The Rochester Protocol: End-to-end Fair Division for the CHIPS Act

Richard Huang* and Anson Kahng+
*Brown University Dept. of CS, +University of Rochester Dept. of CS and Goergen Institute of Data Science
richard_huang2@brown.edu, anson.kahng@rochester.edu

Executive Summary

Motivated by the challenge of fair and economical provisioning of EDA tools for CHIPS Act programs, we apply methods of *mechanism design*, a domain at the nexus of computer science and economics. We propose a well-defined procedure, the **Rochester Protocol**, with potential application to multiple CHIPS Act and other government chip design enablement programs, including "CHES" (Cross-Hub Enablement Solution) in the DoD Microelectronics Commons.

Our procedure, shown in Figure 1, is executable by a single *mediator* entity who maintains relevant confidentiality measures, and can be flexibly melded with government- and industry-driven considerations (e.g., market shares of vendors, or tool diversity to enable best-in-class design flows). Importantly, it provides a well-defined, public mechanism that solves for best-possible, cost-efficient EDA enablement.

We highlight four requirements of the proposed procedure, which we believe are mandatory in any CHIPS Act EDA provisioning context: (1) a starting total budget range from the government; (2) binding prices in EDA vendor offerings; (3) substitutability between tool alternatives; and (4) a "rental" cost structure for projects that incentivizes reporting truthful and cost-efficient tool needs. If desired, a working implementation of the **Rochester Protocol** can be made available and supported for use in CHIPS Act EDA provisioning.

1 Introduction

The CHIPS and Science Act of 2022 has allocated tens of billions of dollars toward semiconductor research and workforce training. A substantial fraction of these dollars will of necessity be spent on commercial EDA tool licenses that are used for research and development in programs such as the DoD Microelectronics Commons, the NSTC Design Enablement Gateway, the NAPMP advanced packaging program, etc. Other sizable government investments toward a strong domestic, on-shore semiconductor design and manufacturing ecosystem (DARPA NGMM, DoD D2TA, etc.), and a world-leading trained workforce (NSF-DoC NNME, NSF Chip Design Hub, etc.) will also require provisioning with large quantities of EDA licenses.

Our discussion focuses on EDA provisioning but extends to IP and other design enablement aspects as well. We draw on the example of the DoD Microelectronics Commons (MEC), and specific challenges of mechanism design illuminated by the MEC Cross-Hub Enablement Solution (CHES) efforts to support Hub operations and projects.

In the ME Commons, the bulk of awarded funding thus far has been allocated to *Hubs* based at universities or at public-private entities. Each Hub houses multiple *projects* that are competed Commons-wide; approximately $240M of awards spanning 30+ projects at the eight MEC Hubs were awarded in Fall 2024. Each Hub also pursues "Hub Operations" to support its members and achieve differentiated long-term Hub sustainability; such operations include development of design flows, design enablements, unique IP blocks, and design prototypes that enhance competitiveness for future revenue capture. Both projects and Hub Operations require *tools* that are licensed by EDA software *companies*. As of Fall 2024, the MEC's CHES construct obtained approximately $34M of EDA tools for projects and Operations at six Hubs, from five vendors (Cadence, Synopsys, Siemens EDA, Ansys, Keysight).

Challenges of Fairness and Mechanism Design for CHIPS Act Design Enablement

Initial experiences in the ME Commons have illuminated several challenges of fairness and mechanism design when (A) a collection of projects and tool users must (B) be provisioned by a collection of tools obtained from multiple EDA vendors, (C) using dollars that are supplied separately (from Hub and project dollars) by the government.

Lack of clear rules of engagement. The absence of important constraints (e.g., overall EDA budget, or binding price quotes that cannot be changed dynamically) will prevent efficient convergence to a provisioning solution; the mechanism design must comprehend this and operate within agreed-upon "rules of the game."

