the task of finding local utility rules that minimize the price of anarchy for these classes of problems is discussed in [11]. In this work, we are interested in amending these design techniques to improve the performance in the presence of defective agents. To accomplish this, in Theorem 2 we generalize the linear program from [11] to compute the optimal local utility rule in the face of stubborn agents.

Proposition 2. *Let $w \in \mathbb{R}^{n+m}_{\geq 0}$ be a local welfare function for the set of resource allocation games $\mathcal{G}^m(w)$, with at most n regular agents and m stubborn agents. The local utility design problem*

$$f^{\mathrm{opt}} \in \underset{f \in \mathbb{R}^{n+m}}{\arg\max}\ \mathrm{PoA}(\mathcal{G}^m(w), f) \tag{13}$$

is equivalent to the following linear program

$$\begin{aligned}(f^{\mathrm{opt}}, \mu^{\mathrm{opt}}) \in\ & \underset{f \in \mathbb{R}^n,\ \mu \in \mathbb{R}}{\arg\max}\ \mu \\ & s.t.\ w(z+y) - \mu w(x+y) \\ & \quad + x f(x+y+d) - z f(x+y+d+1) \leq 0 \\ & \quad \forall (x,y,z) \in \mathcal{I}_n,\ d \in \{0, \dots, m\},\end{aligned} \tag{14}$$

where $I_n = \{(x,y,z) \in \mathbb{N}^3_{\geq 0} \mid 1 \leq x+y+z \leq n\}$. Additionally, the price of anarchy associated with f^{opt} is $\mathrm{PoA}(\mathcal{G}^m(w), f^{\mathrm{opt}}) = 1/\mu^{\mathrm{opt}}$.

The linear program (14) is a generalization of [11, Theorem 4] which solves the local utility design problem in the nominal setting and is recovered when $m = 0$. The parameterization using tuples of the form (x, y, z, d) is described in the appendix. The proof is omitted for brevity; the claim can be verified by following the same proof as in [11]. By noting that the game with stubborn agents is a potential game with potential function $\Phi(a, \overline{a}) = \sum_{r \in \mathcal{R}} \sum_{k=|\overline{a}|_r}^{|a|_r + |\overline{a}|_r} v_r f(k)$, it can be shown that the price of anarchy bound found from the solution to (14) is tight. Additionally, by fixing the vector f in (14), the solution to the same linear program can evaluate the price of anarchy guarantee of a given local utility rule.

Theorem 2 provides a tool for computing local utility rules that are robust to stubborn agents for general local welfare functions. Additionally, we can amend the local welfare function and use the same linear program to compute the optimal local utility rule in the face of failure prone agents.

Corollary 1. *In a class of resource allocation problems $\mathcal{G}^p(w)$, with welfare function w, the optimal local utility rule, when agents have a probability of failure p, can be found by solving (14) with $m = 0$ and the amended welfare function*

$$\overline{w}(x) = \sum_{k=0}^{x} w(k) \binom{x}{k} (1-p)^k\ p^{x-k}. \tag{15}$$

The solution to (14) will provide the optimal local utility rule as well as the associated price of anarchy guarantee.

As is the case with stubborn agents, design modifications need to be made to the local utility rule to improve robustness to failure prone agents. The results of this section provide tractable linear programs that can be used to compute robust utility designs in the presence of defective agents.

V. Conclusion

This work studies the robustness of local utility rules to sub-system failures in the form of stubborn and failure-prone agents. We provide linear programs that compute and evaluate the optimal local utility rules in the face of these defective agents. Our results show that there is a trade-off in designing utility rules that are robust and that give good nominal performance, which is characterized for the setting of covering problems.

References

[1] H. Kitano, S. Tadokoro, I. Noda, H. Matsubara, T. Takahashi, A. Shinjou, and S. Shimada, "RoboCup Rescue: search and rescue in large-scale disasters as a domain for autonomous agents research," in *IEEE SMC'99 Conference Proceedings.*, vol. 6, oct 1999, pp. 739–743 vol.6.

[2] C. E. Nugraheni and L. Abednego, "Multi Agent Hyper-Heuristics based framework for production scheduling problem," in *International Conference on Informatics and Computing*, oct 2016, pp. 309–313.

