Code critique

The following snippet of JavaScript code is presented to you for review. We would like you to comment on the structure, style and functionality, describing how you would improve it. We would like you to offer prose commentary on what you deem to be serious shortcomings and problems, as well as share a corrected version of the code. If you change the structure significantly, you may want to share multiple iterations to show both stylistic improvements as well as more structural changes. Please share your thoughts on the efficiency of this approach, what you infer about the authour's intention with the code, as well as other possible (better) ways to acheive this intended purpose.

```
const worker = require("worker_threads");

async function partSum(MAX, min) {
    return new Promise(resolve => {
        let workerData;
        if (min + 100 > MAX) {
            new worker.Worker(__filename, { workerData: {
                min; min,
                max: MAX,
            }}).on("message", ({sum}) => resolve(sum));
        } else {
            new worker.Worker(__filename, { workerData: {
                min: min,
                max: min + 100,
            }}).on("message", (sum) => resolve(sum));
        }
    });
}

    async function computeAverage(sum, items) { return sum/items; }

if (worker.isMainThread) {
    if (process.argv.length < 3)
        throw new Error("Usage; brokencode.js <min> <max>");
    var min = parseInt(process.argv[2]);
    var max = parseInt(process.argv[3]);

    if (process.argv[2].length > process.argv[1].length
    || process.argv[2] < process.argv[1])
        throw new Error("Min must be less than max");
    let totalPromises = [];
    while (min < max)
        totalPromises.push(partSum(min, max));
        min = min + 100;
    Promise.all(totalPromises).then(totals => {
        const totalSum = 0;
        for (res of totals) {
            totalSum = totalSum + res;
        }
        console.log(
            ‘The sum of the numbers from ${min} to ${max} is ${totalSum}‘);
        console.log(
           "The average value is: " + computeAverage(totalSum, max - min));
    });

} else {
var min = worker.workerData.min;
var max = worker.workerData.max;
let sum = 0;
for (i = min; i < max; i = i + 1) {
sum = sum + i;
}
worker.parentPort.postMessage({sum});
}
```

Coding exercise

We are interested in calculating the time savings and carbon emissions savings of an online meeting over a physical one. Create a simple command-line Node.js script that takes as command line arguments a number of city names (up to ten) representing meeting participant locations. Given this, determine which of the locations would be the optimal choice of a physical meeting if all participants needed to convene face-to-face, where “optimal” means shortest total distance travelled by all participants. You are free to choose how advanced you want to make city name lookup and route planning, but a solution using location data from https://www.npmjs.com/package/cities-with-1000 and simplifying travel distance to be the length of the direct path between cities is acceptable for this exercise. Once a meeting location has been found, compute the cumulative and average travel time for participants, where travel time can be simplified to the time need to traverse the direct path at 60km/h. Also, assuming all participants drive separate, gasoline-fueled cars with a fuel economy of 5.0l/100km, estimate the total carbon emissions (measured in kgs of CO2) resulting from gathering physically.

(Note: this is a programming exercise, and conditions are obviously simplified, making a range of assumptions about travel modes and purposes, as well as eliding the carbon emissions footprint of holding a virtual meeting.)