Incentives for companies to inflate EDA prices. In the ME Commons, EDA companies receive revenue through the CHES construct or through project awards to non-CHES Hubs. Ultimately, there is an available "pie" (e.g., ∼$1.6B over five years) that is split by Hubs and project performers, alongside EDA (and IP, and cloud) suppliers. If tool prices, project tool needs, project selections, and the overall Commons/CHES EDA budget are *all* allowed to be in flux simultaneously, then EDA companies will seek to grow their "share of the pie" by *increasing* prices (relative to competitors' prices) or reducing the discounting of tools used in ME Commons R&D.

Incentives for projects to misreport tool needs. Projects face competing incentives to misreport true tool needs. When EDA companies increase tool prices, projects are incentivized to *under-report* tool needs, so as to pay less for EDA and be more cost-competitive in proposals to the government. (For example, each $100K/year that is not spent on EDA is "convertible" to an extra Ph.D. student or postdoc hired on to the project.) On the other hand, if projects will share a fixed total cost of EDA, and receive licenses in proportion to their requests, then they are incentivized to *over-report* tool needs. A "government pays" regime can also incentivize over-reporting of projects' EDA tool needs. Over-reporting or under-reporting of the number of licenses a project needs, projects hogging tools, etc. can all harm cost-efficiency and technical progress of projects. Thus, mechanism design for EDA provisioning in CHIPS Act programs should seek to reduce such incentives for misreporting of tool needs.

Our Contributions

In our work, we seek to answer the following questions.

Q1. What are mandatory requirements of the CHIPS Act EDA provisioning context?

Q2. How can we incentivize projects to truthfully report the tools needed to complete their proposals?

Q3. How can we fairly divide resources among companies?

With respect to **Q1**, we identify three non-negotiable items: (1) a program-wide budget range for each Hub, (2) binding prices for tools provided by companies to a trusted mediator and projects, and (3) exactly specified substitutability between tool alternatives. We seek to address **Q2** and **Q3** by using tools from the field of *mechanism design*, which seeks to design rules for "games" in which individually rational and strategic agents are incentivized to behave in a way that benefits society as a whole. In particular, we propose separate interventions to address each of **Q2** and **Q3**. To incentivize projects to truthfully report tool needs, we propose a *rental model* in which projects have "skin in the game" and must pay a higher cost to use tools they do not initially request. To fairly divide resources among companies, we propose a fair division scheme that draws on literature from impartial dollar division and claims problems in economics in order to directly distribute each Hub's budget for tools among companies in a way that bypasses company-specific prices for tools.

2 Suggested Process (e.g., for DoD Microelectronics Commons)

We now propose a specific process for the DoD Microelectronics Commons that draws heavily on the concepts of impartial fair division and claims problems described in greater detail in Appendices A.3 and A.4.

2.1 The Provisioning Process

We begin by sketching the provisioning process, along with its immutable constraints.

Rules of the Process

A provisioning process consists of three classes of players (equivalently, agents): (1) *projects*, (2) EDA *companies*, and (3) a trusted *mediator* (i.e., the entity running the allocation process). Each project requires a particular set (or bundle) of tools to complete their work. Each company provides licenses to tools that they own. The goal of each project is to obtain a satisfactory bundle of tools; the goal of each company is to maximize the amount of money they are paid for tools that they provide. The goal of the mediator is to ensure that projects receive tool licenses and pay companies, and that companies send tool licenses and receive payment that effectively comes from projects.

We also establish the following crucial elements of the provisioning process based on context from the CHIPS Act.

1. **Budget specification**: Each collection of projects, or Hub, must provide a budget range for the amount of money they are willing to spend on tool licenses, e.g., "Out of $200M in overall project Year 2 funds, an estimated range of between $15M and $40M is anticipated to be used by the USG for EDA licenses in support of awarded projects."