[3] R. Brown, F. Rossi, K. Solovey, M. Tsao, M. T. Wolf, and M. Pavone, "On Local Computation for Network-Structured Convex Optimization in Multi-Agent Systems," *IEEE Transactions on Control of Network Systems*, p. 1, 2021.

[4] J. R. Marden, "The Role of Information in Distributed Resource Allocation," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 3, pp. 654–664, 2017.

[5] F. Bullo, E. Frazzoli, M. Pavone, K. Savla, and S. L. Smith, "Dynamic Vehicle Routing for Robotic Systems," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1482–1504, 2011.

[6] L. Bertizzolo, S. D'Oro, L. Ferranti, L. Bonati, E. Demirors, Z. Guan, T. Melodia, and S. Pudlewski, "SwarmControl: An Automated Distributed Control Framework for Self-Optimizing Drone Networks," in *IEEE INFOCOM 2020*, jul 2020, pp. 1768–1777.

[7] J. S. Shamma, *Cooperative control of distributed multi-agent systems.* Wiley Online Library, 2007.

[8] N. Li and J. R. Marden, "Designing games for distributed optimization," *IEEE Journal on Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 230–242, 2013.

[9] R. Chandan, D. Paccagnan, and J. R. Marden, "Optimal mechanisms for distributed resource-allocation," *CoRR*, vol. abs/1911.0, 2019.

[10] M. Gairing, "Covering games: Approximation through non-cooperation," in *Internet and Economics*, 2009, pp. 184–195.

[11] D. Paccagnan, R. Chandan, and J. R. Marden, "Distributed resource allocation through utility design - Part {I:} optimizing the performance certificates via the price of anarchy," *CoRR*, vol. abs/1807.0, 2018.

[12] D. Paccagnan and J. R. Marden, "Utility Design for Distributed Resource Allocation – Part II: Applications to Submodular, Covering, and Supermodular Problems," 2021.

[13] A. Vetta, "Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions," *IEEE Symposium on Foundations of Computer Science, 2002.*, pp. 416–425, 2002.

[14] D. Grimsman, M. S. Ali, J. P. Hespanha, and J. R. Marden, "Impact of information in greedy submodular maximization," *IEEE Conference on Decision and Control, 2017*, vol. 2018-Jan, pp. 2900–2905, 2018.

[15] H. Jaleel, W. Abbas, and J. S. Shamma, "Robustness Of Stochastic Learning Dynamics To Player Heterogeneity In Games," in *2019 IEEE 58th Conference on Decision and Control*, dec 2019, pp. 5002–5007.
[16] E. Yildiz, D. Acemoglu, A. Ozdaglar, A. Saberi, and A. Scaglione, "Discrete Opinion Dynamics with Stubborn Agents," *Public Choice: Analysis of Collective Decision-Making eJournal*, 2011.
[17] D. S. Hochba, "Approximation Algorithms for NP-Hard Problems," *ACM SIGACT News*, vol. 28, no. 2, pp. 40–52, jun 1997.
[18] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions-I," *Mathematical Programming*, 1978.
[19] T. Roughgarden, "Intrinsic robustness of the price of anarchy," *Communications of the ACM*, vol. 55, no. 7, pp. 116–123, 2012.

APPENDIX

In this appendix, we give the full proof of Theorem 1 as well as several supporting lemmas. In looking for price of anarchy bounds we note that a class of resource covering problems $\mathcal{G}$ with utility rule f has the same price of anarchy as the class of problems $\mathcal{G}^*$ where each agent has exactly two actions $\mathcal{A}_i = \{a_i^{\mathrm{Ne}}, a_i^{\mathrm{opt}}\}$,[1] thus we will search for price of anarchy bounds in these two-action games and note they hold more generally. The price of anarchy bounds in (9) and (10) are tight along the Pareto-optimal frontier. To prove that each is an upper bound, we will make use of several examples; three structures of parameterized problem instances are shown in Fig. 3a, Fig. 3b, and Fig. 3c. To show that these are lower bounds, we will make use of smoothness inequalities introduced in [19]. If, given a utility rule f, each Nash equilubria $a^{\mathrm{Ne}} \in \mathrm{NE}(G, f)$ satisfies

