Table of Content

[1. Introduction 2](#_Toc500781413)

[2. Metal Discovery 3](#_Toc500781414)

[2.1. Metal Finding 3](#_Toc500781415)

[2.2. Optimization 5](#_Toc500781416)

[3. Metal Digging 10](#_Toc500781417)

[3.1. Metal Found 10](#_Toc500781418)

[3.2. Digger Selection 10](#_Toc500781419)

[3.3. Coalition Formation 14](#_Toc500781420)

[3.4. Coalition next move calculation 14](#_Toc500781421)

[3.5. Movement of free DAs without MF assigned 15](#_Toc500781422)

[3.6. Optimization 15](#_Toc500781423)

[4. Metal manufacturing 16](#_Toc500781424)

[4.2. Selection of and journey to a manufacturing center 16](#_Toc500781425)

# Introduction

One of the most important things when designing our system is that, despite the fact that it is heterogenous, all agents will be cooperative, with the aim to achieve the best value of the following evaluation parameters given a limited number of turns:

* **Benefits:** amount of points received.
* **Manufactured metal:** current amount of each metal already manufactured.
* **Average benefit for unit of metal:** amount of points received per metal unit, regardless of the metal's type.
* **Average time for discovering metal:** the amount of time spent from metal appearance until a prospector discovers it.
* **Average time for digging metal:** the amount of time spent from metal discovery until the first digger gets to that point.
* **Ratio of discovered metal:** the ratio of the metal that is already discovered by prospectors from the total.
* **Ratio of collected metal:** the ratio of the metal that is already collected by diggers, including that in the digger and that already disposed in manufacturing centers

The output of each turn will be the movements of the mobile agents, the collection, or the manufacturing of a metal.

The main problem is the distributed metal management, divided into 3 different actions:

1. Metal detection
2. Digging
3. Manufacturing

To solve this problem, we will need **distributed data sources** (given by the information retrieved by PA and DA) and a **distributed expertise** (given by PCA and DCA). With the purpose of **efficiency** maximization, it could require different agents working at the same time in different parts of the problem.

# Metal Discovery

## Metal Finding

*Description:* Prospector coordinator will be in charge of maximizing the area inside the visual field, given by cells wrapping prospector agents. Also, the prospector agents should be organized with the aim to go to the map locations where there are higher probabilities of founding metal. We have considered 2 different optimization methods.

Prospector coordinator (PCA) will be in charge of the metal discovery optimization. Firstly, we will create some possible game scenarios in order to clarify what contour variables will define our system.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 |  |  |  |  |  |
| 2 |  | PA1 |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |

In this scenario, composed by 12 path cells, a prospector agent is located in the path cell (2,2). To distinguish between path cells (PC) and field cells (FC), we have leaved the first ones with white background while shadowing the second type with blue background. This lead that at the moment, field cells (1,1), (1,2), (1,3), (2,1) and (2,2) are inside the visual field of the PA1, leaving all other field cells out of its visual field and, therefore, susceptible to have undiscovered metal.

Given that for each turn each field cell with no metal has a random provability to produce metal, an undiscovered field cell will have a utility worth given by the formula:

Where N is the number of turns of the cell out of visual field.

Furthermore, will add another term to reduce the probabilities of collision between a collecting DA and the moving PA. Taking into account that DA only could collect on PC where there is a contiguous metal FC, we will penalize the PA to move to that cells. It led to the next formula:

Where MFCi is the i metal field cell adjacent to the PC (where there are NP metal fields). Beta ( is a parameter to weight this function and it will be adjusted based on an optimization criterion.

This formula could be used to measure the utility of a PA to be in a concrete PC:

Where x and y are the coordinates of the FC wrapping the path cell, Nx is the maximum x map coordinate and Ny is the maximum y coordinate.