2. **Fixed prices**: Each company must provide the mediator with "a la carte" book prices for each tool at the start of the fair division process. They may also provide "all-you-can-eat" (AYCE) prices for unlimited licenses to all of their tools. These prices are binding and may not be changed, and each company's prices are private to themselves, the mediator, and Hub or project leads. Prices will be kept from competitors through standard NDAs.

3. **Substitutions**: Projects must specify acceptable substitutions between tools. The mediator will provide a list of "default" substitutions among tools of the form, "1 copy of tool T_j^1 is substitutable by 2 (or more) copies of tool T_k^3." Projects may override default substitution rates during the project request portion of the process.

4. **Tool rental model**: Each project will pay for the tools they request at a base rate per tool; they may also buy extra tools beyond those initially requested at a higher rate per tool. The base rate will be determined based on the fixed prices provided by the companies and the amount of money paid to each company at the end of the suggested fair division process.

5. **No tool contention**: Hubs will purchase the union of all project requests in order to prevent (potentially strategic) contention for scarce tools.

6. **No vetoes**: The results of the mechanism are binding, i.e., companies must accept their payments, and projects must accept the tools acquired on their behalf. If necessary, it may be useful to explicitly disallow side channel deals between individual projects and companies. Hubs, projects, and companies must commit to satisfying their EDA needs through the provisioning process, and it is understood that companies give "best price" deals to the mediator.

2.2 End-to-End Flow

We now describe the overall flow of the **Rochester Protocol**; see Figure 1 for a visual representation.

1. **Company pricing**: EDA companies provide "a la carte" prices per tool to the moderator of the process. They may also provide "all-you-can-eat" (AYCE) pricing at the whole-Hub or whole-Commons level of granularity.

2. **Project requests**: Projects provide tool demands and substitutions (if different from default rates). All requests will be anonymized for privacy reasons.

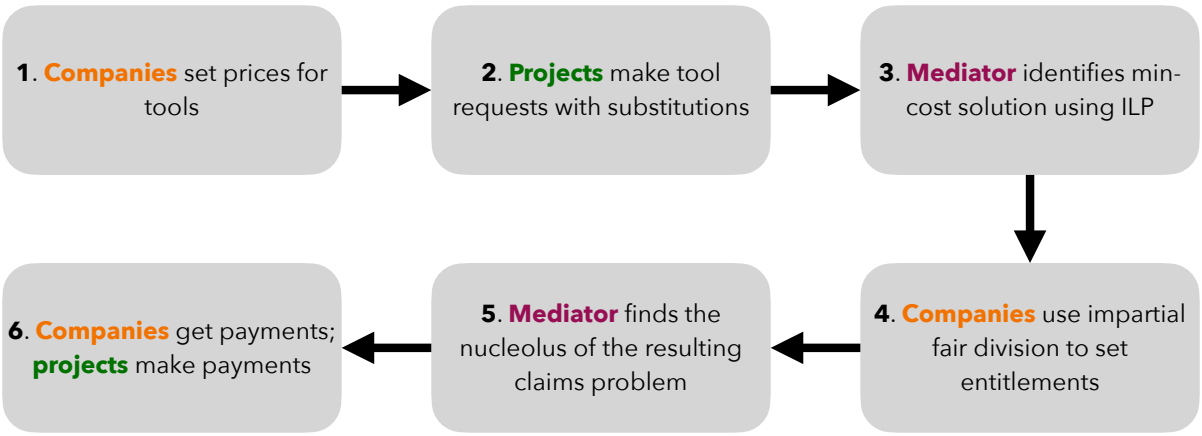

Figure 1: Illustration of the **Rochester Protocol**. Steps are described in Section 2.2.

3. **Bundle optimization**: The mediator solves a minimum cost integer-linear program (ILP)[1] to find the least expensive bundle of tools to acquire that satisfies the union of all project demands. Here, one can view AYCE offers by companies as establishing a cap on their total *entitlement*, i.e., the amount of money they are judged to "deserve" at the end of Step 4. For further details, see Section 2.2.1.