$$W(a^{\mathrm{Ne}}) \geq \lambda \cdot W(a^{\mathrm{opt}}) + \mu \cdot W(a^{\mathrm{Ne}}), \tag{16}$$

for some $\lambda, \mu \in \mathbb{R}$, then the price of anarchy will satisfy $\mathrm{PoA}(\mathcal{G}, f) \geq \frac{\lambda}{1-\mu}$. We will provide lower bounds by finding values of λ and μ for different settings (e.g., with and without stubborn agents); often, to do so, we will utilize the fact that the welfare of a Nash equilibria can be lower bounded by

$$W(a^{\mathrm{Ne}}) \geq \sum_{i\in N} u_i(a_i^{\mathrm{opt}}, a_{-i}^{\mathrm{Ne}}) - \sum_{i\in N} u_i(a^{\mathrm{Ne}}) + W(a^{\mathrm{Ne}}). \tag{17}$$

Additionally, using the prameterization discussed in Section IV, where, in an allocation $(a, \overline{a})$, each resource $r \in \mathcal{R}$ is given a label (x_r, y_r, z_r, d_r) defined by $x_r = |a^{\mathrm{Ne}} \backslash a^{\mathrm{opt}}|_r$, $z_r = |a^{\mathrm{opt}} \backslash a^{\mathrm{Ne}}|_r$, $y_r = |a^{\mathrm{opt}} \cap a^{\mathrm{Ne}}|_r$, and d_r is the number of stubborn agents. This parameterization allows us to write $W(a^{\mathrm{Ne}}) = \sum_{r\in\mathcal{R}} v_r \mathbb{1}_{[x_r+y_r]}$ and $W(a^{\mathrm{opt}}) = \sum_{r\in\mathcal{R}} v_r \mathbb{1}_{[y_r+z_r]}$; additionally, (17) can be rewritten as

$$W(a^{\mathrm{Ne}}) \geq \sum_{r\in\mathcal{R}} v_r [z_r f(x_r + y_r + d_r + 1) - x_r f(x_r + y_r + d_r) + \mathbb{1}_{[x_r+y_r]}], \tag{18}$$

where the welfare function $w(x) = \mathbb{1}_{[x]}$ is the indicator function that the argument is greater than zero in covering games. Manipulating the right hand side of (18) into the form of (16) will be the primary method of lower bounding the price of anarchy of a utility rule f in a class of games.

From [10], the optimal utility rule in covering games with no stubborn agents and arbitrarily many regular agents $\mathcal{G}^0$ is

$$f^0(j) := (j-1)! \frac{e - \sum_{i=0}^{j-1} \frac{1}{i!}}{e-1} \ \forall j \geq 1, \tag{19}$$

and $f^0(0) = 0$. This can also be seen by taking n to infinity in (6). The performance guarantee of f^0 is $\mathrm{PoA}(\mathcal{G}^0, f^0) = 1 - \frac{1}{e}$, which can be seen from the following lemma.

Lemma 1 (Gairing 2009 [10])**.** *In the class of problems $\mathcal{G}^0$, with utility rule*

$$f^0(j) = (j-1)! \frac{e - \sum_{i=0}^{j-1} \frac{1}{i!}}{e-1} \ \forall j \geq 1, \tag{20}$$

(16) *is satisfied with $\lambda = 1$ and $\mu = -1/(e-1)$.*

This utility rule is useful in constructing the optimal utility rules in the setting with stubborn agents. Additionally, the following claim is useful in proving several lower-bounds.

Claim 1. *The local utility rule f^0 defined in* (19) *satisfies*

$$j f^0(j) - f^0(j+1) = \frac{1}{e-1} \quad \forall\, j \in \mathbb{N}. \tag{21}$$

Proof. The claim can be proven directly by substitution:

$$\begin{aligned} jf^0(j) - f^0(j+1) &= j\left((j-1)! \frac{e - \sum_{i=0}^{j-1} \frac{1}{i!}}{e-1}\right) - (j)! \frac{e - \sum_{i=0}^{j} \frac{1}{i!}}{e-1} \\ &= \frac{j!}{e-1}\left(\sum_{i=0}^{j} \frac{1}{i!} - \sum_{i=0}^{j-1} \frac{1}{i!}\right) = \frac{1}{e-1}. \quad \square \end{aligned}$$

The following several lemmas will define and quantify the smoothness coefficients of some useful local utility rules.