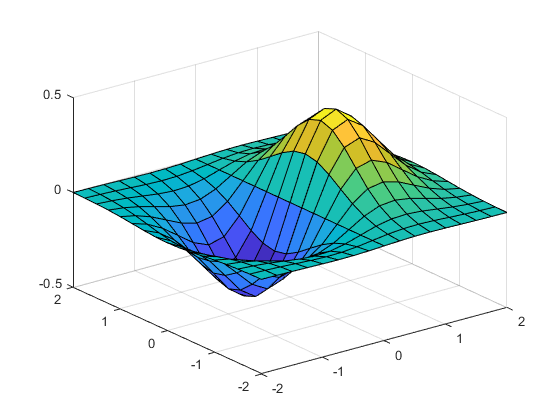
From this formula we could extract the utility formula of sending a concrete PA to a concrete PC:

Where x and y are the coordinates of all PC of the minimum way to go from the current x, y location of the PA to the target x’, y’ PC, also taking into account all PC wrapping the way. M is the needed movements of the agent to reach the target PC and is a weighting parameter.

We will also introduce an additional term to improve the efficiency by maximizing the separation between PA, pursuing a gravitational equilibrium between all PA and therefore maximizing the visual field. It led to the next formula:

Where r is the distance between all PA(i) at less than distance d but the PA(k) that is the PA at one movement distance, to that PC(j). Alpha ( is a parameter to weight this function and it will be adjusted based on an optimization criterion. We will only consider

So, an interesting approach could be that PCA collect all current scenario information and then generate a matrix with the utility functions of each PC are represented. This matrix could be visually represented as the following figure:



Then, PCA will try to flatten the most possible the surface of this 3D area, or what is the same, to reduce the sum of utility functions of all PC in the map.

## Optimization

As previously stated, we have defined 2 different approximations:

**Optimization method 1: GLOBAL APPROXIMATION**

*Description:* In this case the strategy of global approximation, we will perform a **parallel** **combinatorial auction** **directed**, with **PCA** as **auctioneer**.

*Cooperation:* PCA will decide which **PA** could **bid** in that auction and also what **PC** are **auctioned**. It could be possible that different auctions have to be performed in each turn. This is due that if we take into account all possible map initializations, we have to consider separated PC groups in such manner it will be impossible for a PA to go from one PC to another PC of the map.

This scenario could be created by 2 different ways:

1. When the initial map doesn’t offer a path from one PC to each other PC.
2. When somewhere in the map there is a way of only 1 PC width and a digger is collecting metal through one of this PC.

Scenario 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 |  |  |  |  |  |
| 2 |  | PC group 1 |  | PC group 2 |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |

Scenario 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 |  |  |  |  |  |
| 2 | PC group 1 |  | DA collecting |  | PC group 2 |
| 3 |  |  | Metal |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |

Even that in the second case we could make the digger to stop digging one turn to allow another agent to pass through that PC, sometimes it could not be the most optimal solution.

Considering this **auction** **admission rules**, we also have to remark that as in all combinatorial auctions, all achievable PC from each of the PA in the auction are the items to auction, separated individually. Then, each PA will present bids for different combinations of PC (**auction interaction rules**), where the fixed bid for each PA will be exactly it’s utility formula to go to that PC.

Finally, PCA will calculate the **best bids combination** to achieve the **higher benefit** and will accept those **clearance prices**. In case of a draw, the combination with less agents involved will be the chosen one.

*Communication:* The goal of the communication is to inform all PA susceptible to take part in the auction, so PCA will do a broadcast though Contract Net mediated by the Directory Facilitator (DF) agent. The posterior communication will be point to point as stablished by Contract Net auction protocol un till a clearance price is reached.

**Optimization method 2: LOCAL APPROXIMATION**

*Description:* In this case we only will consider the PC where each PA could move next turn, so we will only calculate the utility function of that PC with the previously defined formula plus the term, which give us the new formula:

PAC will calculate this formula for each cell and then will order the next movement to perform to each PA. This method acts as a gradient descent, so each PA will take the next movement to the direction of more gradient descent.