4. **Impartial fair division**: Companies are provided with the minimum-cost bundle from the ILP optimization step, and each company must evaluate how much of the total tool budget their competitors deserve. This leverages industry-specific expertise (competitors should have a good sense of what each company's bundle is "worth") and allows companies to fight back against a company that price-gouges an unsubstitutable tool. In order to ameliorate strategic misreporting, we use an impartial division rule with the property that each company's report does not affect their entitlement. For further details and extensions, see Appendix A.3 and Section 2.2.2.

5. **Claims problem**: The mediator then divides the (likely insufficient) tool budget among companies based on their entitlements at the end of Step 4 according to the Talmudic fair division rule, which recovers the nucleolus. The target division is the result of the impartial fair division step. For further details, see Appendix A.4.

6. **Project payments**: Projects are effectively charged based on the a la carte prices of the tools that they request and the amount of money allocated to each company after solving the claims problem. For further details about the rental model, see Section 2.2.3.

2.2.1 Bundle Optimization Per Project (Step 3)

The ILP that the mediator solves for bundle optimization for each project is as follows. Assume that there are a total of T tools collectively offered by all m companies, and there are also $0 \leq k < m$ AYCE offers from some subset of companies. The mediator would define one ILP for each project (because they may have different substitution matrices S) and return the sum of all $\vec{x}$ solutions as the overall bundle of tools.[2]

- Input: $\vec{t}$ is a $1 \times T$ vector of the quantities of each tool requested by the project.
- Input: $\vec{p}$ is a $1 \times (T + k)$ vector of the price per tool (where the last k slots are prices of AYCE tools). Note that $k \leq m$ because each company can make at most one AYCE offer.

[1] Due to the structure of substitutions, this is potentially an easy problem.

[2] If all substitutions are of a linear form (i.e., each tool is equivalent to some set of linear combinations of other tools), then this ILP becomes quite simple. For each tool T_k, there is a most cost-effective "a la carte" substitution. We may use these most cost-effective substitutions directly. In this alternative formulation, AYCE offers become upper bounds on the amount of money that will be paid to each company.

- Input: S is a $(T+k) \times T$ matrix that encodes the accepted substitution rates between tools specified by the project. Entry S_{ij} represents the value of one copy of tool i in terms of number of copies of tool j. The last k rows represent AYCE substitutions, where entries in row S_ℓ for $\ell \in [T+1, T+k]$ are 0 for tools not offered by the company making the AYCE offer and ∞ for tools offered by the company making the AYCE offer.
- Output: $\vec{x}$ is a $1 \times (T+k)$ vector of the quantities of each tool produced by the solver.

The ILP solves for $\vec{x}$ to minimize the total price of all tools subject to (1) tool requests being met, and (2) integral numbers of each tool requested:

$$\begin{aligned} &\min \vec{x} \cdot \vec{p} \\ &\text{subject to} \\ &\quad \vec{x} \cdot S_{\cdot j} \geq t_j, \ \forall j \in [T] \\ &\quad x_i \in \mathbb{Z}_{\geq 0}, \ \forall i \in [T+k]. \end{aligned}$$

In the constraints, the notation $[x] := \{1, \ldots, x\}$.

2.2.2 Democratizing Entitlements (Step 4)

One downside of the impartial mechanism used in Step 4 is that it does not actively disincentivize any company from providing (ignorant or spiteful) misreports about how much their competitors deserve; impartiality only guarantees that a company's share is not affected by their judgments of others. This leaves the door open for suboptimal outcomes where, for instance, all companies spitefully determine that the most powerful company in the pool deserves no money at all. However, further democratization of Step 4 may address this issue by, for instance, taking into consideration market share and Hub consensus about how much money each company deserves, and determining entitlements based on a weighting scheme over multiple constituent elements (e.g., 30% market share, 20% Hub consensus, and 50% impartial mechanism), as has been done in, e.g., college football rankings. The current setup allows for such democratization to happen essentially "for free" in the sense that the result of Step 4 would still be impartial (company reports still do not affect their entitlements) and the subsequent steps of the division process proceed unchanged.