Lemma 2. *In the class of problems $\mathcal{G}^m$, with utility rule*

$$\overline{f}^m(j) = \begin{cases} \frac{m}{j} f^0(m), & \text{if } j \in \{1, \dots, m\} \\ f^0(j), & \text{otherwise}, \end{cases} \tag{22}$$

(16) *is satisfied with $\lambda = f^0(m+1)$ and $\mu = 1 - m f^0(m)$.*

Proof. Let $\mathcal{R}_a \subset \mathcal{R}$ be the set of all resources where $x_r + y_r + d_r \geq m+1$ and let $\mathcal{R}_b \subset \mathcal{R}$ be the set of all resources where $x_r + y_r + d_r \leq m$, forming a partition of $\mathcal{R}$.

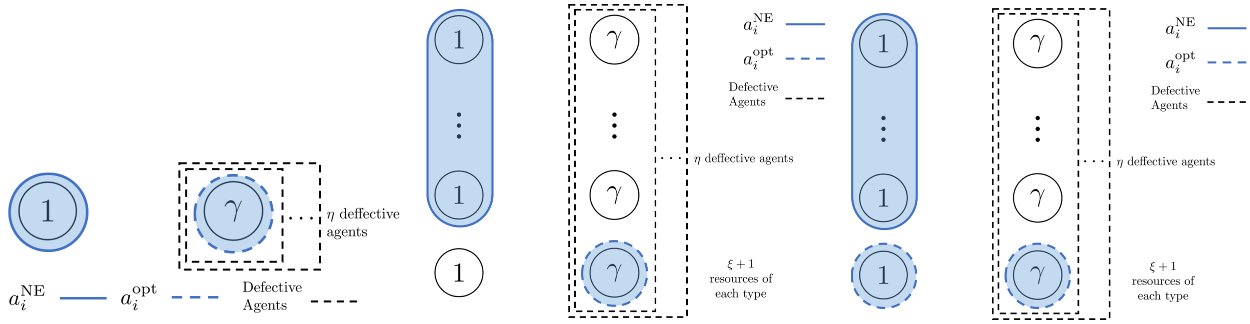
For the resources in $\mathcal{R}_a$,

$$\begin{aligned} &\sum_{r\in\mathcal{R}_a} v_r[z_r f^0(x_r+y_r+d_r+1) - x_r f^0(x_r+y_r+d_r)+1] \\ &\geq \sum_{r\in\mathcal{R}_a} v_r[(z_r + y_r) f^0(x_r + y_r + d_r + 1) - (x_r + y_r + d_r) f^0(x_r + y_r + d_r) + 1] && \text{(23a)} \\ &\geq \sum_{r\in\mathcal{R}_a} v_r[f^0(x_r + y_r + d_r + 1)\mathbb{1}_{[z_r+y_r]} - (x_r + y_r + d_r) f^0(x_r + y_r + d_r) + 1] && \text{(23b)} \\ &\geq \sum_{r\in\mathcal{R}_a} v_r[f^0(x_r+y_r+d_r+1)\mathbb{1}_{[z_r+y_r]} - f^0(x_r+y_r+d_r+1) - \frac{1}{e-1} + 1] && \text{(23c)} \\ &\geq \sum_{r\in\mathcal{R}_a} v_r[f^0(m+1)\mathbb{1}_{[z_r+y_r]} - f^0(m+1) - \frac{1}{e-1} + 1] && \text{(23d)} \\ &= \sum_{r\in\mathcal{R}_a} v_r[f^0(m+1)\mathbb{1}_{[z_r+y_r]} + (1 - m f^0(m))\mathbb{1}_{[x_r+y_r]}], && \text{(23e)} \end{aligned}$$

where (23a) and (23d) hold from f^0 decreasing, (23b) holds from f^0 positive, and (23c) and (23e) hold from claim 1.

For the resources in $\mathcal{R}_b$,

[1] This can be seen by transforming each game $G \in \mathcal{G}$ into one with two actions by removing all actions but the worst equilibrium a^{Ne} and the optimal allocation a^{opt}. Because a^{Ne} remains a Nash equilibrium, the price of anarchy is unchanged.