After applying to each cell that PA could move next turn, we will obtain something like:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 5 | 5 | 16 | 24 | 25 | 0 | 19 |
| 0 | **PA1** | 5 | 10 | 26 | 45 | 65 | 49 | 58 | 12 |
| 0 | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 17 | 2 |
| 20 | 40 | 0 | 50 | 65 | 65 | 15 | 0 | 9 | 3 |
| 20 | 64 | 0 | 95 | 50 | 15 | 32 | 0 | 16 | 4 |
| 24 | 66 | 0 | 95 | 45 | 17 | 32 | 0 | 29 | 9 |
| 22 | 67 | 0 | 45 | 62 | 62 | 17 | 0 | 30 | 16 |
| 21 | 68 | 0 | 0 | 0 | 0 | 0 | 0 | 45 | 5 |
| 25 | 70 | 42 | 35 | 52 | 56 | 58 | 41 | 59 | 24 |
| 2 | 19 | 3 | 20 | 12 | 20 | 24 | 14 | 3 | 13 |

For the sake of simplicity, we have not taken into account any metal field nor other PA in this example. As could be seen, there is a huge problem due that if a PC isn’t in contact with any FC, it’s utility is 0 and therefore PA1 will never go to the FC of the center of the figure. This lead to the possibility of leaving FC without visiting during the whole game and this should be avoided. Furthermore, there is the problem that with that approximation, the cells surrounding the PA will always be 0 and then the algorithm to find the next movement will have

We have reached the solution given by the next formula:

The will be only computed for all PC with 0 utility value. is a weight factor, is the sum of the utilities of all cells in contact with the cell for which we are calculating the utility and the division by 8 is because there are 8 cells in contact with each cell. It has to be noted that this formula will only be calculated by PC with 0 value, not for any FC, it will have no sense.

The obtained result will give us a fuzzier approach:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 5 | 5 | 16 | 24 | 25 | 0 | 19 |
| 0 | **PA1** | 5 | 10 | 26 | 45 | 65 | 49 | 58 | 12 |
| 0 | 20 | 16 | 20 | 33 | 35 | 30 | 27 | 17 | 2 |
| 20 | 40 | 34 | 50 | 65 | 65 | 15 | 11 | 9 | 3 |
| 20 | 64 | 51 | 95 | 50 | 15 | 32 | 17 | 16 | 4 |
| 24 | 66 | 54 | 95 | 45 | 17 | 32 | 20 | 29 | 9 |
| 22 | 67 | 43 | 45 | 62 | 62 | 17 | 19 | 30 | 16 |
| 21 | 68 | 41 | 30 | 39 | 38 | 29 | 31 | 45 | 5 |
| 25 | 70 | 42 | 35 | 52 | 56 | 58 | 41 | 59 | 24 |
| 2 | 19 | 3 | 20 | 12 | 20 | 24 | 14 | 3 | 13 |

Now we have the map properly distributed, allowing prospector agents to go to any location of the map. As previously stated, PCA is the agent in charge of performing all that computations. Following, a brief explanation about how to decide the next movement of each PA will be done:

|  |  |  |
| --- | --- | --- |
|  | 15 |  |
| 3 | **Prospector Agent (PA)**  Resultat d'imatges de prospector | FC |
|  | 6 |  |

With red background meaning that moving to that cell is not allowed in the next turn, due that diagonal movements are not allowed, blue background means a FC and then also movement through it is not allowed. So, following a “gradient ascent” method, in this case PA will choose to move to the above cell with 15 utility value. If there is the case that there are more than one cells with the possibility to move with the same utility value, PCA will choose between them randomly.

*Cooperation:* This method will involve communication point to point with each PA and PCA, where PCA will calculate the utility map and then it will send the next movement to each PA.

*Communication:* In this case only point to point communication between PA and PCA is needed.