2.2.3 Tool Rental Model (Step 6)

In order to determine the rental price of each tool, the mediator linearly scales the book prices of each company's tools such that the total cost of all tools requested by that company matches the amount of money allocated to that company at the end of the provisioning process. The details are laid out below.

For simplicity, consider a single Hub with a total of n projects, $\{1, \ldots n\}$. For each project $i \in [n]$, let the solution of its min-cost ILP be $\vec{x}_i$. Let $\vec{x} = \sum_{i \in [n]} x_i$ be the union of all project tool demands. Slightly overloading notation, let $\vec{x}_{C_j}$ for $j \in [m]$ denote the restriction of the total demand vector to tools licensed by company C_j. Furthermore, let d_j be the amount of money allocated to each company C_j for $j \in [m]$ at the end of Step 5 in the provisioning process. Lastly, let $\vec{p}_j$ denote the book prices of tools provided by company C_j.

To determine the effective price for each tool offered by company C_j, we first find the scaling factor $\alpha_j \in \mathbb{R}^+$ such that $\alpha_j(\vec{x}_{C_j} \cdot \vec{p}_j) = d_j$, and then calculate the rental model price vector, $\vec{r}_j$, by rescaling the original price vector, $\vec{r}_j = \alpha_j \vec{p}_j$. This procedure exactly re-scales book prices to match the amount of money assigned to company C_j during the provisioning process.

3 Benefits

We revisit our initial motivation: Does the **Rochester Protocol** ameliorate concerns mentioned earlier in the document?

Constraining the Provisioning Process Through "Rules of Engagement"

By requiring the government to provide an initial budget range, and by requiring companies to provide binding price quotes for their tools, two crucial sources of non-convergence are eliminated. Further, by requiring each tool to have a substitute from a different vendor, the protocol avoids the possibility of having budget "held hostage" by tools that uniquely serve a project's needs.

Companies Won't Inflate Prices

Companies will no longer inflate the book prices they report to the mediator. This is because reporting higher prices is strictly punished by the min-cost ILP solution in Step 3 (having higher prices increases the chance that a company's tools will be substituted for by others' tools). In fact, companies are incentivized to *decrease* their prices in order to encourage the ILP to include their tools in the solution. While in the limit this means that companies may price their tools too low, the companies are protected by the budget specification lower bound. Company prices also do not affect their entitlements, due to the impartial fair division used in Step 4, or their overall profits, which are set at the end of the claims problem in Step 5, so there is no additional pressure to inflate (or deflate) prices.

Projects Will Report Tool Needs Accurately

Additionally, under the rental model and with the stipulation that Hubs will purchase the union of all requested tools, projects are likely less incentivized to over- or under-report the number of needed tools.

To sketch an informal argument, consider a project that deviates from truthfully reporting its tool needs. There are two cases: The project may either over-report its tool needs and request extra tools, in which case it pays for more tools than it needs, or under-report its tool needs and request too few tools, in which case it must rent additional tools at a markup to complete the work. Due to the complexity of Steps 3 through 5 (solving the min-cost ILP, running impartial fair division, and finding the nucleolus of the claims problem), it is possible that over- or under-reporting will be beneficial (for instance, it could be the case that over-reporting tool needs leads to a different enough set of payments to companies that the project's rental model payment decreases). However, this would be extremely difficult to predict, much less precisely manipulate. Without full information, projects are unlikely to benefit through such strategic misreporting.