(a) Example A: G^A
A problem instance with one agent having two choices: a resource with value one and a resource with value γ covered by η defective agents. When $\gamma \leq 1/f(\eta+1)$, the agent may pick the resource of value one in equilibrium leading to $\mathrm{PoA}(G^A, f) = \frac{1}{\gamma} \geq f(\eta+1)$.

(b) Example B: G^B
A problem with $\xi+1$ agents each with two choices: selecting ξ resources of value 1, a_i^{Ne}, or one resource of value γ with η defective agents, a_i^{opt}. The agents' equilibrium and optimal actions are distinct from one another, implying in the allocation a^{Ne}, xi agents cover each resource of value 1, and in a^{opt}, each resource of value γ is covered by one agent. When $\gamma \leq \frac{\xi f(\xi)}{f(\eta+1)}$, a^{Ne} is an equilibrium allocation with $\mathrm{PoA}(G^B, f) = \frac{1}{\gamma}$.

(c) Example C: G^C
A problem with $\xi+1$ agents each with two choices: selecting ξ resources of value 1, a_i^{Ne}, or the remaining resource of value 1 and one resource of value γ with η defective agents, a_i^{opt}. The agents' equilibrium and optimal actions are distinct from one another, implying in the allocation a^{Ne}, ξ agents cover each resource of value 1, and in a^{opt}, every resource is covered by one agent. When $\gamma \leq \frac{\xi f(\xi) - f(\xi+1)}{f(\eta+1)}$, a^{Ne} is an equilibrium allocation with $\mathrm{PoA}(G^C, f) = \frac{1}{1+\gamma}$.

$$\begin{aligned}
&\sum_{r\in\mathcal{R}_b} v_r[z_r\overline{f}^m(x_r+y_r+d_r+1) - x_r\overline{f}^m(x_r+y_r+d_r) + \mathbb{1}_{[x_r+y_r]}] \\
&= \sum_{r\in\mathcal{R}_b} v_r\Big[\frac{z_r}{x_r+y_r+d_r+1}(mf^0(m)) \\
&\qquad - \frac{x_r}{x_r+y_r+d_r} mf^0(m) + \mathbb{1}_{[x_r+y_r]}\Big] \qquad (24a)\\
&\geq \sum_{r\in\mathcal{R}_b} v_r\Big[(z_r+y_r)f^0(m+1) \\
&\qquad - \frac{x_r}{x_r+y_r+d_r}\big(f^0(m+1)+\frac{1}{e-1}\big) + \mathbb{1}_{[x_r+y_r]}\Big] \qquad (24b)\\
&\geq \sum_{r\in\mathcal{R}_b} v_r\Big[f^0(m+1)\mathbb{1}_{[z_r+y_r]} + \big(1-f^0(m+1)-\frac{1}{e-1}\big)\mathbb{1}_{[x_r+y_r]}\Big] \qquad (24c)\\
&= \sum_{r\in\mathcal{R}_b} v_r[f^0(m+1)\mathbb{1}_{[z_r+y_r]} + (1-mf^0(m))\mathbb{1}_{[x_r+y_r]}]. \qquad (24d)
\end{aligned}$$

where (24b) holds from claim 1 and (24c) holds from $x_r/(x_r+y_r+d_r) \leq 1$, providing the same lower bound for the price of anarchy.

It follows that $\lambda = f^0(m+1)$ and $\mu = 1 - mf^0(m)$ satisfy (16). ☐

Lemma 3. *In the class of problems $\mathcal{G}^0$, with utility rule*

$$\overline{f}^m(j) = \begin{cases} \frac{m}{j} f^0(m), & \text{if } j \in \{1,\dots,m\} \\ f^0(j), & \text{otherwise}, \end{cases} \qquad (25)$$

(16) is satisfied with $\lambda = mf^0(m)$ and $\mu = \frac{e-2}{e-1} - mf^0(m)$.