In the following table pros and cons of this cooperation method are mentioned:

|  |  |  |
| --- | --- | --- |
|  | pros | CONS |
| Approx 1 | * It quickly considers all moving possibilities * Best combination of performance evaluation parameters could be reached. * Global minimums are reached | * Inefficiencies can result from future uncertainties * Difficult to keep track of several simultaneous auctions with substitutable and interdependent goods * Huge computational load |
| Approx 2 | * It quickly considers all moving possibilities * Relatively computational load. * Local minimums are reached. | * Inefficiencies can result from future uncertainties * Global minimums might not be achieved. |

Finally, we have reached the conclusion that because of the superior efficiency, the second approximation will be the one that we will implement in our project

# Metal Digging

## Metal Found

*Description:* The Prospector Coordinator Agent (PCA) will inform the Coordinator Agent (CA) that a new metal field (MF) has been found. Afterwards, the CA will transmit such information to the Digger Coordinator Agent (DCA).

*Cooperation:* There are no cooperation mechanisms involved.

*Communication:* The goal of the communication is to inform another agent. It will be a communication between these 3 agents mediated by the Directory Facilitator (DF) agent.

## Digger Selection

*Description:* Once the DCA is informed of the existence of a MF that has to be dug, it has to choose the optimal digger to perform the excavation. To do so, there are multiple mechanisms of cooperation that can be used. We will make a comparison between them below.

*Cooperation:*

**Option 1:** For each iteration of the system, we perform a Multi-Unit First Bid Auction for each of the MF. Only free diggers or diggers carrying the same type of metal than the one of the field will be allowed to participate in the auction. The bid of each digger will depend on different parameters (available slots, distance to the MF, possibility of manufacturing the metal he is carrying, …). Once all the metal units of the field have been assigned to the diggers, the DCA-G/DCA-S will inform the DCA that they are ready to perform another auction for another MF.

To avoid that a single digger wins 2 auctions simultaneously, we will perform the auctions sequentially, i.e., until all the metal units of a field have been assigned the auction of the next MF won’t start.

We should also decide whether a digger that has already been assigned to a MF can take part in auctions for other MFs.

If we don’t allow such digger to participate in other auctions, we will make sure that a MF is not auctioned multiple times during the same iteration and therefore the computational performance of the code will be improved. However, this could also imply that some MFs won’t have any digger candidates, and therefore the order in which the fields are auctioned would influence the performance of the system.

On the other hand, even if we allowed the digger to participate in other auctions, we are not guaranteed to obtain a better metal recollecting performance. E.g.:

|  |  |  |
| --- | --- | --- |
| Metal Field | Digger 1 Bid | Digger 2 Bid |
| MF1 (Gold) | 12 | 10 |
| MF2 (Gold) | 13 | 12 |

In this case, if the first DA wins the auction for MF1 and it is not allowed to participate in the second auction, the “global” bid would be 24. On the contrary, if it is allowed to participate, the global bid would be 23.

For this reason, we think it would be better not to allow a digger to participate in an auction if it already has a MF assigned in that turn, but that all MFs have to be ranked by the DCA to decide the order in which they will be auctioned.

We also considered the possibility for DA to form coalitions to bid for a certain Metal field. For example, if a MF has 9 units and 2 DAs that are approximately at the same distance from the MF have 4 and 5 slots available, it would make sense for them to work together. To form the coalition, a proactive DA would use a Contract Net protocol to find its partners. We discarded this possibility since we thought it would overload the DAs.

**Option 2:** Another possible solution for this task would be to use a hybrid mechanism (voting + auction). By doing this, we would avoid having to establish a global order of which MFs are better, but instead, each digger ranks the whole set from its point of view.

Since the digger will be able to compare the benefits of all available MFs, it will establish a linear ordering of the possibilities (instead of a weak ordering).