Additional Comments

We make two additional comments. First, as noted in Section 2.2.2, the impartial fair division step brings a possibility of an "unpopular" company being arbitrarily punished through collusive action of its peers. However, we can augment Step 4 with other considerations such as market share and/or Hub consensus, so that companies have limited recourse to punish unpopular peers. Second, we note that budget constraints are guaranteed to be met. The claims problem in Step 5 guarantees that the budget will not be exceeded. If the minimum budget constraint is not met, then it is possible to scale up payments uniformly to satisfy the minimum budget constraint.

References

[AM85] Robert J Aumann and Michael Maschler. Game theoretic analysis of a bankruptcy problem from the talmud. *Journal of Economic Theory*, 36(2):195–213, 1985.

[DCMT08] Geoffroy De Clippel, Herve Moulin, and Nicolaus Tideman. Impartial division of a dollar. *Journal of Economic Theory*, 139(1):176–191, 2008.

A Technical Background

A.1 Model and Notation

We begin with notation. Throughout, let $[n] := \{1, \ldots, n\}$ for all $n \in \mathbb{N}$.

We assume a collection of h different *Hubs*, $H_1, \ldots, H_h$, where Hub H_i consists of p_i different *projects*, $P_i^1, \ldots, P_i^{p_i}$. There are also m different *companies*, $C_1, \ldots, C_m$, and each company C_j provides t_j different *tools*, $T_j^1, \ldots, T_j^{t_j}$. Let $t_{tot} = \sum_{j=1}^{m} t_j$ be the total number of distinct tools offered by all companies.

Each project requires some number of copies of each type of tool to be completed. Let $a(i, j) \in \mathbb{N}^{t_{tot}}$ represent the true vector of demands for project p_i^j for all $i \in [h]$, $j \in [p_i]$, i.e., $a(i, j)$ is the minimum tool demand needed to complete project p_i^j to specifications. Additionally, we assume that each Hub H_i has a maximum budget B_i to spend on tools.

A.2 Pipeline

We assume the following sequence of events occurs independently for each Hub H_i, for all $i \in [h]$. These may not be the only steps; in fact, see Section 2 for a much more detailed and complex instantiation.

1. Each Hub H_i asks its projects to provide tool demands. These tool demands are then aggregated, anonymized, and made available to the companies supplying the tools. This aggregation may be done at the Hub level or across all Hubs; our pipeline works for both cases.

2. The companies then collectively decide how to split H_i's (insufficient) tool budget, B_i. Note that here, they directly divide the budget among themselves without explicitly considering per-tool prices. In general, an EDA company will not reveal its "price book" to a competing company. So, it is reasonable to assume that companies do not know any per-tool prices other than their own.

Our process for dividing awards among the companies is a two-step process directly applying two well-known fair division rules for different settings. We use the first rule to determine how much each agent is entitled towards out of a value larger than the total award amount, and the second rule determines how much each agent is actually allocated. We describe them in the following subsections.

A.3 Impartial Division of a Dollar

Suppose we are trying to divide a divisible item (e.g., a a dollar) among $n \geq 4$ agents. In [DCMT08], they propose a division rule that considers each agent's evaluation of the allocations that other agents should receive, and divides the entire dollar (i.e., is non-wasteful) and maintains *impartiality*: no agent reports anything about their own allocations, and their report has no effect on their own allocation i.e., their allocation is determined solely by the other agents' reports.

To understand the impartial division rule, it is first instructive to introduce the simpler case of wasteful impartial division where the mechanism may distribute less than the entire divisible item. In such wasteful settings, it is possible to build an impartial mechanism where the amount that agent i receives is determined based on the ratios of what i and j are judged to deserve, averaged over all other agents j. The exact method of aggregating ratios for i and a particular j is flexible, and using different aggregators (e.g., the arithmetic mean, geometric mean, median, etc.) leads to different wasteful impartial rules.

In order to turn wasteful impartial rules into non-wasteful impartial rules, consider dividing the initial item into n pieces, where n is the number of agents, and running a wasteful impartial division on each piece with the additional stipulation that agent i's opinion is not taken into account for piece i. Then, for each piece, the agent whose opinion was not taken into account receives the remainder of the piece after the wasteful impartial division process. It is possible to see that such a process is still impartial and now non-wasteful.