Proof. Let $\mathcal{R}_c \subset \mathcal{R}$ denote the set of resources where $x_r > 0$ or $y_r > 0$, and let $\mathcal{R}_d \subset \mathcal{R}$ be the set of resources where $x_r = y_r = 0$. First recall the bound from (23e) and (24d) that together give

$$\sum_{r\in\mathcal{R}} v_r[z_r\overline{f}^m(x_r+y_r+1) - x_r\overline{f}^m(x_r+y_r) + \mathbb{1}_{[x_r+y_r]}] \qquad (26)$$

$$\geq \sum_{r\in\mathcal{R}} v_r[f^0(m+1)\mathbb{1}_{[z_r+y_r]} + (1-mf^0(m))\mathbb{1}_{[x_r+y_r]}], \qquad (27)$$

in the special case where $d_r = 0$, as is the case for games the class $\mathcal{G}^0$. For the set $\mathcal{R}_c$,

$$\begin{aligned}
&\sum_{r\in\mathcal{R}_c} v_r[f^0(m+1)\mathbb{1}_{[z_r+y_r]} + (1-mf^0(m))\mathbb{1}_{[x_r+y_r]}] \\
&= \sum_{r\in\mathcal{R}_c} v_r\Big[\Big(mf^0(m)\mathbb{1}_{[z_r+y_r]} + \frac{1}{e-1}\Big) + (1-mf^0(m))\mathbb{1}_{[x_r+y_r]}\Big] \qquad (28a)\\
&\geq \sum_{r\in\mathcal{R}_c} v_r\Big[(mf^0(m))\mathbb{1}_{[z_r+y_r]} + \frac{1}{e-1} + (1-mf^0(m))\mathbb{1}_{[x_r+y_r]}\Big] \qquad (28b)\\
&= \sum_{r\in\mathcal{R}_c} v_r\Big[(mf^0(m))\mathbb{1}_{[z_r+y_r]} + \Big(\frac{e-2}{e-1} - mf^0(m)\Big)\mathbb{1}_{[x_r+y_r]}\Big], \qquad (28c)
\end{aligned}$$

where (28a) holds from claim 1, (28b) holds from $\mathbb{1}_{[x]} \leq x$ for all non-negative integer x, and (28c) holds from definition of $\mathcal{R}_c$ that $\mathbb{1}_{[x_r+y_r]} = 1$. For the remaining resources in $\mathcal{R}_d$,

$$\begin{aligned}
&\sum_{r\in\mathcal{R}_d} v_r[z_r\overline{f}^m(x_r+y_r+1) - x_r\overline{f}^m(x_r+y_r) + \mathbb{1}_{[x_r+y_r]}] \\
&= \sum_{r\in\mathcal{R}_d} v_r[z_r f^0(1)] \geq \sum_{r\in\mathcal{R}_d} v_r[z_r mf^0(m)] \qquad (29a)\\
&\geq \sum_{r\in\mathcal{R}_d} v_r\Big[(mf^0(m))\mathbb{1}_{[z_r+y_r]} + \Big(\frac{e-2}{e-1} - mf^0(m)\Big)\mathbb{1}_{[x_r+y_r]}\Big], \qquad (29b)
\end{aligned}$$

where (29a) holds from the definition of $\overline{f}^m$ and $mf^0(m) < f^0(1) = 1$, and (29b) holds from f^0 positive and $x_r = y_r = 0$. From (28c) and (29b), $\lambda = mf^0(m)$ and $\mu = \frac{e-2}{e-1} - mf^0(m)$ satisfy (16). ☐

Lemma 4. *In the class of problems $\mathcal{G}^m$, with utility rule*

$$f^0(j) = (j-1)!\frac{e - \sum_{i=0}^{j-1}\frac{1}{i!}}{e-1} \;\; \forall j \geq 1, \qquad (30)$$

(16) is satisfied with $\lambda = f^0(m+1)$ and $\mu = 0$.

Proof. As in Lemma 2, let $\mathcal{R}_a \subset \mathcal{R}$ be the set of all resources where $x_r + y_r + d_r \geq m+1$ and let $\mathcal{R}_b \subset \mathcal{R}$ be the set of all resources where $x_r + y_r + d_r \leq m$, forming a partition of $\mathcal{R}$. For the resources in the set $\mathcal{R}_a$, follow the steps of (23a)-(23e) and note that $jf^0(j) \leq 1$ for all j, therefore (23e) is further lower-bounded by