Then, each digger will be assigned a to a MF following a “Selectivitat”-like procedure. We add an example as further explanation:

Assume there are 5 MFs, 3 of which are gold-type: G1 (5 units), G2 (4 units), G3 (4 units), and the rest are silver-type: S1 (4 Units), S2 (5 units). Also, we have 2 DA carrying gold, 2 DA carrying silver and 1 free DA. Also, assume the following votes are summited:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DA(Type, Slots) | 1st-mark | 2nd-mark | 3rd-mark | 4th-mark | 5th-Mark |
| DA1(G,5) | **G1-13** | G2-6 | G3-2 | - | - |
| DA2(G,4) | G1-12 | **G2-10** | G3-8 | - | - |
| DA3(S,4) | **S2-10** | S1-6 | - | - | - |
| DA4(S,4) | **S1-11** | S2-8 | - | - | - |
| DA5(F,5) | G2-8 | **S2-7** | S1-6 | G2-5 | G3-4 |

Then DA1 would be assigned to the G1 field as its mark is higher than DA2 & for the same field (although both rank such field as its first preference) and he is able to collect the whole field on its own. Therefore, DA2 would be assigned its second preference which is field G2 and since it would be able to mine it on its own, DA5 that has a lower mark for G2, would not be assigned to G2.

DA3 would be assigned to S2, and since it is not able to fully mine the S2 field, DA5 would also be assigned to S2. Finally, DA4 would be assigned to S1.

Note that either DA5 or DA3 won’t be able to fill all its empty slots, and therefore, the mark they calculated when ranking the fields won’t reflect its real performance. We will talk about this phenomenon and propose a solution in the next described task. The same would happen in option 1 if 2 DA are assigned to the same MF.

Also, note that if we used this mechanism, the assignation would be performed by the DCA instead of the DCA-G & DCA-S agents.

Finally, note that the mark that a DA assigns to a MF in option 2 can be computed in the same way than the bid such agent would make for the MF in option 1.

We now present a comparison of the pros and cons we observe for each option.

|  |  |  |
| --- | --- | --- |
|  | pros | CONS |
| Option 1 | · Simpler implementation. | · Requires “objective” evaluation of metal field.  · Performance depends on ordering. |
| OPTION 2 | · Metal Field evaluation is specialized (independent by each DA). No global ordering by the DCA utilized.  · DCA-G & DCA-S agents are free to work on optimization tasks or coalition formations. | · Higher computational load for the DCA. |

By comparing both options we think option 2 will optimize the metal recollection statistical features while having a similar computational performance than option 1. Therefore, we select option 2 for our implementation.

**Mark Computation**

It only remains to define how will DA compute the mark they assign to each MF. We would expect such value to depend on the following independent parameters:

· *Distance DA-MF:* The first parameter the DA should take into account is the distance that it would travel in order to mine the field. Ideally, we would like to compute the shortest path connecting both fields, but depending on the distribution of path fields, this could end up being a very costly computation (breadth-first approach). For this reason, we might be forced to use the following approximation: The Euclidean distance between 2 fields behaves in the same way than the shortest path. That is, a shorter Euclidean distance implies a shorter shortest path. Finally, it is clear that that a higher distance makes the MF less attractive to the DA. Thus, we propose the following dependency between the Mark and the Distance DA-MF:

· *DA* *Available Slots (AS) vs Available Metal Units (AU):* The second parameter that the DA should consider when rating a MF is the number of slots he will be able to fill by mining it. We would expect the DA to be more attracted to the MF if it can completely fill its available slots. More precisely, the higher the proportion of slots filled the better. With all this into account we would propose:

Where is the Heaviside function. By doing this, we make it such that free diggers will have a smaller mark than diggers that are carrying metal if both will mine the field completely and they are at the same distance. On the other hand, if none of the DAs can mine a field completely, then we would like it to send the one that can mine the most (which would be a Free Digger). Therefore:

By adding these two terms we also reinforce the mark of a digger that has the exact same number of available slots than the available units of metal of the field.