A.4 Claims Problems

In the bankruptcy claims problem, a person dies and leaves behind an estate of money $E \in \mathbb{R}_{>0}$. They owed money to n creditors where their estate is less than the total debt they owed, and we want to fairly divide E among the creditors. Specifically, the debts of each creditor is denoted as $d := \{d_i\}_{i\in[n]}$ and the total debt is $D = \sum_i d_i$.

The Talmud, a religious text, gave a mysterious rule to do so. In [AM85], they showed that the Talmud rule always gives the *nucleolus* of a corresponding coalitional game that is derived by the bankruptcy problem, making it a natural solution for allocation problems where the total amount entitled by the agents exceeds the amount that can be allocated. However, their study does not delve into the case of agents misreporting their debts (because in these bankruptcy situations the debts are typically already well-documented).

We formally describe the Talmud rule (also known as the contested garment rule) next. We are given $E, d := \{d_i\}_{i\in[n]}$ where $0 \le E \le D, D := \sum_{i\in[n]} d_i$. We first describe two subroutines by the Talmud rule, known as constrained equal awards (CEA) and constrained equal losses (CEL). In CEA, each agent starts with 0 and "eats" the estate E at an equal rate and stops eating once they've eaten their entitled amount, or if no more estate remains. CEL is a dual procedure, in which each agent starts with their entitled amount and "reverse-eats" until they hit 0 or the total allocation drops from D to E.

We define $\mathrm{CEA}(d, E) := \{a_i\}_{i\in[n]}$ and $\mathrm{CEA}_i = a_i$ where a_i is the allocation of agent i by the CEA procedure given input d, E, and define $\mathrm{CEL}, \mathrm{CEL}_i$ analogously. When the input is clear we may overload the notation as simply CEA (and so on) for cleanliness. The pseudocodes for CEA and CEL are given below.

Algorithm 1 Constrained equal awards

Agent i is active and $a_i \leftarrow 0 \; \forall \, i \in [n]$.
while $E > 0$ **do**:
    $A \leftarrow$ number of currently active agents
    $a_i \leftarrow a_i + \epsilon$ for all active agents i.
    $E \leftarrow E - \epsilon A$
    Deactivate each agent j such that $a_j = d_j$.

Algorithm 2 Constrained equal losses

Agent i is active and $a_i \leftarrow d_i \; \forall \, i \in [n]$.
while $D > E$ **do**:
    $A \leftarrow$ number of currently active agents
    $a_i \leftarrow a_i - \epsilon$ for all active agents i.
    $D \leftarrow D - \epsilon A$
    Deactivate each agent j such that $a_j = 0$.

CEA and CEL are *duals* in the sense that the following holds:

$$\mathrm{CEL}(d, E) = d - \mathrm{CEA}(d, \sum d - E) \tag{??}$$

One perspective of the Talmud rule is to run either CEA or CEL depending on if $E \le D/2$ or not. If $E \le D/2$, then we run CEA with $(\{d_i/2\}_{i\in[n]}, E)$ as input; otherwise, we first allocate $d_i/2$ to agent i for all i, and then run constrained equal losses with $(\{d_i/2\}_{i\in[n]}, E - D/2)$ as input. We denote the output of the Talmud rule as $\mathrm{CG}, \mathrm{CG}_i$.

Algorithm 3 Contested garment rule (Talmud rule)

if $D/2 \ge E$ **then**
    Agent i is allocated $\mathrm{CEA}_i(E, \{d_i/2\}_{i\in[n]})$ for every $i \in [n]$
else
    Agent i is allocated $d_i/2 + \mathrm{CEL}_i(E - D/2, \{d_i/2\}_{i\in[n]})$ for every $i \in [n]$