$$\sum_{r\in\mathcal{R}_a} v_r[\mathbb{1}_{[z_r+y_r]} f^0(m+1)]. \tag{31}$$

For the resources in $\mathcal{R}_b$,

$$\begin{aligned}
&\sum_{r\in\mathcal{R}_b} v_r[z_r f^0(x_r+y_r+d_r+1) - x_r f^0(x_r+y_r+d_r) + \mathbb{1}_{[x_r+y_r]}] \\
&= \sum_{r\in\mathcal{R}_b} v_r[(z_r+y_r) f^0(x_r+y_r+d_r+1) \\
&\qquad -(x_r+y_r) f^0(x_r+y_r+d_r) + \mathbb{1}_{[x_r+y_r]}] && \text{(32a)}\\
&\geq \sum_{r\in\mathcal{R}_b} v_r[(z_r+y_r) f^0(x_r+y_r+d_r+1)] && \text{(32b)}\\
&\geq \sum_{r\in\mathcal{R}_b} v_r[\mathbb{1}_{[z_r+y_r]} f^0(m+1)], && \text{(32c)}
\end{aligned}$$

where (32a) holds from f^0 decreasing, (32b) holds from $jf^0(j) \leq 1$ for all $j \geq 0$, and (32c) holds from $\mathbb{1}_{[x]} \leq x$ for all non-negative integer x. From (31) and (32c), $\lambda = f^0(m+1)$ and $\mu = 0$ satisfy (16). □

Proof of Theorem 1: To prove that the curve defined by (9) and (10) represent a Pareto-optimal frontier of the multi-criterion problem of minimizing $\mathrm{PoA}(\mathcal{G}^m, f)$ and $\mathrm{PoA}(\mathcal{G}^0, f)$, we first give a parameterized utility rule that draws the curve then show a tight lower and upper bound on it's price of anarchy, and finally show this utility rule is indeed Pareto-optimal. Let $f^t(j) = t\overline{f}^m(j) + (1-t)f^0(j)$ for some $t \in [0,1]$, be a local utility rule parameterized by $t \in [0,1]$. Through some rearanging, this is equivalent to (11). We will show the price of anarchy guarantees of this utility rule draw the Pareto-optimal frontier.

Part 1: Upper Bound We will give problem instances that upper bound the price of anarchy over the set $\mathcal{G}^m$ and $\mathcal{G}^0$ for the utility rule f^t. For the nominal price of anarchy, let $G^C \in \mathcal{G}^0$ be a covering game as described in Fig. 3c with $\eta = 0$, $\gamma = \frac{\xi f(\xi) - f(\xi+1)}{f(1)}$. By selecting $\xi \geq m+1$ agents in the game (where m is the number of defective agents for which f^t is designed), from claim 1

$$\gamma = \frac{1}{(e-1)f^t(1)} = \frac{1}{(e-1)(tmf^0(m) + (1-t))}.$$

Defining $\Gamma_m = mf^0(m) - 1 = m!\frac{e - \sum_{i=0}^{m-1}\frac{1}{i!}}{e-1} - 1$, the price of anarchy of the described game is

$$\mathrm{PoA}(G^C, f^t) = \frac{1}{1+\gamma} = \frac{(e-1)(1+t\Gamma_m)}{1+(e-1)(1+t\Gamma_m)}.$$

Because $G^C \in \mathcal{G}^0$, $\mathrm{PoA}(\mathcal{G}^0, f^t) \leq \mathrm{PoA}(G^C, f^t)$. For the price of anarchy in the perturbed agent setting, let $G^A \in \mathcal{G}^m$ be a covering game as described in Fig. 3a with $\eta = m$ and $\gamma = f^t(1)/f^t(m+1)$. From the definition of f^t and claim 1, the price of anarchy of this game, with utility rule f^t is

$$\mathrm{PoA}(G^A, f^t) = \frac{f^0(m+1)}{1 + t(mf^0(m) - 1)} = \frac{\Gamma_m + \frac{e}{e-1}}{1+t\Gamma_m}.$$

Because $G^A \in \mathcal{G}^m$, $\mathrm{PoA}(\mathcal{G}^m, f^t) \leq \mathrm{PoA}(G^A, f^t)$. This provides our upper bounds for the price of anarchy over $\mathcal{G}^0$ and $\mathcal{G}^m$ while using the utility rule f^t.