*· DA Used Slots (US):* The last parameter that will be considered is the amount of metal being carried by a digger. Naturally, DAs would also like to manufacture the metal they are carrying as quickly as possible. For this reason, we finally define the mark function as:

There are other parameters that could be taken into account, for example, the distance of the DA to the closest manufacturing center, or the points it would receive at such manufacturing center, but we will not use them as it would require the calculation of yet another distance, which would worsen the computational performance of the system.

*Communication:* The cooperation process will be guided by the Contract Net Protocol.

## Coalition Formation

*Description:* Once the digger selection process has finished, the DCA will inform the DCA-G & DCA-S of the distribution of diggers for that turn. Afterwards, both the DCA-S & DCA-G agents will look for MFs that have more than one DA assigned and force the formation of a coalition between those DAs. Such coalition will be centralized and its “leader” will be the agent with the higher mark for that MF. This is an optimization that whose effects will be felt as the system scales with higher number of digger agents.

*Cooperation:* Coalition between DAs. The coalition will be imposed by the DCA-G, DCA-S which acts as the external agent.

*Communication:* Between the DCA-G/DCA-S and the DAs of the coalition.

## Coalition next move calculation

*Description:* Once a coalition is formed, the DAs of the coalition choose the order in which they will mine the MF. By doing so, each DA will be aware of the number of turns that he will have to wait before mining the metal, which will be added in the calculation of the distance (number of turns) that he will have to travel. Therefore, the DAs of the coalition will decrease its mark on the MF and might reconsider mining it if a new MF appears. Once a DA mines the MF, it abandons the coalition.

*Cooperation:* The agents in the coalition exchange information with the “leader”, which ranks the marks and establishes the digging order.

*Communication:* The leader of the coalition will broadcast the plan to the other member of the coalition which will have previously sent its mark for the MF in a P2P communication.

## Movement of free DAs without MF assigned

*Description:* When a free DA does not have any MF assigned it will follow the path of the closest PA.

*Cooperation:* The DA will proactively use the Contract Net Protocol to find the closest PA and form a coalition with it. Once this is done, the PA will share its next move with the DA which will try to follow him (reduce its distance).

*Communication:* Guided by the Contract Net Protocol before the coalition formation, and P2P afterwards.

## Optimization

*Description:* The DCA-G/DCA-S will each compute the values of the parameters but for its metal. Thus, we will have 6 parameters in total: . By using different parameters for each type of metal, we take into account the different distribution of metals in the map.

*Cooperation: None.*

*Communication:* The DCA-G/DCA-S will broadcast the parameters to all the DAs.

# Metal manufacturing

## Selection of and journey to a manufacturing center

*Description*

Metal collecting regards the transportation of a mined metal from the mining cell to a certain manufacturing center, where the metal can be exchanged for points depending on the center. Each manufacturing center only deals with certain kinds of metal.

Selecting the best manufacturing center consists of maximizing the points obtained from the metal, whilst minimizing the turns to get to the manufacturing center. Once the best manufacturing center and route has been chosen and communicated, cooperation and communication are minimal as the agent travels to the manufacturing center.

*Optimization*

Let manufacturing center M, generate points Pm for a given metal that a digger agent has mined. And let the travel distance from the digger to the center be denoted as Rm. Then a simple equation to calculate the efficiency score S of going to center M could be:

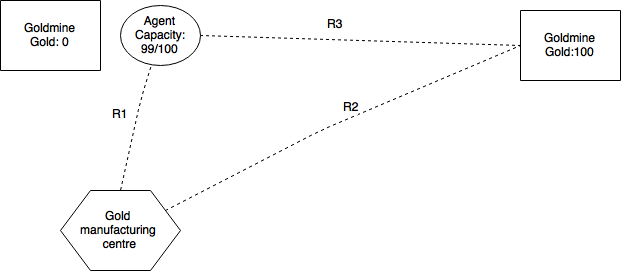
Where alpha (𝛂) denotes a weighting multiplier on the points P and beta (β) denotes a weighting multiplier on the penalization of distance 1/R.