Part 2: Lower Bound To lower bound the price of anarchy, we again look for coefficients λ, μ that satisfy (16). From the definition of f^t, (18) can be rewritten

$$\begin{aligned}
W(a^{\mathrm{Ne}}) \geq \sum_{r\in\mathcal{R}} & t v_r[z_r\overline{f}^m(x_r+y_r+d_r+1) \\
& - x_r\overline{f}^m(x_r+y_r+d_r) + \mathbb{1}_{[x_r+y_r]}] \\
& + (1-t)v_r[z_r f^0(x_r+y_r+d_r+1) \\
& - x_r f^0(x_r+y_r+d_r) + \mathbb{1}_{[x_r+y_r]}],
\end{aligned} \tag{33}$$

where $\overline{f}^m$ is as defined in (22). For any game in $\mathcal{G}^m$, from Lemma 2 and Lemma 4, (33) can be lower bounded by

$$f^0(m+1)\cdot W(a^{\mathrm{opt}}) + t(1 - mf^0(m))\cdot W(a^{\mathrm{Ne}}),$$

producing for the lower bound on the price of anarchy of

$$\mathrm{PoA}(\mathcal{G}^m, f^t) \geq \frac{f^0(m+1)}{t(1-mf^0(m))} = \frac{\Gamma_m + \frac{e}{e-1}}{1+t\Gamma_m}. \tag{34}$$

For the price of anarchy over the nominal setting $\mathcal{G}^0$ with utility law f^t, (10) needs to be lower bounded for the case where $d_r = 0$ for all $r \in \mathcal{R}$. From Lemma 3 and Lemma 1, this lower bound is

$$\begin{aligned}
&(tmf^0(m) + (1-t))W(a^{\mathrm{opt}}) \\
&\quad + \left(t\left(\frac{e-2}{e-1} - mf^0(m)\right) + (1-t)\frac{-1}{e-1}\right)W(a^{\mathrm{Ne}}).
\end{aligned}$$

This gives a lower bound on the nominal price of anarchy while using f^t of

$$\begin{aligned}
\mathrm{PoA}(\mathcal{G}^0, f^t) &\geq \frac{tmf^0(m) + (1-t)}{1 - \left(t\left(\frac{e-2}{e-1} - mf^0(m)\right) + (1-t)\frac{-1}{e-1}\right)} \\
&= \frac{(e-1)(1+t\Gamma_m)}{1+(e-1)(1+t\Gamma_m)}.
\end{aligned}$$

Part 3: Pareto-Optimality Consider a local utility rule f with nominal price of anarchy guarantee

$$\mathrm{PoA}(\mathcal{G}^0, f) > x \tag{35}$$

for some $x \in [0,1]$. Consider a game $G^C \in \mathcal{G}^0$ following Fig. 3c where $\eta = 0$ and $\xi = m+1$. If $\gamma = ((m+1)f(m) - f(m+2))/f(1)$, then

$$\mathrm{PoA}(G^C, f) = \frac{1}{1 + \frac{1}{(e-1)f(1)}},$$

from the assumption that $f(j) = f^0(j)\ \forall j \geq m+1$ and claim 1. To satisfy the price of anarchy guarantee in (35),

$$f(1) > \frac{x}{(e-1)(1-x)}. \tag{36}$$

Now, consider the game $G^A \in \mathcal{G}^m$ described by Fig. 3a where $\eta = m$ and $\gamma = f(1)/f(m+1) = f(1)/f^0(m+1)$. The price of anarchy of this game is $\mathrm{PoA}(G^A, f) = 1/\gamma$. From (36),

$$\mathrm{PoA}(G^A, f) < \frac{(e-1)f^0(m+1)(1-x)}{x}. \tag{37}$$

In (37), choose $x = \frac{(e-1)(1+t\Gamma_m)}{1+(e-1)(1+t\Gamma_m)}$ for some $t \in [0,1]$ and

$$\mathrm{PoA}(\mathcal{G}^m, f) \leq \mathrm{PoA}(G^A, f) < \frac{\Gamma_m + \frac{e}{e-1}}{1+t\Gamma_m} \tag{38}$$

from the fact $\Gamma_m = f^0(m+1) + \frac{1}{e-1}$. The monotonicity of each price of anarchy expression shows the logic is reversible, matching the theorem. A similar argument could be followed for other values of the utility rule. □