It is clear here that the further a center is from the digger agent, the less incentive for the agent to go there, unless there are also a high number of points associated with manufacturing the metal there.

When choosing which manufacturing center to go to, let’s imagine centers M1 and M2. The scores for the two manufacturing centers would be:

The higher the score, the more efficient the choice of manufacturing center, and therefore the digger should choose the center which gives the highest score. The equation to optimize the choice of manufacturing center is therefore:

However, there is also another factor to consider. A digger can also decide to not go straight to a manufacturing center, but rather if they still have capacity, go to another mine. For example, consider the example below.

[](https://www.draw.io/#G1t0L0FcSrSYKKbnnamW-YYBPQomDsHLZq)

It is seen that if the agent decides to go to the manufacturing center first, the agent must travel a distance of R1 to manufacture 99 gold, compared to if the agent decides to mine to max capacity first, which leads to a distance travelled of R3 +R2, to mine 100 gold.

Therefore, the previous equations have to be modified to take in the different paths and amounts of metal that could be manufactured. For one manufacturing center, we now have several scores, each corresponding to a different path needed to collect metal. The overall score of the manufacturing center, can then be calculated as the maximum of the scores of the different paths:

Rather than a constant number of metal we introduce a variable G, for the metal that will be taken to the manufacturing center by the agent along a path of distance R. Now the best manufacturing center is calculated by calculating the maximum score for each path, and then the maximum score for each manufacturing center:

*Cooperation*

In order to calculate the scores needed to optimize the problem, for an agent, all the possible paths to all the possible manufacturing centers has to be calculated. This leads to the following sum of total calculations which need to be performed:

Where A is the total number of digger agents that are selecting a manufacturing center, Ma is the number of manufacturing centers that manufacture the metal type of agent A, Nam is the number of paths from agent a to manufacturing center m. This can become very expensive if the paths to a manufacturing center are numerous, if there are many manufacturing centers, or if there is a high number of agents. In order to reduce the computational cost, we can use a cooperation system so that the computation of the score is distributed over several agents.

The manufacturing plants have to perform the calculations for many diggers, whereas the digger only has to perform the calculations for itself. Therefore, it is decided to perform the score calculation on the digger agent. Once the digger has calculated his maximum score it knows which manufacturing center it needs to go to, and can proceed to go to the center.

That is not all, however. When calculating a path that includes a visit to another metal mine before reaching a manufacturing center, the digger must be sure that they will be able to go there. This is because the allocation of mines to diggers is done through an auction, and therefore a digger cannot assume that they can mine metal at a mine. Therefore, in order to cooperate with the other agents, once the digger has calculated that the maximum score is achieved when visiting another mine, it must make a bid in that mine’s auction (as described in the metal mining section) to verify that it can take the path. If it loses the auction then it must fall back to the next best path, until it finds one which is allowed.

*Communication*

The communication of the agents can be done over the contract-net protocol. In order to calculate the maximum score:

The digger needs to know and . Also, the points of the manufacturing plants Pm must be communicated to the diggers as well as their position, to calculate Rn. The other field Ga does not need to be communicated by another agent to the digger agent, as it is calculated by the digger agent whilst they dig.

The alpha and beta are communicated to the digger agents by their respective coordinator agents at the initialisation of the system. There can also be subsequent broadcasts of alpha and beta when there is a certain situation where the point and distance weighting have to be changed. For example, when the system is nearing a turn limit to mine metal, beta can be increased, which will penalize taking longer paths to the manufacturing center, and encourage diggers to manufacture metal before the turn limit.

The points that the manufacturing plants yield and their positions are unchanging and therefore can be communicated to the diggers upon the systems initialization by their respective coordinators, and stored in the digger agent’s memory.